NeuronBot
User's Manual

Rapid Robotic Development Platform

Manual Rev.:	1.0
Revision Date:	October 29, 2020
Part Number:	50-1Z334-1000

Leading **EDGE COMPUTING**

Preface

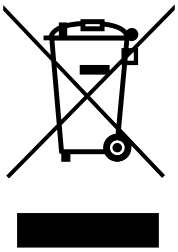
Copyright

Disclaimer

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer. In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Environmental Responsibility

ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.



Battery Labels (for products with battery)

California Proposition 65 Warning

WARNING: This product can expose you to chemicals including acrylamide, arsenic, benzene, cadmium, Tris(1,3-dichloro-2-propyl)phosphate (TDCPP), 1,4-Dioxane, formaldehyde, lead, DEHP, styrene, DINP, BBP, PVC, and vinyl materials, which are known to the State of California to cause cancer, and acrylamide, benzene, cadmium, lead, mercury, phthalates, toluene, DEHP, DIDP, DnHP, DBP, BBP, PVC, and vinyl materials, which are known to the State of California to cause birth defects or other reproductive harm. For more information go to www.P65Warnings.ca.gov.

Trademarks

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Revision History

Revision	Description	Date	By
1.0	Initial release	2020-10-29	TS

Table of Contents

List of Figures

This page is intentionally left blank.

1. Introduction

The NeuronBot is an affordable, miniature, autonomous rapid robotic development platform with integrated computational unit, LiDAR sensor, high payload capacity and dynamic motion capability, and is ideal for enabling a wide range of exciting research, training, and educational activities.

2. Product Overview

2.1. Main Specifications

NeuronBot NB-SK	
Main System	
Processor	Intel® Celeron® Processor G3900TE
Memory	2x 4G DDR4
Storage	64GB mSATA
Inertial Motion Sensor	GY85 9-axis IMU 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer
Motor Control Unit	Arduino Mega 2560
Encoder	7N14P 2-channel for motor control
Laser Distance Sensor	
Sensor Model	2D 360° RPLIDAR A1, 12 meters
Sample Frequency	8000Hz
Scan Rate Range	1-10Hz, 5.5Hz typical
Communication Interface	USB/UART
Front Side	
Status LED Bar (front)	The status LED bar lights one of the following colors according to the system status: • Amber: NeuronBot is powered on. • Blue: Base driver is active. • Off: NeuronBot is powered off.
Camera Area	For Intel® RealSense™ Depth Camera D435 (optional side stand bracket required for installation)
Rear Side	
Battery Panel	Battery status display
Power Button	Power On/Off button
GPIO	1x GPIO connector
Power Requirements	
Motherboard DC Input	24V DC ±5% via ATX power connector
Battery	Provided by user if required

NeuronBot NB-SK	
Mechanical	
Payload	3 kg
Wheel Diameter	83 ±2 mm
Wheel Center Distance	218 ±3 mm
Translational Velocity Max.	0.6 m/s
Rotational Velocity Max.	0.6 m/s
Climbing Threshold	0° ±1°
Actuator	DC carbon-brush motor (1:139)
Dimensions	260 x 270 x 260 mm (10.24 x 10.63 x10.24 inches)
Weight	7.8 kg
Environmental	
Operating Temperature	0°C to 50°C (32°F to 122°F)
Operating Humidity	10% to 95%, non-condensing
Storage Temperature	-20°C to 80°C (-4°F to 176°F)
EMC	Compliant with CE, FCC Class B
Vibration	Package random vibration: IEC 60068-2-64, 5-500Hz, 5Grms, 1hr/axis
Drop	ISTA-1A
Software	
SDK	Neuron SDK (optional)
Environment	Ubuntu 18.04 LTS
Middleware	ROS/ROS 2 Intel® OpenVINO™

2.2. Power Specifications

Motherboard Power Input	12VDC ±5% with ATX power connector
EOS1300-PWBD Power Board	**Input:** 24VDC ±5%, **Output:** 12VDC ±5%, 24VDC ±10%, 5Vsb ±5%
	Output protection: 12V and 5Vsb short protection by DC-DC converter

2.3. Software

Environment	Ubuntu 18.04 LTS
Middleware	ROS/ROS2
SDK	Neuron SDK (optional)

2.4. Package Contents

Device	1x NeuronBot Rapid Robotic Development Platform
Documentation	1x Quick Start Guide
Cable	1x DisplayPort to HDMI adapter cable

2.5. Optional Accessories

ADLINK provides all the necessary parts and accessories for NeuronBot. You can purchase additional accessories according to your needs.

Item	Description
Camera	Intel® RealSense™ Depth Camera D435
Support Plate	Attachable top plate for LIDAR protection, object transport, and top camera installation
Brackets	Front bracket for Intel® RealSense™ Depth Camera D435
	Top bracket for Intel® RealSense™ Depth Camera D435
Stand-offs	M3 stand-offs for attaching support plate
	Large, top-side stand-off for attaching top camera

2.6. Mechanical Dimensions

Note: All are dimensions shown in millimeters.

2.6.1. Front View Dimensions

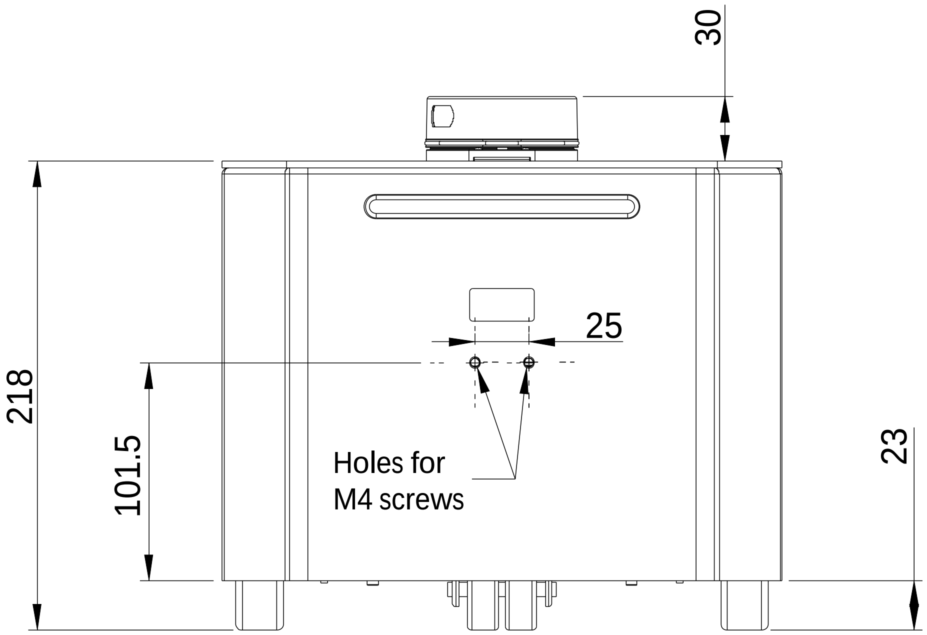


Figure 1: Front View Dimensions (No Accessories)

Note: M4 screw holes are for attaching the front camera bracket to the chassis (see Front Camera Installation on page 21).

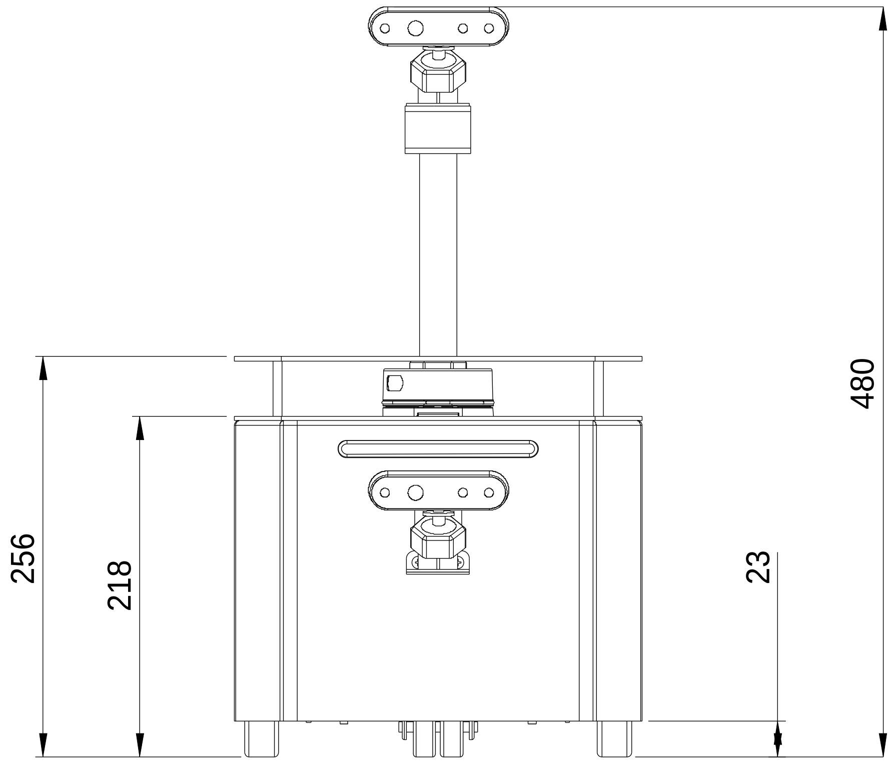


Figure 2: Front View Dimensions (with Accessories)

2.6.2. Side View Dimensions

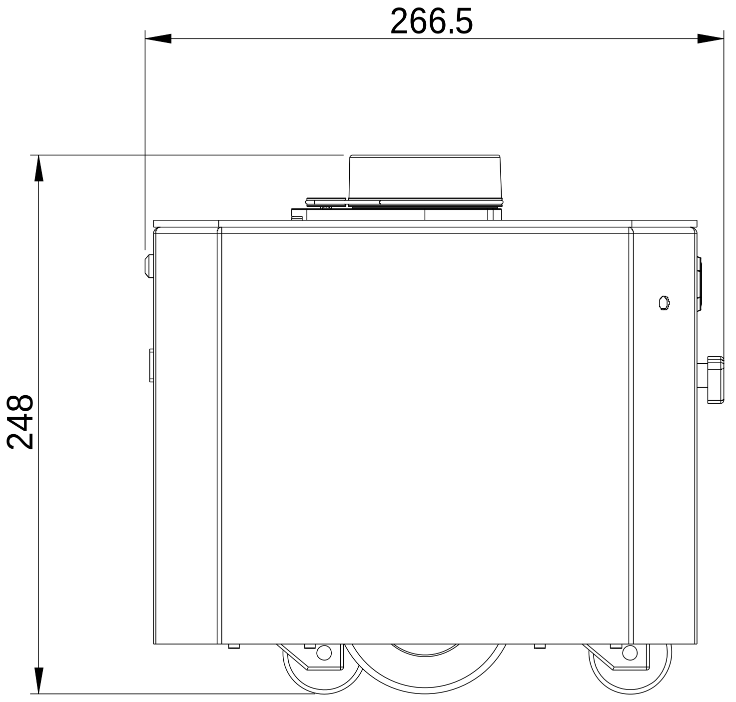


Figure 3: Side View Dimensions (No Accessories)

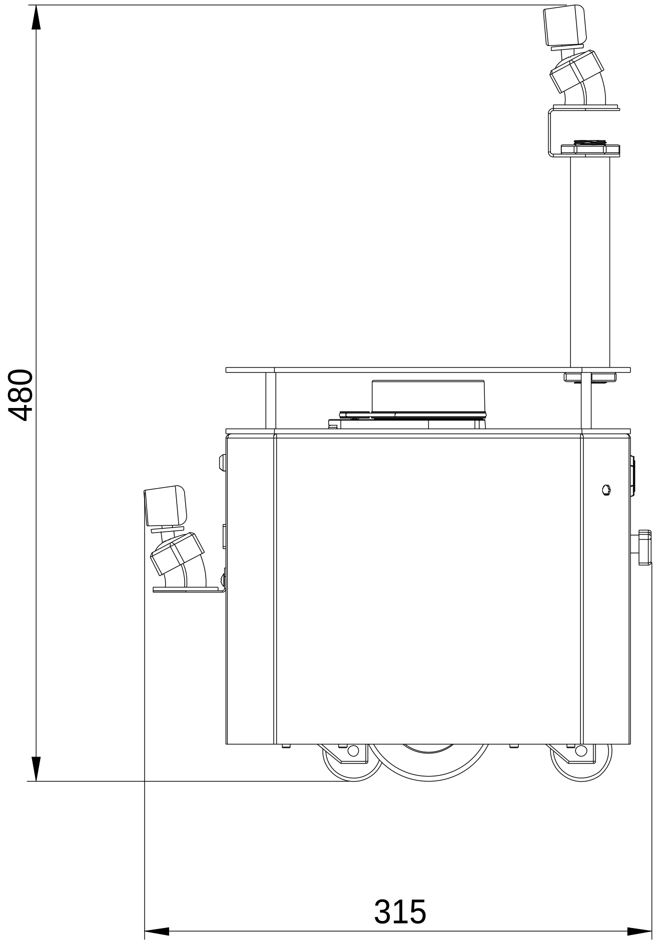


Figure 4: Side View Dimensions (with Accessories)

2.7. System Layout

2.7.1. Front View Layout (No Accessories)

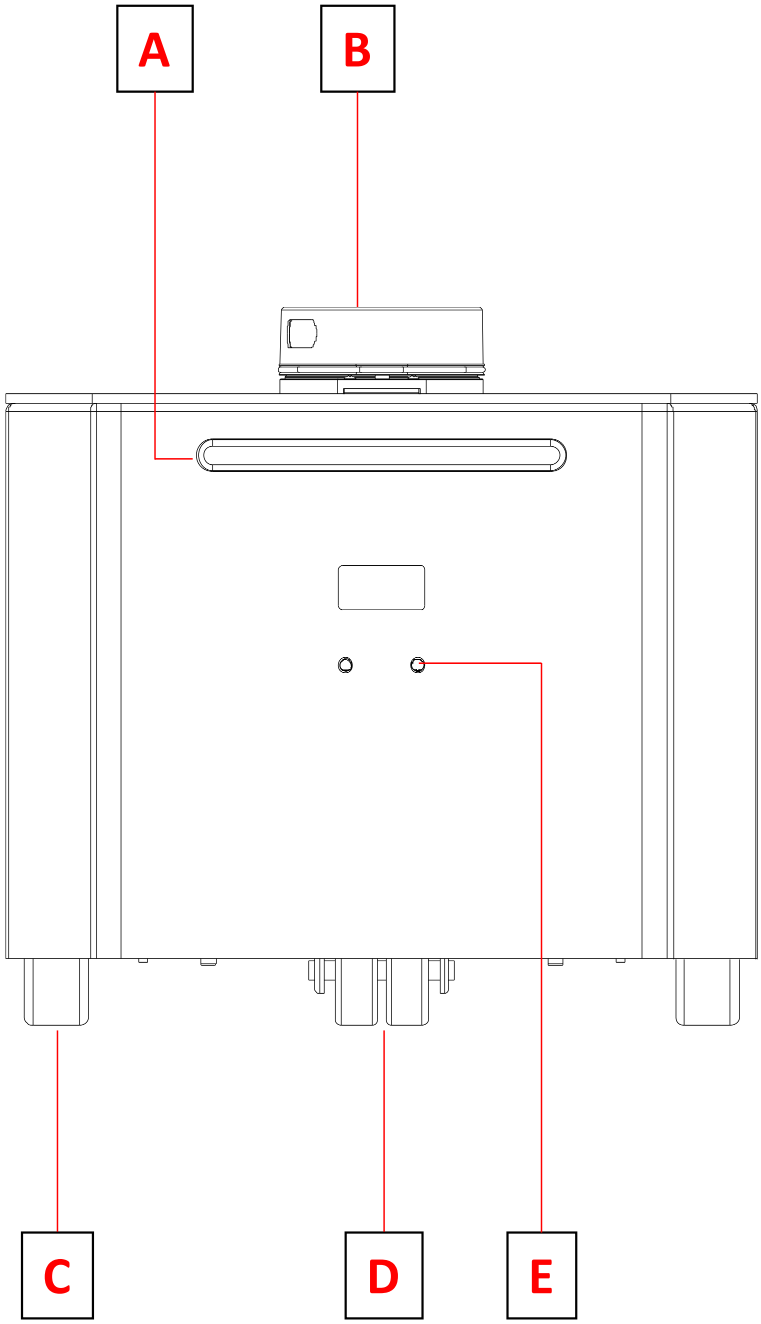


Figure 5: Front View Layout (No Accessories)

A	Status LED bar	**D**	Wheel
B	LIDAR component	**E**	Screw holes for front camera bracket
C	Wheel	**-**	-

2.7.2. Front View Layout (with Accessories)

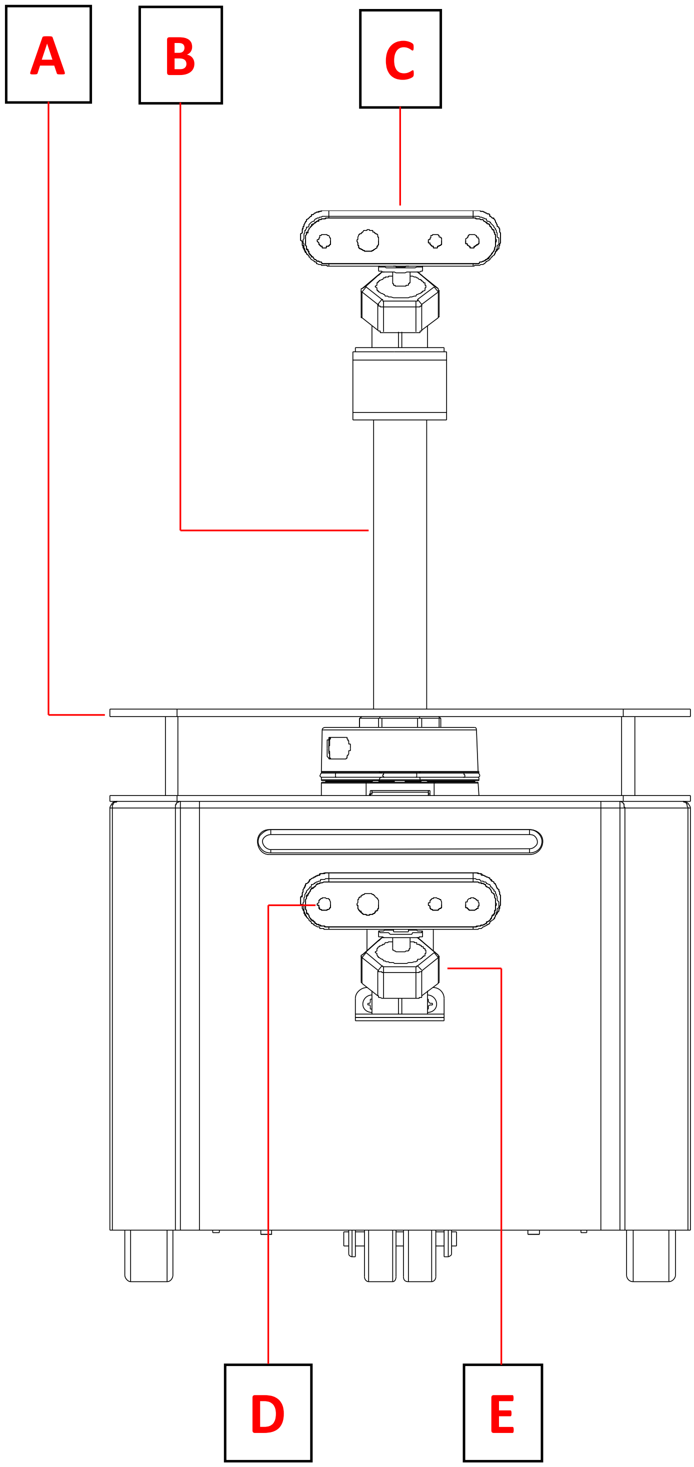


Figure 6: Front View Layout (with Accessories)

A	Support plate	**D**	RealSense camera
B	Top-side stand-off	**E**	Front bracket for RealSense camera
C	RealSense camera	**-**	-

2.7.3. Rear View Layout

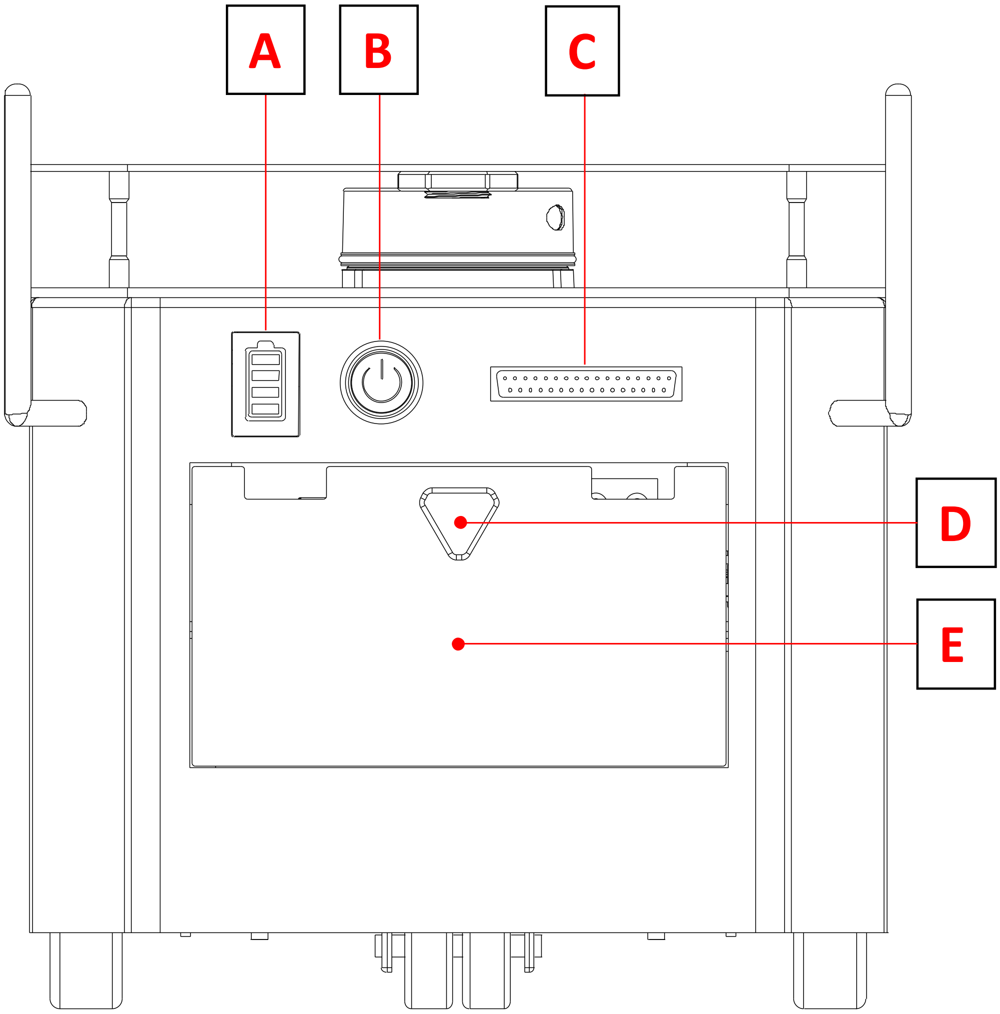


Figure 7: Rear View Layout

A	Battery status indicator	**D**	Rear cover knob
B	Power button	**E**	Rear cover
C	GPIO connector	**-**	-

Note: The rear cover is attached to the chassis by two magnetic fasteners. To access the rear panel I/O, remove the rear cover by grasping the knob and gently pulling the cover away from the chassis.

For details on I/O connectors and internal components, refer to the AmITX-SL-G User's Manual, downloadable from the ADLINK website.

2.7.4. Rear View I/O Layout (Rear Cover Removed)

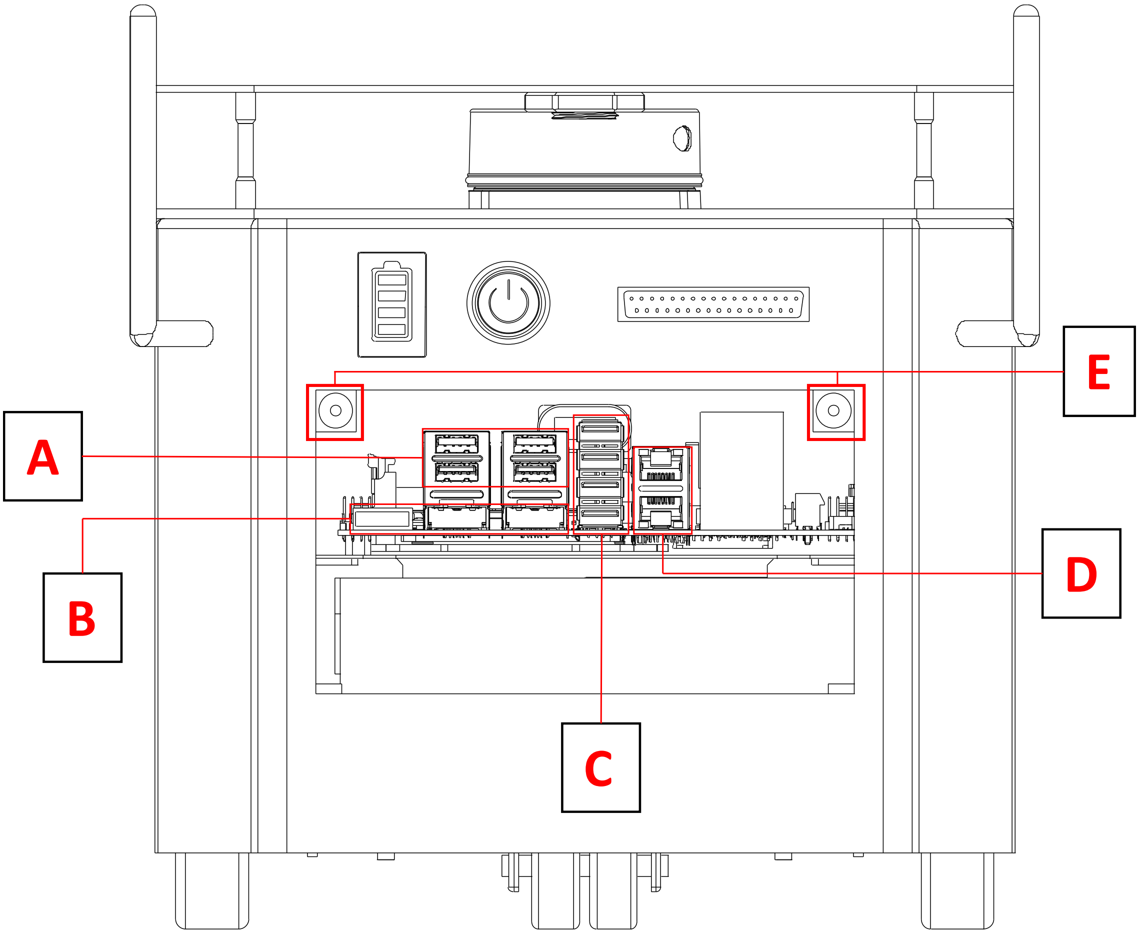


Figure 8: Rear View I/O Layout (Rear Cover Not Attached)

A	USB 3.0 ports (4x)	**D**	LAN ports (2x)
B	DisplayPorts (3x)	**E**	Magnetic fasteners for rear cover
C	USB 2.0 ports (4x)	**-**	-

3. Hardware Accessory Installation

This section describes how to install the following accessories on the NeuronBot:

- Support plate
- Intel® RealSense™ Depth Camera D435

  Note: You can install up to two cameras according to your needs.

