MYBOTSHOP

QUADRUPED ROBOTICS

Unitree B2

User Manual

CONTACT

info@quadruped.de

support@quadruped.de

quadruped.de

Contents

1 Attention

Attention: Read this document carefully. In case of doubt, it is essential to 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 all uncertainties are clarified. Contact information is available in Chapter 12, Getting Help & RMA.

Do not allow persons who are not familiar with robotics or these instructions to operate the robot. Robots are dangerous in the hands of untrained users.

Do not operate the robot if you are intoxicated, under the influence of drugs, or unable to concentrate. Familiarize yourself with the characteristics of each gait mode and the emergency braking method (press **L1 + A** to enter damping mode) before switching the robot on. Check that there are no foreign objects (water, oil, sand, soil) inside the robot or its components before every startup, and ensure the remote control and battery pack are fully charged.

Caution: The B2 is a large, powerful quadruped — maintain a safe distance during startup and operation, especially when it stands up automatically or transitions between AI Mode and Sports Mode. Crossing Mode (40 cm ditch-crossing) is explicitly flagged by the manufacturer as dangerous: no one may stand in front of the robot, and every execution requires app authorization.

2 Disclaimer

The B2 and B2-W robots documented here are sold and supported by *MYBOTSHOP / QUADRUPED Robotics* as the German distributor for Unitree Robotics. 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 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 designed for civilian use only. Users must avoid making hazardous modifications or utilizing the robot in unsafe ways. Some features described in this manual are still under development or testing (the Unitree Z1 arm driver and Crossing Mode are explicitly called out as such by the manufacturer/distributor) and will be released to users in future updates. For terms, policies and compliance with local laws and regulations, refer to the Unitree Robotics website and the RMA/support process in Chapter 12.

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 B2 & B2-W

B2 Robot Tutorial

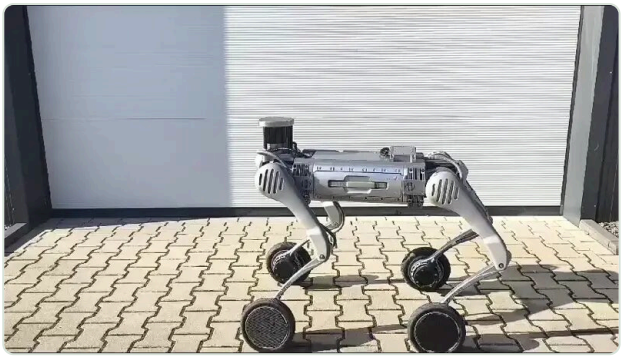
The provided tutorial will assist you with setting up and operating your B2 quadruped robot. The tutorial topics are listed in the left column and presented in the suggested reading order.

B2 Robot

The Unitree B2 is an innovative quadruped robot designed for exceptional agility, adaptability, and durability. Developed by Unitree Robotics, the B2 features advanced sensors, powerful computing capabilities, and a robust modular design, making it well-suited for diverse applications such as industrial inspection, search and rescue, surveillance, and academic research. Whether maneuvering through challenging environments or executing complex tasks, the B2 integrates state-of-the-art technology with outstanding functionality, redefining the standards of modern robotics.

Joint Mobility

The Unitree B2 features advanced joint mobility, powered by high-torque actuators that enable precise and dynamic movement. Each joint is designed with high power density and precision control, allowing the robot to execute agile and stable locomotion across various terrains.The B2's joint actuators incorporate advanced force feedback and torque control, enabling smooth transitions between walking, running, and climbing. This adaptive mobility ensures the robot can navigate uneven surfaces, obstacles, and slopes** while maintaining stability. Additionally, the high-speed joint response enhances real-time adjustments, making it capable of rapid motion changes for industrial and research applications. With modular and reinforced joint structures, the B2's leg mobility supports both flexibility and durability, ensuring long-term reliability even under high-load conditions. This makes it well-suited for autonomous exploration, logistics, and industrial inspection tasks where high mobility is crucial.



