MYBOTSHOP

QUADRUPED ROBOTICS

Unitree B1

User Manual

CONTACT

info@quadruped.de

support@quadruped.de

quadruped.de

Contents

1 Attention

Attention: Read this document carefully before operating the robot. In case of doubt, consult specialists, experts, or the manufacturer (Unitree Robotics) or distributor (MYBOTSHOP / QUADRUPED Robotics) of the assemblies used. The robot must not be put into operation before clarification. Contact information is available in Chapter 9, Getting Help & RMA.

It is highly recommended to view the official B1 quick start manual and user manual (linked in Chapter 6, Operation) before using the robot, to ensure safe and correct operation.

Note: This documentation assumes you are comfortable working with ROS. If you are not familiar with ROS already, we recommend starting with the official ROS tutorials first.

2 Disclaimer

The B1 robot documented here is sold and supported by *MYBOTSHOP / QUADRUPED Robotics* as the German distributor for Unitree Robotics. Basic ROS understanding is required to operate the software stack covered in this manual — if you are not familiar with ROS, we recommend checking the ROS documentation first.

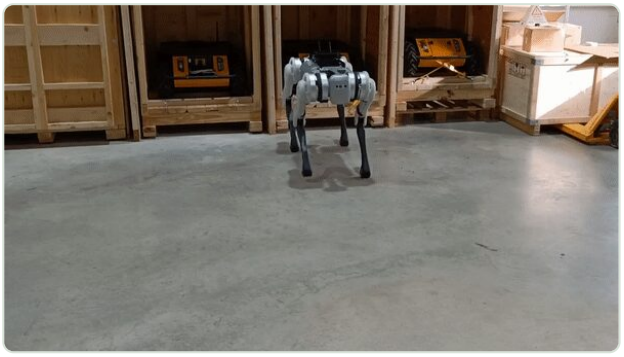
The client acknowledges and agrees that any information or materials provided by MYBOTSHOP / QUADRUPED Robotics are for R&D and development purposes only. Any services are provided "AS IS" and without any representation or warranty of any kind, express or implied, including but not limited to any warranty of merchantability, fitness for a particular purpose, non-infringement, or any other warranty.

This product is intended for civilian use only. Users should avoid making dangerous modifications or using the robot in hazardous ways. Some software components described in this manual are still under active development (noted inline where applicable — e.g. low-level driver mode and the camera package) — always confirm current status with MYBOTSHOP / QUADRUPED Robotics support before relying on them. For terms, policies and compliance with local laws and regulations, refer to the Unitree Robotics website and the RMA/support process in Chapter 9.

MYBOTSHOP / QUADRUPED Robotics shall not be liable for any damages, including but not limited to direct, indirect, special, incidental, or consequential damages, arising out of or in connection with the use or inability to use the information or materials in this document. This limitation on liability shall apply regardless of the form of action, whether in contract, tort, or otherwise.

3 About the B1

Overview



_images/b1_movearound.gif

This package supplies Sphinx-based tutorial content to assist you with setting up and operating your B1 quadruped robot. The tutorials topics are listed in the left column, and presented in the suggested reading order.

Warning: These tutorials assume that you are comfortable working with ROS. We recommend starting with our ROS tutorial if you are not familiar with ROS already.

4 Pre-requisite Software – ROS Setup

Pre-requisite Software

The ROS driver for the robots is initially supported for the ROS Melodic . All of ROS related software should be install either on remote PC/Rasberry Pi/Nvidia board on robot. The robot ships either with Rasberry Pi or Nvidia board as an upgradation kit. The steps for shown below are for remote PC both boards usually come with pre-installed ROS distribution.

Note: All the ROS related software should be installed/run on remote PC/Rasberry Pi/Nvidia board, nothing needed to be installed on robots control board.

[Ubuntu 20.04 - ROS NOETIC]

_images/ros_noetic.png

The ROS driver for the robots is initially supported for the ROS Noetic . All of ROS related software should be installed in a Remote-PC . The steps for shown below are for Remote-PC or B1-PC .

> **Warning:** Care should be taken when installing ROS so that it matches with your CPU's architecture (i.e. armhf, amd64, arm6, etc.)