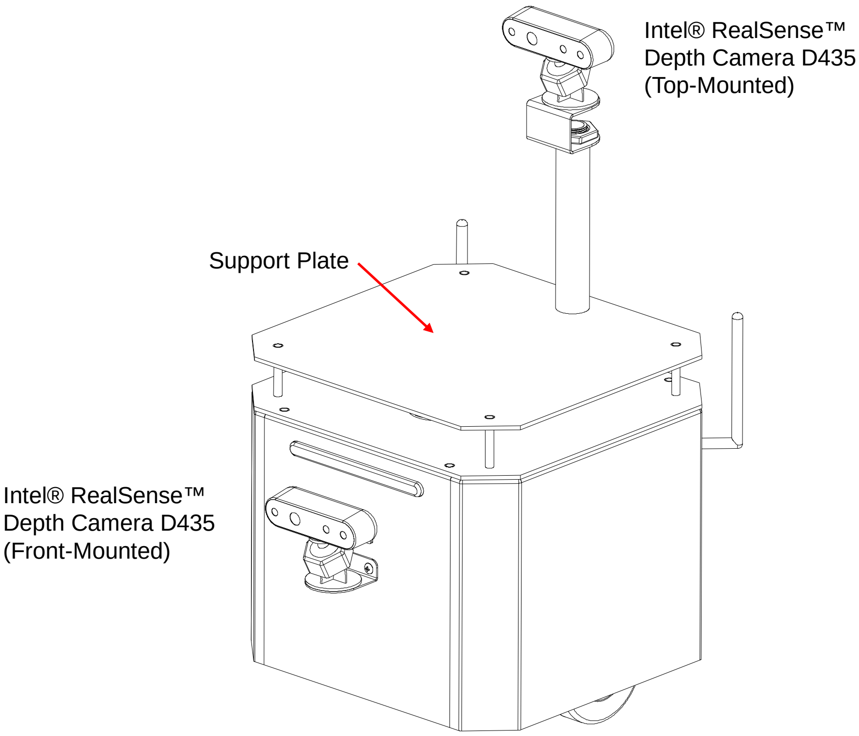
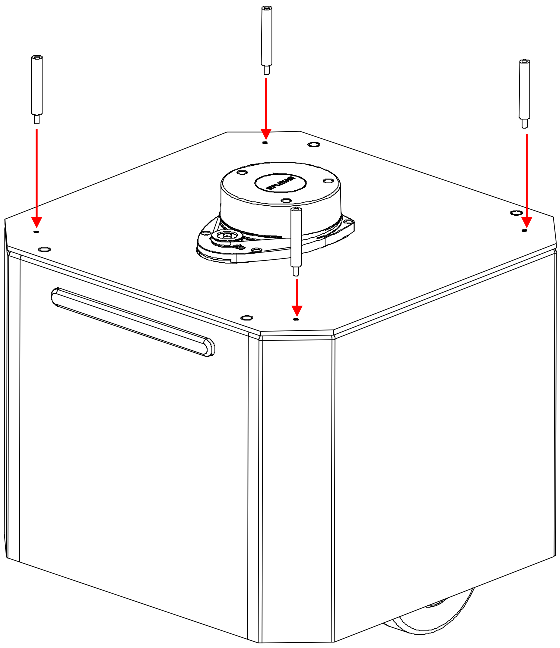


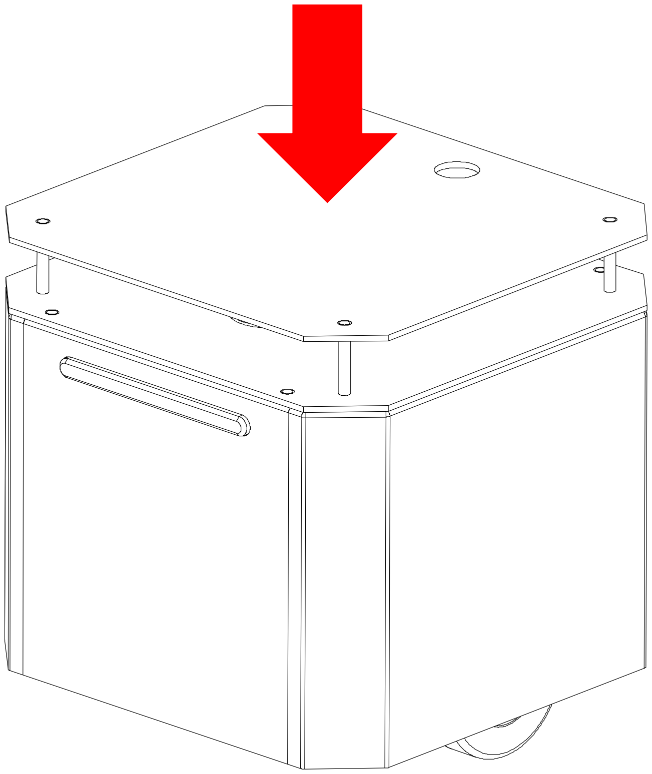
Figure 9: Isometric View with Accessories

3.1. Support Plate Installation

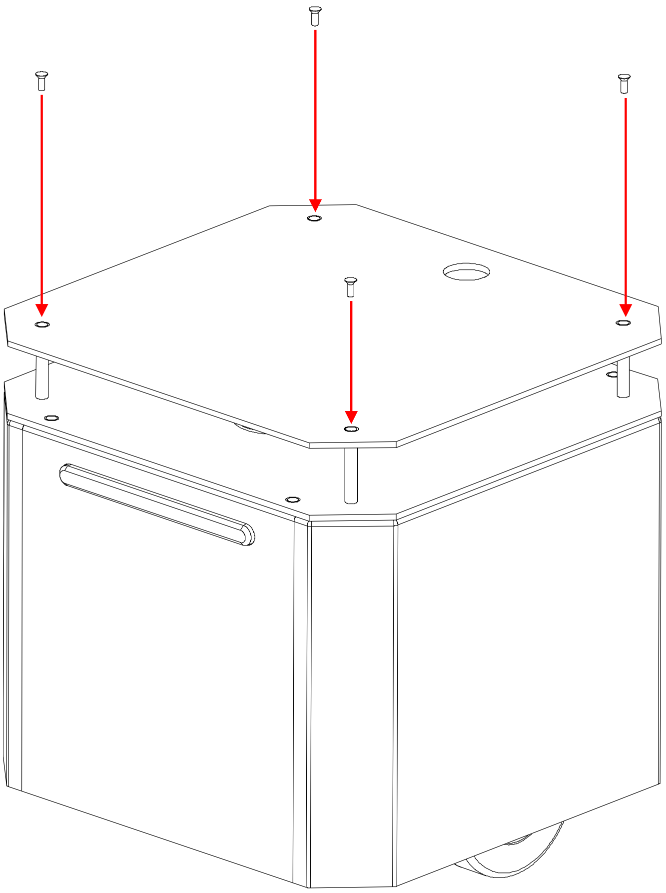
1. Attach the support plate stand-offs to the NeuronBot.



2. Align the holes on the support plate with the stand-offs and place the support plate on top of the stand-offs.

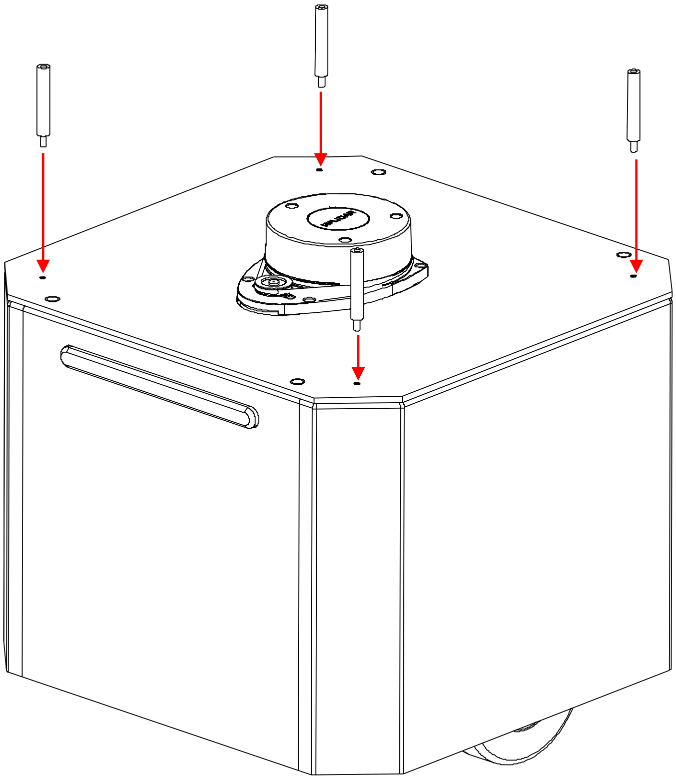


3. Attach the screws to the support plate.

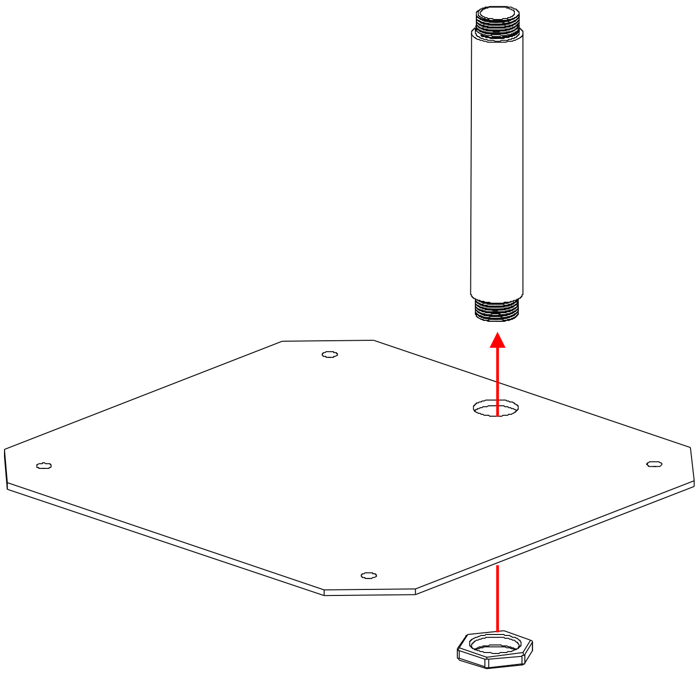


3.2. Top Camera Installation

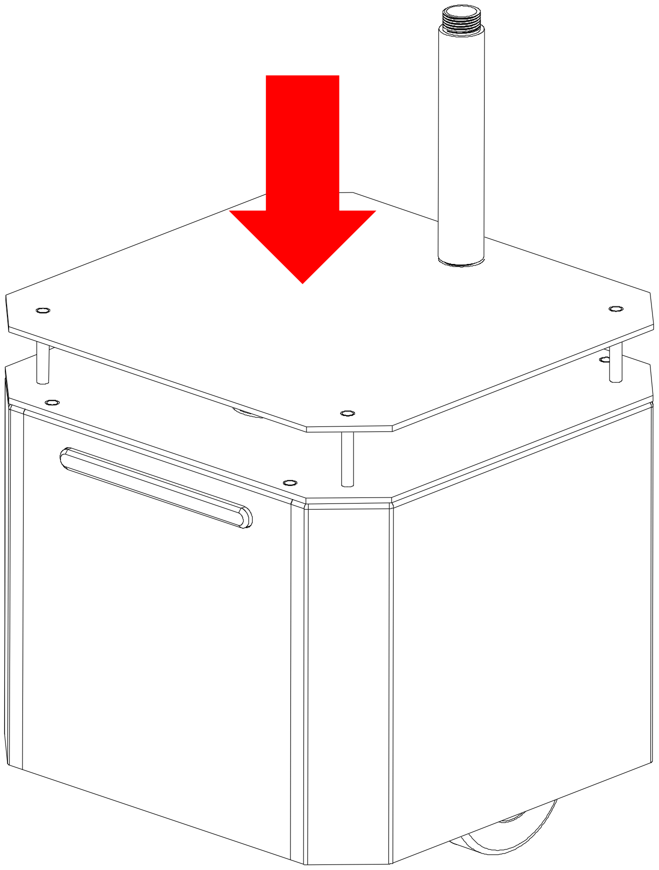
1. Attach the support plate to the NeuronBot.
    a. Attach the support plate standoffs to the NeuronBot.



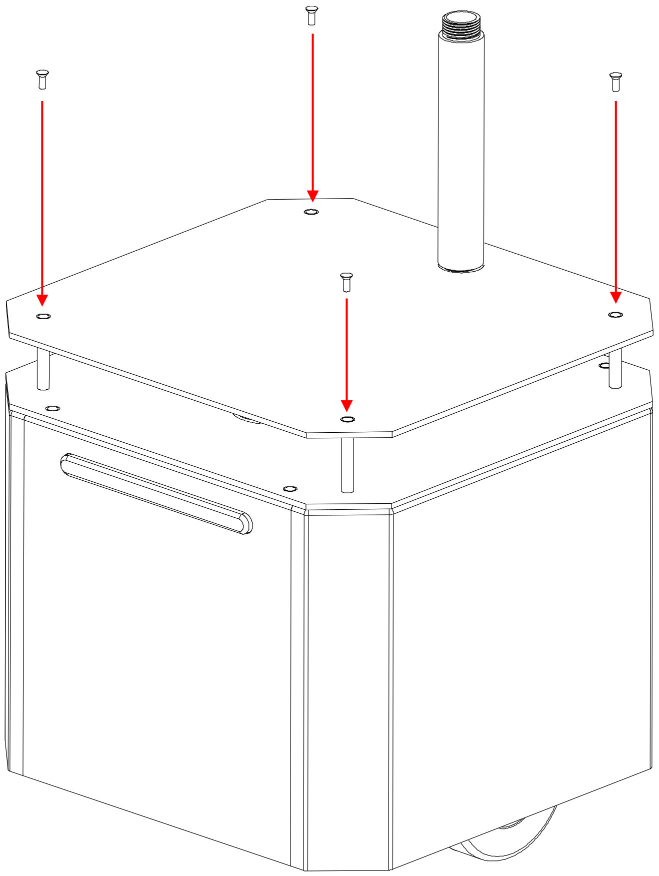
    b. Attach the large, top-side stand-off to the support plate.



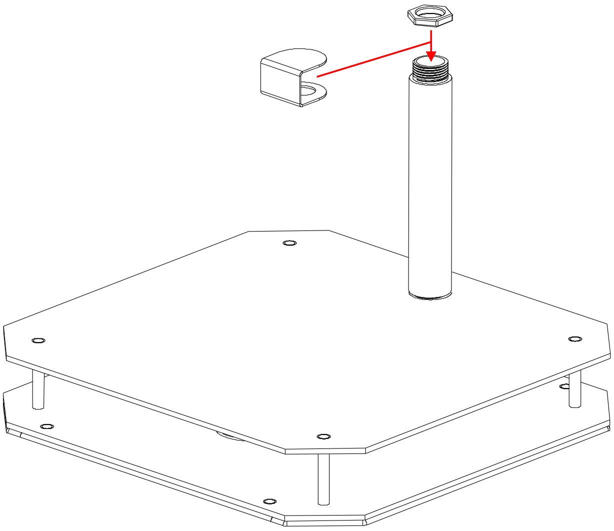
c. Place the attached stand-off and support plate on the NeuronBot.



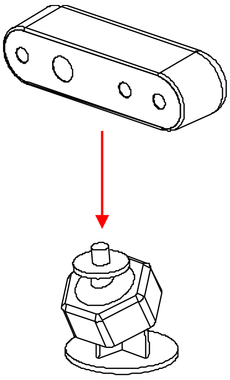
d. Align the holes on the support plate with the stand-offs and attach the screws.



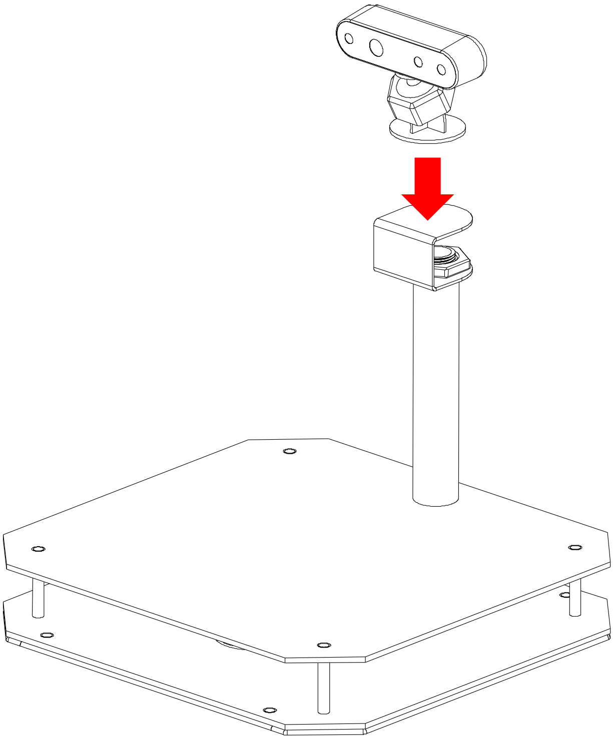
2. Attach the top bracket to the large, top-side stand-off.



3. Assemble the camera by screwing the camera head onto the camera base.

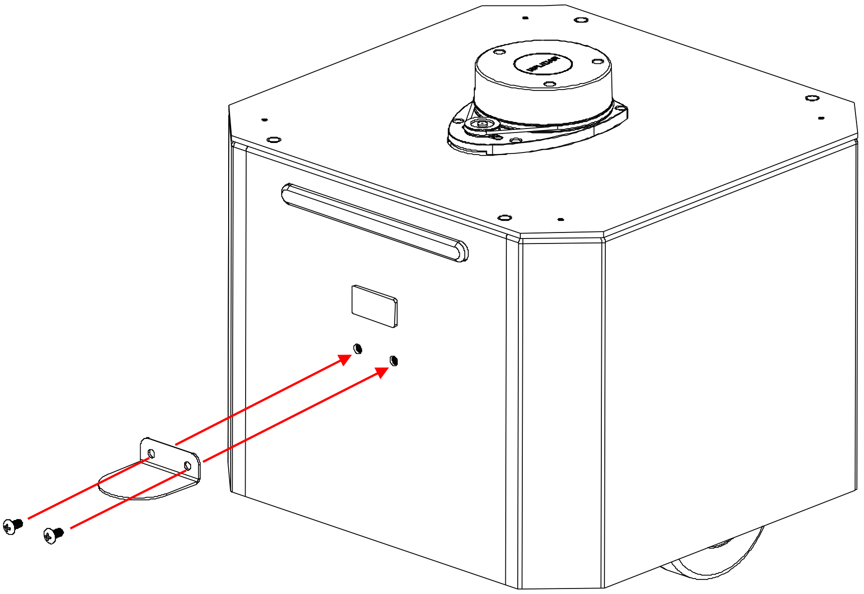


4. Attach the assembed camera to the top bracket using the adhesive sticker provided with the NeuronBot.

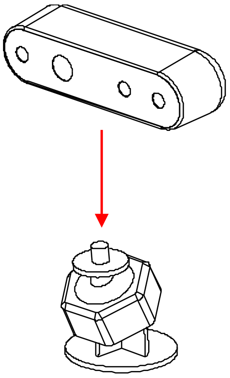


3.3. Front Camera Installation

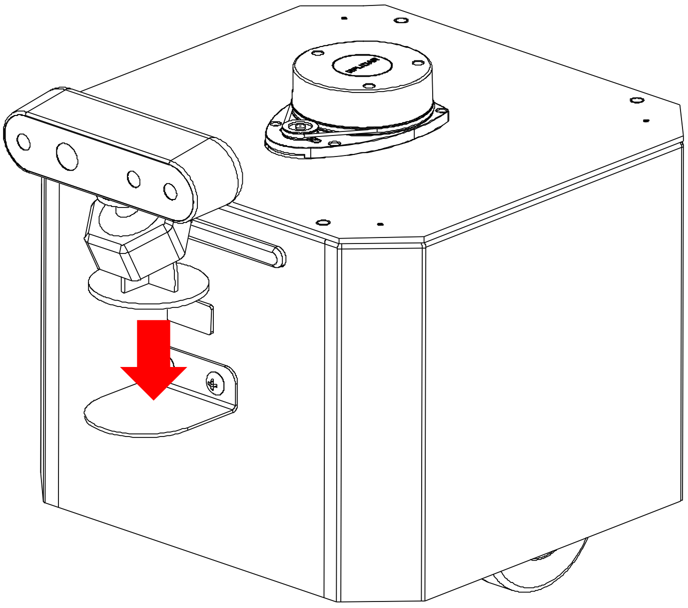
1. Attach the front bracket to the NeuronBot.



2. Assemble the camera by screwing the camera head onto the camera base.

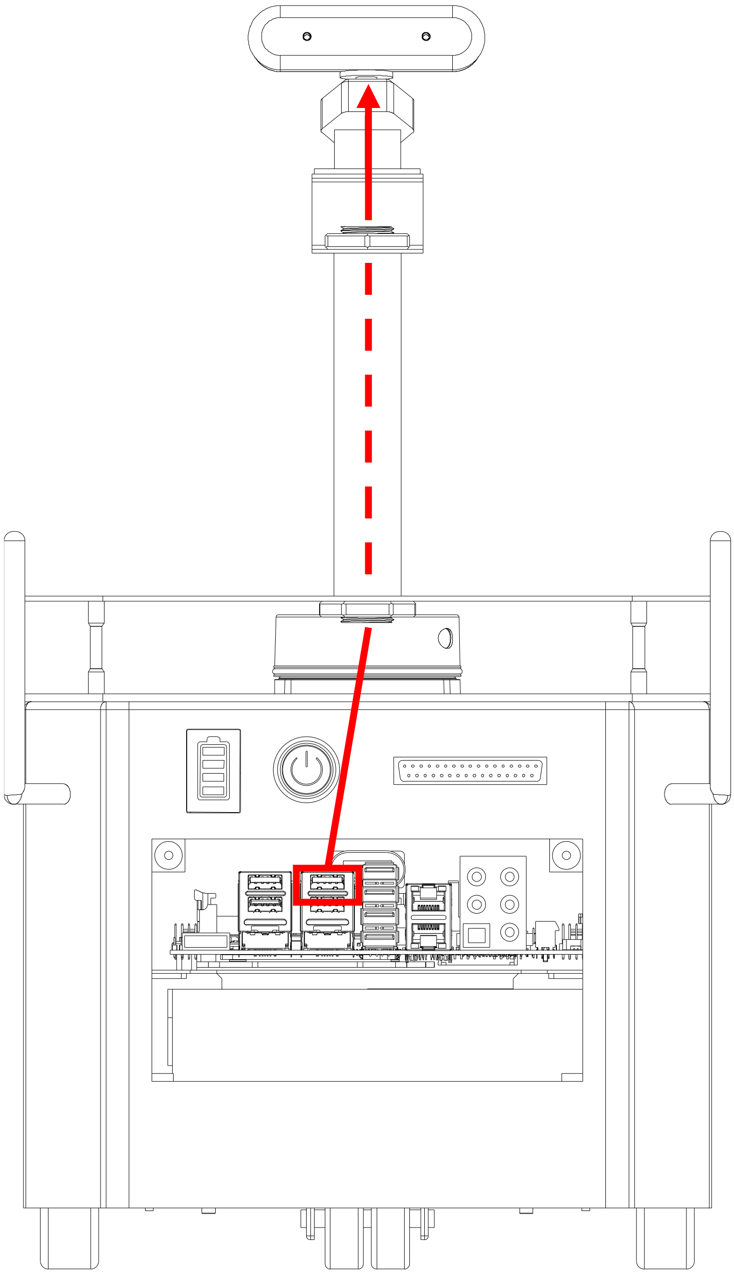


3. Attach the assembled camera to the front bracket using the adhesive sticker provided with the NeuronBot.



3.4. Top Camera Cable Routing

1. Connect a USB cable to a USB 3.0 port on the rear I/O panel (see Rear View Layout on page 14).
2. Route the cable through the top-side stand-off, from the bottom of the support plate to the top bracket, and connect the USB cable to the top camera.

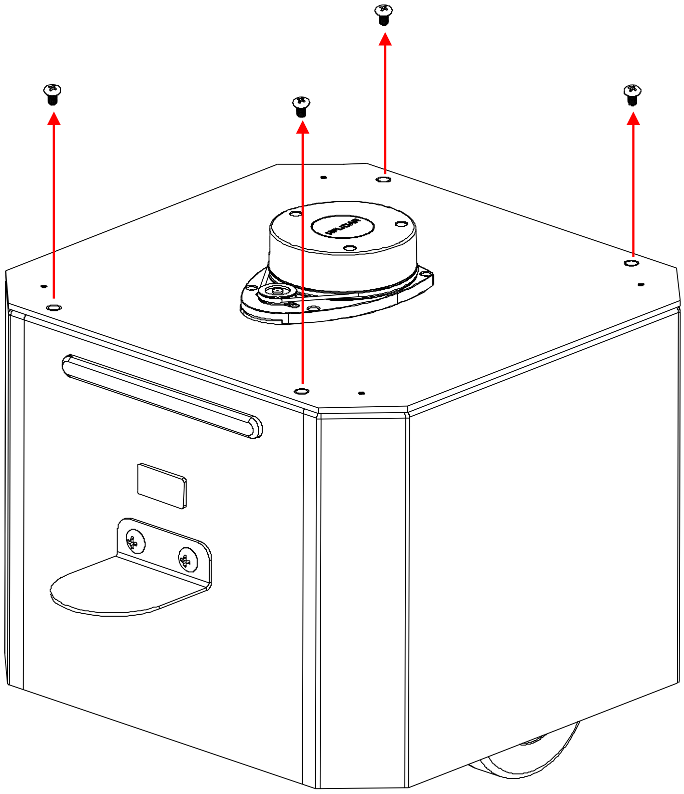


3.5. Front Camera Cable Routing

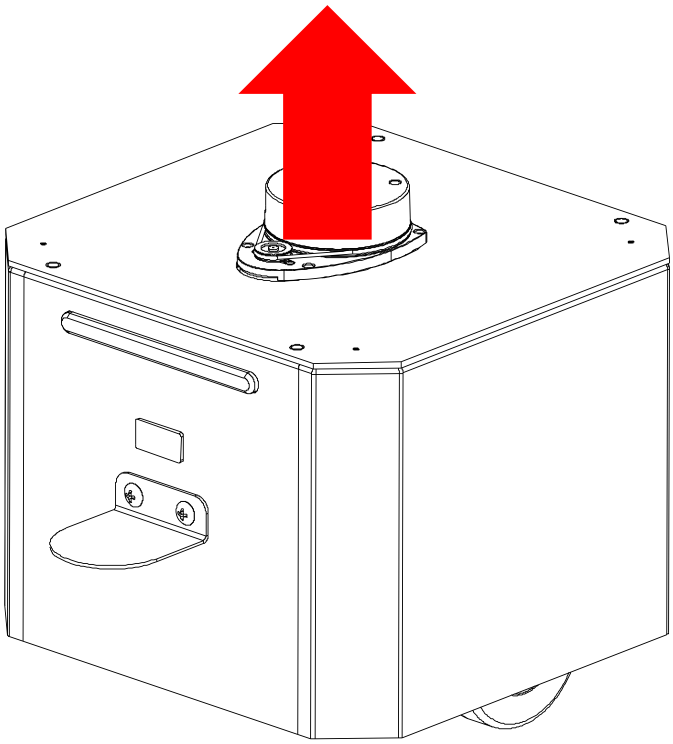
1. Remove the top cover from the chassis.

Note: Ensure that the LiDAR module is disconnected from the I/O before removing the top cover.

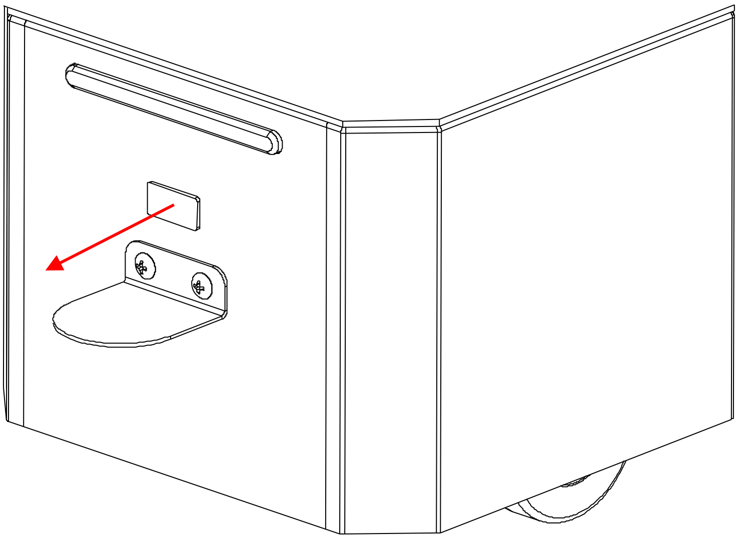
a. Remove the screws that secure the top cover to the chassis.



b. Lift the cover off.

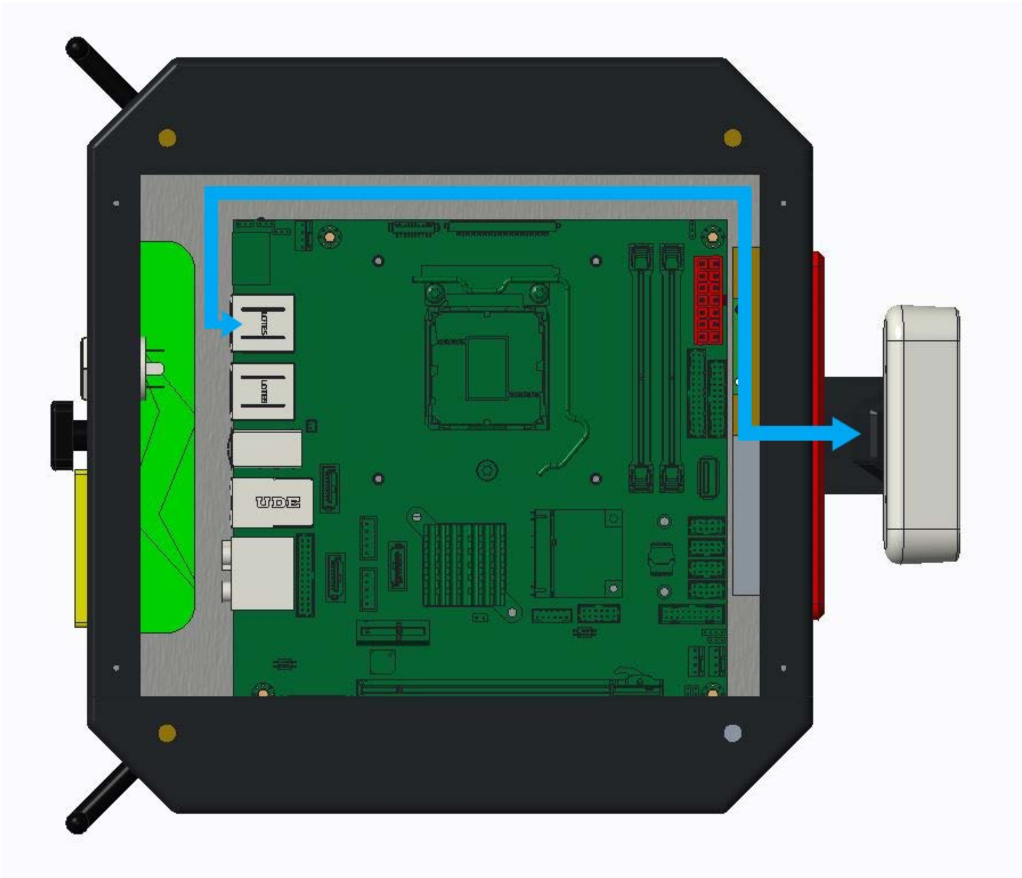


2. Remove the plug from the front opening of the chassis.



3. Connect a USB cable to a USB 3.0 port on the rear I/O panel (see Rear View I/O Layout on page 15).

4. Route the cable to the front of the chassis internally and connect the cable to the front-mounted camera.



Note: When connecting the RealSense camera, do not allow the connector cable to pass over the main board. As shown in the image above, ensure that you run the cable along the interior edge of the chassis between the USB port and the front opening of the chassis.