B2 Robot

AI-Driven Learning and Adaptation

The Unitree B2 uses AI-driven learning to enhance autonomy and efficiency. Equipped with deep learning and reinforcement learning, it continuously improves navigation, obstacle avoidance, and locomotion across complex terrains.Its sensor fusion system, combining depth cameras, LiDAR, and IMUs, enables real-time SLAM and intelligent decision-making. The B2 adapts its gait using reinforcement learning, optimizing stability and agility on various surfaces.With gesture recognition and human-robot collaboration, the B2 enhances industrial and security applications. Cloud-based learning further refines performance, making it a versatile and intelligent quadruped for advanced automation and research.

B2 Robot

Technical Specifications

Body and Dimensions	
Weight: 60 kg (Incliding Battery)	
Standing Height: 645 mm	
Folded Height: 758 mm	
Total Degrees of Freedom: Up to 12	

Perception and Sensing	
360° Detection: Equipped with 3D LIDAR and depth camera.	

Cameras: Intel RealSense D435i for depth sensing.
Cameras: Optical Cameras
LIDAR: RS-Helioshelios 5515 for precise mapping.

Mobility and Power
Movement Speed: Up to 5m/s
Battery Life: Approx. 3 hours 20 minutes with a 13-string lithium battery.
Battery Weight: 12.2kg
Battery Capacity: 45Ah (2268Wh)
Quick Release Battery: Long-lasting power and easy replacement.

Model Variants

B2 Robot

B2: Standard model with 12 degrees of freedom and normal foot.

B2-W: The wheeled model features specialized wheeled feet, combining speed, adaptability, and cutting-edge design. Engineered to navigate various terrains and tackle complex tasks with ease, it pushes the boundaries of robotic mobility, offering enhanced efficiency and versatility in motion.

Model	**B2**	**B2-W**
Mechanical Dimensions		
Height, Width and Thickness (Stand)	1098mm×450mm×645mm	1098mm×550mm×758mm
Height, Width and Thickness (Fold)	880mm×460mm×330mm	950mm×550mm×450mm
Weight (With Battery)	About 60kg	About 85kg
Total Degrees of Freedom (Joint Freedom)	12	12

Single Leg Degrees of Freedom	3	3
Maximum Torque of Knee Joint	360N.m	360N.m
Maximum Standing Load	About 40Kg	About 40kg
Continuous Walking Load	About 120Kg	About 120kg
Joint Movement Space Hip joint: Thigh joint: Calf joint:	-0.87~0.87° -0.94~4.69° -2.82~ -0.43°	-0.87~0.87° -0.94~4.69° -2.82~ -0.43°
Electrical Characteristics		
Full Joint Hollow Electrical Routing	YES	YES
Joint Encoder	Dual encoder	Dual encoder
Cooling System	Local air cooling	Local air cooling
Power Supply	13 string lithium battery	13 string lithium battery
Basic Computing Power	Intel Core i5 (platform function) and Intel Core i7 (user development) CPU	Intel Core i5 (platform function) and Intel Core i7 (user development) CPU
Sensing Sensor	Depth Camera+ Optical camera+ 3D LiDAR	Depth Camera+ Optical camera+ 3D LiDAR
WiFi 6, Bluetooth 5.2	YES	YES
Accessories		
High Computing Power Module	NVIDIA Jetson Orin NX (optional)	NVIDIA Jetson Orin NX (optional)
Smart Battery (Quick Release)	45Ah	45Ah
Charger	58.8V 10A	58V 10A
Manual Controller	YES	YES
Enviroment		
Operating Temperature | -20℃~55℃		-20℃~55℃
Climbing Angle	> 45°	> 45°
Maximum Step Height	20~25cm	20~25cm
Maximum Jumping Distance	> 1.6m	
IP67 Protection	YES	YES
Others		
Battery Life	**About 4~6h | About 4~6h**	

