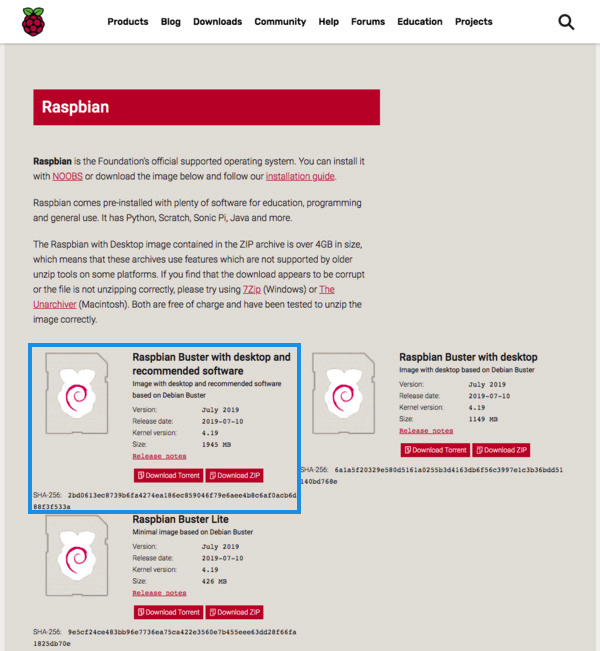
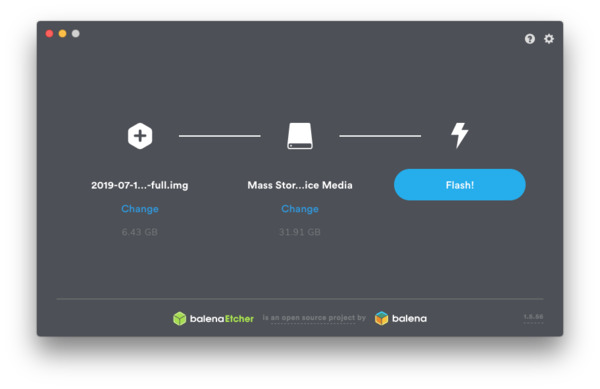
ROS Melodic on Raspbian Buster

This tutorial assumes you have a properly configured Raspberry Pi 4 running a Debian Buster OS. If not, follow steps 1-4.

1. **[](https://pyimagesearch.com/wp-content/uploads/2019/09/install_opencv4_buster_download_os.jpg)**Head on over to the [**official BusterOS download page**](https://www.raspberrypi.org/downloads/raspbian/) (**Figure 2**), and start your download. I recommend the “Raspbian Buster with Desktop and recommended software”.

**Figure 1:** Download Raspbian Buster for your Raspberry Pi and OpenCV 4.

1. Download [**Balena Etcher**](https://www.balena.io/etcher/) — software for flashing memory cards. It works on every major OS.
2. **[](https://pyimagesearch.com/wp-content/uploads/2019/09/install_opencv4_buster_etcher.jpg)**Use Etcher to flash BusterOS to your memory card (**Figure 2**).

**Figure 2:** Flash Raspbian Buster with Etcher. We will use BusterOS to install OpenCV 4 on our Raspberry Pi 4.

1. Once the flash process is complete, you should be able to insert the microSD card into your Pi and power on.

# Expand filesystem and reclaim space

For the remainder of this tutorial I’ll be making the following assumptions:

1. You are working with a brand new, fresh install of **Raspbian Buster** (see the previous section to learn how to flash Buster to your microSD).
2. You are comfortable with the command line and Unix environments.
3. You have an [**SSH or VNC connection established with your Pi**](https://pyimagesearch.com/2019/07/01/remote-development-on-the-raspberry-pi-or-amazon-ec2/). Alternatively, you could use a keyboard + mouse + screen.

Go ahead and insert your microSD into your Raspberry Pi and boot it up with a screen attached.

Once booted, configure your WiFi/ethernet settings to connect to the internet (you’ll need an internet connection to download and install required packages for OpenCV).

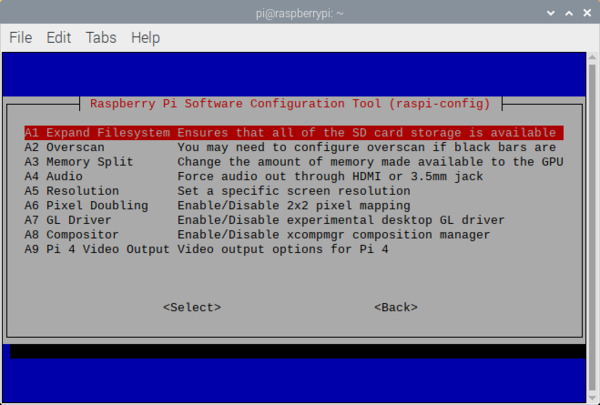
From there you can [**use SSH**](https://pyimagesearch.com/2019/07/01/remote-development-on-the-raspberry-pi-or-amazon-ec2/) as I have done, or go ahead and open a terminal.