4. Controls and I/O

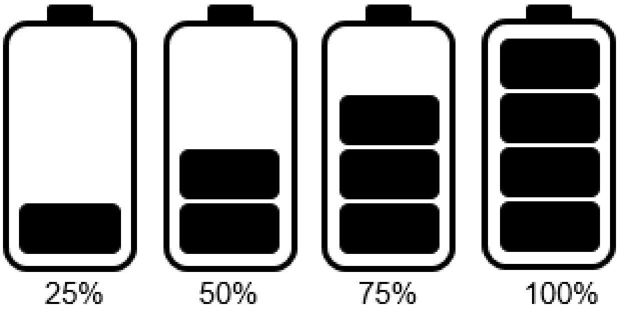
4.1. Power Button

The power button is a non-latched push button located on the rear side of the NeuronBot (see Rear View Layout on page 14). The system powers on when the button is pressed and the LED status bar lights (see Front View Layout on page 12). When depressed, the power button lights white.

If the system hangs, depressing the power button for five seconds turns the system off completely.

4.2. Battery Status Indicator

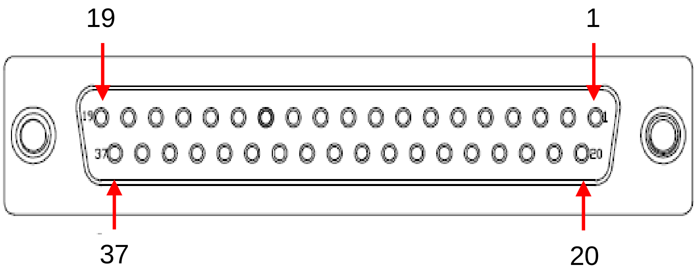
The battery status indicator is located on the rear side of the NeuronBot (see Rear View Layout on page 14). The indicator displays the battery charge percentage as follows:



Note: Ensure that you charge the battery when the charge drops below 25%.

4.3. GPIO Pinouts

The GPIO connector is located on the rear side of the NeuronBot (see Rear View Layout on page 14). The NeuronBot connects GPIO, I2C, and GND pins from the internal computer to the external D-Sub connector.



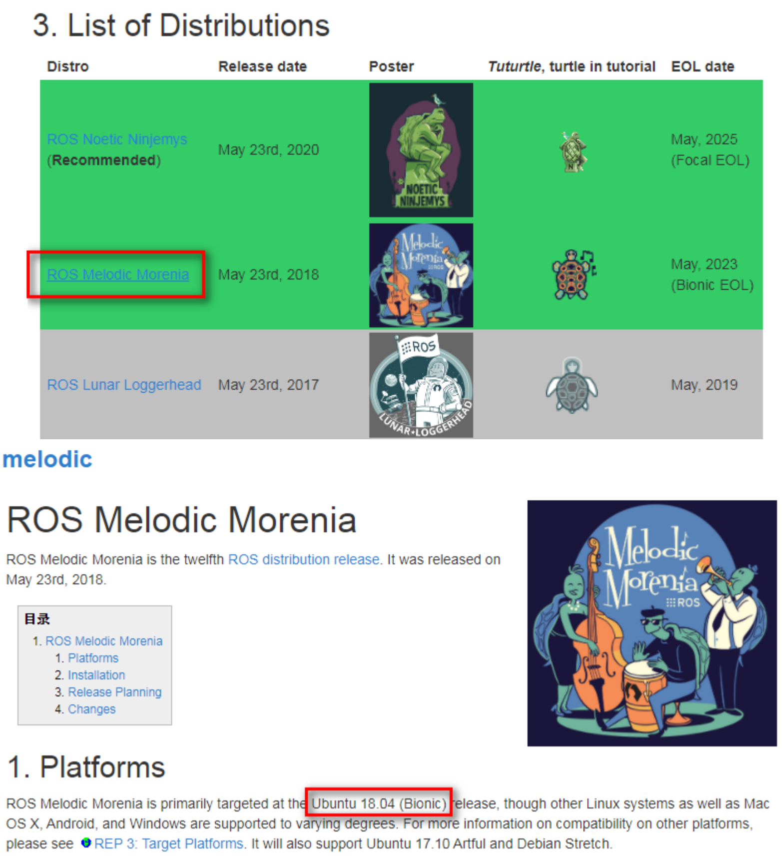
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	GND	9	GPIO3	17	NC	25	NC	33	NC
2	I2CC	10	GPIO4	18	NC	26	NC	34	NC
3	I2CD	11	GPIO5	19	NC	27	NC	35	NC
4	GND	12	GPIO6	20	NC	28	NC	36	NC
5	GND	13	GPIO7	21	NC	29	NC	37	NC
6	GPIO0	14	GPIO8	22	NC	30	NC	-	-
7	GPIO1	15	GPIO9	23	NC	31	NC	-	-
8	GPIO2	16	GND	24	NC	32	NC	-	-

5. Operating Instructions

5.1. Getting Started

Before installation, you must create an Ubuntu USB installation stick on a Windows PC. To install Ubuntu and ROS on the NeuronBot, you need to connect a USB keyboard, mouse, and monitor to the DisplayPort connector. ROS has many different distributions, each requiring a different Ubuntu version. For example, if you want to use ROS 1 Kinetic, you have to install Ubuntu 16.04. For ROS 1 Melodic, Ubuntu 18.04 is required. Confirm the ROS distribution and required Ubuntu version before starting the development of your ROS application.

Refer to the ROS wiki (http://wiki.ros.org/Distributions) to find the right Ubuntu version for your ROS 1 application:



3. List of Distributions

Distro	Release date	Poster	*Tuturtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019

melodic

ROS Melodic Morenia

ROS Melodic Morenia is the twelfth ROS distribution release. It was released on May 23rd, 2018.

目录
1. ROS Melodic Morenia
1. Platforms
2. Installation
3. Release Planning
4. Changes

1. Platforms

ROS Melodic Morenia is primarily targeted at the Ubuntu 18.04 (Bionic) release, though other Linux systems as well as Mac OS X, Android, and Windows are supported to varying degrees. For more information on compatibility on other platforms, please see REP 3: Target Platforms. It will also support Ubuntu 17.10 Artful and Debian Stretch.

You can also check https://index.ros.org/doc/ros2/Releases/ to to find the right Ubuntu version for your ROS 2 application.

5.1.1. Ubuntu Installation

Follow the steps below to install Ubuntu on the NeuronBot.

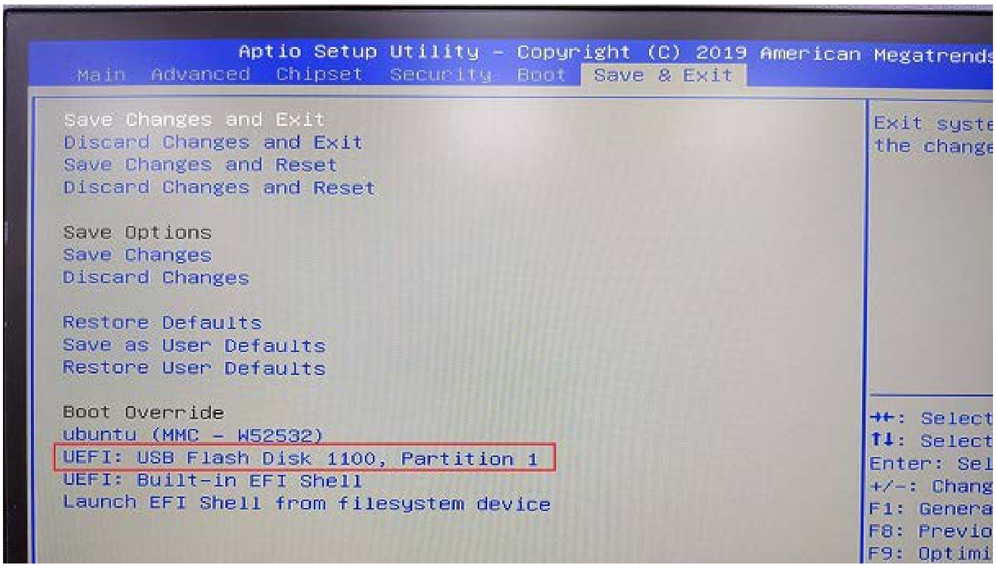
1. Download an Ubuntu desktop image to your Windows PC:
   a. Get ubuntu-18.04.4-desktop-amd64.iso from https://releases.ubuntu.com/18.04/
   b. Or, get ubuntu-20.04-desktop-amd64.iso from https://releases.ubuntu.com/20.04/

2. Create a bootable Ubuntu USB installation stick on your Windows PC. Refer to the following tutorial: https://ubuntu.com/tutorials/create-a-usb-stick-on-windows

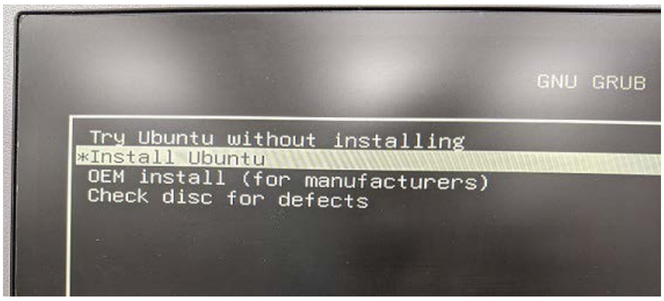
3. Insert the USB stick into the NeuronBot and power it on.

4. When the ADLINK logo appears onscreen, press the Delete key to enter the BIOS menu.

5. In the BIOS menu, select the USB stick as the boot device, and then press the Enter key to boot from the device.



6. Select “Install Ubuntu” and press the Enter key to start the installation.



7. When the installation wizard appears, follow the instructions on the Ubuntu website to complete installation: https://ubuntu.com/tutorials/install-ubuntu-desktop#5-prepare-to-install-ubuntu

5.1.2. ROS Installation

After installing Ubuntu on the NeuronBot, follow the instructions below to set up the ROS environment.

5.1.2.1. ROS Distributions

Please visit the ROS official website to get the latest installation guide for both ROS 1 and ROS 2.

- For Ubuntu 18.04 Desktop:

  ROS 1 Melodic – http://wiki.ros.org/melodic/Installation/Ubuntu

  ROS 2 Dashing – https://index.ros.org/doc/ros2/Installation/Dashing/Linux-Install-Debians/

  ROS 2 Eloquent – https://index.ros.org/doc/ros2/Installation/Eloquent/Linux-Install-Debians/

- For Ubuntu 20.04 Desktop:

  ROS 1 Noetic – http://wiki.ros.org/noetic/Installation/Ubuntu

  ROS 2 Foxy – https://index.ros.org/doc/ros2/Installation/Foxy/Linux-Install-Debians/

Because different ROS distributions are installed in different paths (e.g., /opt/ros/melodic, /opt/ros/dashing, and /opt/ros/eloquent), installing different ROS distributions on the same disk will not cause any issues. This means that you can use one of the installed ROS distributions as long as you “**source**” the specific setup.bash. For example, if you have installed Melodic, Dashing, and Eloquent, and you want to run ROS applications with Eloquent, you can start with the following:

```
source /opt/ros/eloquent/setup.bash
```

5.1.2.2. Verifying the ROS 1 Installation

To verify the ROS 1 installation, you need to open three terminals and execute the following commands.

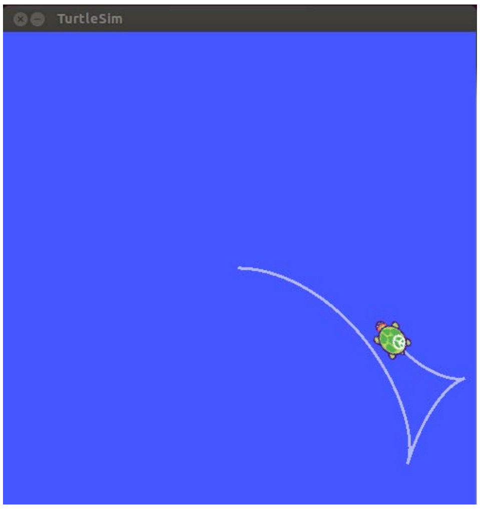
Note: <YOUR_ROS_DISTRO> here stands for the ROS distribution (e.g., /opt/ros/melodic/setup.bash for ROS 1 Melodic; /opt/ros/dashing/setup.bash for ROS 2 Dashing.

- Terminal 1: Load the ROS 1 environment and execute **roscore** (ROS Master).

```
source /opt/ros/<YOUR_ROS_DISTRO>/setup.bash
roscore
```

- Terminal 2: Load the ROS 1 environment and execute the turtlesim program.

```
source /opt/ros/<YOUR_ROS_DISTRO>/setup.bash
rosrun turtlesim turtlesim_node
```



- Terminal 3: Load the ROS 1 environment and use the arrow keys to tele-opreate the turtle.

```
source /opt/ros/<YOUR_ROS_DISTRO>/setup.bash
rosrun turtlesim turtle_teleop_key
```

5.1.2.3. Verifying the ROS 2 Installation

To verify the ROS 2 installation, you need to open two terminals and execute the following commands.

- Terminal 1: Load the ROS 2 environment and execute the turtlesim program.

```
source /opt/ros/<YOUR_ROS_DISTRO>/setup.bash
ros2 run turtlesim turtlesim_node
```

If you see the error "*Package 'turtlesim' not found*", then you have not installed turtlesim for ROS 2. You can manually install turtlesim by executing the following command:

```
sudo apt install ros-<YOUR_ROS_DISTRO>-turtlesim
```

After installation, you can execute **turtlesim_node** again to check that the turtlesim program opened successfully.

- Terminal 2: Load the ROS 2 environment and use the arrow keys to tele-opreate the turtle

```
source /opt/ros/<YOUR_ROS_DISTRO>/setup.bash
ros2 run turtlesim turtle teleop key
```

5.1.2.4. Verifying the ROS Bridge

ROS 1 and ROS 2 can share the same topics if their message types are the same. To bridge ROS 1 and ROS 2, you have to install **ros1_bridge** for your ROS 2 distribution. Execute the following command to install **ros1_bridge**:

```
sudo apt install ros-<YOUR_ROS_DISTRO>-ros1-bridge
```

To bridge **ROS 1 Listener** and **ROS 2 Talker**, enter the commands in four terminals. For example:

- Terminal 1: Execute ROS 1 roscore

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
roscore
```

- Terminal 2: Execute ROS 1 Listener

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
rosrun roscpp_tutorials listener
```

- Terminal 3: Execute ROS 2 Talker

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
ros2 run demo_nodes_cpp talker
```

- Terminal 4: Execute ROS Bridge

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
ros2 run ros1_bridge dynamic_bridge
```

If done correctly, ROS 1 Listener will successfully receive the messages from ROS 2 Talker.

5.1.3. Linux Tools Installation

After the ROS environment is verified, it is recommended that you install the following tools for help developing ROS applications.

```
sudo apt update && sudo apt install -y \
  build-essential \
  cmake \
  git \
  libbullet-dev \
  python3-colcon-common-extensions \
  python3-flake8 \
  python3-pip \
  python3-pytest-cov \
  python3-rosdep \
  python3-setuptools \
  python3-vcstool \
  openssh-server \
  wget \
  curl \
  byobu
```

5.1.4. Neuron Startup Menu

You may notice that when you execute ROS commands, you have to source the ROS environment for each terminal. This is not an ideal solution, so ADLINK provides an easy way to load different development environments using the **Neuron Startup Menu**.

UNIX based systems often store parameters in a format of "shell variables", or more commonly, "environmental variables". Scripts and programs constantly load these variables either as execution parameters or for internal usage. Configuring environmental variables can be tedious, especially when there are conflicting settings between ROS/ROS 2, Python 2/Python 3, OpenCV 2/OpenCV 3, etc. The Neuron Startup Menu prepares the environment for you.

After installing the Neuron Startup Menu, a menu appears at each terminal startup, allowing you to select ROS/ROS 2/ROS Bridge. This way, you will no longer need to set ROS environmental variables every time you open the terminal.

Note: If you want to use the ADLINK Neuron Startup Menu, you should remove the line "`source /opt/ros/<ROS_DISTROS>/setup.bash`" from ~/.bashrc, which you may have added when installing ROS.

For more information, please visit our GitHub: https://github.com/Adlink-ROS/ros_menu

5.1.4.1. Menu Installation

Clone the repository and execute the install command:

```
cd ~
sh -c "MENU_VERSION=v1.4.0 $(curl -fsSL
https://raw.githubusercontent.com/Adlink-ROS/ros_menu/master/scripts/setup.sh)"
```

5.1.4.2. Menu Usage

After installation, the terminal will display the following menu:

```
************ Neuron Startup Menu for ROS *************
* Usage: To set ROS env to be auto-loaded, please    *
*        assign ros_option in ros_menu/config.yaml   *
******************************************************
0) Do nothing
1) ROS 1 melodic
2) ROS 2 dashing
3) ROS2/ROS1_bridge
Please choose an option:
```

The menu items perform the following actions, respectively:

- **Do nothing**:
  - Does not set up any environment.
- **ROS 1 melodic**:
  - Sets up the ROS 1 environment.
  - Sets the ROS_IP and ROS_MASTER_URI, which is your host IP.
- **ROS 2 dashing**:
  - Sets up the ROS 2 environment.
  - Loads the DDS settings and select the DDS you want to use.
- **ROS2/ROS1_bridge**:
  - Sets up the ROS bridge environment.
  - Runs ROS bridge automatically.

After selecting an option, you can view the settings via environmental variables:

- Check the ROS version (1 or 2):

```
echo $ROS_VERSION
```

- Check the ROS distribution:

```
echo $ROS_DISTRO
```

- Check the DDS implementation (only for ROS 2):

```
echo $RMW_IMPLEMENTATION
```

5.1.4.3. Menu Configuration

You can configure the menu by modifying ~/.ros_menu/config.yaml. You can perform the following actions:

- **Enable the menu**:

  menu_enable: "true" to enable the menu, “false” to disable the menu

- **Set the default ROS option**:

  ros_option: "menu" to show all the options of the menu.

  Note: You can also set an option number to this variable and the menu will automatically apply the option every time you open the terminal.

- **Modify ROS options separately:**

  The following parameters are needed to create a new option for the menu:

  ROS 1:

  ROS_version: 1

  distro_name: the name of the ROS 1 version you are using

  ros1_path: the path where ROS 1 is located

  master_ip: sets the IP address of the master if the master isn't on the current computer

  cmds: source your ROS 1 workspace here

  ROS 2:

  ROS_version: 2

  distro_name: the name of the ROS 2 version you are using

  ros2_path: the path where ROS 2 is located

  domain_id: sets the Domain ID for DDS communication. Keep this empty to use *$default_ros_domain_id(30)*

  cmds: source your ROS 2 workspace here. Remarks: `source_plugin dds_bashrc` is necessary every time you use ROS 2

  ROS2/ROS1_bridge:

  ROS_version: bridge

  ros1_version_name: the name of the ROS 1 version you are using

  ros2_version_name: the name of the ROS 2 version you are using

  ros1_path: the path where ROS 1 is located

  ros2_path: the path where ROS 2 is located

  master_ip: sets the IP address of the master if master isn't on the current computer

  domain_id: sest the Domain ID for DDS communication. Keep empty to use *$default_ros_domain_id(30)*

  cmds: any command you want to run every time using ROS Bridge. Remarks: `source_plugin dds_bashrc` and `ros2 run ros1_bridge dynamic_bridge --bridge-all-topics` is necessary every time using ROS Bridge

5.1.4.4. Menu Upgrade

You can upgrade the Neuron Startup Menu by executing the following command:

```
ros_menu_upgrade
```

The new version will load the next time you open the terminal.

5.1.4.5. Menu Uninstallation

You can uninstall the Neuron Startup Menu by executing the following command:

```
ros_menu_uninstall
```

Note: Remember to remove the Neuron Startup Menu configuration in ~/.bashrc.

5.1.5. NeuronBot Setup

NeuronBot source code is available on our GitHub. Visit the following webpage to get the latest updates:
https://github.com/Adlink-ROS/neuronbot2

5.1.5.1. Getting the NeuronBot Software

For ROS 1 Melodic:

1. Create a workspace:

```
mkdir -p ~/neuronbot2_ros1_ws/src
cd ~/neuronbot2_ros1_ws/src
```

2. Git-clone the package with a melodic-devel branch:

```
git clone https://github.com/Adlink-ROS/neuronbot2.git -b melodic-devel
```

3. Install dependencies:

```
cd ~/neuronbot2_ros1_ws/
rosdep update
rosdep install --from-paths src --ignore-src -r -y --rosdistro melodic
```

For ROS 2 Eloquent:

1. Create a workspace:

```
mkdir -p ~/neuronbot2_ros2_ws/src
cd ~/neuronbot2_ros2_ws/
```

2. Get the latest packages with an eloquent-devel branch:

```
wget https://raw.githubusercontent.com/Adlink-
ROS/neuronbot2_ros2.repos/eloquent-devel/neuronbot2_ros2.repos

vcs import src < neuronbot2_ros2.repos
```

3. Install dependencies:

```
cd ~/neuronbot2_ros2_ws/
source /opt/ros/eloquent/setup.bash
rosdep update
rosdep install --from-paths src --ignore-src -r -y --rosdistro eloquent
```

5.1.5.2. LiDAR and TTY Initialization

Find **neuronbot2_init.sh** and run the following script with root permission:

```
cd ~/neuronbot2_ros1_ws/src/neuronbot2/neuronbot2_tools/neuronbot2_init/
# or
cd ~/neuronbot2_ros2_ws/src/neuronbot2/neuronbot2_tools/neuronbot2_init/

sudo ./neuronbot2 init.sh
```

5.1.5.3. NeuronBot Installation

For ROS 1 Melodic:

```
source /opt/ros/melodic/setup.bash
cd ~/neuronbot2_ros1_ws/
catkin_make
```

Use **catkin_make** to compile NeuronBot under the ROS 1 Melodic environment.

After successful compilation, the NeuronBot environment is created. Afterward, if you want to run NeuronBot applications, you have to "source" **setup.bash** in ROS 1 Melodic and the NeuronBot workspace.

```
source /opt/ros/melodic/setup.bash
cd ~/neuronbot2_ros1_ws/
source devel/setup.bash
```

```
source /opt/ros/eloquent/setup.bash
cd ~/neuronbot2_ros2_ws/
colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TYPE=Release
source ~/neuronbot2_ros2_ws/install/local_setup.bash
```

For ROS 2 Eloquent:

Use **colcon** to compile NeuronBot under the ROS 2 Eloquent environment.

After successful compilation, the NeuronBot environment is created. Afterward, if you want to run NeuronBot applications, you have to "source" **setup.bash** in ROS 2 Eloquent and **local_setup.bash** in the NeuronBot workspace.

```
source /opt/ros/eloquent/setup.bash
cd ~/neuronbot2_ros2_ws
source install/local_setup.bash
```

We highly recommend that you add the above "source" command to the ADLINK Neuron Start Menu so the environment loads automatically.

5.1.5.4. NeuronBot Verification

Go to the self-diagnosis section for instructions on verifying and diagnosing NeuronBot.

5.2. Remote Control and Monitoring

Before teaching ROS to control and monitor NeuronBot, you must remove the keyboard, mouse, and monitor from the NeuronBot. To get started, install Ubuntu and set up the ROS environment on your computer and connect the NeuronBot and computer to the same WiFi router.

Note: We recommend using a portable computer (i.e., a laptop) to remotely control the NeuronBot for added mobility and accessibility.

5.2.1. Wireless Setup

After connecting the NeuronBot and computer to the same WiFi router, you need to obtain the IP address of the NeuronBot and computer to remotely control the NeuronBot.

1. Power on the NeuronBot and connect it to a monitor and keyboard.
2. Connect to the WiFi.
3. Execute the following command to obtain the IP address of the NeuronBot:

```
ip address show
```

The IP address is at the following location:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s31f6: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 84:7b:eb:43:c2:03 brd ff:ff:ff:ff:ff:ff
3: wlp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether e4:b3:18:2b:c3:a2 brd ff:ff:ff:ff:ff:ff
    inet 172.16.202.148/22 brd 172.16.203.255 scope global dynamic noprefixroute wlp1s0
       valid_lft 22934sec preferred_lft 22934sec
    inet6 fe80::1621:3a66:495b:d8b2/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Use the same command to get the IP address of the computer. Take note of these IP addresses for use in the following sections.

Note: NeuronBot has multiple network interfaces because it comes installed with a wireless card (wlp1s0) and wired (enp0s31f6) card. The IP address obtained from the WiFi router must be aligned with the wlpXXX interface.

5.2.2. ROS 1 Remote Control Settings

In ROS 1, the IP address is needed to tell ROS 1 applications (e.g., rviz and rqt) where to find NeuronBot. Therefore, when you want to connect NeuronBot remotely, make sure that ROS_MASTER_URI and ROS_IP are set correctly.

- ROS_MASTER_URI

This is a required setting that tells nodes where they can locate the ROS Master. In this example, NeuronBot is the Master because we will run roscore on the NeuronBot.

- ROS_IP

This tells other nodes where they can locate themselves.

Use the following commands to set up ROS_MASTER_URI and ROS_IP.

- NeuronBot:

```
export ROS_MASTER_URI=http://<IP_OF_NEURONBOT>:11311
```

```
export ROS_IP=<IP_OF_NEURONBOT>
```

- Computer:

```
export ROS_MASTER_URI=http://<IP_OF_NEURONBOT>:11311
export ROS_IP=<IP_OF_LAPTOP>
```

For example, if the IP address of the NeuronBot is 192.168.50.26, and the IP address of the computer is 192.168.50.110, then you should execute the following commands:

- NeuronBot:

```
export ROS_MASTER_URI=http://192.168.50.26:11311
export ROS_IP=192.168.50.26
```

- Computer:

```
export ROS_MASTER_URI=http://192.168.50.26:11311
export ROS_IP=192.168.50.110
```

To make this easier, you can set up the ADLINK Neuron Startup Menu on the NeuronBot and computer. Use the following command to edit the Neuron Startup Menu. Here, we used gedit as as the editor; however, you can use any editor you want.

```
gedit ~/.ros_menu/config.yaml
```

The Master IP addresses of the NeuronBot and computer are the same. Modify the Neuron Startup Menu on both the NeuronBot and computer as follows:

```
Menu:
  ROS 1 melodic:
    option_num: 1
    ROS_version: 1
    distro_name: melodic
    ros1_path: /opt/ros/melodic
    master_ip: 192.168.50.26
    cmds:
    # - source ${HOME}/catkin_ws/devel/setup.${shell}
```

5.2.3. ROS 2 Remote Control Settings

ROS 2 uses DDS as the underlying transport protocol, and DDS supports physical segmentation of the network based on the **Domain ID**. The ROS_DOMAIN_ID helps keep each machine group's ROS 2 nodes from interfering with other groups. The function of ROS_DOMAIN_ID is similar to the principle of a walkie-talkie. Execute the following command to set up the ROS_DOMAIN_ID:

```
export ROS_DOMAIN_ID=<YOUR_DOMAIN_ID>
```

For example, if you choose "30" for the ROS_DOMAIN_ID, enter the following command on both the NeuronBot and the computer:

- NeuronBot:

```
export ROS_DOMAIN_ID=30
```

- Computer:

```
export ROS_DOMAIN_ID=30
```

To make this easier, you can set up the ADLINK Neuron Startup Menu on the NeuronBot and computer. Use the following command to edit the Neuron Startup Menu. Here, we used gedit as the editor; however, you can use any editor you want.

```
gedit ~/.ros_menu/config.yaml
```

The Domain ID of the NeuronBot and computer should be the same for DDS communication. Modify the Neuron Startup Menu on both the NeuronBot and computer as follows:

```
Config:
  menu_enable: true
  ros_option: menu
  default_ros_domain_id: 30
Menu:
  ROS 2 dashing:
    option_num: 2
    ROS_version: 2
    distro_name: dashing
    ros2_path: /opt/ros/dashing
    domain_id: # set if you don't want to use default domain id
    cmds:
    # - source ${HOME}/ros2_ws/install/local_setup.${shell}
      - source_plugin dds_bashrc 1
```

5.2.4. SSH Remote Session

Typically, it is not convenient to connect a mouse, keyboard, and monitor to the NeuronBot while the NeuronBot is moving around. To address this, you can connect to an SSH server, which is a secure shell for network connections. An SSH server will allow you to remotely connect to a NeuronBot for the secure transfer of files or to perform administrative tasks like teleoperating the NeuronBot. First you must install an SSH server on NeuronBot.