Upgraded Intelligent OTA	YES	YES
Secondary Development	YES	YES
Maximum speed	> 6m/s	50rad/s

Important: The listed parameters may vary based on configuration and usage scenarios. Please refer to actual operating conditions for precise specifications. The quadruped robot features a complex structure and advanced capabilities. Users should maintain a safe distance and operate it with caution. In case of any modifications to the product's appearance, the actual product should be considered as the reference. Some features mentioned on this page are still under development and testing. They will be released to users in future updates. This product is designed for civilian use only. Users must avoid making hazardous modifications or utilizing the robot in unsafe ways. For further details on terms, policies, and compliance with local laws and regulations, please visit the Unitree Robotics website.

4 Quick Start – Unpacking, Power & Network

B2 Manuals

_images/B2_possing.webp

The following text contains hyperlinks. Click on them to be redirected to the websites.

- Unitree Documentation
- B2 SDK

B2 Mobile Apps

The QR codes for Android and iOS to download the B2 app are provided below. Please scan the respective QR code to access the download link.

Scan to download for Android	Scan to download for iOS

B2 Itinerary and Transport

The Unitree B2 is designed for efficient transport and quick deployment, making it ideal for field operations, industrial inspections, and research applications. Its foldable legs and compact structure allow for easy packing, and its approximately 60 kg weight makes it transportable in medium-sized vehicles or provided industrial cases. The Unitree provided case comes with wheels for easy transport and maneuverability. Position the B2 inside the case, ensuring it rests securely without movement. Place the removed, battery pack and charger into designated compartments within the transport case, ensuring they are secured and will not shift during transit.Store additional accessories, such as controllers or spare parts, in separate compartments or pouches within the case to prevent contact with the robot and potential damage.

_images/B2_trolley1.webp

_images/B2_trolley2.webp

Note: Always handle the Case with care to prevent damage to the robot and accessories during transport.

B2 Charging

Proper charging procedures are essential to ensure the safe and efficient operation of your Unitree B2 Quabruped Robot. Follow the guidelines below for charging the main battery and the companion remote control.

Battery Charging

Due to self-discharge during transportation and storage, the battery may have low or no power initially. Use the following steps to charge the main battery:

1. Remove the battery : Detach the battery from the B2 body. Ensure the battery pack strap is pulled out.
2. Prepare the charger : Connect the charger to an AC power source (100-240V, 50/60Hz). Verify that the external power supply voltage matches the charger's rated input voltage (marked on the charger's nameplate). Failure to match voltages may damage the charger.
3. Connect the battery : Plug the charger into the AC power supply first. Ensure the battery pack is switched off before connecting it to the charger to avoid damage.
4. Monitor charging status : During charging, the battery indicator on the battery pack will flash at a frequency of 1Hz (once per second), displaying the current charge level. When fully

charged, the battery indicator will turn off. Manually disconnect the charger and battery to complete the process.

5. Temperature precautions : Allow the battery to cool to room temperature before charging if it is warm from recent use.

Warning: Only use official chargers to charge the B2 battery. Using unofficial chargers may cause damage or safety hazards.

Remote Control Charging

The companion remote control requires proper charging to maintain functionality. Follow these steps:

1. Use a 5V/2A USB charger that complies with FCC/CE standards.
2. Make sure the remote control is turned off before charging.
3. Connect the remote control to the charger using a USB Type-C cable.
4. Monitor the charging indicator: - The power indicator will blink at 1Hz (once per second) while charging, displaying the current charge level. - Once the indicator turns off, charging is complete. Disconnect the charger.

Remote Charging Indicator

The charging indicator lights display the current charge level as shown in the table below:

Indicator Lights	Current Battery	Description
LED1	0%-25%	Lowest battery level
LED1, LED2	25%-50%	Partial charge
LED1, LED2, LED3	50%-75%	Moderate charge
All LEDs	75%-100%	Fully charged

Note: For optimal battery life, avoid overcharging and ensure all devices are disconnected after charging.