The first step is to run raspi-config and expand your filesystem.

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ raspi-config

And then select the ***“7 Advanced Options”*** menu item:

**[](https://pyimagesearch.com/wp-content/uploads/2019/09/install_opencv4_buster_raspi_config_expand_fs.jpg)**Followed by selecting ***“A1 Expand filesystem”***:

**Figure 3:** The `A1 Expand Filesystem` menu item allows you to expand the filesystem on your microSD card containing the Raspberry Pi Buster operating system. Then we can proceed to install OpenCV 4.

Once prompted, you should select the first option, ***“A1 Expand File System”***, hit enter on your keyboard, arrow down to the ***“<Finish>”*** button, and then reboot your Pi — you may be prompted to reboot, but if you aren’t you can execute.

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo reboot

After rebooting, your file system should have been expanded to include all available space on your micro-SD card. You can verify that the disk has been expanded by executing df -h and examining the output.

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ df -h

Filesystem Size Used Avail Use% Mounted on

/dev/root 29G 5.3G 23G 20% /

devtmpfs 1.8G 0 1.8G 0% /dev

tmpfs 1.9G 0 1.9G 0% /dev/shm

tmpfs 1.9G 8.6M 1.9G 1% /run

tmpfs 5.0M 4.0K 5.0M 1% /run/lock

tmpfs 1.9G 0 1.9G 0% /sys/fs/cgroup

/dev/mmcblk0p1 253M 40M 213M 16% /boot

tmpfs 386M 0 386M 0% /run/user/1000

As you can see, my Raspbian filesystem has been expanded to include all 32GB of the micro-SD card.

# Increase Swapspace

Since this will be a hefty installation, I recommend increasing your swapspace. Once your Pi is running, open a terminal window and access your dphys-swapfile with the following command.

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo nano /etc/dphys-swapfile

Scroll down until you see this variable:

CONF\_SWAPSIZE=100

And edit it to this:

CONF\_SWAPSIZE=1024

Or, if you have the 4GB Raspberry Pi 4, change it to this instead:

CONF\_SWAPSIZE=2048

(This allows the Pi to use up to 1024 Mb (1 Gb) of space on the SD card for extra RAM. However, it also lowers the life of the SD card if left this high so we will put it back after we're done.)

Exit the file by typing ‘CTRL+X’, and then ‘Y’

Restart swapspace

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo /etc/init.d/dphys-swapfile stop

$ sudo /etc/init.dphys-swapfile start

# ROS Dependencies

Setup proper certificate management

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo apt-get install dirmngr

Add ROS to the repository

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo sh -c ‘echo “deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main” > /etc/apt/sources.list.d/ros-latest.list’

$ wget http://packages.ros.org/ros.key -O - | sudo apt-key add-

Make sure your system is up to date

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo apt-get update

$ sudo apt-get upgrade

# Install bootstrap dependencies

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo apt-get install python-rosdep2 python-rosinstall-generator python-wstool python-rosinstall build-essential cmake

# Install rosdep

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo rosdep init

$ rosdep update

# Resolve Missing Packages

Install missing package for embedding Python expressions and statements in template text

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo apt-get install python-empy

Install missing packages for qt\_gui

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo apt-get install sudo apt-get install sip-dev pyqt5-dev python-sip-dev pyqt5-dev-tools

Install missing packages for related to running roscore

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

sudo apt-get install python-defusedxml

sudo apt-get install python-netifaces

# Configure and Download Software

Create workspace

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ mkdir ros

$ cd ros

Download core ROS packages into src folder. Note this installation is for desktop version, as opposed to the headless comm option

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rosinstall\_generator desktop --rosdistro melodic --deps --wet-only --tar > melodic-desktop-wet.rosinstall

$ wstool init -j8 src melodic-desktop-wet.rosinstall

Resolve ROS dependencies with rosdep (this will take 10-15 minutes)

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rosdep install -y --from-paths src --ignore-src --rosdistro melodic -r --os=debian:buster

Build the catkin workspace (error-free, this will process will take at least an hour)

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo ./src/catkin/bin/catkin\_make\_isolated -j4 --install -DCMAKE\_BUILD\_TYPE=Release --install-space /opt/ros/melodic

Source the installation with ~/.bashrc

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

$ source /opt/ros/melodic/setup.bash

Verify the installation with the following command

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ export | grep ROS

A screenshot of a computer

Description automatically generated

**Figure 4:** Terminal output of ‘export | grep ROS’ after a valid ROS installation

Change swapspace back to 100 mb (or not yet if you want to install rosserial or OpenCV)

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo nano /etc/dphys-swapfile

CONF\_SWAPSIZE = 100

CTRL+X, Y

# Install the Arduino IDE for ROS

The Arduino platform is a convenient way to interface the Raspberry Pi with ROS to control external peripherals. Install Arduino with the following

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ sudo apt-get update

$ sudo apt-get dist-upgrade

$ sudo apt-get install arduino

Ensure Arduino IDE is accessible (i.e. Desktop icon), and has created a ~/sketchbook/libraries or ~/Arduino/libraries directory in your home folder. The directory’s only contents should be a ‘readme.txt’.