- NeuronBot:

```
sudo apt install openssh-server
```

Before you connect to the SSH server, make sure that the NeuronBot and computer are connected to the same WiFi router. Then, execute the following command on the computer to remotely connect to the Neuronbot.

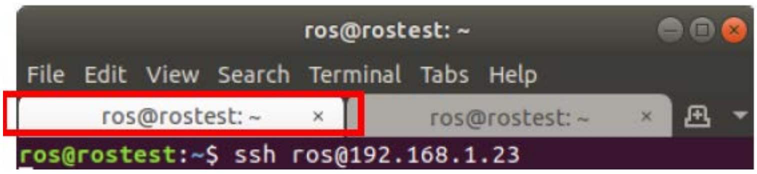
- Computer:

```
ssh -X <USER_NAME>@<IP_ADDRESS>
```

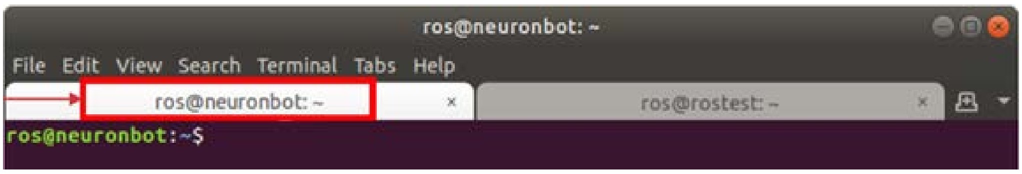
For example, the USER_NAME of the NeuronBot is “ros” and the IP_ADDRESS is “192.168.50.1.23”:

```
ssh -X ros@192.168.1.23  # add capital -X to enable Linux X11 forwarding
```

Before connecting to the SSH server, your host name will be your computer. For example, in the following picture, the host name of the computer is “rostest”.



After connecting to the SSH server, the host name will change to the NeuronBot’s name. For example, in the following picture, the host name of NeuronBot is “neuronbot”.



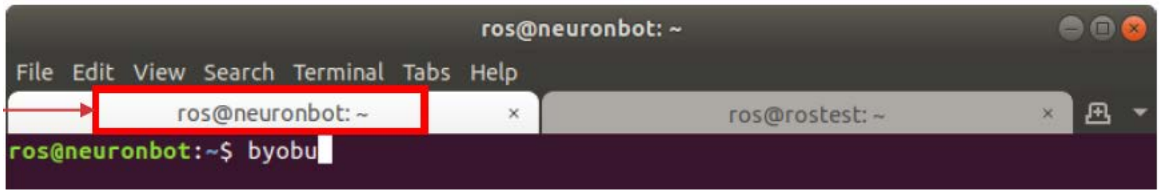
5.2.5. Byobu

Byobu is an easy-to-use wrapper for the tmux terminal multiplexer and can open multiple windows within a single terminal. With Byobu, you can open new windows without having to connect to an SSH server again and again. To install Byobu on the NeuronBot, execute the following command:

- NeuronBot:

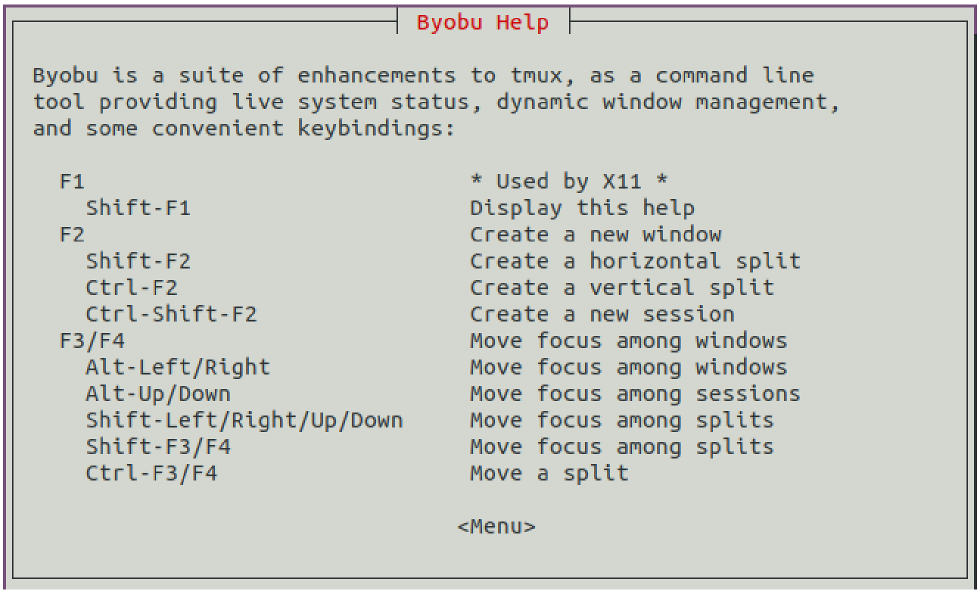
```
sudo apt install byobu
```

After installing, type “byobu” on NeuronBot to start Byobu.



In Byobu, press `F1` to open the Byobu Help window for detailed instructions on using Byobu.

Press `ESC` to exit the Byobu Help menu.



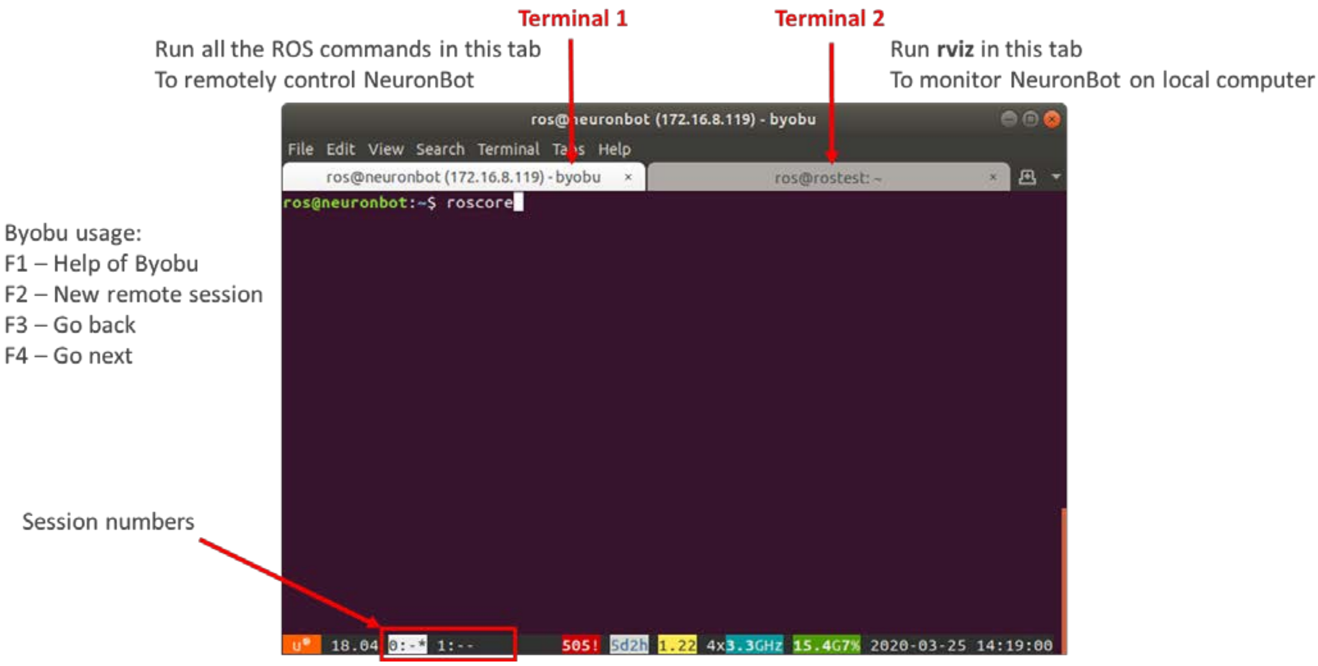

```
Byobu is a suite of enhancements to tmux, as a command line
tool providing live system status, dynamic window management,
and some convenient keybindings:

  F1                             * Used by X11 *
    Shift-F1                     Display this help
  F2                             Create a new window
    Shift-F2                     Create a horizontal split
    Ctrl-F2                      Create a vertical split
    Ctrl-Shift-F2                Create a new session
  F3/F4                          Move focus among windows
    Alt-Left/Right               Move focus among windows
    Alt-Up/Down                  Move focus among sessions
    Shift-Left/Right/Up/Down     Move focus among splits
    Shift-F3/F4                  Move focus among splits
    Ctrl-F3/F4                   Move a split

                                <Menu>
```

We highly recommend that you run all ROS commands except RVIZ in the terminal with SSH connections. Run RVIZ in a new terminal using a local terminal on a computer.

Note: “Session” means a window in a single terminal with an SSH connection to a NeuronBot. You can confirm your session number at the bottom of the window.



5.2.6. ROS 1 Remote Control Verification

To verify whether you have successfully setup remote control parameters for ROS 1, you can run a talker on NeuronBot and a listener on the host computer.

- Terminal 1:

1. Open the terminal.
2. Set up the SSH connection to access the NeuronBot remotely on your computer.

```
ssh -X ros@192.168.50.26
```

3. Start Byobu to run multiple sessions on a single SSH connection.

Note: “Session” means a window in a single terminal with an SSH connection to a NeuronBot.

```
byobu
```

- Session 0:

4. Source the environment.

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
```

5. Launch the roscore.

Note: The “roscore” is the “core” of the ROS. We recommend manually starting the core in a separate window for improved access and control.

```
roscore
```

6. Press F2 to create a new session in Byobu.

- Session 1:

7. Source the environment.

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
```

8. Start the talker.

After starting the talker on NeuronBot, it will publish “hello world” messages continuously.

```
rosrun roscpp_tutorials talker
```

You should see messages similar to those in Session 1:

```
[INFO] [1596772167.770081]: hello world 1596772167.77
[INFO] [1596772167.870652]: hello world 1596772167.87
[INFO] [1596772167.972095]: hello world 1596772167.97
[INFO] [1596772168.071039]: hello world 1596772168.07
[INFO] [1596772168.170968]: hello world 1596772168.17
[INFO] [1596772168.271473]: hello world 1596772168.27
[INFO] [1596772168.370915]: hello world 1596772168.37
[INFO] [1596772168.471067]: hello world 1596772168.47
[INFO] [1596772168.570971]: hello world 1596772168.57
[INFO] [1596772168.671001]: hello world 1596772168.67
[INFO] [1596772168.771006]: hello world 1596772168.77
[INFO] [1596772168.870979]: hello world 1596772168.87
[INFO] [1596772168.971656]: hello world 1596772168.97
[INFO] [1596772169.070738]: hello world 1596772169.07
```

9. Press Ctrl+Alt+T to create a new terminal on the host computer.
   You can also press Ctrl+Shift+T to create a new terminal tab.

- Terminal 2:

10. Source the environment.

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
```

11. Run the listener.

The listener will run on the computer and subscribe to messages from the talker.

```
rosrun roscpp_tutorials listener
```

You should see messages similar to those in Terminal 2:

```
[INFO] [1596772167.771804]: /listener_27816_1596772161347I heard hello world 1596772167.77
[INFO] [1596772167.876104]: /listener_27816_1596772161347I heard hello world 1596772167.87
[INFO] [1596772167.978625]: /listener_27816_1596772161347I heard hello world 1596772167.97
[INFO] [1596772168.077289]: /listener_27816_1596772161347I heard hello world 1596772168.07
[INFO] [1596772168.177649]: /listener_27816_1596772161347I heard hello world 1596772168.17
[INFO] [1596772168.277493]: /listener_27816_1596772161347I heard hello world 1596772168.27
[INFO] [1596772168.377027]: /listener_27816_1596772161347I heard hello world 1596772168.37
[INFO] [1596772168.476484]: /listener_27816_1596772161347I heard hello world 1596772168.47
[INFO] [1596772168.577383]: /listener_27816_1596772161347I heard hello world 1596772168.57
[INFO] [1596772168.677679]: /listener_27816_1596772161347I heard hello world 1596772168.67
[INFO] [1596772168.777763]: /listener_27816_1596772161347I heard hello world 1596772168.77
[INFO] [1596772168.878287]: /listener_27816_1596772161347I heard hello world 1596772168.87
[INFO] [1596772168.977759]: /listener_27816_1596772161347I heard hello world 1596772168.97
[INFO] [1596772169.073819]: /listener_27816_1596772161347I heard hello world 1596772169.07
[INFO] [1596772169.173241]: /listener_27816_1596772161347I heard hello world 1596772169.17
[INFO] [1596772169.277652]: /listener_27816_1596772161347I heard hello world 1596772169.27
[INFO] [1596772169.377547]: /listener_27816_1596772161347I heard hello world 1596772169.37
```

If the listener successfully subscribed to the messages, then the remote control parameters have been successfully set for ROS 1.

5.2.7. ROS 2 Remote Control Verification

To verify that remote control parameters have been set up for ROS 2, you can run a talker on NeuronBot and a listener on the computer.

- Terminal 1:

1. Open the terminal.
2. Setup the SSH connection to access the NeuronBot remotely.

```
ssh –X ros@192.168.50.26
```

3. Start Byobu to run multiple sessions on a single SSH connection.

Note: “Session” means a window in a single terminal with an SSH connection to a NeuronBot.

```
byobu
```

- Session 0:

4. Source the environment.

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
```

5. Start the talker.

After starting the talker on NeuronBot, it will publish “hello world” messages continuously.

```
ros2 run demo_nodes_cpp talker
```

You should see messages similar to those in Session 0:

```
[INFO] [talker]: Publishing: 'Hello World: 1'
[INFO] [talker]: Publishing: 'Hello World: 2'
[INFO] [talker]: Publishing: 'Hello World: 3'
[INFO] [talker]: Publishing: 'Hello World: 4'
[INFO] [talker]: Publishing: 'Hello World: 5'
[INFO] [talker]: Publishing: 'Hello World: 6'
[INFO] [talker]: Publishing: 'Hello World: 7'
[INFO] [talker]: Publishing: 'Hello World: 8'
[INFO] [talker]: Publishing: 'Hello World: 9'
[INFO] [talker]: Publishing: 'Hello World: 10'
[INFO] [talker]: Publishing: 'Hello World: 11'
[INFO] [talker]: Publishing: 'Hello World: 12'
[INFO] [talker]: Publishing: 'Hello World: 13'
```

6. Press Ctrl+Alt+T to create a new terminal on the host computer
   You can also press Ctrl+Shift+T to create a new terminal tab.

- Terminal 2:

7. Source the environment.

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
```

8. Run the listener.

The listener will run on the computer and subscribe to messages from the talker.

```
ros2 run demo_nodes_cpp listener
```

You should see messages similar to those in Terminal 2:

```
[INFO] [listener]: I heard: [Hello World: 1]
[INFO] [listener]: I heard: [Hello World: 2]
[INFO] [listener]: I heard: [Hello World: 3]
[INFO] [listener]: I heard: [Hello World: 4]
[INFO] [listener]: I heard: [Hello World: 5]
[INFO] [listener]: I heard: [Hello World: 6]
[INFO] [listener]: I heard: [Hello World: 7]
[INFO] [listener]: I heard: [Hello World: 8]
[INFO] [listener]: I heard: [Hello World: 9]
[INFO] [listener]: I heard: [Hello World: 10]
[INFO] [listener]: I heard: [Hello World: 11]
[INFO] [listener]: I heard: [Hello World: 12]
[INFO] [listener]: I heard: [Hello World: 13]
```

If the listener successfully subscribed to the messages, then the remote control parameters have been successfully set for ROS 2.

5.3. ROS 1 Applications

This section describes how to build, compile, and run several applications with ROS 1 on NeuronBot, and provides instructions on configuring NeuronBot for the following applications:

- **Teleoperation**: Move the NeuronBot using a keyboard and scan the surrounding environment using 2D LiDAR.
- **RViz (ROS-VIsualiZation) monitoring**: Monitor Neuronbot during movement and laser scanning using RViz, a powerful 3D visualization environment for ROS.
- **Simultaneous Localization And Mapping (SLAM)**: Configure NeuronBot to build a map during teleoperation.
- **Guided navigation**: Navigate NeuronBot from a starting point to a destination using a map created with a SLAM package.

5.3.1. Driver Startup and Teleoperation

To begin teleoperating NeuronBot, you must start the ROS driver in addition to all IO connections and sensory devices such as the motor controller, encoder odometry, laser scanner, and IMU state estimation.

- Terminal 1:

1. Open the terminal.
2. Set up the SSH connection to access the NeuronBot remotely on your computer.

```
ssh –X ros@192.168.50.26
```

3. Start Byobu to run multiple sessions on a single SSH connection.

Note: “Session” means a window in a single terminal with an SSH connection to a NeuronBot.

```
byobu
```

- Session 0:

4. Source the environment.

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
```

5. Launch the roscore.

Note: The “roscore” is the “core” of the ROS. We recommend manually starting the core in a separate window for improved access and control.

```
roscore
```

- Session 1:

6. Source the environment.

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
source ~/neuronbot2_ros1_ws/devel/setup.bash
```

7. Launch NeuronBot.

Note: This launch file contains multiple nodes and enables communication between the motor controller, laser SLAM, and all NeuronBot TF definitions. If you end the node with `ctrl + c`, remember only to press once and allow it to shut down automatically. The rplidarNode node requires some time to shut down the serial port.

```
roslaunch neuronbot2_bringup bringup.launch
```

- Session 2:

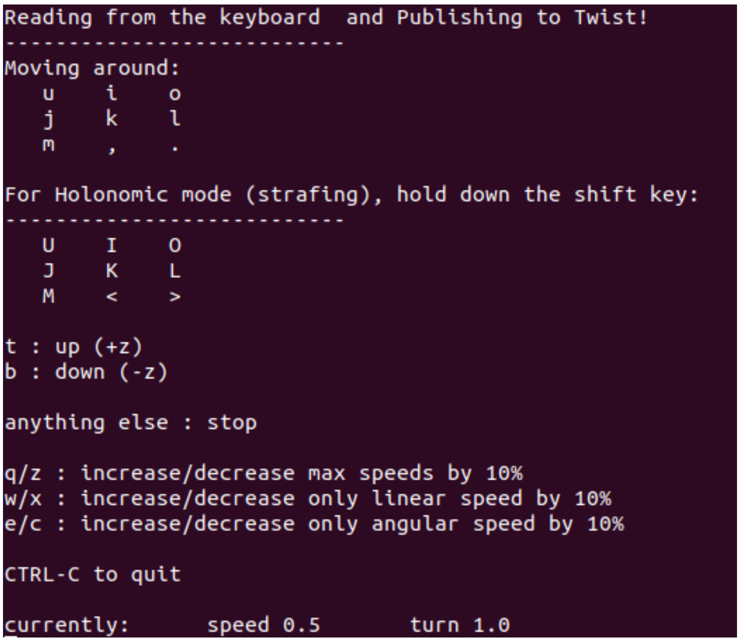
8. Source the environment.

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
```

9. Start teleoperation.

Note: The manual driver used for this scenario is teleop_twist_keyboard. The default command is a little too fast, so press **x** and **c** to decrease the linear speed to 0.4 m/s and the angular speed to 0.4 rad/s. Press **k** or **s** to immediately stop.

```
rosrun teleop_twist_keyboard teleop_twist_keyboard
```



```
Reading from the keyboard  and Publishing to Twist!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

Figure 4-1-3-1: teleop_twist_keyboard

- Terminal 2:

10. Set up the environment by setting ROS_MASTER_URI to the NeuronBot's IP address.

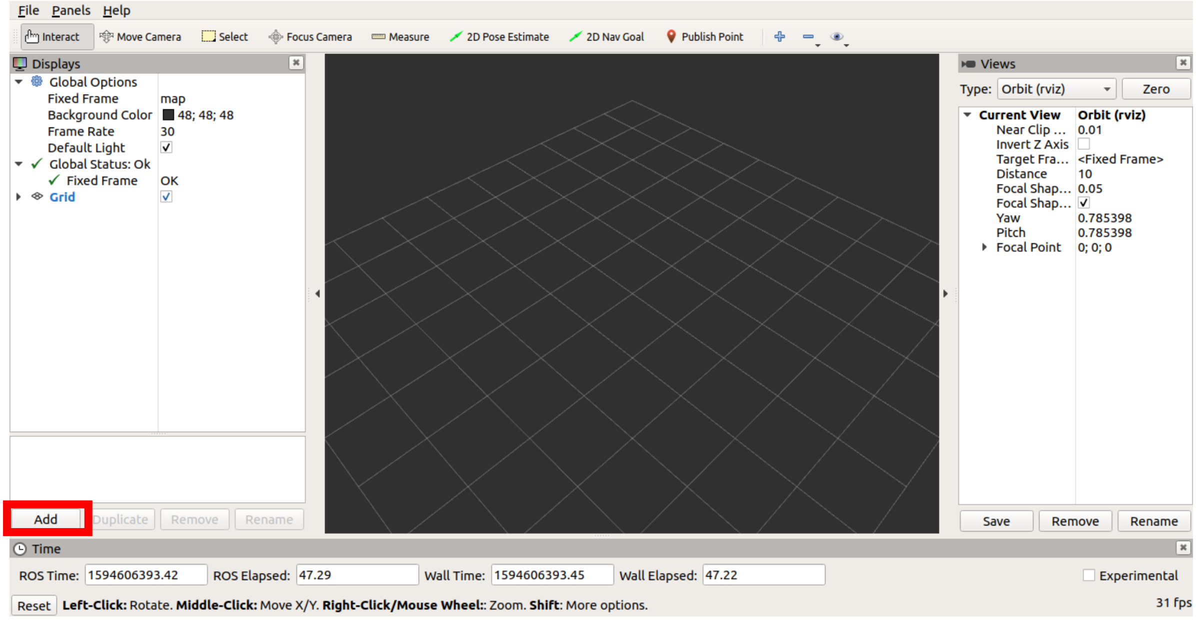
```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
export $ROS_MASTER_URI=http://192.168.50.26:11311
```

11. Launch RViz.

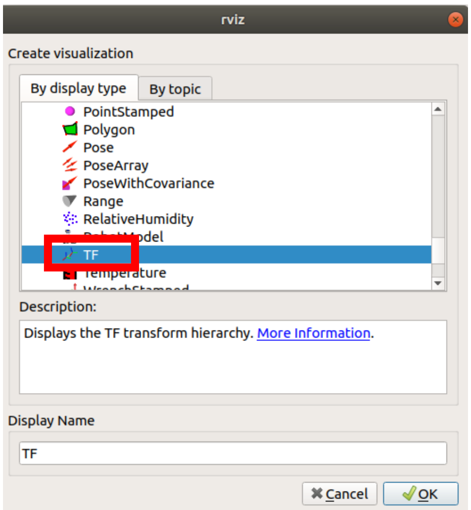
```
rviz
```

- RViz:

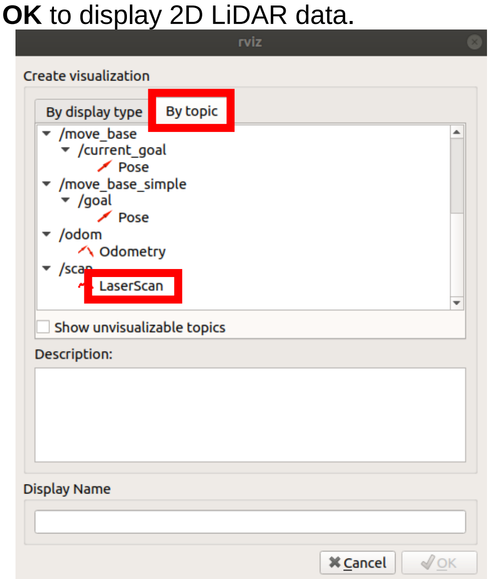
12. Click the **Add** button in the lower left.



13. Select **TF** and click **OK** to display the frames.



14. Click the **Add** button again in the lower left.
15. Click the **By topic** tab to display available topics.
16. Select **LaserScan** and click **OK** to display 2D LiDAR data.



Tip: In `ros_menu/config.yaml` in the ADLINK Neuron Startup Menu, you can add commands to the "cmds" section. For example, in ROS 1 Melodic, we added a "source" command so that every time a new session and terminal is opened, the menu automatically loads the NeuronBot workspace environment. This way, there is no longer a need to "source" ROS and NeuronBot anymore.

```
Menu:
  ROS 1 melodic:
    option_num: 1
    ROS_version: 1
    distro_name: melodic
    ros1_path: /opt/ros/melodic
    master_ip: 192.168.50.26
    cmds:
     - source ~/neuronbot2_ros1_ws/devel/setup.bash
```

5.3.2. Laser SLAM

This section describes how to build a map using a 2D laser scanner.

Note: Ensure that everything in the base driver has been launched before running SLAM.

- Terminal 1, Session 3:

1. Source the environment.

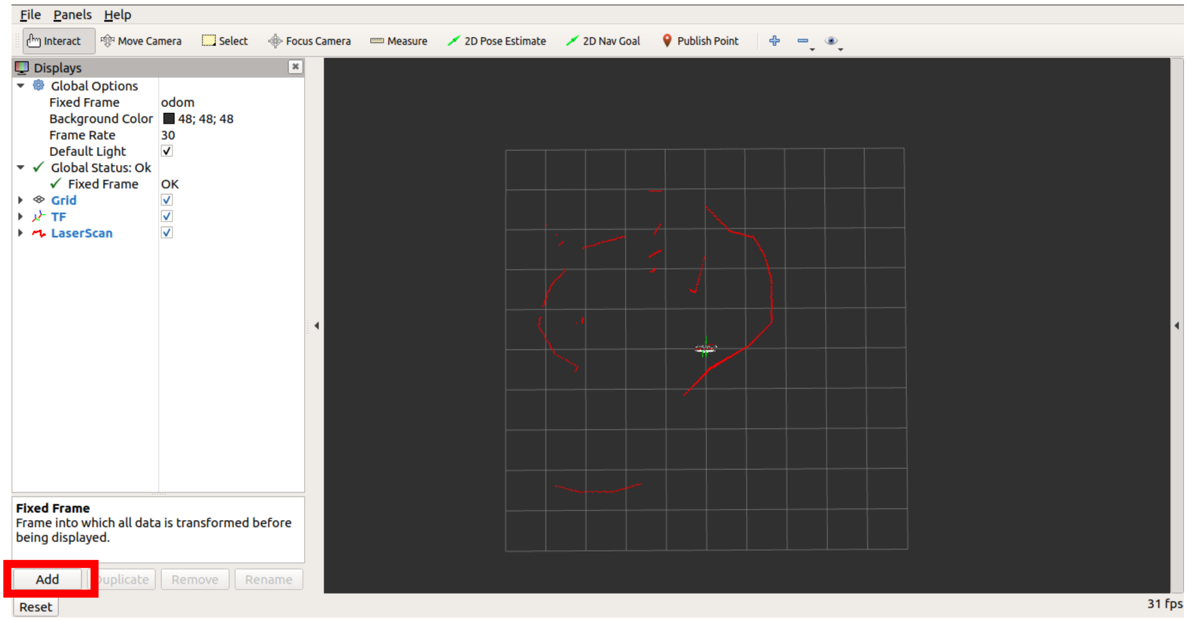
```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
source ~/neuronbot2_ros1_ws/devel/setup.bash
```

2. Start the laser localization and mapping procedure with gmapping by executing the following command.

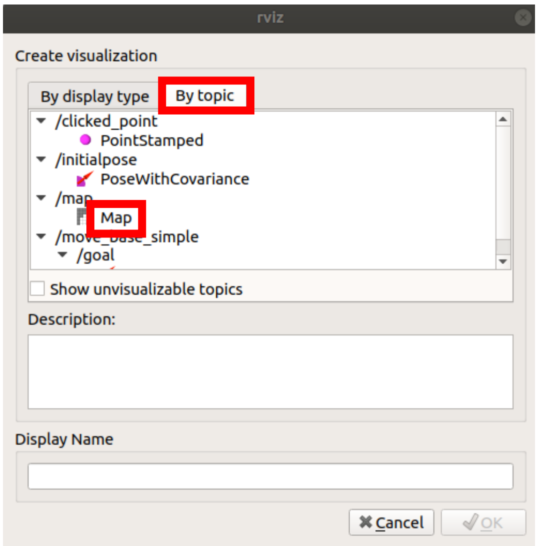
```
roslaunch neuronbot2_slam gmapping.launch
```

- RViz:

3. Click the **Add** button in the lower left.



4. Click the **By topic** tab to display available topics.
5. Select **Map** and click **OK** to display the map created using gmapping.



- Session 2:

6. Go back to the teleop_twist_keyboard session. Press **x** and **c** to decrease the linear speed to 0.3 m/s the angular speed to 0.2 rad/s, and then drive the NeuronBot around using the keyboard driver. After mapping the environment, remember to save the map **before** closing gmapping.

- Session 4:

7. Source the environment.

```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
source ~/neuronbot2_ros1_ws/devel/setup.bash
roscd neuronbot2_nav/maps/
```

8. Save the map.

A map file and a config file will be saved to **neuronbot2_nav/maps**.