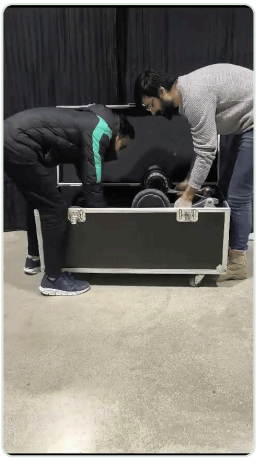
B2 Unpacking and Initial Setup

Follow the steps below to properly unpack and prepare the Unitree B2 Quabruped Robot for operation.

Unpacking

1. Position the box on a stable, flat surface with the front side facing up.

2. Open the top of the box and gently lift the robot out as a whole.
3. Take out all included components, such as the robot, remote control, charger, and accessories.
4. Place the robot securely on a stable surface, ready for activation.



B2 Remote Charge

Important: Handle all components with care to avoid damage during unpacking.

Pre-Power-On Checklist

Before powering on the B2, ensure the following:

1. Use only authentic Unitree Robotics parts and confirm that all parts are in good working condition.
2. Do not operate the robot if you are intoxicated, under the influence of drugs, or unable to concentrate.
3. Familiarize yourself with: Characteristics of each gait mode. The emergency braking method in case of instability or loss of control.
4. Check that there are no foreign objects (e.g., water, oil, sand, or soil) inside the robot or its components.
5. Ensure the remote control and battery pack are fully charged.

B2 Powering On

Attention: If the B2 enters an unexpected state, press L1 + A . The robot will enter damping mode and slowly fall to the ground .

Installing Battery Pack

1. Insert the battery into the battery compartment on the side of the robot.
2. Ensure the power switch faces the back of the robot.

3. If the battery does not fully insert, adjust its direction. Do not forcefully press it to avoid damaging the battery interface and buckle.
4. Listen for a “click” sound, indicating the battery is securely installed. Verify the buckle is in place.

B2 Battery Insert

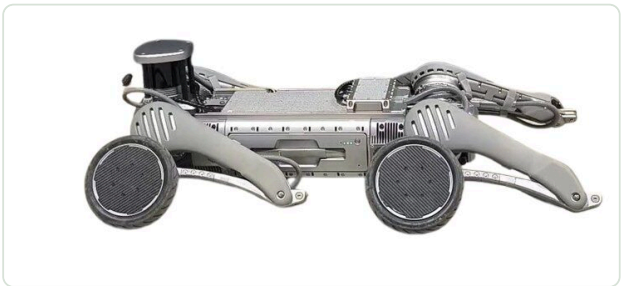
Powering On

1. Short-press the power button on the battery once.
2. Press and hold the power button for more than 2 seconds to power on the battery.

B2 Power On

Preparation Before Startup

1. Place the B2 steadily on the ground.
2. Operate in a flat and obstacle-free environment for initial startup.
3. Avoid wet, dusty, or unstable surfaces that may affect performance.
4. Ensure that all four legs are positioned correctly, with the knees making firm contact with the ground, to maintain stability before powering on the robot.



B2 Ground Placement

B2-W Start-Up

1. Short-press and then Long-press and hold the power button for more than 2 seconds on B2-W battery to power on.
2. Wait approximately 1-2 minutes for the B2-W to fully boot up.
3. Once the B2-W is powered on, it will automatically stand up.
4. After the B2 straightens and stands, use the remote control to press: L2 + R2 + Y for AI Mode(Slow Movement) or L2 + R2 + Start to enter Normal ode(Fast Movement) .

Attention: Maintain a safe distance as the robot will start to get up on its own.

The B2 will rise from the ground, as illustrated in the following GIF:

B2 Wake Up Process

Emergency Stop

If the B2 enters an unexpected state, press L1 + A . The robot will enter damping mode and slowly fall to the ground .

Attention: Always ensure the suspension setup is stable and secure to prevent damage or accidents during operation.