Next, we need the rosserial packages; which will in turn require a separate workspace. Create a directory called ‘catkin\_ws’ within your ‘ros’ folder and run the catkin\_make command to set it u

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ mkdir -p ~/ros/catkin\_ws/src

$ cd ~ros/catkin\_ws/

$ catkin\_make

Source the new workspace (note the ‘echo’ command appends the segment in quotations to the bottom of your system’s ~/.bashrc fil

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ echo “source $HOME/ros/catkin\_ws/devel/setup.bash” >> ~/.bashrc

Navigate to the ‘src’ folder of your ‘catkin\_ws’ directory and download: rosserial drivers from Github

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ cd catkin\_ws/src

$ git clone <https://github.com/ros-drivers/rosserial.git>

Go up one directory and generate the rosserial\_msgs needed for communication

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ cd ..

$ catkin\_make

Make the libraries, which will be stored in the ‘catkin\_ws/devel/lib’ director

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ catkin\_make install

Open new terminal to ensure packages are recognized

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rospack list

The list should contain multiple references to ‘rosserial\_’

**A screenshot of a computer

Description automatically generated**

**Figure 5:** Running ‘rospack list’ should produce a list resembling this

Run the following command to make the ‘ros\_lib’ library which will contain all ‘rosserial\_arduino’ packages. Note the period at the end has a single space between it and the ‘y’; that is no typo, it instructs ‘rosrun’ to place the ‘ros\_lib’ directory within the current folder.

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rosrun rosserial\_arduino make\_libraries.py .

Verify that there is a ros\_lib folder within the ~/sketchbook/libraries folder

Connect Arduino to Raspberry Pi

**A screenshot of a cell phone

Description automatically generated**Open Arduino IDE and determine serial port name under: Tools -> Serial Port

**Figure 6:** Examining the serial port in the Arduino IDE

Make note of Serial Port name: such as **/dev/ttyACM1**

Navigate to File -> Examples -> ros\_lib -> Blink, as shown in Figure 8.

**A screenshot of a cell phone

Description automatically generated**

**Figure 7:** Example Arduino-ROS code can be opened from the IDE

Open the file and upload the Blink code onto the Arduino. Note, sometimes getting a successful upload requires unplugging and reconnecting the Arduino board. If done, be sure to note any changes to the serial port number.

Open a new terminal and start ROS master

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ roscore

Open another terminal and initiate serial connection. Note the reference to serial port ‘/dev/ttyACM1’

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rosrun rosserial\_python serial\_node.py /dev/ttyACM1

Both of the Arduino’s onboard LED’s should blink simultaneously every couple seconds. Congratulations! You now have your ROS/Arduino serial interface established.

Open yet another terminal and enter the following command to list the running ROS nodes

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rosnode list

To stop the rosserial node currently running, use the ‘rosnode kill’ command and add the unique name provided in the previous command

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ rosnode kill /serial\_node

To kill roscore, simply hit CTRL+C

If you already have OpenCV installed in a python virtual environment on your Pi and intend to use ROS with it, create sym-links between ROS and the virtual environment. Be sure to use tab-completion for the ‘ln -s’ commands as your file paths may differ from mine. The three ROS packages that need to be linked are ‘yaml’, ‘rospkg’, and ‘catkin\_pkg’.

Install ROS Melodic on Raspberry Pi 4 and Raspbian Buster

$ cd ~

$ workon cv

$ cd ~/.virtualenvs/cv/lib/python3.7/site-packages/

$ ln -s /usr/lib/python3/dist-packages/mypy/typeshed/third\_party/2and3/yaml

$ ln -s /usr/lib/python2.7/dist-packages/rospkg

$ ln -s /usr/lib/python2.7/dist-packages/catkin\_pkg

# References

|  |  |
| --- | --- |
| [1] | A. Rosebrock, "Install OpenCV 4 on Rasbian Buster," Pyimagesearch, 16 September 2019. [Online]. Available: <https://www.pyimagesearch.com/2019/09/16/install-opencv-4-on-raspberry-pi-4-and-raspbian-buster/>. [Accessed 27 July 2020]. |
| [2] | ROS.org, "Installing ROS Melodic on the Raspberry Pi," Open Source Robotics Foundation, 19 April 2020. [Online]. Available: <http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Melodic%20on%20the%20Raspberry%20Pi>. [Accessed 30 July 2020]. |
| [3] | D. Calin, "How To Install ROS Melodic, rosserial, and more on Raspberry Pi 4 (Raspbian Buster)," Intro Robotics, 15 April 2020. [Online]. Available: <https://www.intorobotics.com/how-to-install-ros-melodic-rosserial-and-more-on-raspberry-pi-4-raspbian-buster/>. [Accessed 30 July 2020]. |