Rover Robotics Computer Setup Guide

Version	Date	Comments
1.0.0	1/16/20	Initial Release
1.0.1	3/26/20	Fix rover UART latency and xbox controller assignment issues
1.0.2	3/26/20	Update ros install instructions
2.0.0	6/11/20	Add RPLidar, move_base, robot localization and gmapping ROS packages and rename main launch files to starterkit_bringup.launch and slampack_bringup.launch
3.0.0	7/14/20	Add ROS 2 install, create user “rover”
3.0.1	7/22/20	Minor updates to add usersection
3.0.2	8/5/2020	Minor usability updates and add RealSense driver install

This document has been tested for the following computers, operating systems, and versions of ROS

Computers

- Nvidia Jetson TX2
- Nvidia Jetson Xavier AGX
- Nvidia Jetson Xavier NX
- Intel NUC7i5DNH1E
- ADLINK ROSCube Pico

Operating Systems

- Ubuntu 18.04

Versions of ROS

- ROS Melodic

Table of Contents

This document was written so you can copy and paste commands in order and it should work.

Commands that should be typed into the terminal are highlighted in light grey

Text that needs to be placed inside a file is colored in light blue

⚠ Be careful not to copy and paste any hidden characters ⚠

Step 0 – Create new User called rover (optional)

Optionally you can choose not to create a new user, you will just need to change the user name in the systemd scripts in step 4

To create a new user run the following command

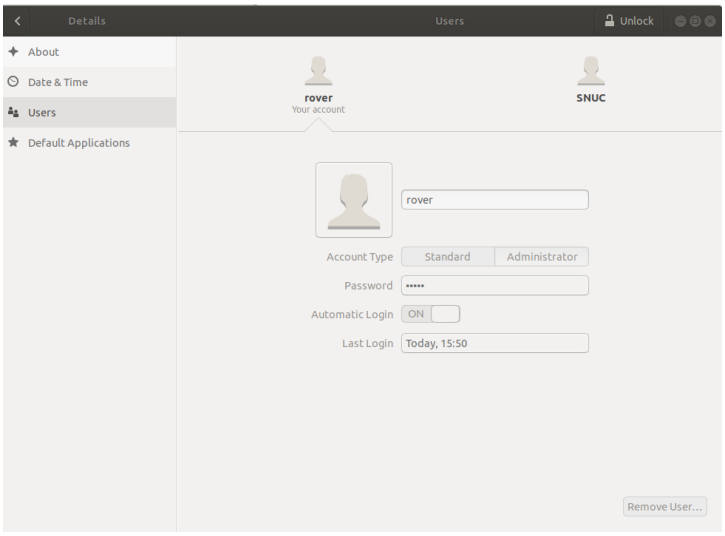
```
sudo adduser rover
```

Set the password to rover
Set the full name to rover
Leave Room Number, Work Phone, Home Phone and Other blank

Then to give the user sudo permissions run this command

```
sudo usermod -aG sudo rover
```

From the users menu select unlock and change the rover user to auto login



Step 1 - Install nice to have programs

Git must be installed to download Rover Robotics code from source

Nano must be installed to create and edit files

```
sudo apt install git

sudo apt install nano

sudo apt install net-tools

sudo apt install openssh-server
```

Step 2 - Install ROS 1 Melodic

The below commands in red are listed here just for reference. It is recommended to use the commands listed online for installing ROS to ensure they are up to date http://wiki.ros.org/melodic/Installation/Ubuntu. Particularly the second command that has the key that gets updated from time to time.

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

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

sudo apt update

sudo apt install ros-melodic-desktop

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

source ~/.bashrc

sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential

sudo apt install python-rosdep

sudo rosdep init

rosdep update
```

Step 2 - Install Rover Robotics packages from source

We recommend installing from source so you can see the driver code that is running and edit it if you need. It’s also available as binaries if you move to production.