```
rosrun map_server map_saver -f map_name
```

9. Stop gmapping by pressing `ctrl + c` in the gmapping session (Session 3).

5.3.3. Navigation

After getting a static map, running a SLAM package is not recommended due to its computational load. This section describes how to use an AMCL package to locate the NeuronBot using a previously generated map and existing laser scan. This will allow you to move the NeuronBot from one location to a specified destination.

Note: Ensure that everything in the base driver has been launched before running SLAM.

- Terminal 1, Session 3:

1. Source the environment.

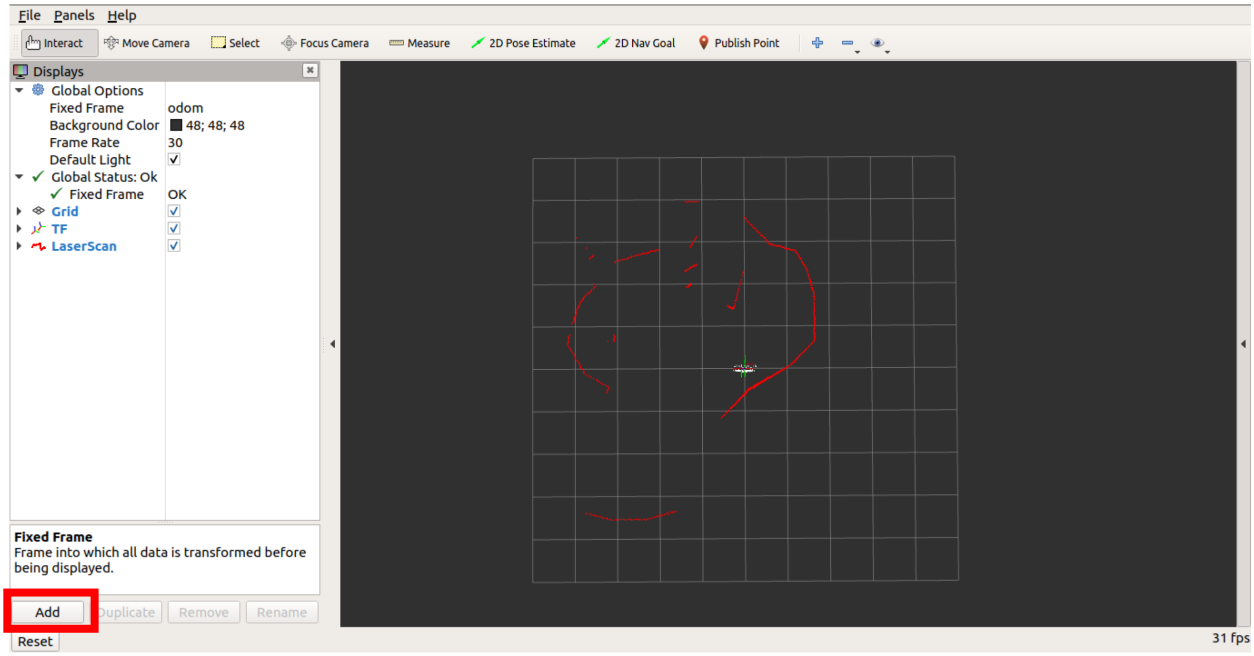
```
source /opt/ros/<YOUR_ROS1_DISTRO>/setup.bash
source ~/neuronbot2_ros1_ws/devel/setup.bash
```

2. Start the navigation procedure by executing the following command.

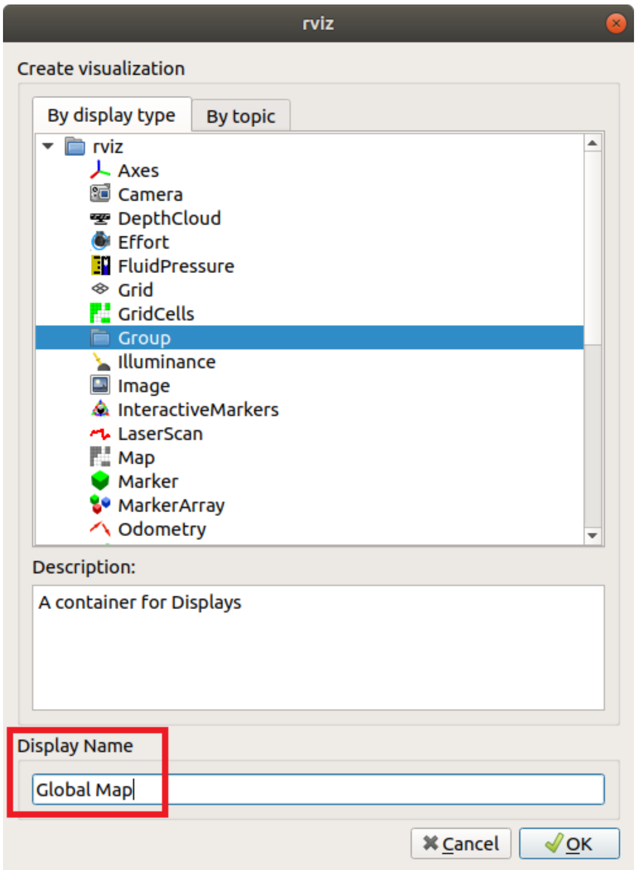
```
roslaunch neuronbot2_nav bringup.launch map:=map_name
```

- RViz:

3. Click the **Add** button in the lower left.



4. Rename **Group** to **Global Map**.



5. Click **OK**.
6. Click the **Add** button in the lower left again.
7. Rename **Map** to **Costmap**.
8. Drag **Costmap** into **Global Map** and set the parameters as follows.

Global Map	[x]
Costmap	[x]
✓ Status: Ok	
Topic	/move_base/global_costmap/costmap
Alpha	0.7
Color Scheme	costmap
Draw Behind	[x]
Resolution	0.05
Width	544
Height	512
Position	-12.2; -13.8; 0
Orientation	0; 0; 0; 1
Unreliable	[ ]
Use Timestamp	[ ]

9. Click **Add**, rename **Path** to **Planner**, drag **Planner** into **Global Map**, and set the parameters as follows.

Global Map	[x]
Costmap	[x]
✓ Status: Ok	
Topic	/move_base/global_costmap/costmap
Alpha	0.7
Color Scheme	costmap
Draw Behind	[x]
Resolution	0.05
Width	544
Height	512
Position	-12.2; -13.8; 0
Orientation	0; 0; 0; 1
Unreliable	[ ]
Use Timestamp	[ ]
Planner	[x]
✓ Status: Ok	
Topic	/move_base/GlobalPlanner/plan
Unreliable	[ ]
Line Style	Billboards
Line Width	0.03
Color	0; 255; 0
Alpha	1
Buffer Length	1
Offset	0; 0; 0
Pose Style	None

10. Click **Add**, rename **Group** to **Local Map**, and click **OK**.

11. Click **Add**, rename **Map** to **Costmap**, drag **Costmap** into **Local Map**, and set the parameters as follows.

Local Map	[x]
Costmap	[x]
✓ Status: Ok	
Topic	/move_base/local_costmap/costmap
Alpha	0.7
Color Scheme	costmap
Draw Behind	[ ]
Resolution	0.05
Width	40
Height	40
Position	-1; -0.95; 0
Orientation	0; 0; 0; 1
Unreliable	[ ]
Use Timestamp	[ ]

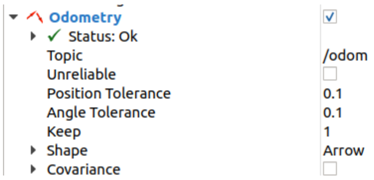
12. Click **Add**, rename **Path** to **Planner**, drag **Planner** into **Local Map**, and set the parameters as follows.

Local Map	[x]
Costmap	[x]
✓ Status: Ok	
Topic	/move_base/local_costmap/costmap
Alpha	0.7
Color Scheme	costmap
Draw Behind	[ ]
Resolution	0.05
Width	40
Height	40
Position	-1; -0.95; 0
Orientation	0; 0; 0; 1
Unreliable	[ ]
Use Timestamp	[ ]
Planner	[x]
✓ Status: Ok	
Topic	/move_base/DWAPlannerROS/local_plan
Unreliable	[ ]
Line Style	Billboards
Line Width	0.03
Color	237; 212; 0
Alpha	1
Buffer Length	1
Offset	0; 0; 0
Pose Style	None

13. Click **Add**, rename **PoseArray** to **Amcl Particle Swarm**, and set the parameters as follows.

Amcl Particle Swarm	[x]
✓ Status: Ok	
Topic	/particlecloud
Unreliable	[ ]
Shape	Arrow (Flat)
Color	0; 192; 0
Alpha	1
Arrow Length	0.2

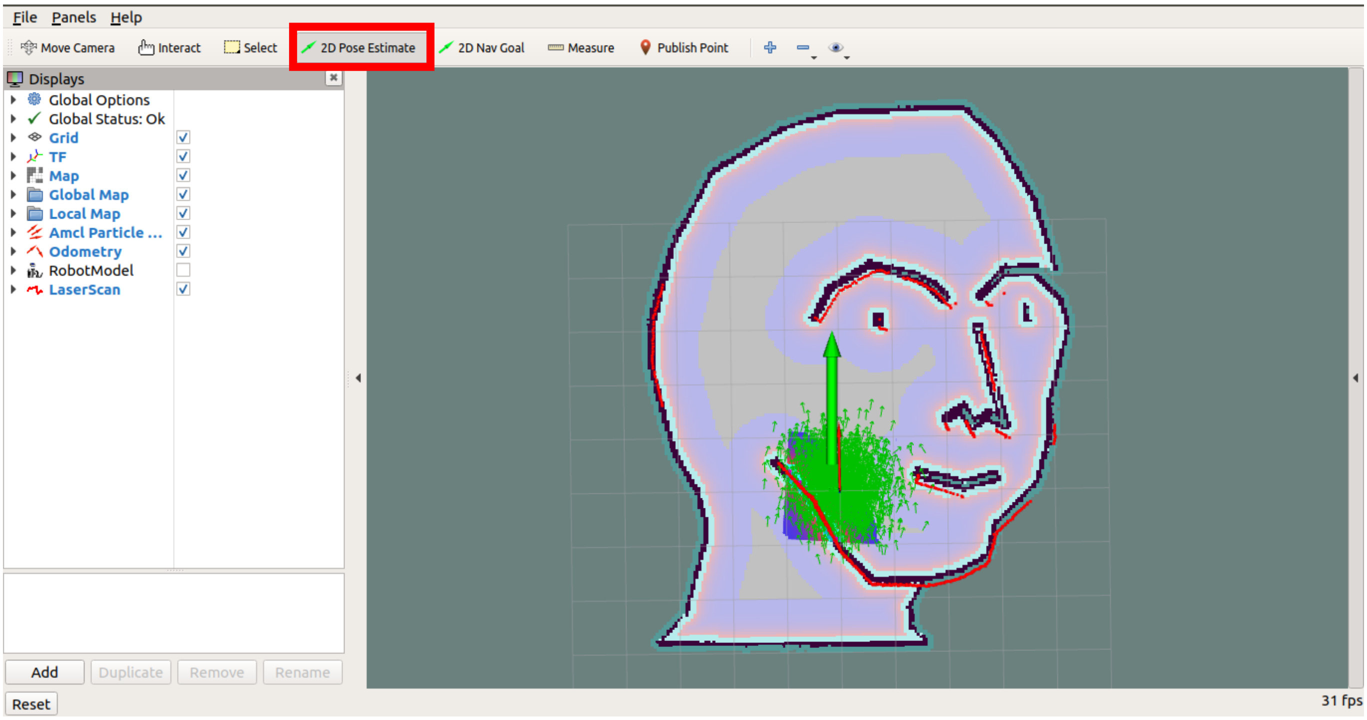
14. Click **Add**, select **Odometry**, click **OK**, and set the parameters as follows.



15. Perform pose estimation.

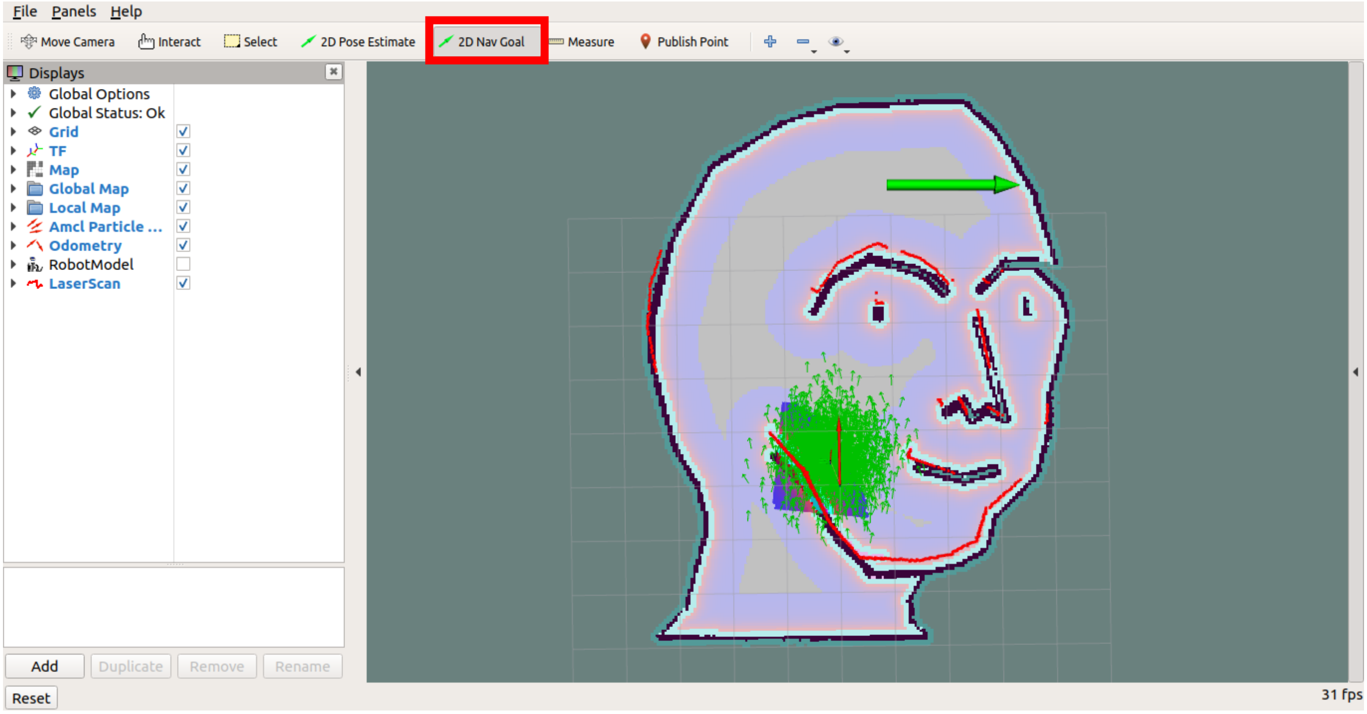
    a. Click "2D Pose Estimate", and set the pose estimation to the approximate location of the NeuronBot on the map.

    Note: By default, the localization package will initialize the NeuronBot at (x,y)=(0,0); i.e., the same as the starting position when the mapping process started. You can also manually assign the starting position by using the "set 2D pose estimation" function in RViz. Select the tool, click on the position, and drag the arrow to its initial heading as shown in the following figure. "2D pose estimation" is marked by a red square in the upper banner.

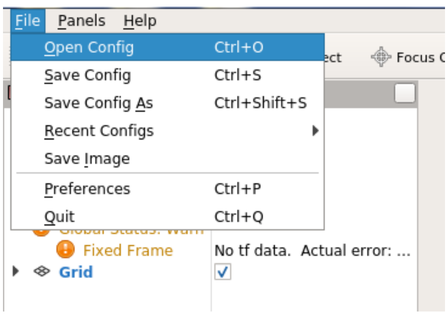


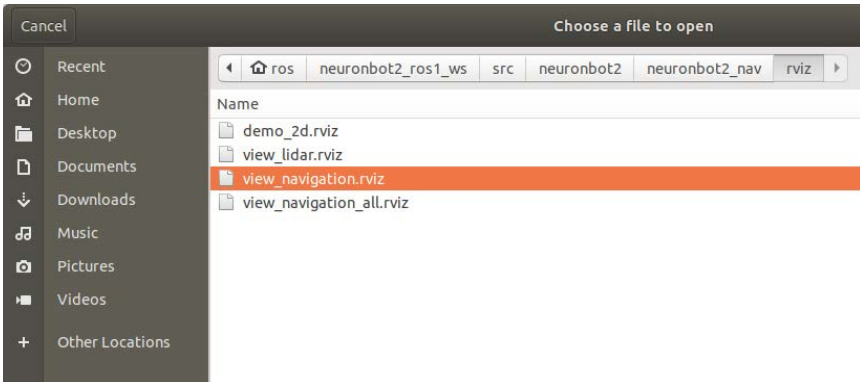
16. Set the goal.

    b. Click and drag "2D Nav Goal" to set the goal and orientation to any free space on the map. The NeuronBot should drive toward the goal by itself.



Tip: Save time by opening the RViz config file in:
`~/neuronbot2_ros1_ws/src/neuronbot2/neuronbot2_nav/rviz/view_navigation.rviz`





5.4. ROS 2 Applications

This section describes how to build, compile, and run several applications with ROS 2 on NeuronBot, and provides instructions on configuring NeuronBot for the following applications:

- **Teleoperation**: Move the NeuronBot using a keyboard and scan the surrounding environment using 2D LiDAR.
- **RViz (ROS-VIsualiZation) monitoring**: Monitor Neuronbot during movement and laser scanning using RViz, a powerful 3D visualization environment for ROS.
- **Simultaneous localization and mapping (SLAM)**: Configure NeuronBot to build a map during teleoperation.
- **Guided navigation**: Navigate NeuronBot from a starting point to a destination with a map created using a SLAM package.

5.4.1. Driver Startup and Teleoperation

To begin teleoperating NeuronBot, you must start the ROS driver in addition to all IO connections and sensory devices such as the motor controller, encoder odometry, laser scanner, and IMU state estimation.

- Terminal 1:

1. Open the terminal.
2. Set up the SSH connection to access the NeuronBot remotely on your computer.

```
ssh -X ros@192.168.50.26
```

3. Start Byobu to run multiple sessions on a single SSH connection.

Note: “Session” means a window in a single terminal with an SSH connection to a NeuronBot.

```
byobu
```

- Session 0:

4. Source the environment.

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
source ~/neuronbot2_ros2_ws/install/local_setup.bash
```

5. Launch NeuronBot.

Note: This launch file contains multiple nodes and enables communication between the motor controller, laser SLAM, and all NeuronBot TF definitions. If you end the node with `ctrl + c`, remember only to press once and allow it to shut down automatically. The rplidarNode node requires some time to shut down the serial port.

```
ros2 launch neuronbot2_bringup bringup_launch.py
```

- Session 1:

6. Source the environment.

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
```

7. Start teleoperation.

Note: The manual driver used for this scenario is teleop_twist_keyboard. The default command is a little too fast, so press **x** and **c** to decrease the linear speed to 0.4 m/s and the angular speed to 0.4 rad/s. Press **k** or **s** to immediately stop.

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

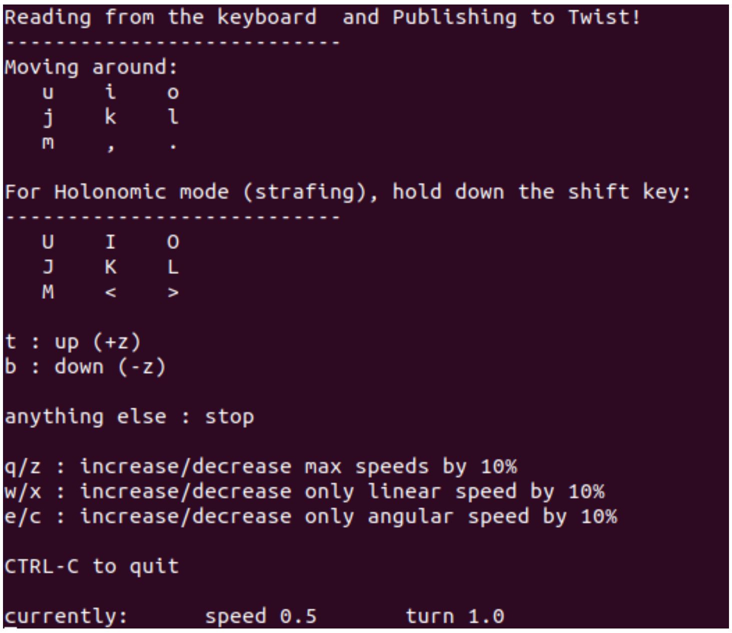


Figure 4-1-4-1: teleop_twist_keyboard

- Terminal 2:

8. Set up the environment.

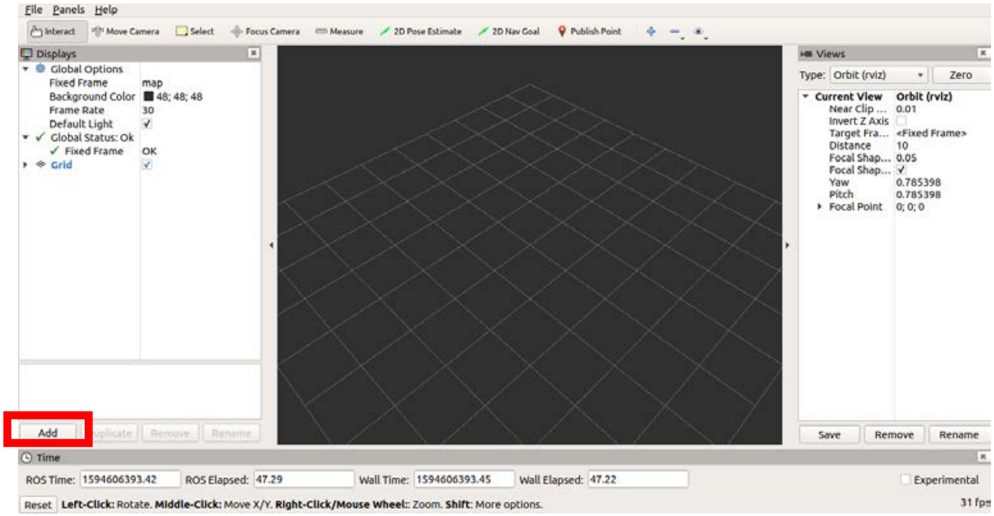
```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
```

9. Launch RViz.

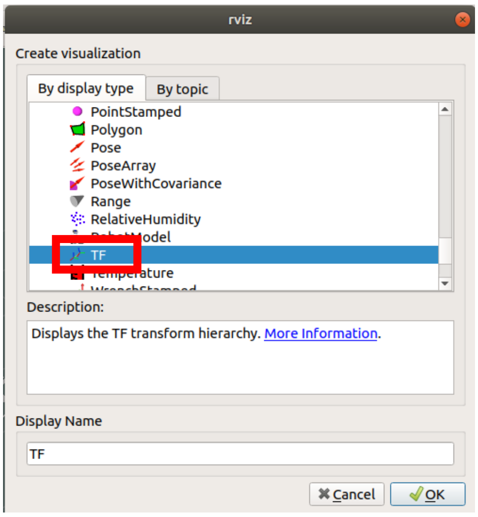
```
rviz2
```

- RViz2:

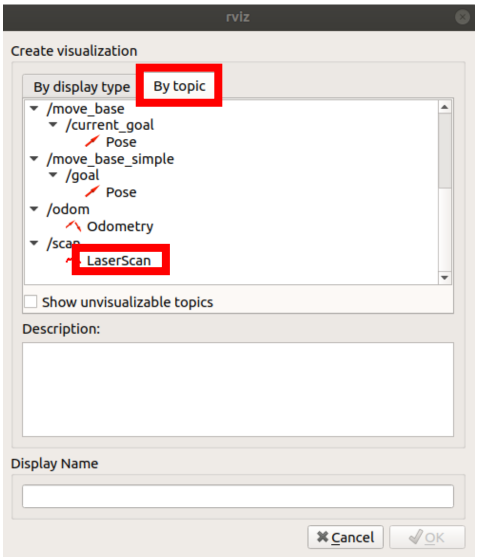
10. Click the **Add** button in the lower left.



11. Select **TF** and click **OK** to display the frames.



12. Click the **Add** button again in the lower left.
13. Click the **By topic** tab to display available topics.
14. Select **LaserScan** and click **OK** to display 2D LiDAR data.



Tip: In `ros_menu/config.yaml` in the ADLINK Neuron Startup Menu, you can add commands to the “cmds” section. For example, in ROS 2 Dashing, we added a “source” command so that every time a new session and terminal is opened, the menu automatically loads the NeuronBot workspace environment. This way, there is no longer a need to “source” ROS and NeuronBot anymore.

```
Menu:
  ROS 2 dashing:
    option_num: 2
    ROS_version: 2
    distro_name: dashing
    ros2_path: /opt/ros/dashing
    domain_id: # set if you don't want to use default domain id
    cmds:
      - source_plugin dds_bashrc 1
      - source ~/neuronbot2_ros2_ws/install/local_setup.bash
```

5.4.2. Laser SLAM

This section describes how to build a map using a 2D laser scanner.

Note: Ensure that everything in the base driver has been launched before running SLAM.

- Terminal 1, Session 2:

1. Source the environment.

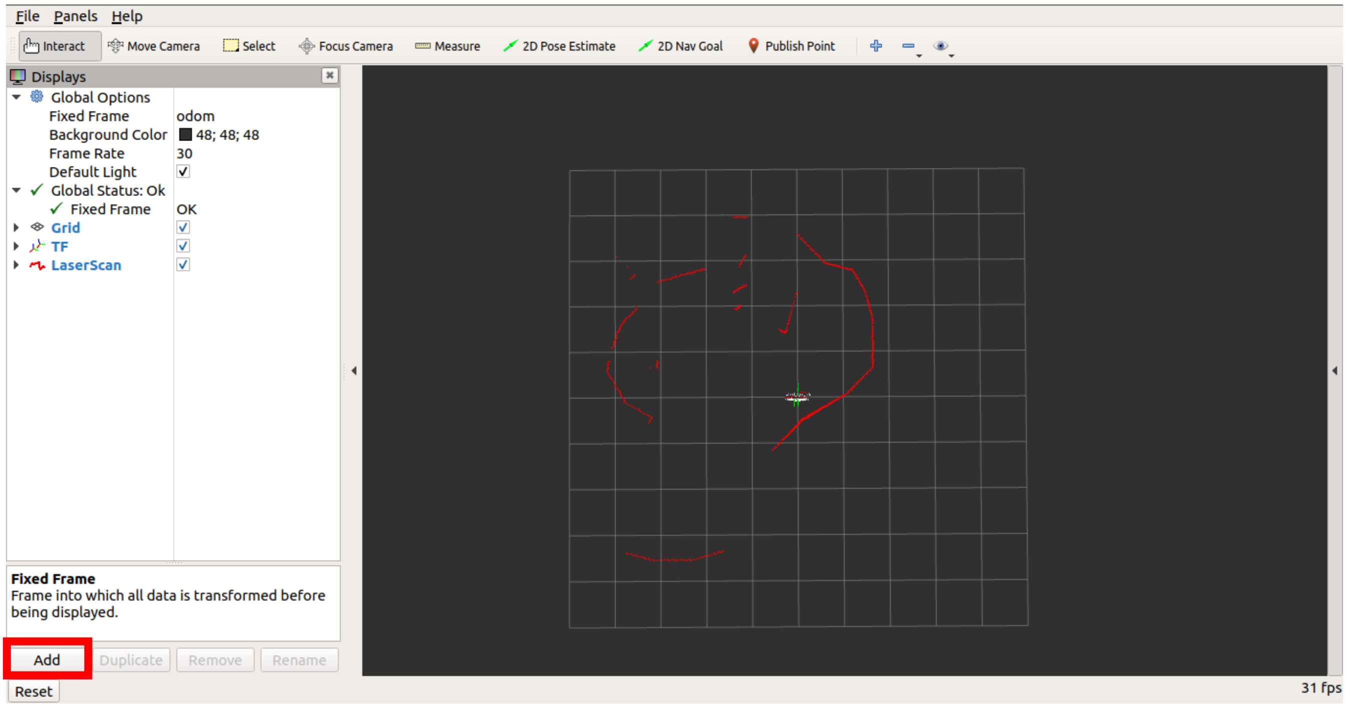
```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
source ~/neuronbot2_ros2_ws/install/local_setup.bash
```

2. Start the laser localization and mapping procedure with the Slam Toolbox by executing the following command.

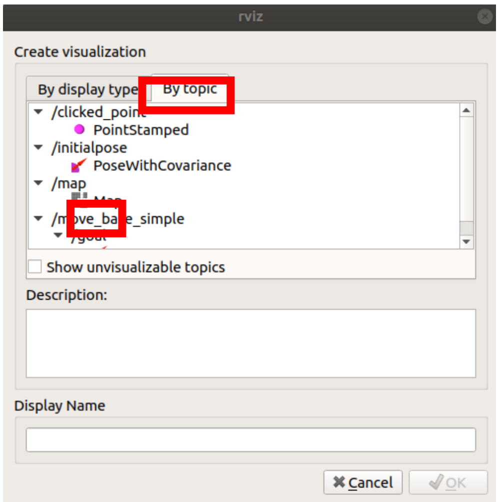
```
ros2 launch neuronbot2_slam slam_toolbox_launch.py
```

- RViz2:

3. Click the **Add** button in the lower left.



4. Click the **By topic** tab to display available topics.
5. Select **Map** and click **OK** to display the map created using the Slam Toolbox.



- Session 2:

6. Use x and c to decrease the linear to 0.3 m/s as well as angular speed to 0.2 rad/s, and then drive the NeuronBot around using the keyboard driver. After mapping the environment, remember to save the map **before** closing the Slam Toolbox.

- Session 3:

7. Source the environment.