B2 Switching Off

To ensure the Unitree B2 is safely powered down without causing system damage or instability, follow these steps:

1. Stop Movement & Enter Idle Mode - Bring the robot to a stable standing or sitting position . - If the robot is moving, halt all operations via the remote controller or command interface.
2. Disable Motors - Use the controller and press L2 + A to disable the actuators and set the robot to a damping state and the robot will sit down. - Ensure all joints are in a relaxed position before

proceeding.

3. Power Down the System - Press and hold the power button until the shutdown sequence begins. - Wait for indicator lights to turn off , signaling that the system has safely powered down.
4. Remove the Battery (if needed) - If storing or transporting the B2, remove the battery after shutdown. - Store the battery in a cool, dry place and avoid extreme temperatures.
5. Post-Shutdown Inspection - Check for loose components or unusual noises that may indicate wear. - If needed, clean sensors and store the B2 in a secure location .

By following this safe shutdown process , you can prolong the lifespan of the B2 and prevent unexpected issues during future startups.

B2-W Packing

The B2-W robot is designed for easy transport and storage, with a compact structure and foldable legs for efficient packing. Follow the steps below to pack the B2-W robot securely for transport.

1. Power Off the Robot : Ensure the B2-W robot is powered off before packing to prevent accidental activation.
2. Fold the Legs : Gently fold the legs of the robot to reduce its size and prepare for packing.
3. Place the Robot in the Case : Position the robot securely inside the provided case, ensuring it rests without movement.
4. Secure the Robot : Use the provided straps or padding to secure the robot in place and prevent shifting during transport.
5. Store Accessories : Place the removed battery pack, charger, and any additional accessories in designated compartments within the case.
6. Close the Case : Seal the case securely and ensure all components are stored safely for transport.

B2 Wake Up Process

Powering-on

To power on the B2 robot, press the power button on the battery once, and then press and hold for 3 seconds to start the robot.

The battery compartment is located on the back of the robot.

Similarly, to turn off the robot, press the power button on the battery once, and then press and hold for 3 seconds to turn off the robot.

> **Note:** Please be cautious with battery consumption. Continuously monitor the battery levels. If the battery is depleted, the robot will fall.

Network Interface

Instructions for interfacing with the robot using Ubuntu 22.04 and ROS2 Humble .

This procedure should be followed after successfully setting up and pairing with the B2 robot. Additionally, all of the B2's functionality should be verified through the accompanying app before proceeding.

The Ethernet port can be used to establish communication via LAN.

Network Setup

To configure the robot's network for the first time, you need to connect using a LAN cable.

To create a static connection on your PC (not on the robot), follow these steps in Ubuntu :

1. Go to Settings → Network , then click on the + button to create a new connection.
2. In the IPv4 settings, change the connection type to Manual .
3. Set the Address to 192.168.123.51 and the Netmask to 24 .
4. Click Save , then restart your network.

After successfully connecting, check the host's local IP by typing the following command in the host PC's terminal:

```
ifconfig
```

Now, ping the robot by entering:

```
ping 192.168.123.164
```

To access the robot via SSH, use the following command:

```
ssh -X unitree@192.168.123.164
```

The default password is:

```
Unitree0408
```

IP Addresses

IP Address	Device	Username	Password
192.168.123.161	B2 MCU	x	x
192.168.123.162	B2 Auxiliary PC 1	unitree	123
192.168.123.163	B2 Auxiliary PC 2	unitree	123
192.168.123.164	B2 Nvidia Orin NX	unitree	Unitree0408
192.168.123.164:8080	B2 Webserver MBS	admin	mybotshop
192.168.123.170	B2 Nvidia Secondary IP	unitree	Unitree0408
192.168.123.120	Livox Mid360 Front	x	x
192.168.123.121	Livox Mid360 Rear	x	x

Note: Sometimes other networks can cause disruptions when connecting to the B2. It is best to have only your connection to the robot active and disable all others.

