

MYBOTSHOP

QUADRUPEL ROBOTICS

Diablo

User Manual



CONTACT

info@mybotshop.de
support@mybotshop.de
mybotshop.de

Contents

1	Attention	3
2	Disclaimer	4
3	About Diablo	5
4	Quick Start – Network & SSH Access	6
5	Host PC Setup – ROS 2 Foxy Installation	8
6	Operation	11
7	Software Package Reference	13
8	Proprietary MBS Drivers	15
9	Quick Links & Resources	16
10	Getting Help & RMA	17
11	FAQ	18

1 Attention

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

These tutorials assume you are comfortable working with ROS. If you are not familiar with ROS already, we recommend starting with the roswiki ROS/Tutorials before operating Diablo.

2 Disclaimer

Diablo is a wheel-legged (“wheeled-bipedal”) self-balancing robot manufactured by *DDTRobot* and distributed/integrated by *MYBOTSHOP GmbH*. Basic ROS 2 understanding is required to operate the software stack covered in this manual — if you are not familiar with ROS 2, we recommend checking the [ROS 2 Foxy documentation](#) first.

The client acknowledges and agrees that any information or materials provided by MYBOTSHOP GmbH 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, or non-infringement.

This product is intended for civilian use only. Users should avoid making dangerous modifications or using the robot in hazardous ways. MYBOTSHOP GmbH 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.

3 About Diablo

DIABLO



Diablo demonstrating its wheel-legged balancing and posture-change capabilities.

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

Attention: These tutorials assume that you are comfortable working with ROS. We recommend starting with [roswiki ROS/Tutorials](#) if you are not familiar with ROS already.

4 Quick Start – Network & SSH Access

Network Setup

To set up a connection to the Diablo, please follow the provided guidelines:

1. Connect an Ethernet cable from your Host PC to the Ethernet port of the Diablo (Raspberry pi).
2. Next in your Host PC, create a static connection from your network manager with an ip of 192.168.0.51 . For the netmask, use 255.255.255.0 .
3. Once the static connection is setup correctly, one can ssh into the on-board PC as described below.

To ssh, enter the command terminal and enter the following:

```
ssh -X diablo@192.168.0.<diablo_ip>
```

The password is:

```
diablo123
```

Network tables

The following table shows Diablo's networks.

Device	Network Address	Users	Password
Diablo	192.168.0.*	diablo	diablo123

Software Setup (Diablo)

Connect to WiFi

Connect to Diablo via ethernet lan cable.

Then use `sudo nmap -sP 192.168.0.*` to find ip. The name diablo in one of the connections should turn up.

Enter diablo via ssh, enter the command terminal and enter the following:

```
ssh -X diablo@192.168.0.###
```

The password is:

```
diablo123
```

Connect the Raspberry pi to your local WiFi network via

```
sudo nmtui
```

The new ip can be located via

```
ifconfig
```

Nav2 Performance

Install and add the export command to your .bashrc file to enable smooth ROS2 nav experience.

```
sudo apt install ros-foxy-rmw-cyclonedds-cpp  
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
```

Also to perform buffer increase for cyclone (do only once)

```
echo "net.core.rmem_max=8388608\nnet.core.rmem_default=8388608\n" | sudo tee  
/etc/sysctl.d/60-cyclonedds.conf
```

Note (from the mbs_diablo repository README, not on the docs site): To drive Diablo under remote/teleop control, flip the physical switch on the back bottom right of the robot down to enable remote mode.

5 Host PC Setup – ROS 2 Foxy Installation

Software Setup (User)



ROS 2 Foxy Fitzroy.

The following installation must be done in the Host PC that will communicate with Diablo as Diablo's Raspberry Pi comes pre-installed with ROS2 Foxy. It is recommended to follow the instructions provide in ROS Foxy Docs for latest info on installation.

Set Up Your Sources List

```
sudo sh -c 'echo "deb [arch=amd64,arm64] http://packages.ros.org/ros2/ubuntu DISTRO main" >
/etc/apt/sources.list.d/ros2-latest.list'
```

Replace DISTRO with your distribution name (e.g., focal for Ubuntu 20.04).

Set Up Your Keys

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

Update Your Package Index

```
sudo apt update
```

Install ROS 2 Foxy Packages

```
sudo apt install -y ros-foxy-desktop
```


This command installs the ROS 2 Foxy Fitzroy desktop packages, including common dependencies and tools.

Source the ROS 2 Setup Script

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

To avoid running this command each time you open a new terminal, consider adding it to your shell's startup script (e.g., ~/.bashrc).

Install ROS 2 Dependencies

You may need to install additional dependencies based on your specific use case. Common dependencies include:

- `python3-colcon-common-extensions` : Useful for building ROS 2 packages.

Install them using apt :

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

Create a ROS 2 Workspace (Optional)

You can create a workspace to organize your ROS 2 packages and build them:

```
mkdir -p ~/ros2_ws/src  
cd ~/ros2_ws  
colcon build --symlink-install
```

This step is optional but recommended for working with ROS 2 packages.

Test Your ROS 2 Installation

To verify that your ROS 2 installation is working correctly, open a new terminal and run:

```
ros2 doctor
```

This command checks the system and reports any issues or missing dependencies.

You can also try running a basic ROS 2 command like:

```
ros2 --help
```

This should display the ROS 2 command-line interface help message.