```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
source ~/neuronbot2_ros2_ws/install/local_setup.bash
cd ~/neuronbot2_ros2_ws/src/neuronbot2/neuronbot2_nav/map/
```

8. Save the map.

A map file and a config file will be saved under **neuronbot2_nav/map**.

```
# for dashing and eloquent
ros2 run nav2_map_server map_saver -f <map_name>
or
# for foxy:
ros2 run nav2_map_server map_saver_cli -f <map_name>
```

9. Stop the Slam Toolbox by pressing `ctrl + c` in the Slam Toolbox session (Session 3).

5.4.3. Navigation

After getting a static map, running a SLAM package is not recommended due to its computational load. This section describes how to use an AMCL package to locate the NeuronBot using a previously generated map and existing laser scan. This will allow you to move the NeuronBot from one location to a specified destination.

Note: Ensure that everything in the base driver has been launched before running SLAM.

- Terminal 1, Session 2:

1. Source the environment.

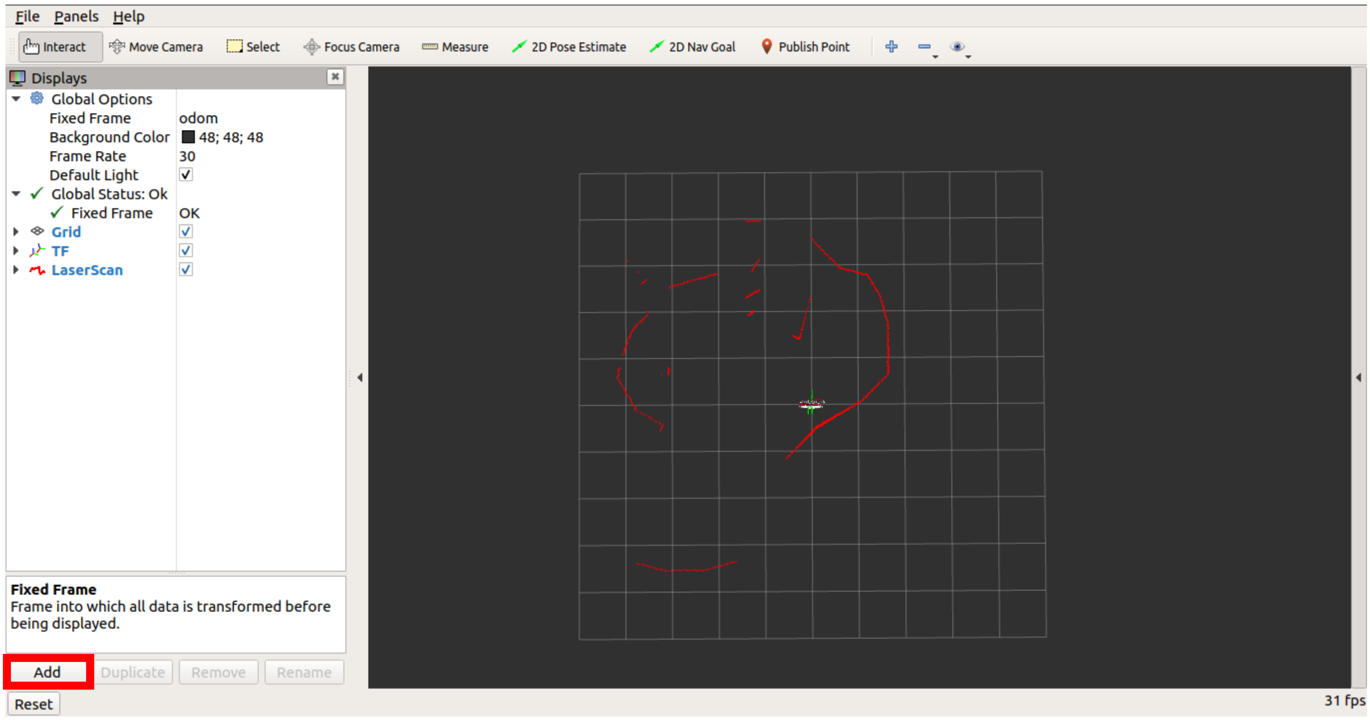
```
source /opt/ros/<YOUR_ROS2_DISTRO>/setup.bash
source ~/neuronbot2_ros2_ws/devel/local_setup.bash
```

2. Start the navigation procedure by executing the following command.

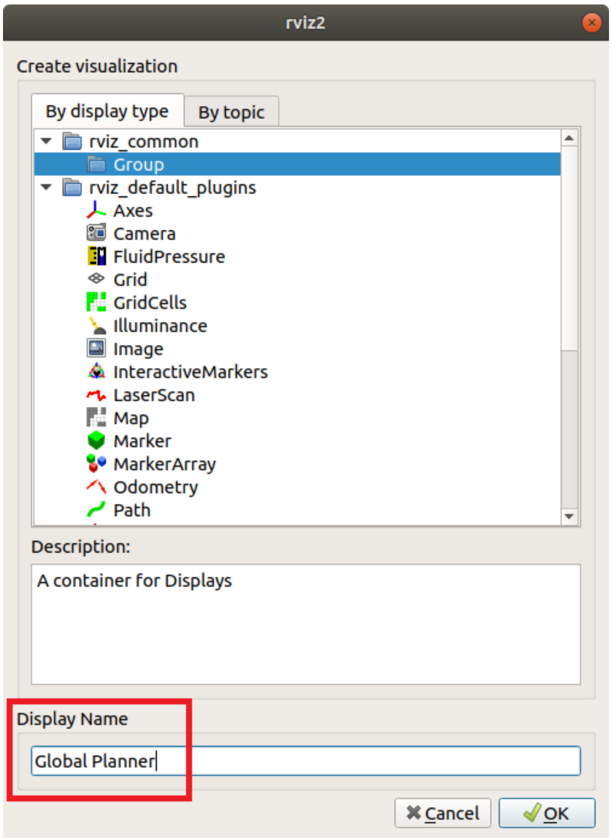
```
ros2 launch neuronbot2_nav bringup_launch.py map:=<full_path_to_your_map_name.yaml>
```

- RViz2:

3. Click the **Add** button in the lower left.

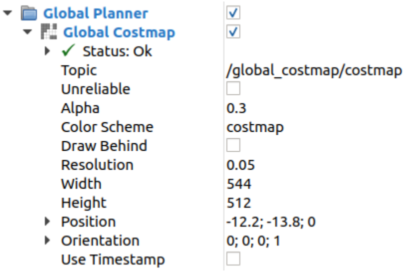


4. Rename **Group** to **Global Planner**.

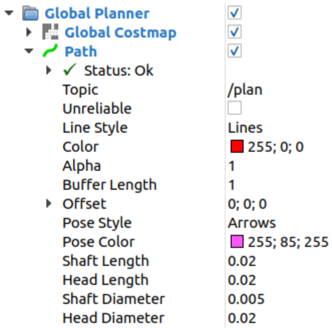


5. Click **OK**.
6. Click the **Add** button in the lower left again.
7. Rename **Map** to **Global Costmap**.

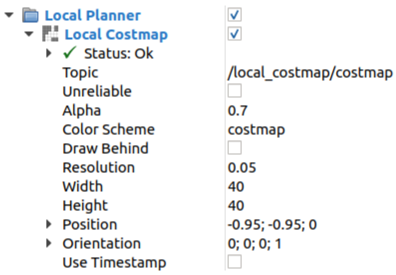
8. Drag **Global Costmap** into **Global Planner** and set the parameters as follows.



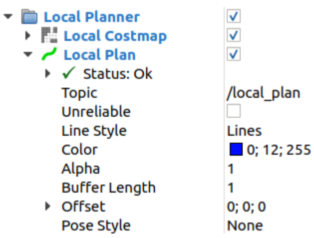
9. Click **Add**, select **Path**, drag **Path** into **Global Planner**, and set the parameters as follows.



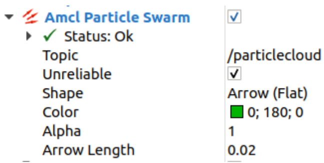
10. Click **Add**, select **Group**, click **OK**, and then rename **Group** to **Local Planner**.
11. Click **Add**, rename **Map** to **Local Costmap**, and then drag it into **Local Planner** and set the parameters as follows.



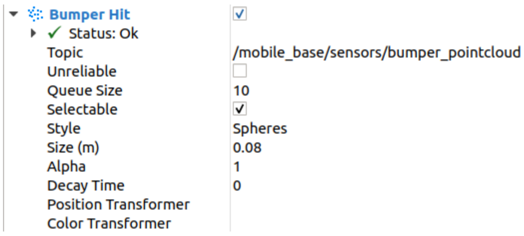
12. Click **Add**, rename **Path** to **Local Plan**, drag **Local Plan** into **Local Planner**, and set the parameters as follows.



13. Click **Add**, rename **PoseArray** to **Amcl Particle Swarm**, and set the parameters as follows.



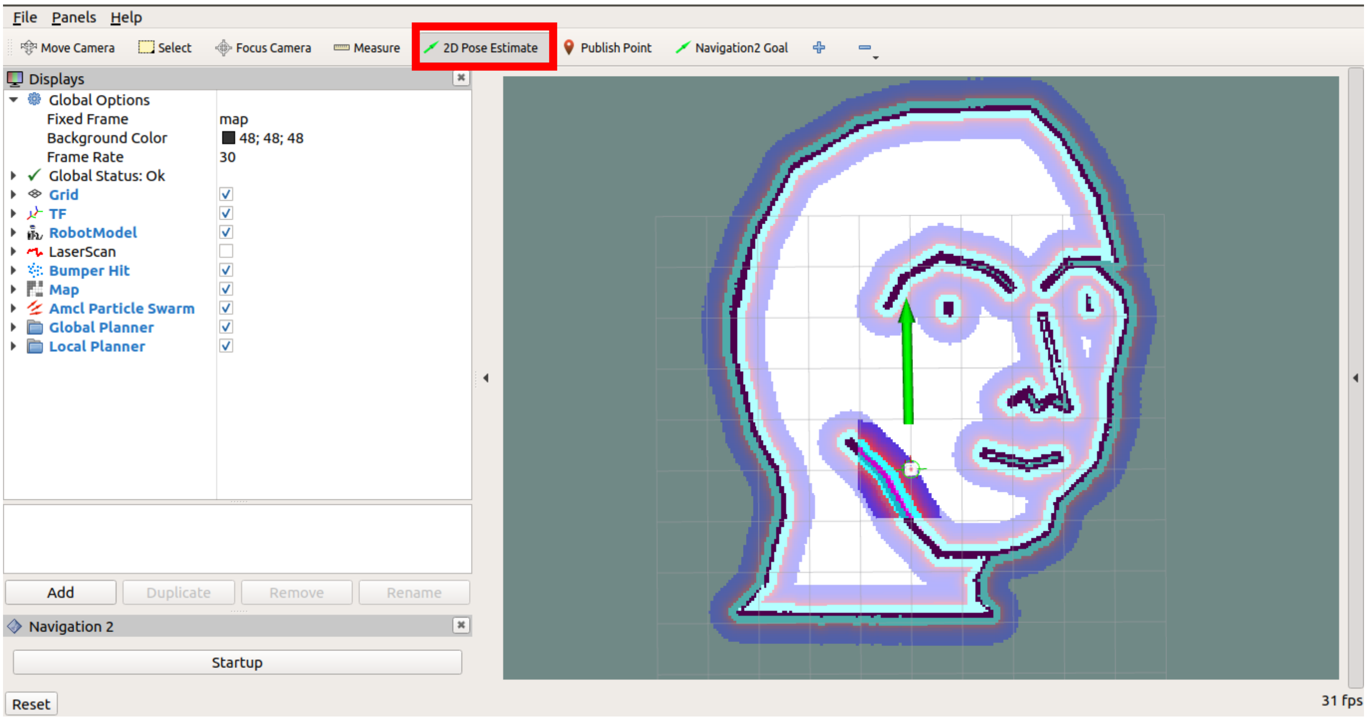
14. Click **Add**, select **PointCloud2**, click **OK**, and set the parameters as follows.
15. Rename **PointCloud2** to **Bumper Hit** and set the parameters as follows.



16. Set the estimation.

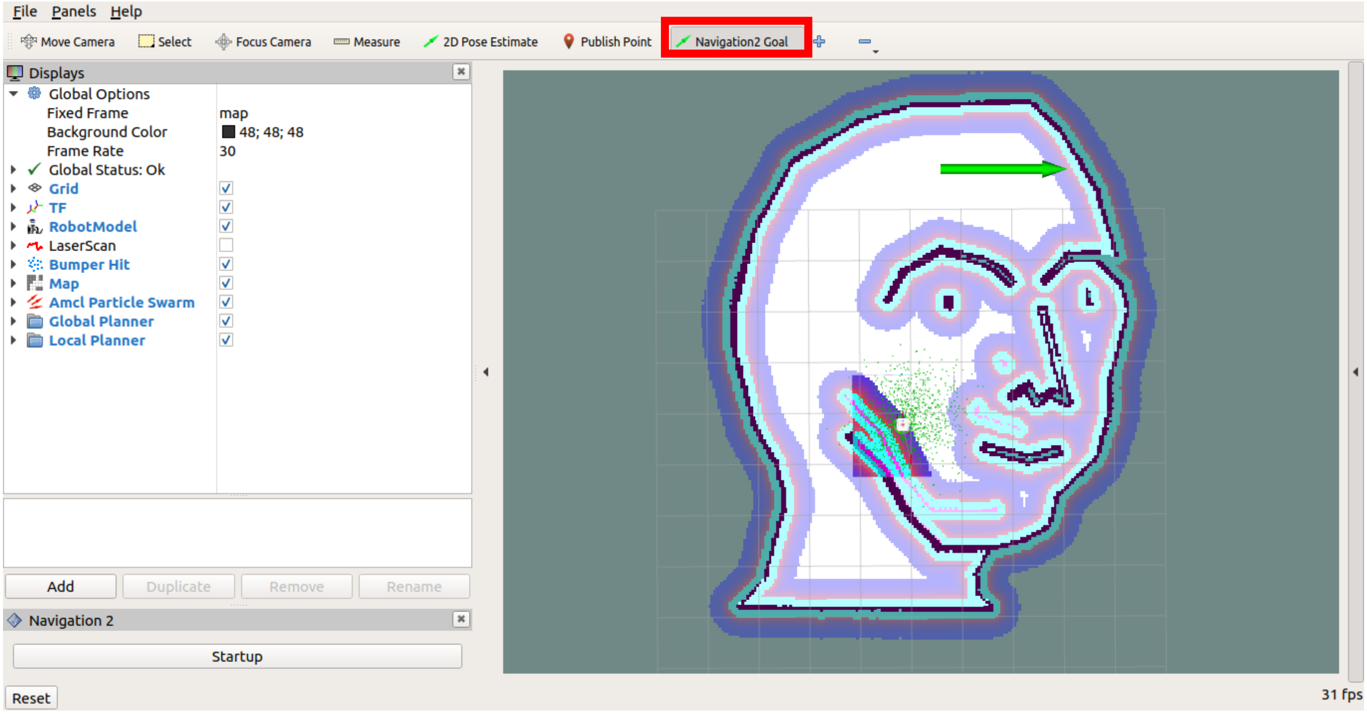
    a. Click "2D Pose Estimate", and set the estimation to the approximate location of the NeuronBot on the map.

    Note: By default, the localization package will initialize the NeuronBot at (x,y)=(0,0); i.e., the same as the starting position when the mapping process started. You can also manually assign the starting position by using the "set 2D pose estimation" function in RViz2. Select the tool, click on the position, and drag the arrow to its initial heading as shown in the following figure. "2D pose estimation" is marked by a red square in the upper banner.

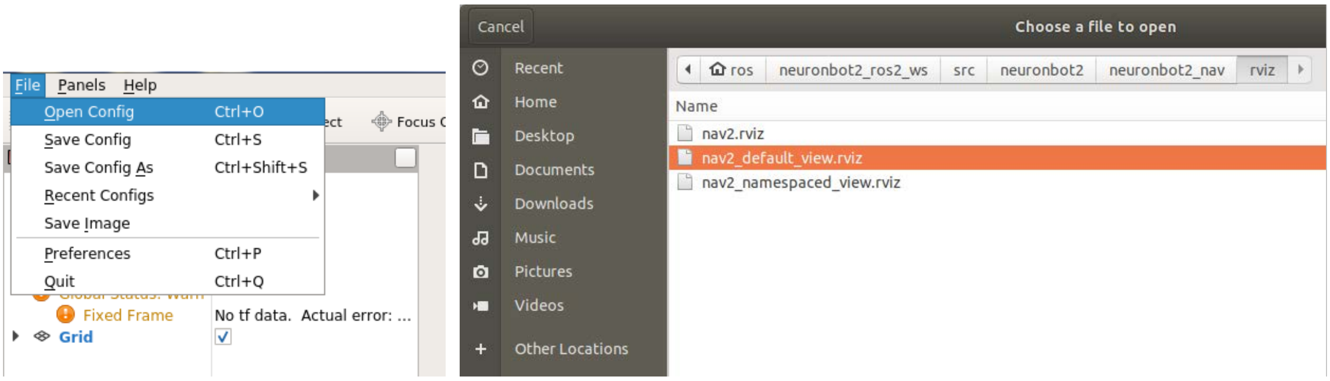


17. Set the goal.

    a. Click and drag "2D Nav Goal" to set the goal and orientation to any free space on the map. The NeuronBot should drive toward the goal by itself.



Tip: You can save time by opening the RViz2 config file in `~/neuronbot2_ros2_ws/src/neuronbot2/neuronbot2_nav/rviz/nav2_default_view.rviz`



6. Troubleshooting

6.1. Self-diagnosis

This section illustrates how to determine whether your NeuronBot is running normally or abnormally. The test scripts provided in this section leverage the NeuronBot ROS 1 package.

Note: Ensure that you download and build the latest source code before troubleshooting.

6.1.1. Motor Test

Execute the motor test diagnostic command as follows:

```
cd ~/neuronbot2_ros1_ws/src/neuronbot2/neuronbot2_tools/neuronbot2_init/
./neuronbot2_test.sh 1
```

“Motor controller version” and non-zero “RobotParameters” will appear, and NeuronBot will automatically spin a few times.

```
ROS_MASTER_URI=http://localhost:11311

process[neuronbot2_driver-1]: started with pid [3683]
process[joint_state_publisher-2]: started with pid [3684]
process[robot_state_publisher-3]: started with pid [3685]
process[rplidarNode-4]: started with pid [3691]
[ INFO] [1596795865.029211873]: [NeuronBot2] port: /dev/neuronbot2 baudrate: 115200
[ INFO] [1596795865.036691030]: [NeuronBot2] out_pid_debug_enable: 1
[ INFO] [1596795865.041410704]: [NeuronBot2] BaseDriver startup
process[led_control-5]: started with pid [3693]
[ INFO] [1596795865.050643410]: [NeuronBot2] Connected to main board
[ INFO] [1596795865.077069458]: RPLIDAR running on ROS package rplidar_ros. SDK Version:1.7.0
[led_control-5] process has finished cleanly
log file: /home/ros/.ros/log/2817bff8-d898-11ea-969c-00051bd1a109/led_control-5*.log
[ INFO] [1596795867.057327250]: [NeuronBot2] Motor controller version: v1.NB2.1 build time: 20200730
[ INFO] [1596795867.065932517]: [NeuronBot2] RobotParameters: 84 225 1428 10 75 2500 0 10 250 40 0 150 69
[ INFO] [1596795867.090422353]: [NeuronBot2] Subscribe cmd topic on [cmd_vel]
[ INFO] [1596795867.097385566]: [NeuronBot2] Advertise odom topic on [odom]
[ INFO] [1596795867.105268619]: [NeuronBot2] Advertise imu topic on [raw_imu]
RPLIDAR S/N: 75E99A87C5E392D2A5E49EF0E53C3D64
[ INFO] [1596795867.593538987]: Firmware Ver: 1.25
[ INFO] [1596795867.593573185]: Hardware Rev: 5
[ INFO] [1596795867.594950618]: RPLidar health status : 0
[ INFO] [1596795868.195467061]: current scan mode: Boost, max_distance: 12.0 m, Point number: 8.0K , angle_compensate: 2
```

If you do not see “Motor controller version” and non-zero “RobotParameters”, try the following:

- Perform TTY initialization again. For details, see NeuronBot Setup on page 33.
- Disconnect and reconnect all USB cables and restart NeuronBot.
- Execute the following command:

```
ls /dev/neuronbot2 -l
```

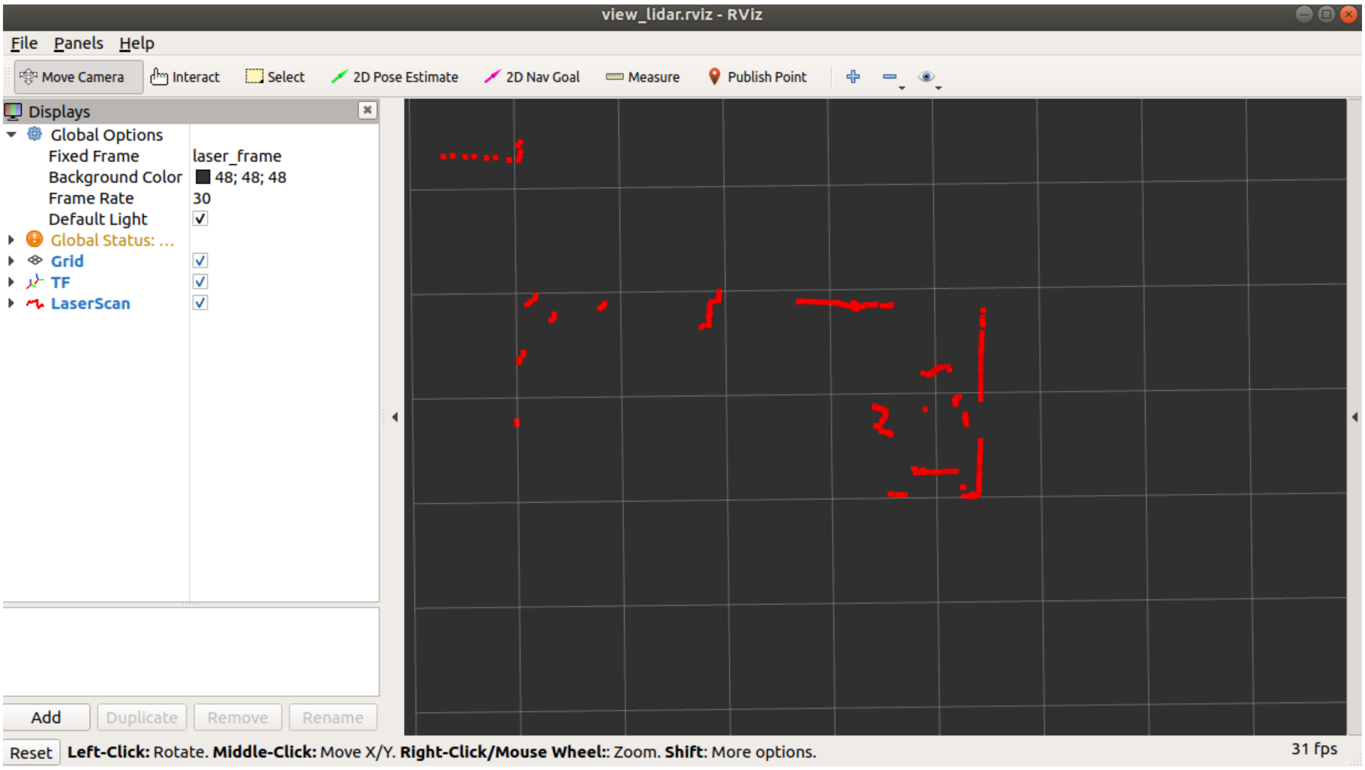
You should find /dev/neuronbot2 linked to /dev/ttyUSB*.

6.1.2. LiDAR Test

Execute the LiDAR diagnostic command as follows:

```
cd ~/neuronbot2_ros1_ws/src/neuronbot2/neuronbot2_tools/neuronbot2_init/
./neuronbot2_test.sh 2
```

RViz will open automatically:



If the LiDAR-generated red lines which do not appear, try the following:

1. Perform LiDAR initialization again. For details, see NeuronBot Setup on page 33.
2. Disconnect and reconnect all USB cables and restart NeuronBot.
3. Execute the following command:

```
ls /dev/rplidar -l
```

You should find /dev/rplidar linked to /dev/ttyUSB*.

6.1.3. LED Test

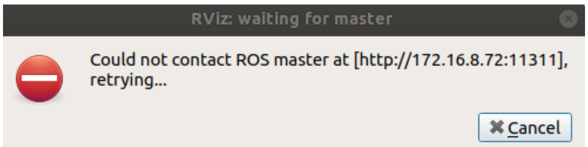
Execute the LED diagnostic command as follows:

```
cd ~/neuronbot2_ros1_ws/src/neuronbot2/neuronbot2_tools/neuronbot2_init/
./neuronbot2_test.sh 3
```

The LED color will change.

6.2. FAQ

1. Q: What causes “Could not contact ROS master…”?



A: The program cannot locate the ROS Master. Please verify that roscore is running and that ROS_IP and ROS_MASTER_URI are correctly set. For details, see ROS 1 Remote Control Settings on page 35.

2. Q: What causes “Command ‘XXX’ not found”?

```
ros@rostest:~/nb2_melodic_ws$ roslaunch neuronbot2_bringup bringup.launch

Command 'roslaunch' not found, but can be installed with:

sudo apt install python-roslaunch
```

or

```
ros@rostest:~/nb2_melodic_ws$ ros2 launch neuronbot2_bringup bringup.launch
ros2: command not found
```

A: The ROS environment may not be sourced. Source the environment or use the Neuron Startup Menu.

```
source /opt/ros/<YOUR_ROS_DISTRO>/setup.bash
```

3. Q: What causes “[xxx.launch] is neither a launch file …”?

```
ros@rostest:~/nb2_melodic_ws$ roslaunch neuronbot2_bringup bringup.launch
RLException: [bringup.launch] is neither a launch file in package [neuronbot2_bringup] nor is [neuronbot2_bringup] a launch file name
The traceback for the exception was written to the log file
```

A: The NeuronBot ROS 1 environment may not be sourced. Source the environment to fix this issue.

```
source ~/neuronbot2_ros1_ws/devel/setup.bash
```

4. Q: What causes “Package ‘neuronbot2_xxx’ not found …”?

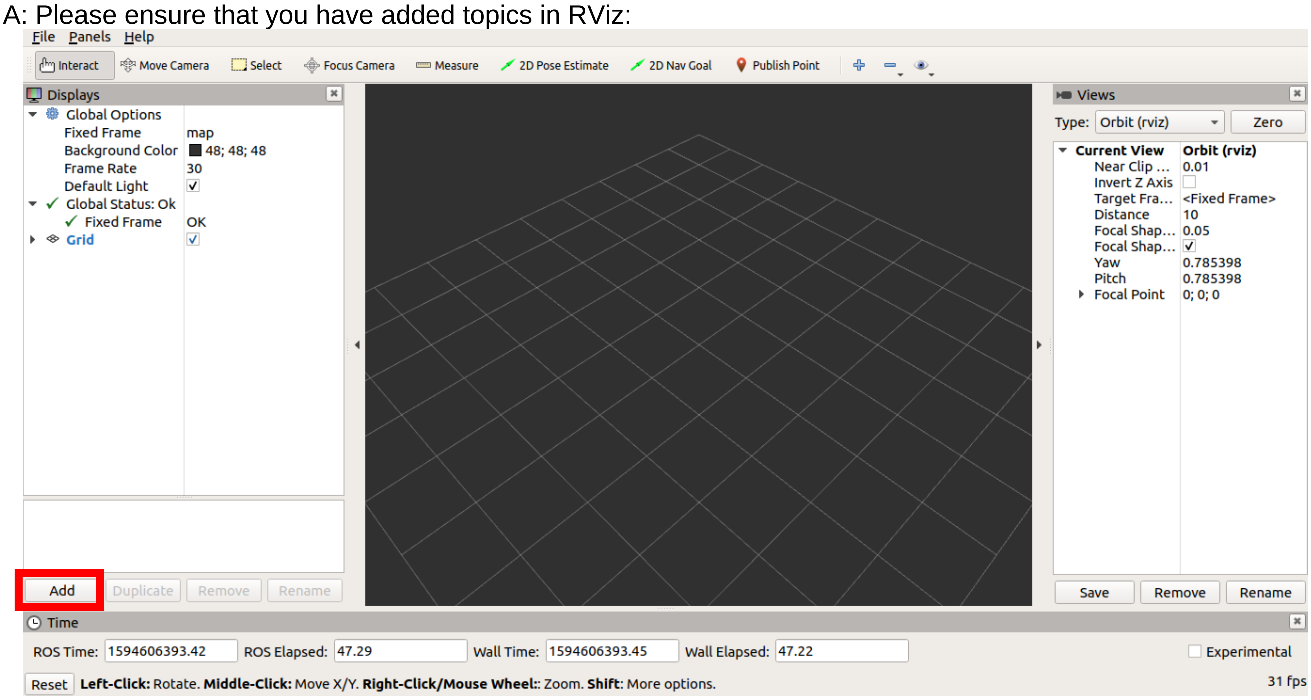
```
ros@rostest:~/nb2_eloquent_ws$ ros2 launch neuronbot2_bringup bringup_launch.py
Package 'neuronbot2_bringup' not found: "package 'neuronbot2_bringup' not found, searching: ['/opt/ros/eloquent']"
```

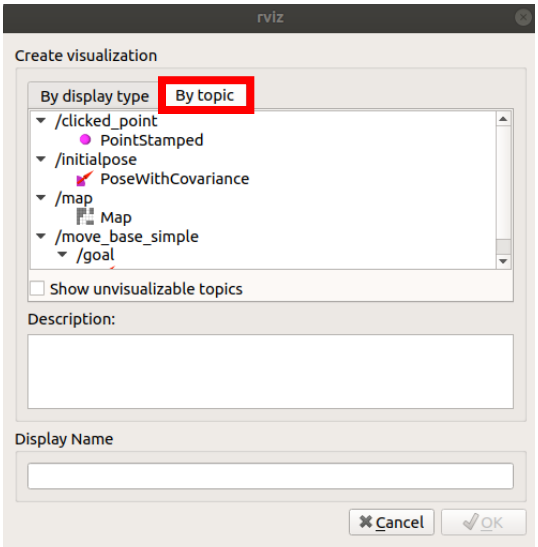
A: The NeuronBot ROS 2 environment may not be sourced. Source the environment to fix this issue.