ROS_DOMAIN_ID

The B2 system uses ROS_DOMAIN_ID 10 for communication. Ensure your host PC is configured with the same domain:

```
export ROS_DOMAIN_ID=10
```

Add this to your .bashrc for persistent configuration.

Secondary IP Configuration

For camera streaming, the robot uses a secondary IP address (192.168.123.170). This is configured via netplan:

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    eth0:
      dhcp4: false
      addresses: [192.168.123.164/24,192.168.123.170/24]
      nameservers:
        addresses: [114.114.114.114,8.8.8.8]
```

5 Remote Control & Manual Operation

B2 Basic Operation

Remote Control

The remote control is part of the B2 remote control module, equipped with a digital transmission module and a Bluetooth module . Using the Unitree Explore App, the robot and remote control can be bound for seamless connection upon powering on.

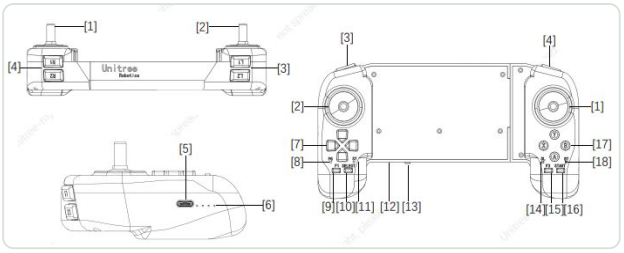
Features:

1. 3-axis posture and position stability control.
2. Movement capabilities include: Forward, backward, left, right, turn in place. Walking in predefined patterns: straight line, circular, linear, rectangular. Navigation on flat surfaces, slopes, and steps.
3. Ergonomic design for improved grip and comfort.

Key Components

The remote control consists of the following key parts:

- [1] Right Rocker
- [2] Left Rocker
- [3] Key L1 / L2
- [4] Key R1 / R2
- [5] Type C Charging Interface
- [6] Data transmission Signal Light
- [7] Arrow Keys
- [8] Power Connect Indicator
- [9] SELECT Key
- [10] F1 (Function Setting Key)
- [11] Charging Status Indicator
- [12] Power Connect Indicator
- [13] Power Button
- [14] Data Transmission Indicator Light
- [15] F3 (Function Setting Key)
- [16] START Key
- [17] Action Keys
- [18] Bluetooth Signal Indicator Light



Remote Control Labels

Technical Specifications

Parameter	Specification
Charging Voltage	5.0V
Charging Current	700mA
Battery Capacity	780mAh Lithium Battery
Communication Mode	Data Transmission, Bluetooth
Running Time	5 hours
Remote Control Range	Above 100m (open environment)

Installation and Calibration

Installing the Joystick

1. Remove the joystick : Pull out the remote control from the storage slot. Remove the two joysticks.
2. Install the joystick : Secure each joystick by rotating it clockwise until tight.

Rocker Calibration

1. Hold the remote control without touching the rocker.
2. Press F1 and F3 simultaneously to enter calibration mode . The remote will emit a continuous “beep-beep” sound (1 beep/second).