PC Setup

ROS Noetic installation requires Ubutnu 20.04.

1. Download Ubuntu 20.04
2. Follow the Ubuntu 20.04 Installation guide

ROS-noetic Installation

The ros driver provided can be run either on a remote PC or on board robot computer. Usually, the on-board computer already has pre-installed ROS distribution, so the instructions below will be applicable to a remote computer. You may run the following commands to install ROS noetic or you can simply follow the instructions from the roswiki.

1. Enter the Ubuntu terminal and typie in the following command:

```
cd
```

1. The next step is setting up the source list:

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

1. After which you will set up the keys:

```
sudo apt install curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

1. Then, you will install ROS-noetic:

```
sudo apt update
sudo apt install ros-noetic-desktop-full
```

1. Finally, you will add ROS environment to your bash file, what this means is that each time you load up your terminal, it will automatically find the built-in ROS packages.

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

1. Install python dependencies and initialize repo:

```
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-
wstool build-essential
sudo rosdep init
rosdep update
```

Once the ROS packages are set-up the next step would be to set up a workspace in which you can add pre-built packages to control your robot.

Catkin Workspace

1. The first step is to ensure that you have correctly setup you ROS noetic. It can be checked by:

```
cd ..
source /opt/ros/noetic/setup.bash
```

1. Next, install catkin_tools:

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

1. The next step is to create, your workspace. This is typically created in the home directory following the standard ROS convention, once built, you should see a build and devel folder next

to your src folder:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/
catkin build
```

1. To enable the workspace, you must type in:

```
source devel/setup.bash
```

1. To avoid typing this everytime you start a terminal, you'll add this into the last line of the bashrc file:

```
gedit ~/.bashrc
```

- Add the following line at the end of the bashrc file:

```
source /home/<your_computer_username>/catkin_ws/devel/setup.bash
```

- Everytime you do a catkin build , it is necessary to either restart all your terminals, or source them.
- To verify that everything is working correctly, type the following into the terminal:

```
echo $ROS_PACKAGE_PATH
```

[Ubuntu 18.04 - ROS MELODIC]

_images/melodic.jpg

The ROS driver for the robots is initially supported for the ROS Melodic . All of ROS related software should be installed in a remote PC/Rasberry Pi/Nvidia board that is onboard the robot. The robot ships either with Rasberry Pi or Nvidia board as an upgradation kit. The steps for shown below are for remote PC both boards usually come with pre-installed ROS distribution.

Note: All the ROS related software should be installed/run on remote PC/Rasberry Pi/Nvidia board, nothing needs to be installed on robots control board.

PC Setup

ROS Melodic installation requires Ubutnu 18.04 Bionic, Artful and supports macOS, Windows upto varying extent. But for this installation Ubuntu 18.04.x Bionic is recommended.

1. Ubuntu 18.04 can be downloaded from here .
2. The installation instruction for Ubuntu can be found here

ROS-Melodic Installation

The ros driver provided can be run either on a remote PC or on board robot computer. Therefore, ROS only needs to be installed on any one of them. Usually, the on board computer have already pre-installed ROS distribution, so below one will only be applicable to a remote computer. Run following commands to install ROS Melodic or you can simply follow the instructions .

1. Enter the Ubuntu terminal and typie in the following command:

```
cd ..
```

1. The next step is setting up the source list:

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

1. After which you will set up the keys:

```
sudo apt install curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

1. Then, you will install ROS-melodic:

```
sudo apt update
sudo apt install ros-melodic-desktop-full
```

1. Finally, you will add ROS environment to your bash file, what this means is that each time you load up your terminal, it will automatically find the built-in ROS packages.

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

Once the ROS packages are set-up the next step would be to set up a workspace in which you can add pre-built packages to control your robot.

Catkin Workspace

1. The first step is to ensure that you have correctly setup you ROS melodic. It can be checked by:

```
cd ..
source /opt/ros/melodic/setup.bash
```

1. Next, install catkin_tools:

```
sudo apt-get install python3-catkin-tools
```

1. The next step is to create, your workspace. This is typically created in the home directory following the standard ROS convention, once built, you should see a build and devel folder next to your src folder:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/
catkin build
```

1. To enable the workspace, you must type in:

```
source devel/setup.bash
```

1. To avoid typing this everytime you start a terminal, you'll add this into the last line of the bashrc file:

```
gedit ~/.bashrc
```

- Add the following line at the end of the bashrc file:

```
source devel/setup.bash
```

- Everytime you do a Catkin build it is necessary to either restart all your terminals, or source them.
- To verify that everything is working correctly, type the following into the terminal:

```
echo $ROS_PACKAGE_PATH
```

5 Quick Start

Quick-Start

Once you are connected to the B1 via LAN or 5G and have completed Pre-requisite System section you can begin operating the B1.

