

Getting Started (Part III)

Approximate Time Investment: 1 hour

Installing ROS2 Humble, Create Workspace, and Setup bashrc for Better Quality of Life

At this stage, we have exhausted the checks we can perform without ROS2. ROS2 is the critical middleware that we will use going forwards for working with the vehicle. It is assumed that the reader has watched the beginners guide in the prerequisites and has basic knowledge of ROS2 (creating packages, creating nodes, building a workspace, publishers/subscribers, topics, etc.)

1. Install ROS2 Humble by following the official installation guide. This step should be a repeat of the installation process from the prerequisite ROS2 beginner's guide by Robotics Back-End.
 - a. [ROS2 Humble for Beginners by Robotics Back-End](#) (11 video series)
 - b. [Official ROS2 Humble Installation Guide](#)
2. Install colcon for building ROS packages. Colcon is an open-source, command-line tool that automates the building, testing, and installation of multiple software packages.
 - a. Open a new terminal and run the following command

```
sudo apt install python3-colcon-common-extensions
```

3. Create and build a ROS2 workspace by running the following commands. I am creating a folder called "ros2_ws" in the Home directory.

```
cd                # this will bring us to the home directory
mkdir ros2_ws     # this will create a folder called "ros2_ws" which will be our workspace
cd ros2_ws
mkdir src
colcon build
```

Our workspace should successfully build and you should see a screen similar to that in the next figure.

```

jeffoh@trc: ~/ros2_ws
jeffoh@trc:~/ros2_ws$ colcon build
Starting >>> ds_dbw_msgs
Starting >>> velodyne_msgs
Starting >>> ouster_sensor_msgs
Starting >>> velodyne_laserscan
Starting >>> arena_camera_node
Starting >>> ouster_alarm
--- stderr: arena_camera_node
-- [ DEBUG ] arena_sdk_installation_root = /home/jeffoh/Arena_SDK/ArenaSDK_Linux_x64
---
Finished <<< arena_camera_node [0.70s]
Finished <<< velodyne_laserscan [0.72s]
Finished <<< velodyne_msgs [1.65s]
Starting >>> velodyne_driver
Starting >>> velodyne_pointcloud
Finished <<< ouster_sensor_msgs [1.67s]
Starting >>> ouster_ros
Finished <<< ouster_alarm [1.91s]
Finished <<< velodyne_driver [0.86s]
Finished <<< velodyne_pointcloud [1.03s]
Starting >>> velodyne
Finished <<< ds_dbw_msgs [2.77s]
Starting >>> ds_dbw_can
Starting >>> mkz_blinker
Finished <<< ouster_ros [1.50s]
Finished <<< velodyne [0.51s]
Finished <<< ds_dbw_can [0.63s]
Starting >>> ds_dbw_joystick_demo
Finished <<< ds_dbw_joystick_demo [0.48s]
Starting >>> ds_dbw
Finished <<< ds_dbw [0.20s]
Finished <<< mkz_blinker [1.57s]

Summary: 14 packages finished [4.76s]
1 package had stderr output: arena_camera_node
jeffoh@trc:~/ros2_ws$