```
source ~/neuronbot2_ros2_ws/install/local_setup.bash
```

5. Q: Why won’t RViz display anything?

A: Please ensure that you have added topics in RViz:





Click **Add**, and then click the **By topic** tab. If you can't find the topics you expect to see, try the following:

- For ROS 1, ensure that ROS_MASTER_URI and ROS_IP are correctly set on both the host computer and NeuronBot.
- For ROS 2, ensure that ROS_DOMAIN_ID is correctly set on both the host computer and NeuronBot.

See ROS 1 Remote Control Settings and ROS 2 Remote Control Settings for details on setting environment variables.

7. System Backup and Restore

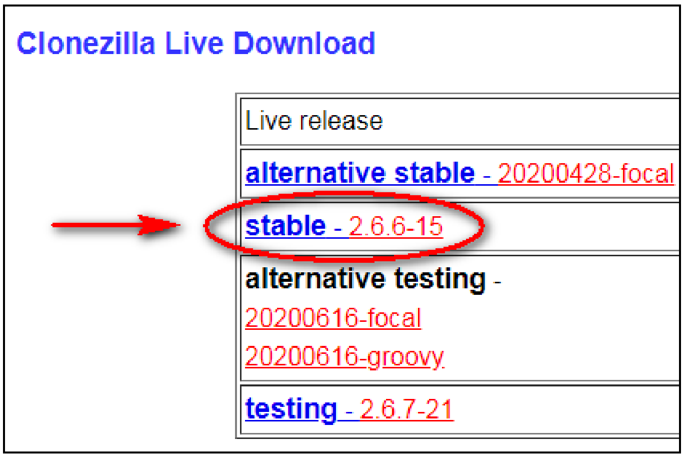
This section explains how to create a bootable USB drive for backing up and restoring the system.

7.1. Preparation

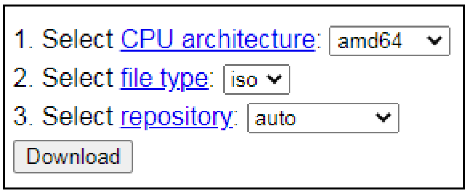
7.1.1. Clonezilla

Clonezilla is an open source tool for backup and restoration.

1. Download the stable version from the official website: https://clonezilla.org/downloads.php



2. Select **amd64** for **CPU architecture** and **iso** for **file type**, and then click **Download**.
   Clonezilla will automatically start to download.



3. After downloading the Clonezilla ISO file, load the ISO file onto your USB drive and make it bootable.
   If you're unable to do this, we recommend using **Rufus** instead.

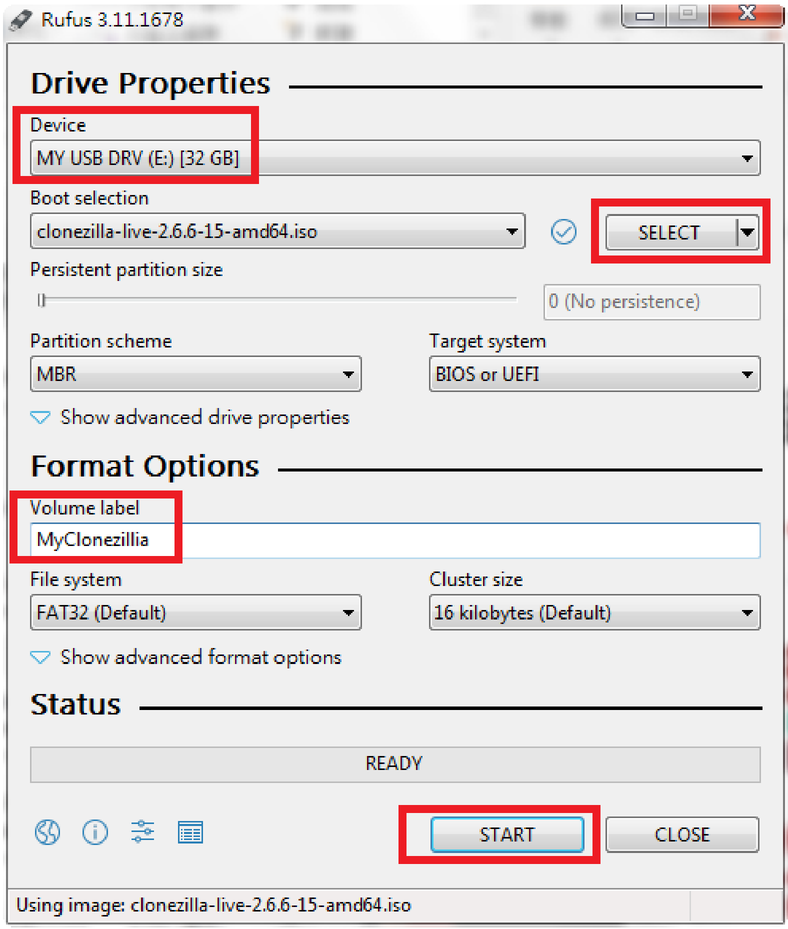
7.1.2. Rufus

Rufus is a Windows program for creating bootable USB drives. This section illustrates how to create a bootable Clonezilla USB drive.

Note:

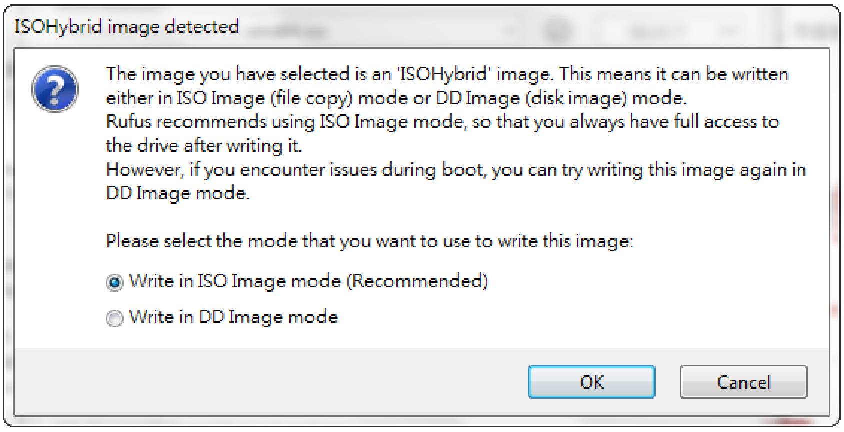
- Ensure that you have downloaded the Clonezilla ISO before performing the following procedure.
- We recommend using an empty USB drive with at least 32GB of free space. Creating a bootable USB drive erases all data on the drive.

1. Download Rufus from the official website: https://rufus.ie/
2. Start Rufus.
3. Select the USB device and Clonezilla ISO
4. Specify a name for **Volume label**.
5. Click the **START** button to load the ISO onto the USB drive.



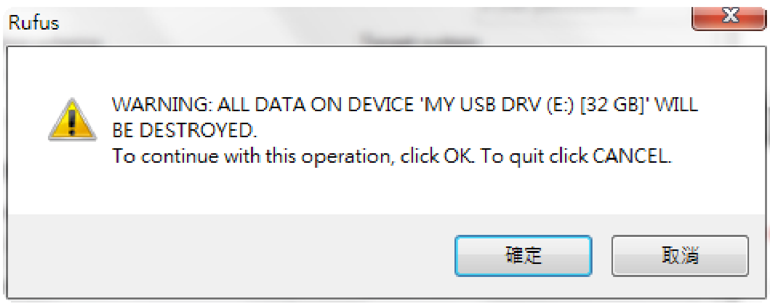
The **ISOHybrid image detected** window will appear.

6. Select **Write in ISO image mode (Recommended)** and click **OK**.

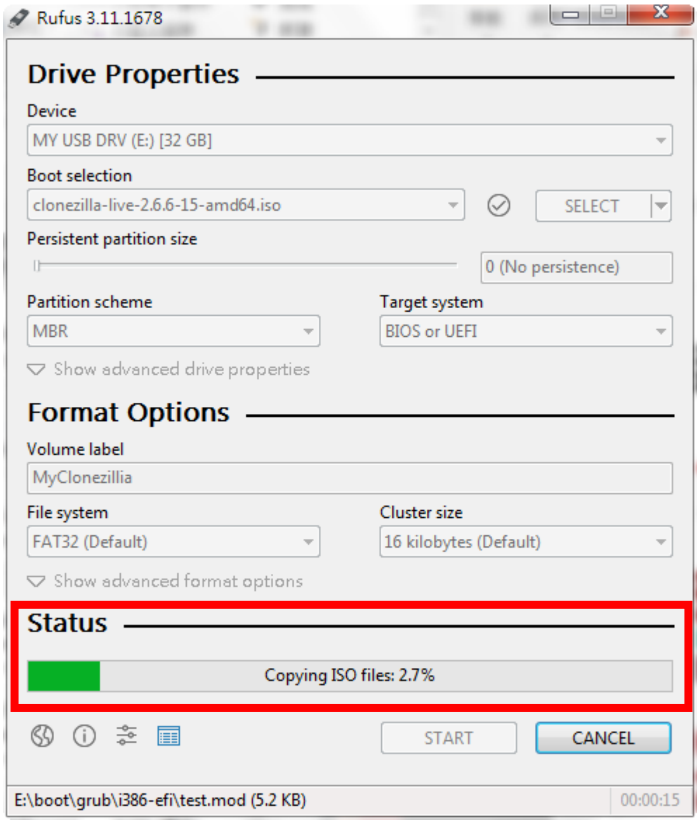


A warning message will appear to notify you that all data on the USB drive will be erased.

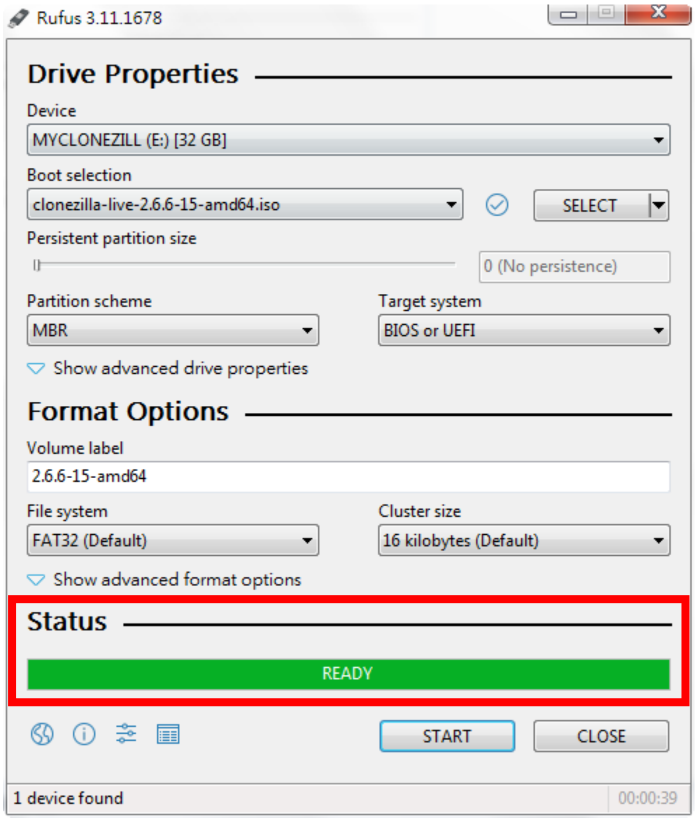
7. Click **OK**.



Rufus will being writing to the USB drive.



When the process is completed, the **Status** will show **READY**.



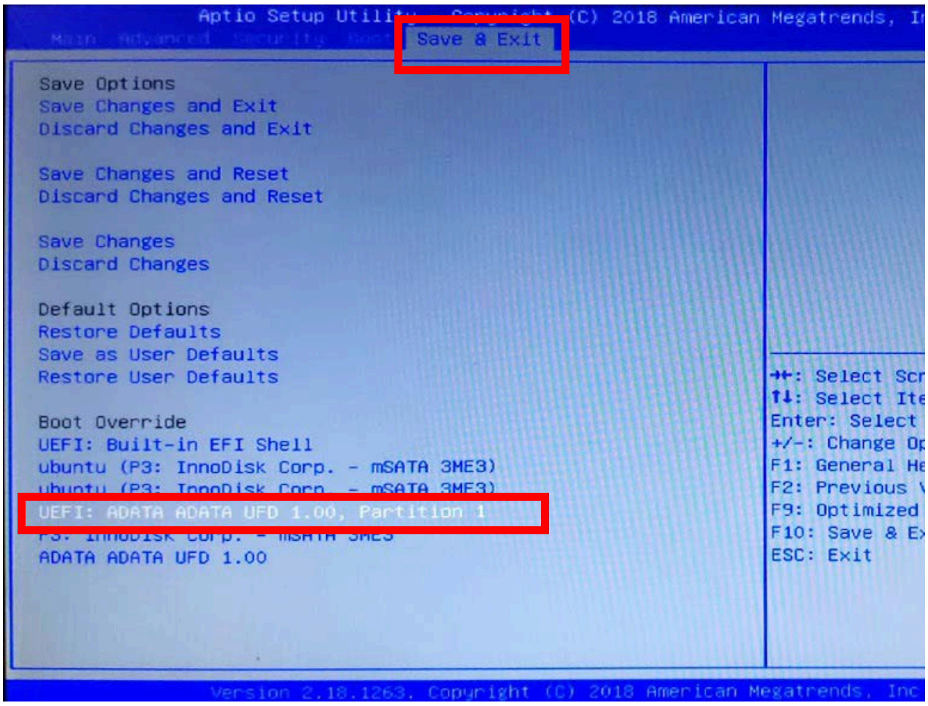
8. Click **CLOSE** to exit the program.

The bootable Clonezilla USB drive is now ready. You can use the Clonezilla USB drive to back up and restore the system.

7.2. Full Disk Backup

The following steps describe how to create a compressed backup image.

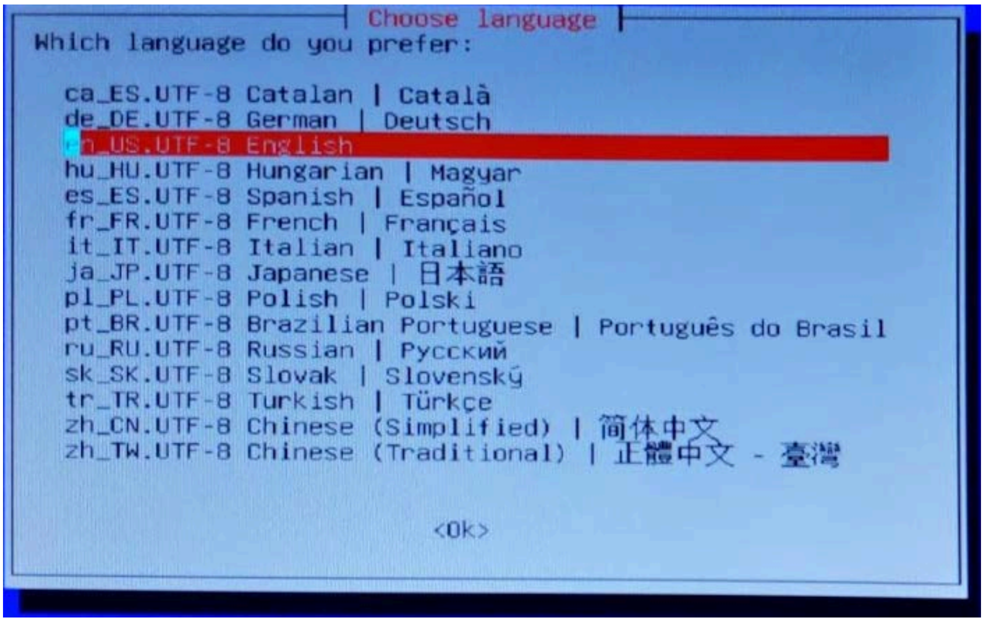
1. Insert the Clonezilla USB drive into the USB port on NeuronBot
2. Power on Neuronbot.
3. When the ADLINK boot logo appears on the screen, press the **Delete** key a few times. NeuronBot will enter BIOS mode.
4. Go to the **Save & Exit** tab and select **UEFI: <YOUR-CLONEZILLA-USB-DRIVE>**.



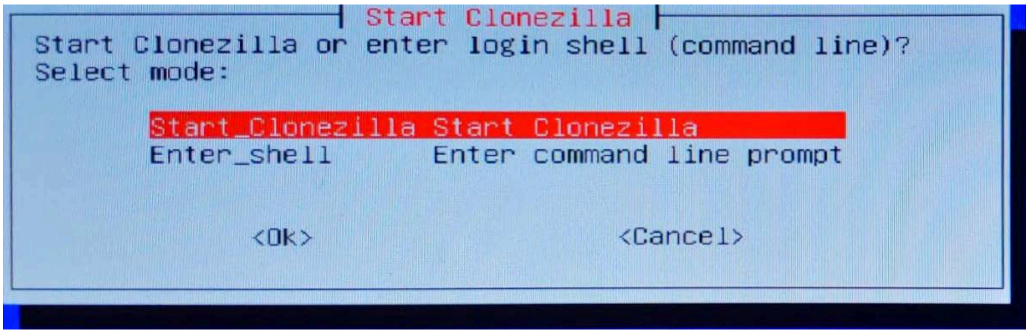
NeuronBot will boot using the inserted Clonezilla USB drive. The screen will display the Clonezilla GNU GRUB.

5. Select **Clonezilla live (To RAM, boot media can be removed later)** in the menu that appears.

6. Select your language.

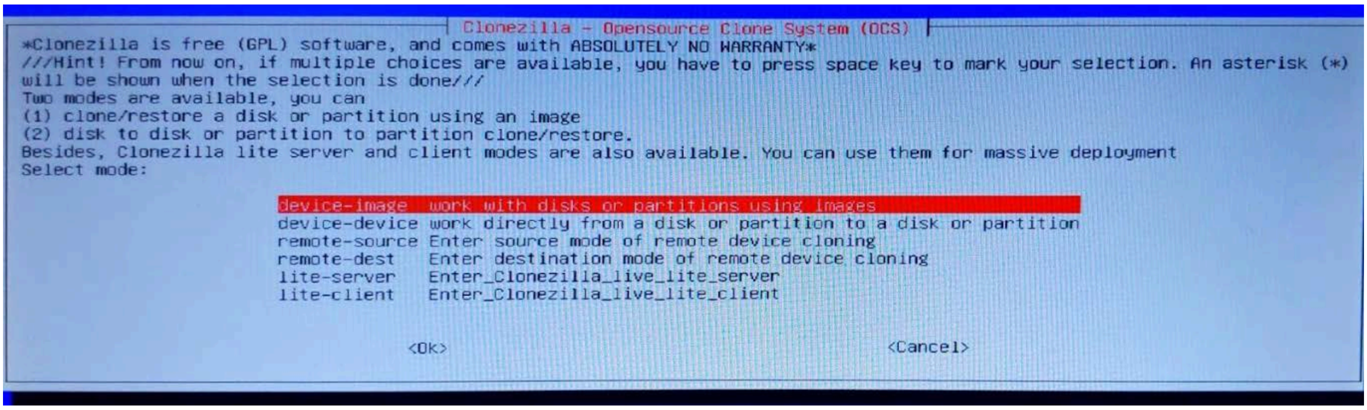


7. Select **Start_Clonezilla**.

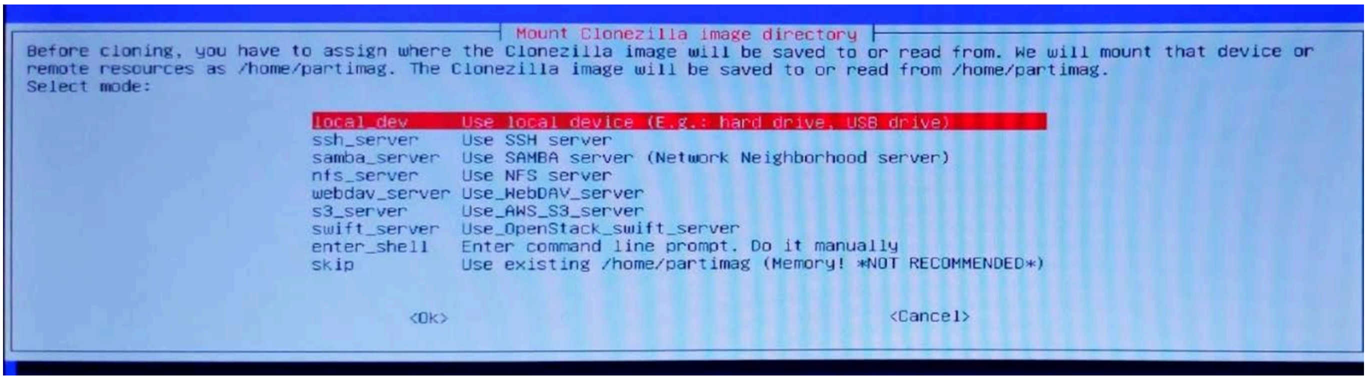


8. Select **device-image** (for creating a backup image).

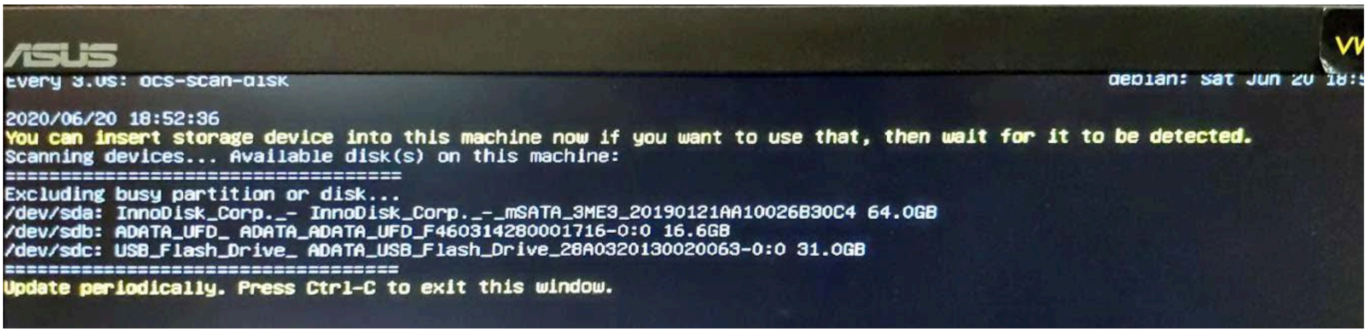
Tip: Use the arrow keys to change options and press the spacebar to confirm your selection.



9. Select **local_dev** (to store the image on the USB drive).

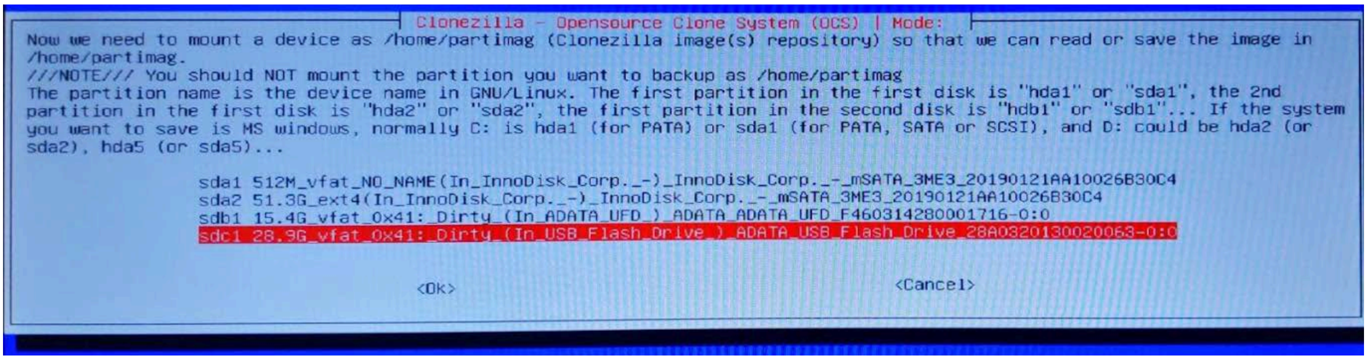


10. (Optional) Insert another USB drive to store the image. To save the image onto the same USB drive as Clonezilla, press **Ctrl-C**.



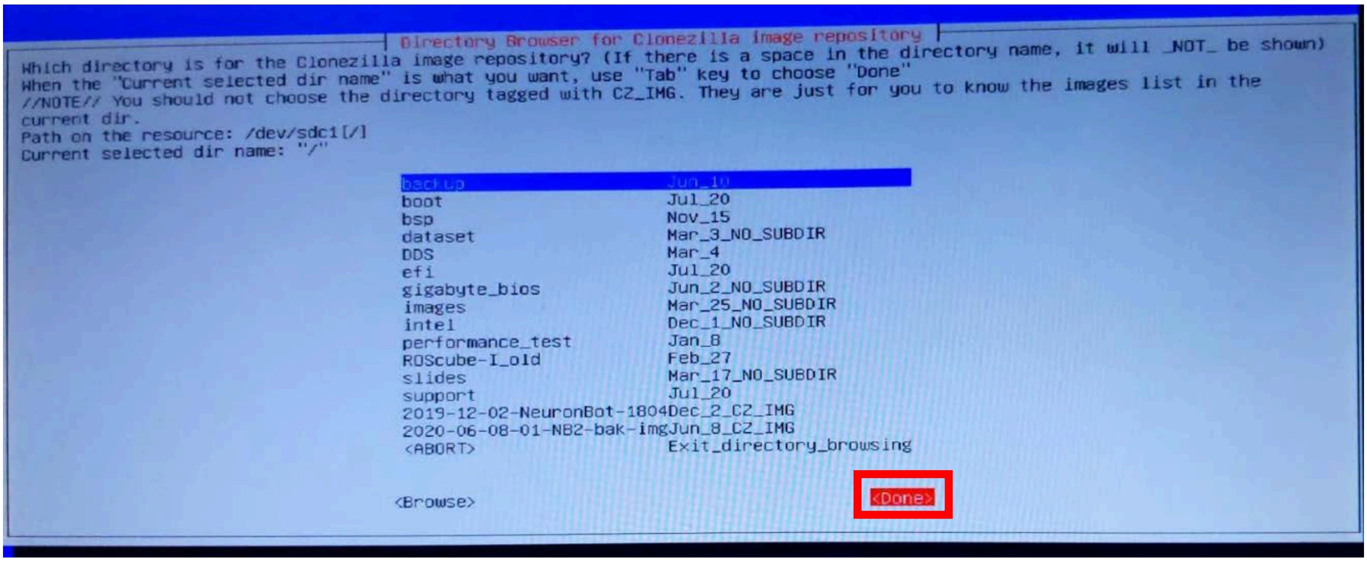
11. Select the disk where you want to store the image.

Tip: You can use the Clonezilla USB drive if it has enough free space.

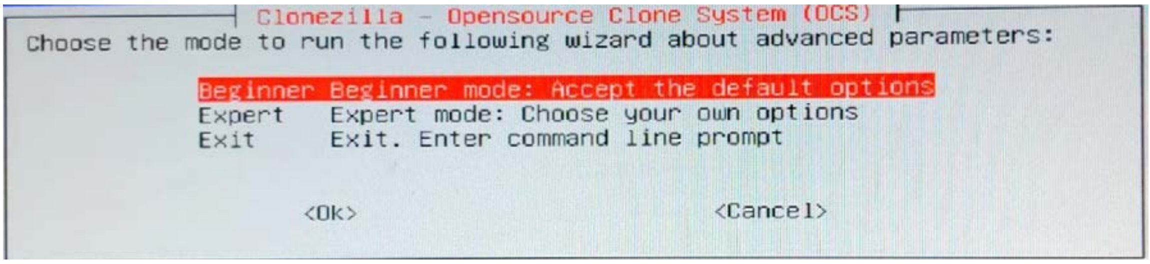