B1 High-level Driver

In high level mode the robot can walk. To start the driver please run:

```
sudo su
source ~/catkin_ws/devel/setup.bash
roslaunch b1_base base.launch
```

B1 Visualization

To launch the b1_viz:

```
roslaunch b1_viz view_robot.launch
```

Quick-Start Teleop

To tele-operate:

```
rosrun teleop_twist_keyboard teleop_twist_keyboard.py cmd_vel:=/b1/cmd_vel
```

6 Operation – Manuals, Apps & Network

B1 Manuals

_images/b1_goaround.gif

Important: The following text are hyperlinks to demonstratory videos. Click on them to get redirected to the websites.

- B1 User Manual
- B1 Unitree Github
- Manufacturers Documentation (Requires Translation Plugin)

Attention: It is highly recommended to view the quick start manual and user manual before using the robot to ensure safe and correct operation.

B1 Apps

- B1 iOS
- B1 Android

In the app, you would have to go to the settings option -> Dogs -> and select B1.

B1 Network

For network setup please connect the ethernet cable that is provided with the B1 from the B1 to your computer. After that set-up a static connection as explained below.

Network-IPs

The B1 has the following networks as of 21-Mar-2024 and may vary in future versions, the following table shows the networks

Device	Network Address	Password

B1-MCU	192.168.12.1	123

Network via Hot-spot

Once the apps are installed users can connect to the native hot-spot of B1 or through the Ethernet port to access its on-board computers. B1 has the network address range of 192.168.12.* . The SSID of the Wifi network of B1's hot-spot begins with UnitreeRoboticsB1-### , where the hash tags represent the B1's model number and the default password is 00000000 or 8 zeros.

Important: It is recommended to connect to the B1's 5G network for better connectivity.

Once connected to the WiFi network, one can access B1's app, web-control, and PCs.

```
ssh -X unitree@192.168.12.1
```

The password is 123

B1 Basic Operations

The basic operations for the B1 are provided in the User manual.

Good to know commands:

- Walking: START
- Prone position: L2+A (Press double times to go onto standing position)

Note: Some features are bound to the version of the B1.

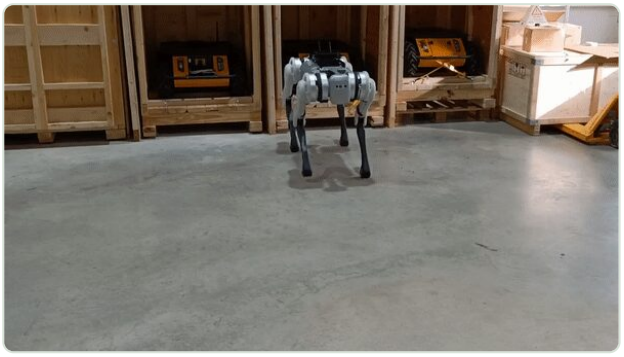
Once connected to the WiFi of the B1. You can access the B1 website by entering the following in the web addess bar 192.168.12.1 . The B1's Firmware version, motor temperature, vision, and control can all be accessed in here.

B1 Leg Calibration

- B1 Calibration

7 Software Packages

B1 Software



_images/b1_movearound.gif

B1 edu is intended for academic and research community. Unitree Robotics has provided an SDK for their robots in C++. A demo is available to the robots, however, it is recommended to directly contact support@mybotshop.de for the latest driver as the old ones are known to have issues.

Requirements

Please follow the pre-requisite software section.

- Pre-requisite Software
- Gazebo9 (melodic)
- Gazebo11 (noetic)

This driver requires a system setup with ROS. It is recommended to use Ubuntu 18.04/Ubuntu 20.04 with ROS melodic/noetic.