```

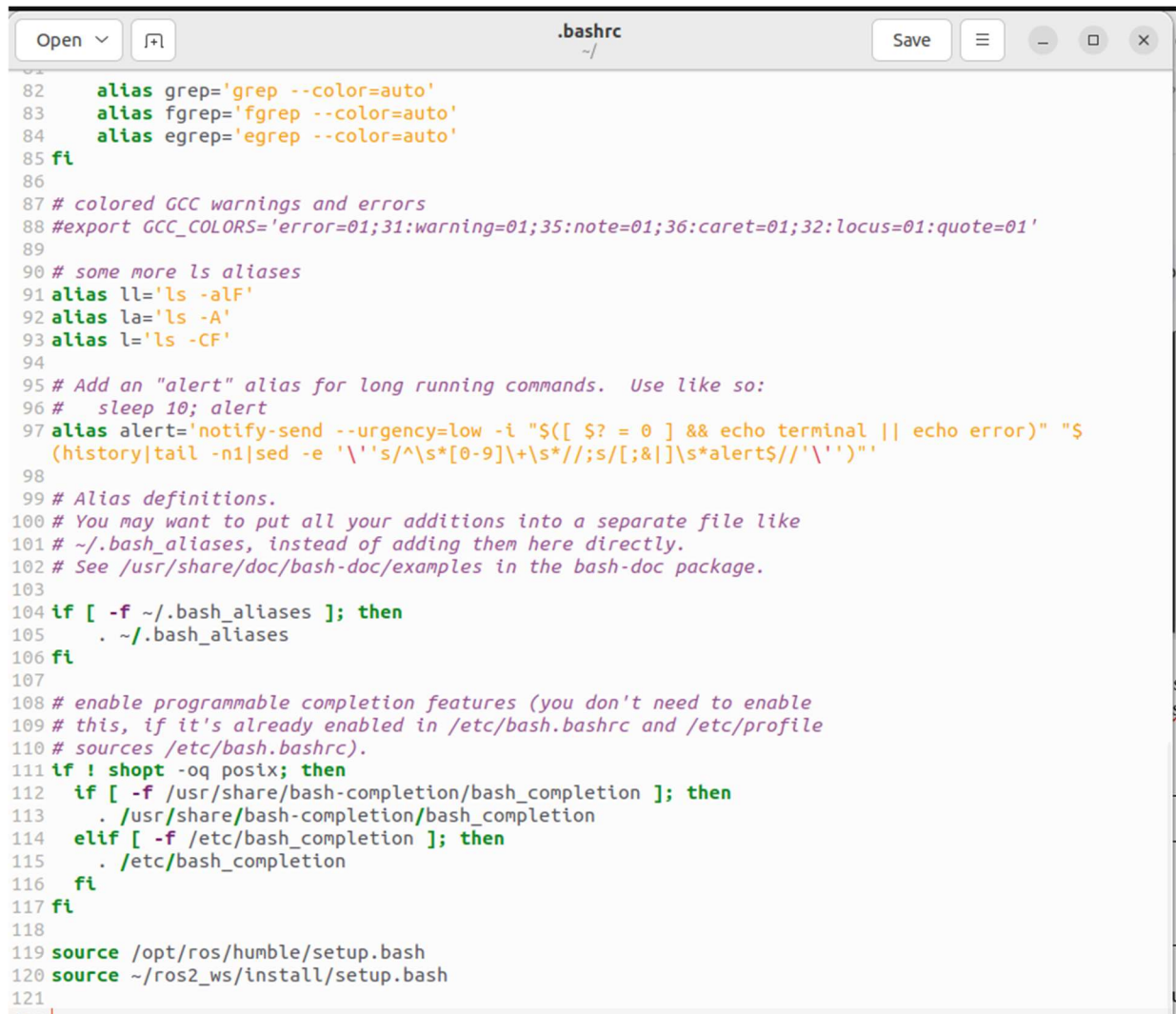
Figure 1: Our ros2_ws colcon build result

4. You may recall from the ROS2 beginner's guide that we must source the ROS2's setup.bash file and our workspace's setup.bash file every time we need to use ROS. We will edit our bashrc file so that those files are automatically sourced every time we open a new terminal.
 - a. In the terminal open bashrc with the following command

```
gedit ~/.bashrc
```

- b. Inside bashrc, add the following lines at the bottom:

```
source /opt/ros/humble/setup.bash
source ~/ros2_ws/install/setup.bash # this location may vary depending on where you created the workspace
```



```

82 alias grep='grep --color=auto'
83 alias fgrep='fgrep --color=auto'
84 alias egrep='egrep --color=auto'
85 fi
86
87 # colored GCC warnings and errors
88 #export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
89
90 # some more ls aliases
91 alias ll='ls -alF'
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # Add an "alert" alias for long running commands. Use like so:
96 # sleep 10; alert
97 alias alert='notify-send --urgency=low -i "${? = 0} && echo terminal || echo error)" "$
98 (history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\`"'
99
100 # Alias definitions.
101 # You may want to put all your additions into a separate file like
102 # ~/.bash_aliases, instead of adding them here directly.
103 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
104
105 if [ -f ~/.bash_aliases ]; then
106     . ~/.bash_aliases
107 fi
108
109 # enable programmable completion features (you don't need to enable
110 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
111 # sources /etc/bash.bashrc).
112 if ! shopt -oq posix; then
113     if [ -f /usr/share/bash-completion/bash_completion ]; then
114         . /usr/share/bash-completion/bash_completion
115     elif [ -f /etc/bash_completion ]; then
116         . /etc/bash_completion
117     fi
118 fi
119
120 source /opt/ros/humble/setup.bash
121 source ~/ros2_ws/install/setup.bash
122

```

Figure 2: The bottom of your bashrc file should have the two source lines.