

ROS Noetic (JetPack >=5.0.1)

ROS Installation :

Jetson

Noetic – Use the "Desktop Install"

X86 (HOST)

Noetic – Use the "Desktop-Full Install"

<http://wiki.ros.org/noetic/Installation/Ubuntu>

From Wiki ROS Noetic Install:

```
#setup computer to accept software from packages.ros.org
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/
/ros-latest.list'

# set up the public key
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -

# update debian repo
$ sudo apt update

#IF ON JETSON PLATFORM:
$ sudo apt install ros-noetic-desktop

#IF ON REMOTE MACHINE:
$ sudo apt install ros-noetic-desktop-full

# this is done in a section further below on this wiki page
#source ros noetic and add it into bashrc
#$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
#$ source ~/.bashrc

#install tools and other dependencies for building ROS packages
$ sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential

#initialize rosdep ( set the ROS parameters before calling it, see section below)
$ sudo rosdep init
$ rosdep update
```

catkin tools :

<https://catkin-tools.readthedocs.io/en/latest/installing.html>

```
$ sudo apt update
$ sudo apt install python3-catkin-tools
```

Building single ROS core package from source

ROS package cv_bridge from debian repositories was built with an open_cv that does not use cuda resources.

*** YOU WILL NEED TO INSTALL OPENCV FROM SOURCE.

```

## Building opencv from source
cd <dir program>
git clone https://gitlab.jhuapl.edu/slamr01/SLAMDev.git
cd SLAMDev/dependencies

# Now edit the install_opencv4.sh script to select the correct platform.

# For Jetson Xavier, uncomment the following (line 30):
# Jetson AGX- volta
CUDA_ARCH=7.2

# For Jetson Orin, uncomment the following (line 32):
# Jetson Orin - Ampere
CUDA_ARCH=8.7

# Now install
#
# NOTE: You may need to disconnect from APLNIS (e.g., disconnect VPN on remote PC if running ENET sharing with
remote PC)
#       in order for OpenCV to access external repos to install properly.
#
#       If you try to run the OpenCV script and it fails before all the install directories are all created
#       (i.e., if the folder "slamdev_temp/opencv" is empty, then delete the "opencv" folder under
"slamdev_temp"
#       before trying again, as the directory handling for the install script isn't perfect and messes up here
#       if only some but not all of the needed directories are created in a single run attempt.
#
./install_opencv4.sh

# Troubleshooting:
# If build errors occur with attempting to include header files that don't exist, it is likely
# that it is trying to build an opencv_contrib module that does not have a required dependency installed.
# To troubleshoot:
# -- check the cmake output generated at the start of the build for whether the required header file is
#    associated with a library in the "Unavailable:" list under "OpenCV modules:"
# -- to see what the dependencies are for a particular opencv_contrib modules, look at the CMakeLists.txt
#    file for that module under path opencv_contrib/modules/<module_name>
# -- then either install the missing dependency, or turn off the build option that requires that dependency if
#    that build option is not needed
#
# BUILD_opencv_viz was at first giving some issues on Jetson Orin, and was turned off to complete install;
# but after building rtabmap, opencv built successfully with viz enabled, so there may be a dependency
# (for vtk?) missing in install_opencv4.sh. When built successfully, vtk7 was present on the system, but
# /usr/bin/vtk did not exist (only vtk7). So for rtabmap build, a symbolic link needed to be created as
follows:
#
#     sudo ln -s /usr/bin/vtk7 /usr/bin/vtk
#
# perhaps adding this symbolic link may have been what helped the OpenCV build as well, but not sure.
#

```

Linking Custom built Opencv for python3

NOTE: this must be done before rebuilding vision_opencv for ROS below

```

# python3 does not link the cv2 wrapper by default:
# you will need to add a symbolic link for the OS to find it
cd /usr/local/lib/python3.8/dist-packages

sudo ln -s /usr/local/opencv_v4.5.5/lib/python3.8/site-packages/cv2 cv2

```

Set up the environment parameter (section below) before going any further!!

ROS environmental parameters:

```
$ mkdir -p /home/slamr01/workspace/ros_home

$ vi ~/.bashrc

# copy the following block into the end of the file:
# -----
# OPENCV
export LD_LIBRARY_PATH=/usr/local/opencv_v4.5.5/lib:$LD_LIBRARY_PATH

# ROS
export ROS_HOME=/home/slamr01/workspace/ros_home
# for running independently:
#export ROS_MASTER_URI=http://localhost:11311
#export ROS_IP=127.0.0.1
# for running with remote PC:
export ROS_MASTER_URI=http://192.168.3.2:11311
export ROS_IP=192.168.3.2
source /opt/ros/noetic/setup.bash
# -----

# apply the new .bashrc settings (or close and reopen the terminal)
source ~/.bashrc
```

Now rebuild vision_opencv as used by the base ROS workspace

```

# assuming you have installed ROS noetic as described above
# you will need to install packages for building from source

sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool
sudo rosdep init
rosdep update

# this was done in the preceding step of this wiki page
# Check that you have exported the library path:
# - you can add to the .bashrc
#export LD_LIBRARY_PATH=/usr/local/opencv_v4.5.5/lib:$LD_LIBRARY_PATH
#echo "export LD_LIBRARY_PATH=/usr/local/opencv_v4.5.5/lib:$LD_LIBRARY_PATH" >> ~/.bashrc

# Now build the core package / packages
mkdir ~/programs/ros_catkin_ws
cd ~/programs/ros_catkin_ws

mkdir src build logs devel

# download the packages from repos ( examples only shows cv_bridge )
cd src
git clone https://github.com/ros-perception/vision_opencv.git
cd vision_opencv/
# checkout the corresponding branch or tag
git checkout noetic

# back to workspace
cd ~/programs/ros_catkin_ws

catkin config --init --install-space /opt/ros/noetic --install --extend /opt/ros/noetic

# OPTIONAL: check the dependencies of the packages have been installed.
# if the packages is overwriting a previously installed packaged ( core debian repo ), the dependencies
should have been installed.
rosdep check --from-paths ./src --ignore-packages-from-source --rosdistro noetic -y

sudo catkin build

# check the newest package has overwritten packages in /opt/ros/noetic
# check logs for the package (it should have used the latest opencv) and check version (it should match the
newly installed version)
rosversion cv_bridge

```

Errors for PC Build of OpenCV

Building OpenCV4.5.5 with CUDA v12 fails: https://www.reddit.com/r/opencv/comments/zh2zhw/bug_problem_compiling_opencv_with_cuda_support_on/

Manually building with OpenCV 4.7.0 solves the OpenCV compile issue. If any CMake files for this application specify a build requirement for OpenCV 4.5.5, then change the OpenCV version specification to 4.7.0 in the CMake files to build the application.