To make sure that robot control isn't affected by system latencies, it is highly recommended to connect the robot via LAN of via 5G connection.

Building

A script can be found below for automatic installation of all the required dependencies mentioned above and the provided software installation. After downloading and changing directory to the downloaded folder, installation script can be run by the following command:

> **Note:** For the B1 installation package please email support@mybotshop.de

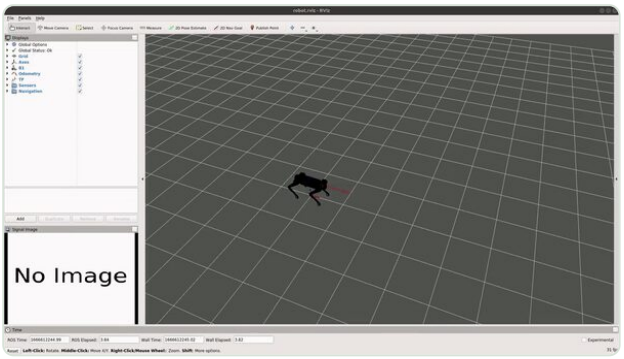
You first have to give the script root permission via:

```
sudo chmod +x installer.bash
```

```
./installer.bash
```

Now everything should be installed and built (assuming ROS and Gazebo installed prior to following this tutorial). If there is any issues please contact support@mybotshop.de .

B1 Base



_images/b1_base_driver.gif

B1 work in two modes:

1. High level mode
2. Low level mode

User can be in one of these modes and they can not be switched after running.

High level mode

High level mode is mainly for walking and running. This mode too has two modes normal and motion mode which can be distinguished from their ips (For more details see Quadruped software guide). At the moment this driver is working with default configuration.

In high level mode the robot can walk. To start the driver please run:

```
sudo su
source ~/catkin_ws/devel/setup.bash
roslaunch b1_base base.launch
```

To launch the b1_viz:

```
roslaunch b1_viz view_robot.launch
```

User can be in one of these modes and they can not be switched after running. With base.launch launches a communication channel between the robot and remote PC. Then later in /base_node/cmd_vel a ROS node is run.

```
Node [high_level_driver]
```

```
Publications: state [b1_legged_msgs/HighState]

Subscriptions: cmd_vel [geometry_msgs/Twist]

Services: set_body_pose [b1_legged_msgs/SetMode]
```

The robot pose can be set using the robot set_body_pose service. Robot state is published in state topic. This message contains a lot of information. For more information see high state message .

Using the driver

The node subscribes to base_cmd_vel topic. Here special attention needs to be paid. As the robot is holonimic so different Twist message fields determine different movement. A nice tool here would be teleop_twist_keyboard . Here in the teleop holonomic and non holonomic modes can be used (Be aware key strokes and their usage).

Low level mode

In development ...

B1 Legged Msgs

This ros package contains custom ros messages used by the B1 Base.

B1 Camera

Coming soon ...

B1 Control

This ros package enables teleop for the Logitech controller.

```
roslaunch b1_control controller.launch
```

Future updates will include localization files required for navigation.

B1 Description

This is a ros package that contains the 3D models of the B1. Additionally, the seperate 3D models are stitched together into a single entity via the universal robot description format (URDF). It contains the models for auxiliary components such as lidars as well. Additions to the robot's 3D model can be made in the xacro/robot.urdf.xacro . To view the robot with simulated dummy drivers. You can run:

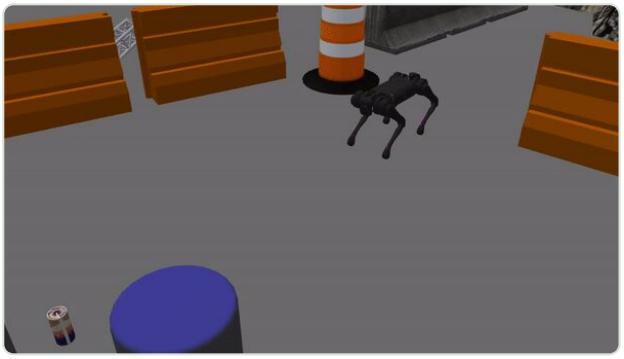
```
roslaunch b1_description check_joint.launch
```

> **Note:** It is ideal to run this on your own PC when the robot is not attached just to adjust the position of your auxiliary components.