Nav2 Performance

Install and add the export command to your .bashrc file to enable smooth ROS2 nav experience.

```
sudo apt install ros-foxy-rmw-cyclonedds-cpp
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
```

Also to perform buffer increase for cyclone (do only once)

```
echo "net.core.rmem_max=8388608\nnet.core.rmem_default=8388608\n" | sudo tee
/etc/sysctl.d/60-cyclonedds.conf
```

ROS2 Diablo Domain ID

Finally in your .bashrc put the ROS DOMAIN ID 5 which is the default domain id of diablo.

```
export ROS_DOMAIN_ID=5
```

Congratulations! You have successfully installed ROS 2 Foxy Fitzroy on your Linux system. You can now start developing and running ROS 2 applications.

For further documentation and tutorials, refer to the official ROS 2 documentation: <https://docs.ros.org/en/foxy/index.html>

6 Operation

Operation

To operate the robot remotely via PC, you need to setup your network as described in the Network Setup section.

Important: Please note that some Diablo may not have the latest drivers and have the old structure layout. In those cases where the instructions are not clear please contact support @ mybotshop . de .

Diablo Driver

To start the hardware-diablo drivers, ssh into diablo and launch:

```
ros2 run diablo_ctrl diablo_ctrl_node
```

Next in your own PC you can launch:

```
ros2 launch diablo_bringup bringup.launch.py
```

Visualize

This turned on by bringup, alternatively you can launch please launch:

```
ros2 launch diablo_viz view_robot.launch
```

Teleop

To operate the Diablo either in simulation or the real robot, the following command may be used:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Odom Navigation

Launch

```
ros2 launch diablo_navigation odom_navi.launch.py
```

Change rviz frame to map and provide goal.

SLAM (from the mbs_diablo repository README, not on the docs site): A SLAM launch file is also available:

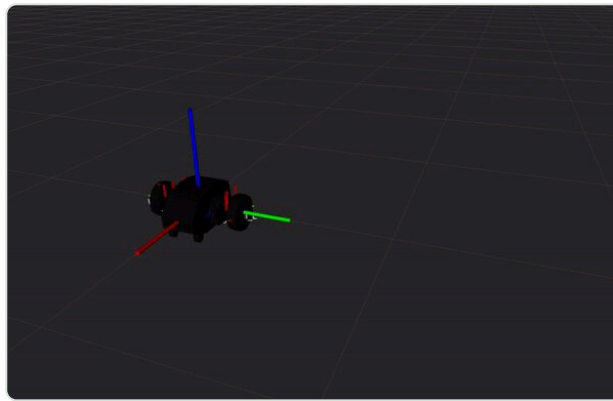
```
ros2 launch diablo_navigation slam.launch.py
```

Drive the robot around to build a map, then save it with the standard `nav2_map_server` `map_saver_cli` tool.

7 Software Package Reference

Correction: The source documentation for this page used `dingo_*` package names in its code samples (leftover from an internal docs template) — corrected below to the real `diablo_*` package names used by the `mbs_diablo` repository.

Diablo Bringup



Diablo's state visualized in RViz via the `diablo_viz` package.

This `ros` package launches the base driver as well as all necessary auxiliary components such as the inertial measurement unit (IMU), Lidar (if available), robot description (required for visualization), and `ekf_localization` (robot position estimation).

Furthermore, this package contains the UDEV rule required by the Diablo's internal PC to detect the communication channel.

```
ros2 launch diablo_bringup system_bringup.launch
```

Diablo Description

This is a `ros` package that contains the 3D models of the Diablo. Additionally, the separate 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/diablo.urdf.xacro`. To view the robot with simulated dummy drivers. You can run:

Warning: Running this command will shut the Diablo Bringup description driver. You may need to restart that if you wish to view the running robots state. 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.

```
ros2 launch diablo_description bringup.launch
```

Diablo Navi

The details of Diablo Navigation are explained in detail in its own section. The main commands are:

1. For odom navigation:

```
ros2 launch diablo_navigation odom_navigation.launch
```

1. For map navigation:

```
ros2 launch diablo_navigation map_navigation.launch
```

Diablo Viz

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

```
ros2 launch diablo_viz view_robot.launch
```

8 Proprietary MBS Drivers

MBS Drivers



Diablo Box Display

The package is designed to work on the Diablo. To install the package request it from MBS Support .

- Copy the provided ROS2 package into your ros2_ws and build

```
colcon build --symlink-install
```

9 Quick Links & Resources

The following external resources are referenced from Diablo's own documentation site.

Resources

- [Diablo Github](#) — DDTRobot's own GitHub organization.

Documentation

- [Diablo Software Docs](#) — DDTRobot's own SDK documentation.

Recommended Resources

- [SSH Tutorial](#)
- [ROS Video Tutorials](#)

Note: The source documentation's "Technical Specification" link points to a Clearpath *Dingo* datasheet PDF, not a Diablo one — almost certainly the same leftover Dingo-template mix-up as the package names above. We have deliberately not carried that link into this manual so as not to present another robot's datasheet as Diablo's own. No official Diablo spec sheet was found on the docs site or in the mbs_diablo repository; the most authoritative source for Diablo's joint layout is its URDF, included with the downloadable robot description package.

10 Getting Help & RMA

RMA / Support

How to Raise a Return Merchandise Authorization (RMA) Case

At MYBOTSHOP GmbH, 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 @ mybotshop . 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: 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.

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 .

11 FAQ

No FAQs have been published yet for Diablo — check back on the [official docs page](#) or ask on the MYBOTSHOP Forum.