```
mkdir -p ~/catkin_ws/src

cd ~/catkin_ws/src

git clone https://github.com/RoverRobotics/rr_openrover_stack.git

git clone https://github.com/Slamtec/rplidar_ros.git

sudo apt-get install ros-melodic-twist-mux

sudo apt-get install ros-melodic-tf2-geometry-msgs

sudo apt-get install ros-melodic-robot-localization

sudo apt-get install ros-melodic-gmapping

sudo apt-get install ros-melodic-move-base

cd ~/catkin_ws

catkin_make

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

source ~/.bashrc

sudo cp ~/catkin_ws/src/rr_openrover_stack/rr_openrover_navigation/rviz/default.rviz /opt/ros/melodic/share/rviz/
```

Step 3 - Install Joystick Drivers

Install the ROS joystick driver

sudo apt-get install ros-melodic-joy

Plug in the Xbox wireless dongle and turn on the controller.

Open the terminal and run this command

sudo apt-get install --install-recommends jstest joystick xboxdrv setserial*

You also need to ensure that xpad is not getting loaded

echo "blacklist xpad" | sudo tee -a /etc/modprobe.d/blacklist.conf

sudo rmmod xpad

Its okay if the command give the error (rmmod: ERROR: Module xpad is not currently loaded)

sudo nano /etc/udev/rules.d/95-xboxdrv.rules

SUBSYSTEM=="usb", DRIVER=="usb", ATTRS{idVendor}=="045e", ATTRS{idProduct}=="0719", ACTION=="add", RUN+="/bin/systemctl restart xboxdrv.service"
SUBSYSTEM=="usb", DRIVER=="usb", ATTRS{idVendor}=="045e", ATTRS{idProduct}=="0719", ACTION=="remove", RUN+="/bin/systemctl stop xboxdrv.service"

sudo nano /etc/systemd/system/xboxdrv.service

[Unit]
Description=Xbox controller driver daemon
Documentation=man:xboxdrv(1)

[Service]
Type=simple
User=root
PIDFile=/var/run/xboxdrv.pid
ExecStartPre=/bin/sleep 2
ExecStart=/usr/bin/xboxdrv --daemon --detach-kernel-driver --pid-file /var/run/xboxdrv.pid --dbus disabled --silent

[Install]
WantedBy=multi-user.target

Step 4 - Create systemd scripts

These scripts start ROS when you boot your computer and they

```
sudo mkdir /etc/roverrobotics

sudo nano /etc/systemd/system/roscore.service
    [Unit]
    After=NetworkManager.service time-sync.target
    [Service]
    Type=forking
    User=rover
    # Start roscore as a fork and then wait for the tcp port to be opened
    # -------------------------------------------------------------
    # Source all the environment variables, start roscore in a fork
    # Since the service type is forking, systemd doesn't mark it as
    # 'started' until the original process exits, so we have the
    # non-forked shell wait until it can connect to the tcp opened by
    # roscore, and then exit, preventing conflicts with dependant services
    ExecStart=/bin/sh -c ". /opt/ros/melodic/setup.sh; . /etc/roverrobotics/env.sh;
    roscore & while ! echo exit | nc localhost 11311 > /dev/null; do sleep 1; done"
    [Install]
    WantedBy=multi-user.target
```

If you skipped step 0, fill in the user name where is says rover

```
sudo nano /etc/roverrobotics/env.sh
    #!/bin/sh
    export ROS_HOSTNAME=$(hostname).local
    export ROS_MASTER_URI=http://$ROS_HOSTNAME:11311

sudo nano /etc/systemd/system/roverrobotics.service
    [Unit]
    Requires=roscore.service
    PartOf=roscore.service
    After=NetworkManager.service time-sync.target roscore.service
    [Service]
    Type=simple
    User=rover
    ExecStart=/usr/sbin/roverrobotics
    [Install]
    WantedBy=multi-user.target
```

```
sudo nano /usr/sbin/roverrobotics
        #!/bin/bash
        source ~/catkin_ws/devel/setup.bash
        source /etc/roverrobotics/env.sh
        export ROS_HOME=$(echo ~rover)/.ros
        roslaunch rr_openrover_driver starterkit_bringup.launch &
        PID=$!
        wait "$PID"

sudo systemctl enable xboxdrv.service

sudo systemctl enable roverrobotics.service

sudo systemctl enable roscore.service

sudo chmod +x /usr/sbin/roverrobotics
```

Step 5 - Setup UDEV Rules (simlinks)

sudo nano /etc/udev/rules.d/roverrobotics.rules

```
# set the udev rule , make the device_port be fixed by rplidar
#
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60",
MODE:="0777", SYMLINK+="rplidar"
# creates fixed name for rover serial communication
KERNEL=="ttyUSB[0-9]", ATTRS{idVendor}=="0403",
ATTRS{idProduct}=="6001", MODE:="0777", SYMLINK+="rover",
RUN+="/bin/setserial /dev/%k low_latency"
KERNEL=="ttyUSB[0-9]", ATTRS{idVendor}=="0403",
ATTRS{idProduct}=="6015", MODE:="0777", SYMLINK+="rover",
RUN+="/bin/setserial /dev/%k low_latency"
# create fixed mapping for xbox control to avoid inconsistent naming
SUBSYSTEM=="input", KERNEL=="js*", ATTRS{name}=="Xbox Gamepad
(userspace driver)", SYMLINK="input/jsX"
```

sudo udevadm control --reload-rules && sudo udevadm trigger

Step 6 – Install RealSense D435i Dev Packages (optional)

Install the realsense librairies

```
sudo apt-key adv --keyserver keys.gnupg.net --recv-key F6E65AC044F831AC80A06380C8B3A55A6F3EFCDE || sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-key F6E65AC044F831AC80A06380C8B3A55A6F3EFCDE
```

```
sudo add-apt-repository "deb http://realsense-hw-public.s3.amazonaws.com/Debian/apt-repo bionic main" -u
```

```
sudo apt-get install librealsense2-dkms librealsense2-utils librealsense2-dev librealsense2-dbg
```

Install RealSense ROS packages from source

```
cd ~/catkin_ws/src
```

```
git clone https://github.com/IntelRealSense/realsense-ros.git
```

```
git checkout tags/2.1.14
```

```
cd ..
```

```
catkin_make
```

Step 6 - Reboot

```
sudo reboot
```

Appendix A - Advice

Don’t turn this document into a PDF and copy and paste commands, the PDF will insert hidden characters