The description is automatically launched with the base driver to show the perceived position of the legs with the robot.

```
roslaunch b1_description description.launch
```

B1 Gazebo



_images/b1_gazebo.gif

This ros package is used for controlling and operating the B1 in Gazebo simulation. The robot can be tele-operated and can be used further for slam and autonomous navigation. This package requires the Champ ROS repository.

To launch Gazebo run the following command:

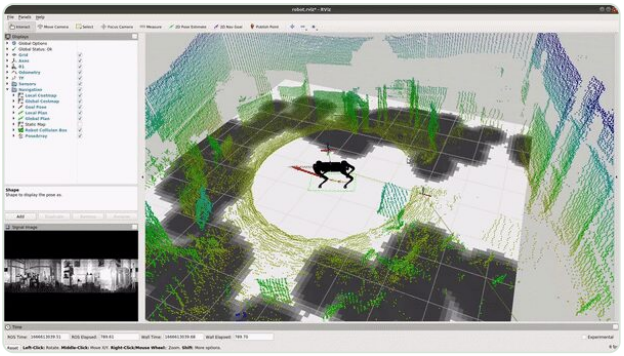
```
roslaunch b1_gazebo bringup.launch rviz:=true
```

To tele-operate:

```
roslaunch champ_teleop teleop.launch
```

This can be used to create and test low-level drivers as well, more information on this is available on the Champ repository.

B1 Viz



_images/b1_ouster.gif

This ros package is used for visualization of the robots current state both in simulation and in the real-hardware.

```
roslaunch b1_viz view_robot.launch
```

8 Augmentations – Z1 Arm & Ouster LIDAR

Augmentations

Quadruped B1 can be upgraded with different sensros and some other accessories. These upgradations require some modifications in the robot software which is detailed below.

Z1 Installation

_images/b1.png

The B1 can be combined with the Z1 manipulator which is a robotic manipulator arm that has three degree-of-freedom (DOF) arm, which means it has three joints that allow it to move and manipulate objects in three-dimensional space. The arm is attached to the robot's body via a rotating joint that allows it to move independently of the robot's locomotion. This enables the B1 to walk on four legs while using its arm to manipulate objects or perform other tasks. Furthermore, the end effector on the Z1's robotic manipulator arm is a gripper, which can open and close to grasp objects. The gripper is capable of exerting a maximum force of 30N, which is sufficient to lift and manipulate objects weighing up to 1kg. The arm can also rotate around its base to adjust the orientation of the gripper, allowing the Z1 to grasp objects from different angles.

Note: For more information please contact info @ mybotshop . de

Ouster Installation

_images/ouster.png

For the ouster attachment, the following package has to be installed in the current workspace. The following tutorial can be followed or the provided description.

Note: This tutorial is to only be followed if the ouster software is not already pre-installed.

1. Install dependencies:

```
sudo apt install -y build-essential libeigen3-dev libjsoncpp-dev  cmake
```

```
sudo apt install -y ros-$ROS_DISTRO-pcl-ros ros-$ROS_DISTRO-rviz ros-$ROS_DISTRO-tf2-
geometry-msgs
```

1. Create a new folder in your catkin workspace src folder named third_party :

```
cd ~/catkin_ws/src
mkdir third_party
cd third_party
```

1. Clone the repository into utils :

```
git clone --recurse-submodules https://github.com/ouster-lidar/ouster-ros.git
```

1. Create your workspace afterwhich you can start configuring the Ouster:

```
cd ..
catkin build
source devel/setup.bash
```

1. Connect the ouster with an ethernet cable, and follow the instructions provided in the software user manual chapter 2. It may take ~5 mins for it to be detected.
2. Next navigate to the utils (if the ouster pkg is provided by us) and in there open a terminal and run:

```
chmod +x set_static_ip.py
python3 set_static_ip.py
```

1. Then make open up your ethernet connection that is in the settings and add a new connection. Give your desired name and then navigate to the IPv4 tab. In it change the connection to Manual . In the Addressess , give the address as 192.168.123.2 and the net mask as 255.255.255.0 , and save. Then connect to this ip.