The files on the disk will display to indicate that you selected the correct disk.

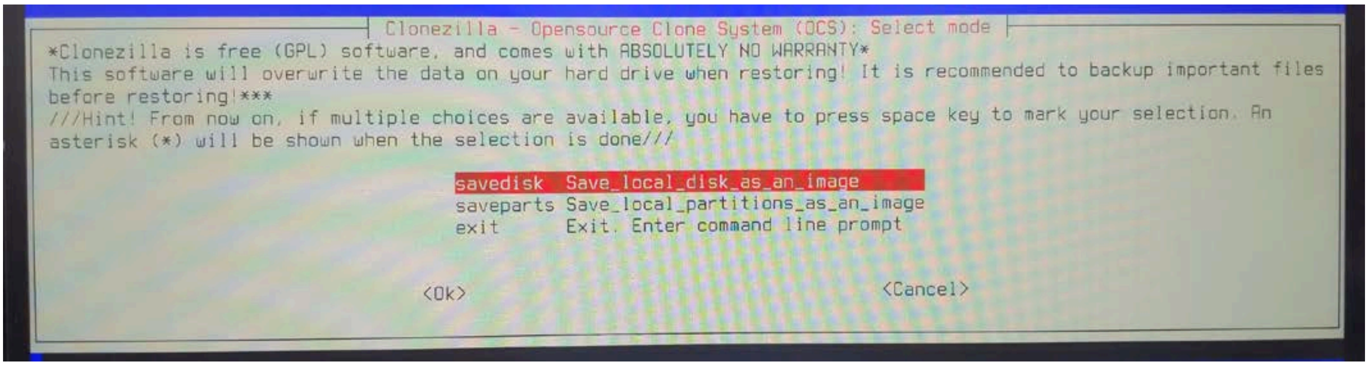
12. Press the right arrow key to highlight <Done>, and then press the spacebar to confirm.



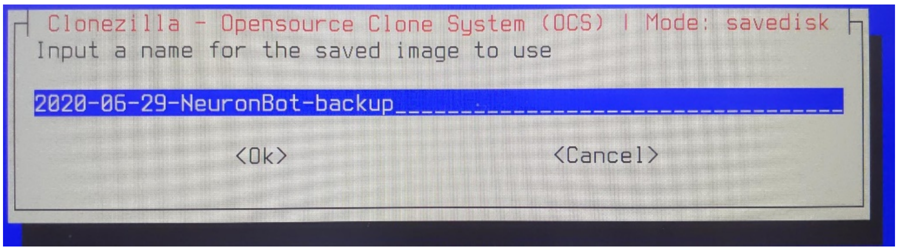
13. Select **Beginner** (to use default options).



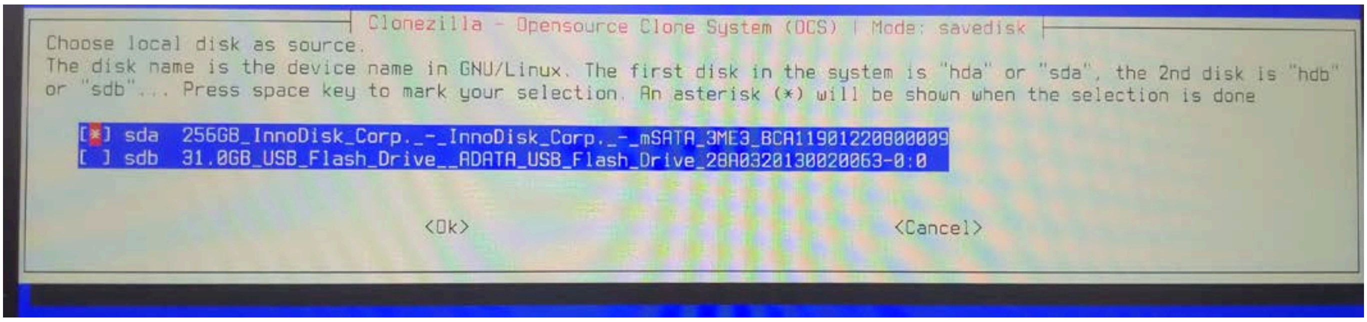
14. Select **savedisk** (to back up the entire disk).



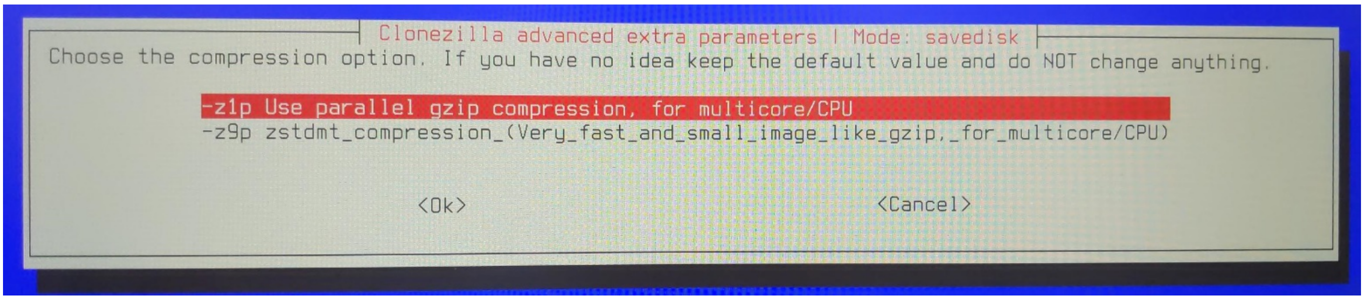
15. Specify a name for the image file.



16. Select the disk where you want to save the image file.

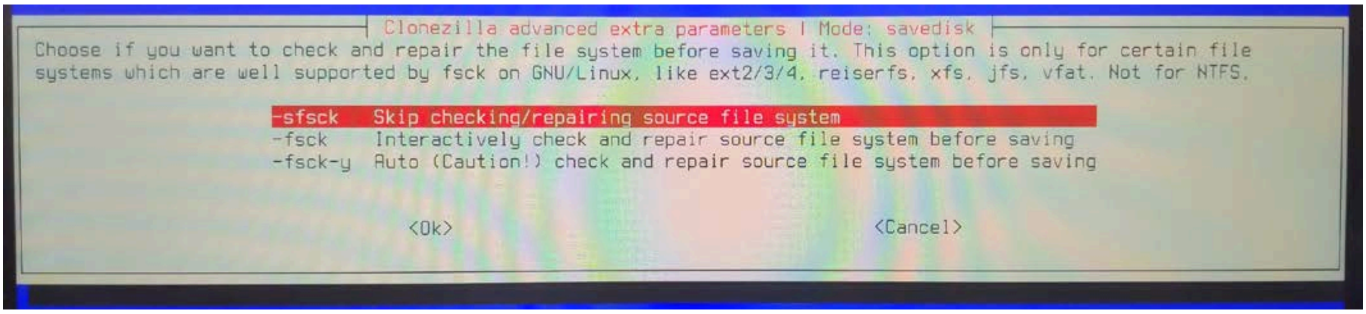


17. Select the default compression option.

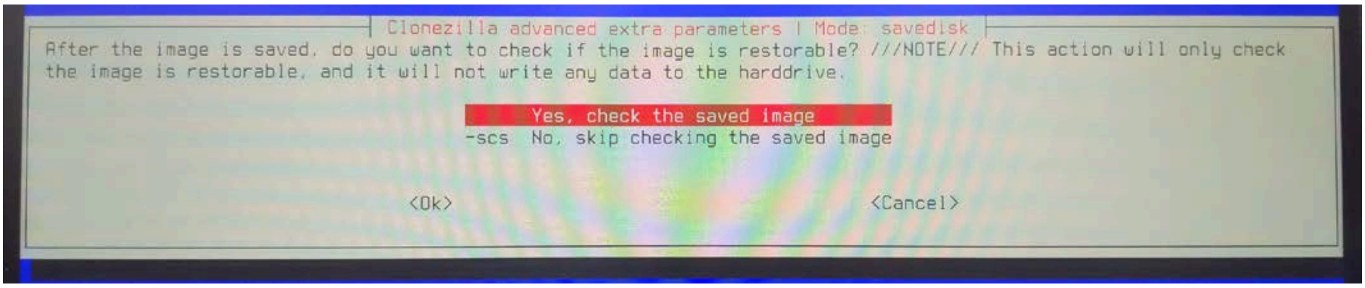


18. Select **-sfsck** to skip checking the file system.

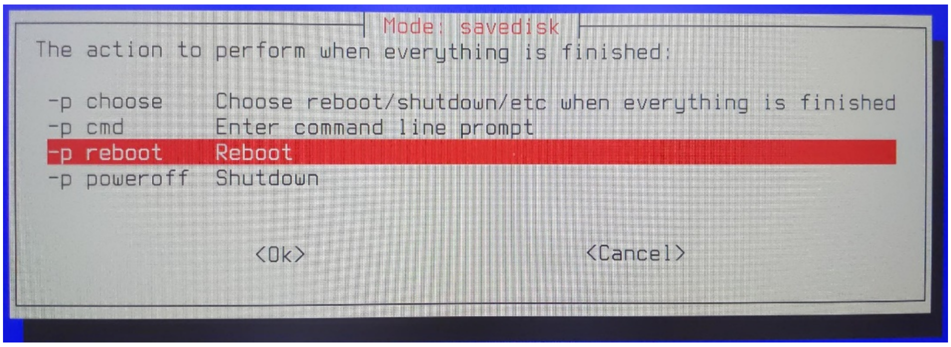
Note: If you want to check the file system, select **-fsck**.



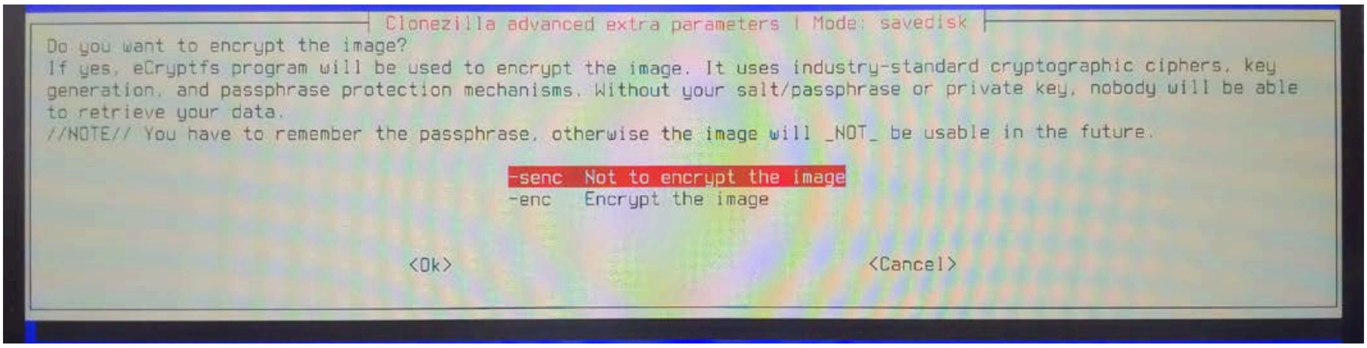
19. Select **Yes** to check whether the saved image is restorable.



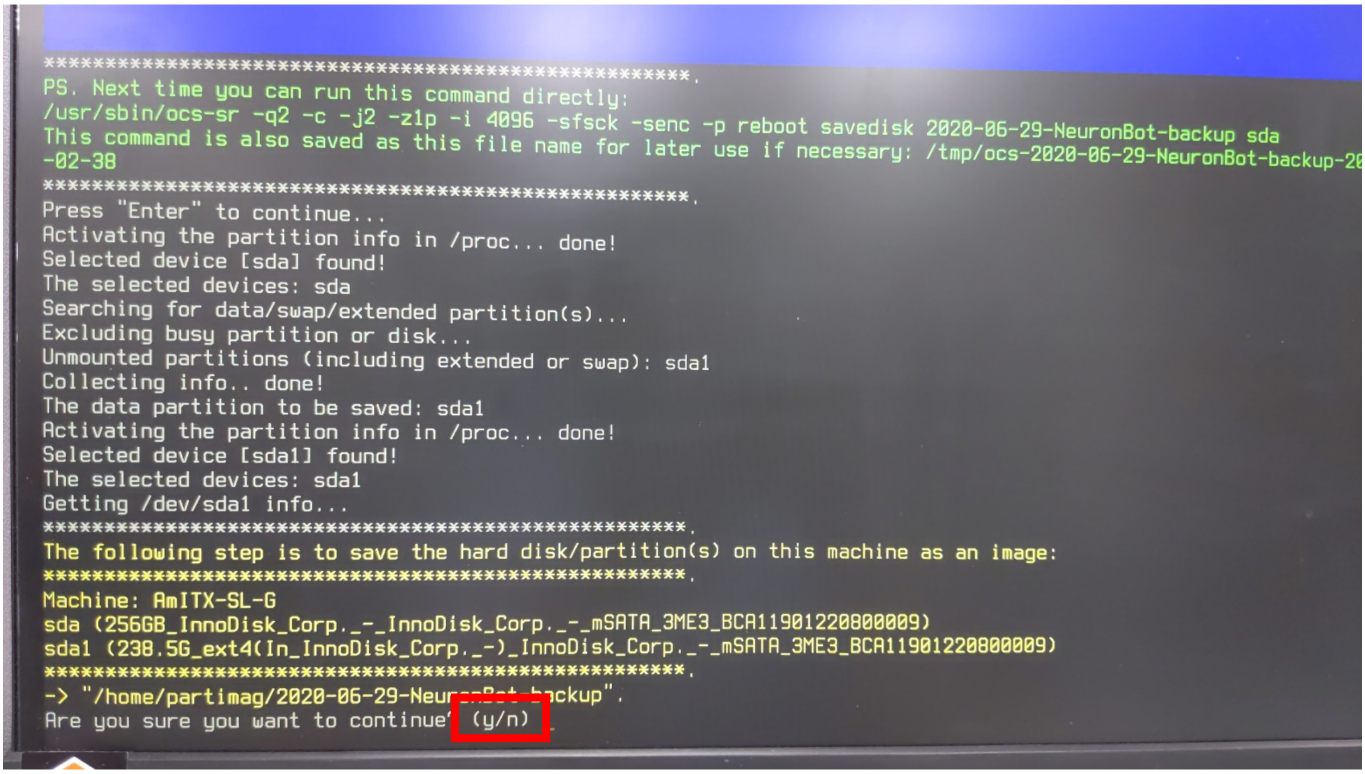
20. Select **reboot**.



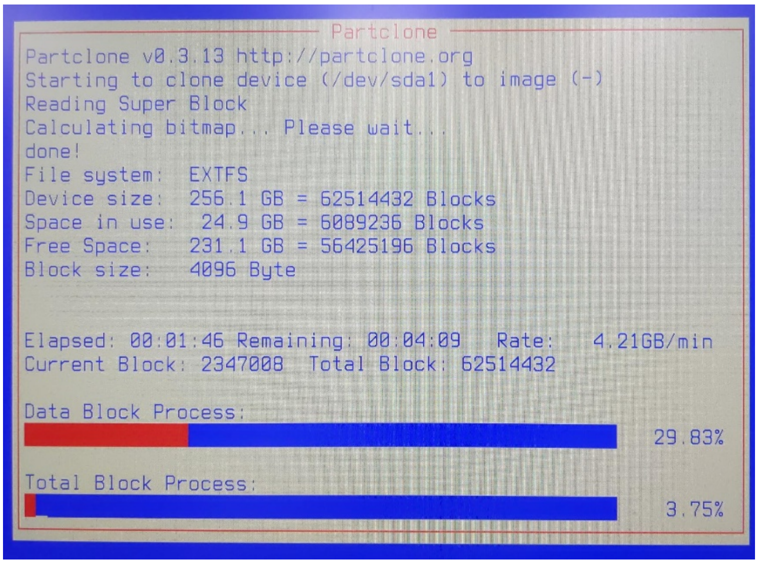
21. Select **-senc** to skip image encryption.



22. Press **y** to confirm savedisk.



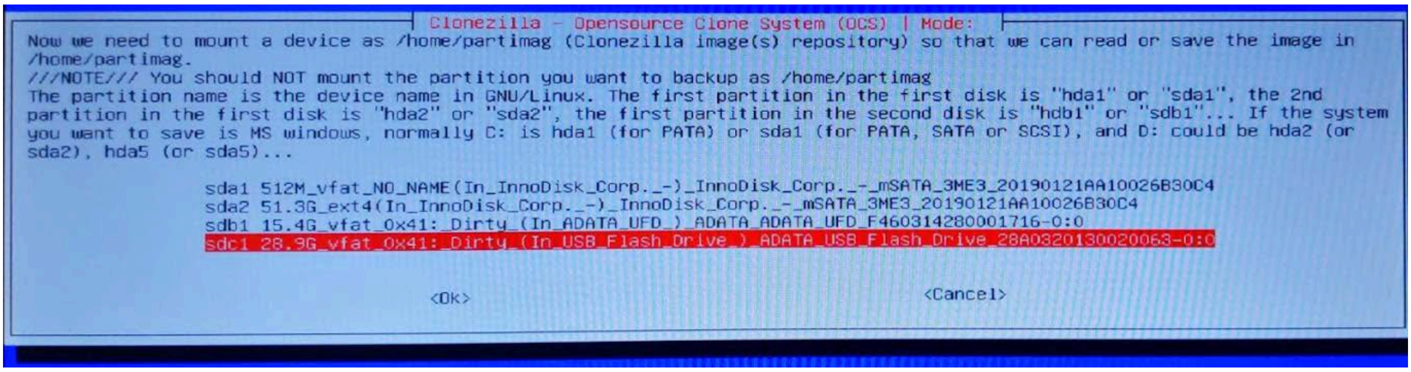
Clonezilla will back up the disk backup to an image file. Please wait while the disk is backed up.



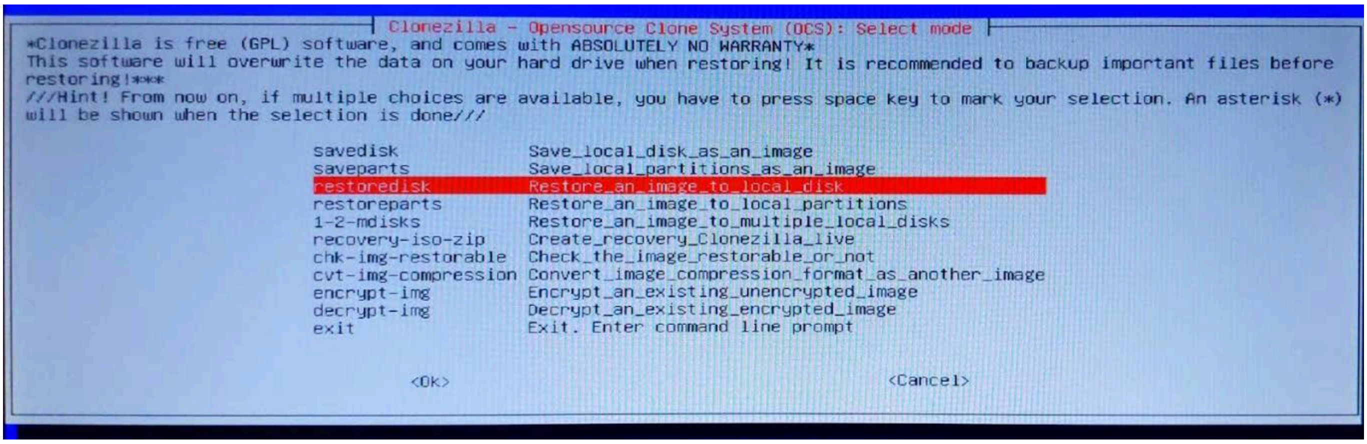
7.3. Full Disk Restoration

Restoring the system from a backup image is similar to the Full Disk Backup process. The following instructions contain only the steps that differ from the backup process.

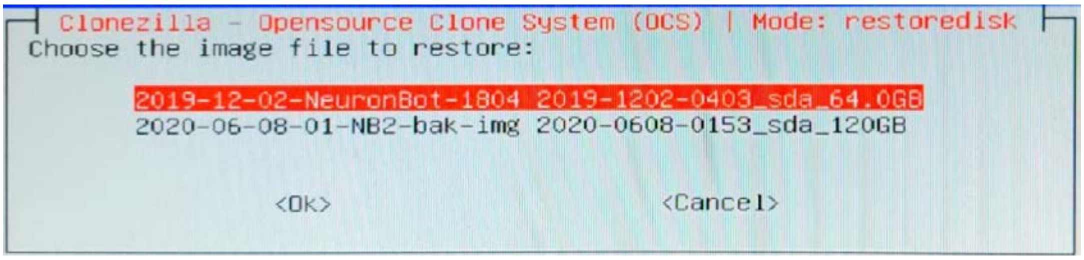
1. Select the disk where the image is saved.



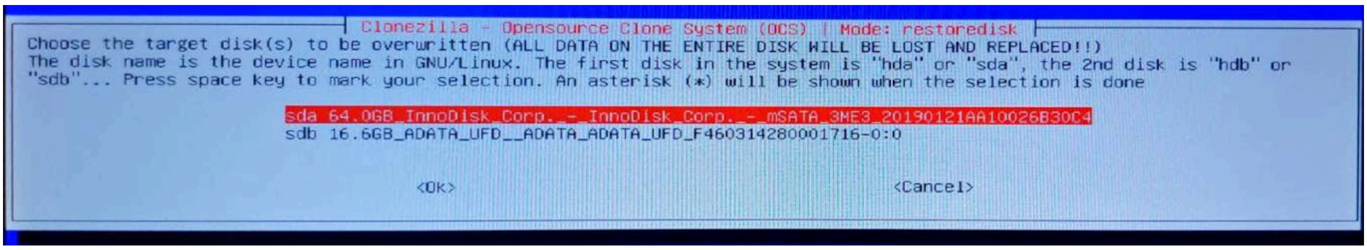
2. Select **restoredisk** to restore the system.



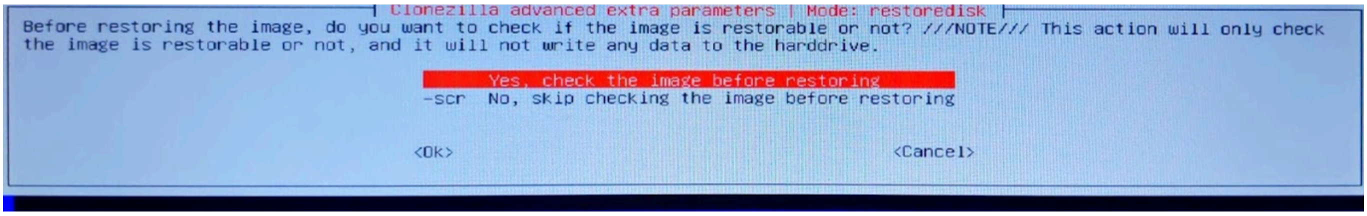
3. Select the image source.



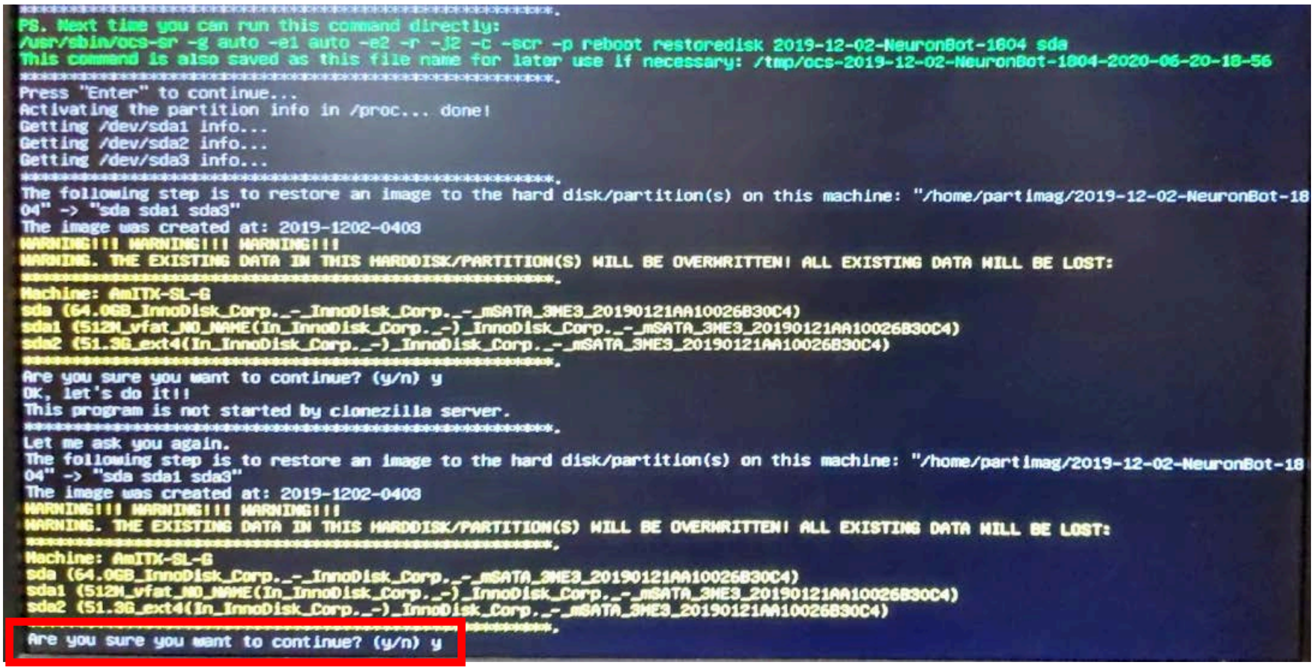
4. Select the target disk to be restored.



5. Select **Yes** to check the image before restoring (recommended if the image has not been used for a long time); otherwise, choose **No** to skip the check.

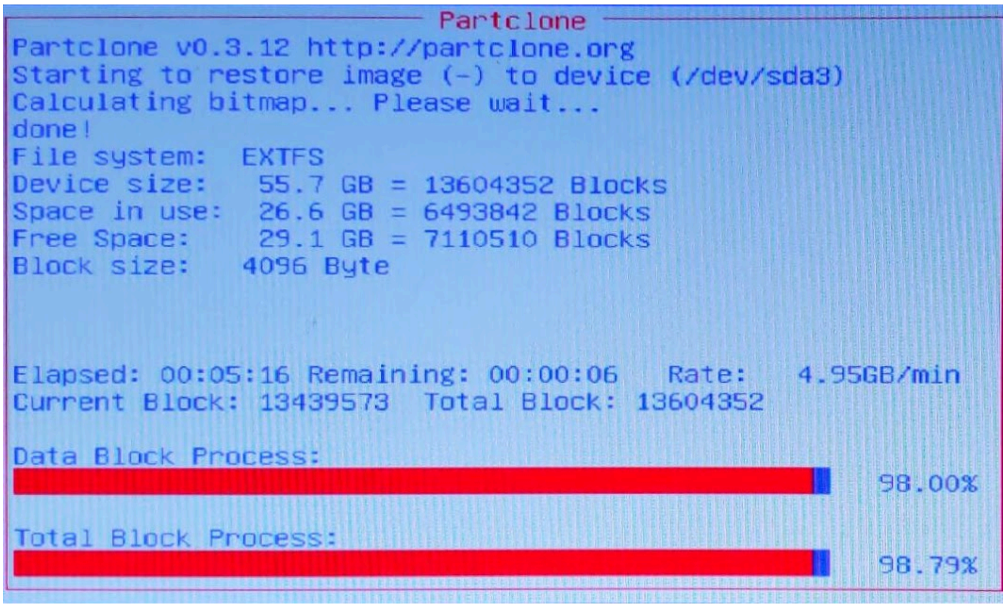


Clonezilla will ask you to confirm (the disk to be restored will be erased).



6. Type **y** and press **Enter** to confirm.

Clonezilla will restore the disk from the backup image. Please wait while the disk is restored.



8. Safety Instructions

For user safety, please read and follow all instructions marked on the product and documentation before handling/operating the device. Please retain all safety and operating instructions for future reference.

- Read these safety instructions carefully
- Keep this User's Manual for future reference
- Read the Specifications section of this manual for detailed information on the recommend operating environment for this equipment.
- When installing/mounting or uninstalling/removing equipment, turn off the power and unplug any power cords/cables.
- To avoid electrical shock and/or damage to equipment:
  - Keep equipment away from water or liquid sources.
  - Keep equipment away from high heat or high humidity.
  - Keep equipment properly ventilated (do not block or cover ventilation openings).
  - Make sure to use recommended voltage and power source settings.
  - Always install and operate equipment near an easily accessible electrical socket-outlet.
  - Secure the power cord (do not place any object on/over the power cord).
  - Only install/attach and operate equipment on stable surfaces and/or recommended mountings.
  - If the equipment will not be used for long periods of time, turn off and unplug the equipment from its power source.
- Never attempt to fix the equipment. Equipment should only be serviced by qualified personnel.

9. Getting Service

Ask an Expert: http://askanexpert.adlinktech.com

ADLINK Technology, Inc.
Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
Address: 5215 Hellyer Avenue, #110, San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
Address: 300 Fang Chun Rd., Zhangjiang Hi-Tech Park, Pudong New Area
Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology GmbH
Hans-Thoma-Straße 11
D-68163 Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: germany@adlinktech.com

Please visit the Contact page at www.adlinktech.com for information on how to contact the ADLINK regional office nearest you.