- Verify that the computer is receiving data from ouster via: catkin build source devel/setup.bash

1. Add a new launch file called ouster.launch to the ouster-ros package in the launch directory, and configure the Lidar to your IP and your specifications:

```
<?xml version="1.0"?>
<launch>

<arg name="ouster_ns" default="ouster"/>
<arg name="sensor_hostname" default="192.168.123.35"/>
<arg name="udp_dest" default="192.168.123.6"/>
<arg name="lidar_port" default="46481"/>
<arg name="imu_port" default="45352"/>
<arg name="udp_profile_lidar" default=""/>
<arg name="lidar_mode" default="512x10" />
<arg name="timestamp_mode" default="TIME_FROM_ROS_TIME"/>
<arg name="metadata" default=""/>
<arg name="viz" default="false"/>
<arg name="rviz_config" default="$(find ouster_ros)/config/viz.rviz"/>
<arg name="tf_prefix" default=""/>
<include file="$(find ouster_ros)/launch/sensor.launch" pass_all_args="true"/>

<node pkg="tf" type="static_transform_publisher" name="lidar_to_robot" args="0 0 0.55 0 0 0
/base /os_sensor 100" />

</launch>
```

1. Verify installation via:

```
roslaunch ouster_ros ouster.launch viz:=true
```

9 Getting Help & RMA

RMA / Support

How to Raise a Return Merchandise Authorization (RMA) Case

At QUADRUPED Robotics, we are committed to providing excellent support to ensure your issues are resolved promptly. If you encounter any problems with our products, please follow the steps below to raise a Return Merchandise Authorization (RMA) case:

Step 1: Open a Topic in Our Forum

1. Visit MYBOTSHOP Forum .
2. Open a new topic describing your issue in detail. Our support team and community members will attempt to assist you in resolving the problem remotely.

Step 2: Contact Support for RMA Number

If the issue cannot be resolved remotely through our forum:

1. Email us at support @ quadruped . de .
2. Provide a detailed description of the issue and include any troubleshooting steps already taken.
3. Our support team will evaluate your case and, if necessary, issue a RMA number along with a return address or a shipping label.

Important Shipping Information

Note: Please note: All goods must be sent back to the following address: QUADRUPED Robotics GmbH c / o MYBOTSHOP GmbH Willy - Messerschmitt - Strasse 12 50126 Bergheim Germany Do not ship goods without obtaining a RMA number. Parcels sent without a RMA number will be declined and returned to the sender. Do not ship goods to QUADRUPED Robotics GmbH Office in Leverkusen!

Help Resources

In case of any issues, help can be taken from any of the following forums/resources:

1. MYBOTSHOP Forum : The forum is actively monitored by support teams at MYBOTSHOP GmbH .
2. Github Issues : Issues can be reported in the repository.
3. Online QA Sites: Questions can be asked on any of the QA sites like StackOverflow and Answers ROS .

10 Getting Involved

Getting Involved

Contact support @ mybotshop . de for more info

11 FAQ

Frequently Asked Questions

Wireless Access B1

The B1 can be wirelessly accessed by its onboard hotspot. It usually has the name UnitreeRoboticsB1-## followed by the model number. The default password is 00000000 .

- The main controller ip:

```
ssh -X unitree@192.168.123.161
```

- The Nvidia controller ip:

```
ssh -X unitree@192.168.123.12
```

Important: The password for both are 123 .

The Unitree App can be used to view the robot as well.

- For recording and visualing its data, it is recommended to use record a rosbag transfer it back to your computer and play it:

- Record rosbag:

```
rosbag record -a
```

- Play rosbag:

```
rosbag play -l <file_name.bag>
```

Display not visible B1

There are two pcs on the robot. The front side of the robot has some ports they are for Rasberry Pi. Here you can connect the HDMI cable and than restart the robot because if the robot is turned on and than you connect an HDMI adapter then it will not show anything. So you have to restart after connecting the HDMI. For Nvidia the same procedure applies.

Note: The Nvidia's moAlstherboard goes to sleep, so if you connect an external mouse or keyboard you can directly wake it up.