3. Rotate the left and right rockers to full rudder several times.
4. When the beeping stops, press F3 once to save the calibration.

Attention: Do not touch the rockers before entering calibration mode. Move the rockers only after calibration mode is active.

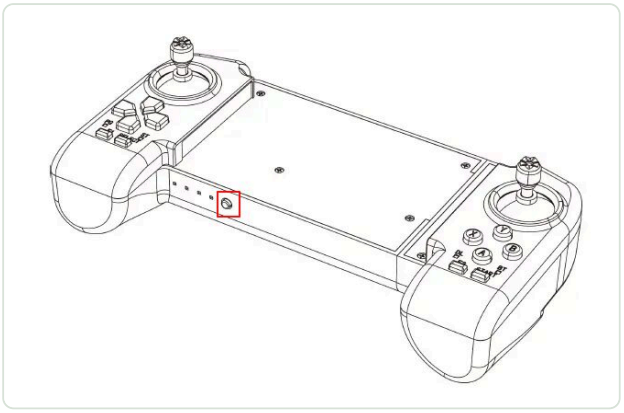
Powering On/Off

Turning On

1. Short-press the power button once.
2. Long-press the power button for 2+ seconds until you hear two “beeps.”

Turning Off

1. Short-press the power button once.
2. Long-press the power button for 2+ seconds until you hear three “beeps.”



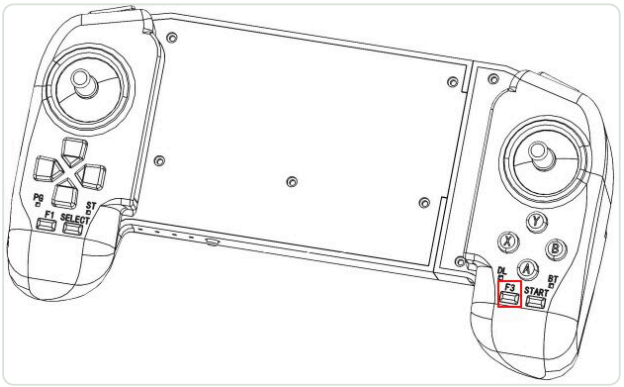
Joystick Installation

> **Tip:** The sequence is short press + long press for 2 seconds or more for both turning on and off.

Vibration and Sound Settings

Vibration/Sound Toggle

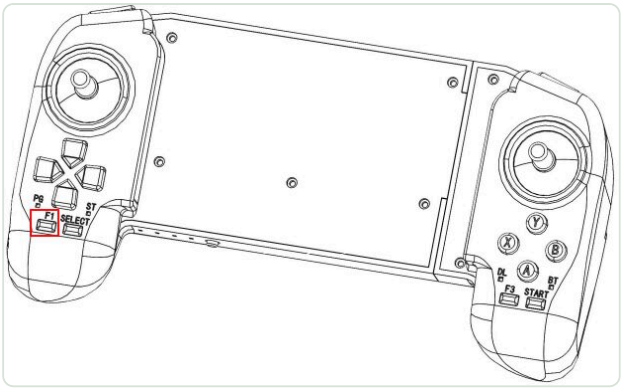
- Switching vibration : Quickly press F3 3 times to switch to vibration mode.
- Switch sound : Quickly press F3 3 times to switch to sound mode.



Joystick Operation

Vibration/Sound Switch

- Turn off vibration/sound : Quickly press F1 3 times to turn off vibration/sound.
- Turn on vibration/sound : Quickly press F1 3 times turn on vibration/sound.



Joystick Operation

Save Feedback Settings

- To save vibration/sound settings for the next startup: - Hold down the F1 button during shutdown.

Attention: By default, feedback settings (vibration/sound) are not saved after modifying.

Important: Always ensure to follow the correct steps to avoid damaging the remote control or affecting calibration accuracy.

Remote Control Binding

First-Time Use: Binding the Remote Control

To use the remote control for the first time:

1. Open the Unitree Explore App and Select “Device”.
2. Go to the “Data” Section.
3. Select “RemoteControl Binding”
4. Tap “Change” and enter the corresponding remote control code.

Navigate to “Device”	Access the “Data” Tab
Open “RemoteControl Binding”	Select “Change”

Once the remote control is bound to the digital transmission module on the robot, it can connect automatically in subsequent uses.

Verifying Connection

- When the remote control is powered on and successfully connected to B2, the right DL indicator light will turn on.
- This indicates the remote control is linked to the B2 digital transmission module, and you can now operate B2 using the remote control.

Remote Control Basic Operation

When using the remote control to maneuver the B2 robot with joysticks, follow these control methods:

Left Joystick

- Push Forward:
- Push Backward:
- Push Left:
- Push Right:

Right Joystick

- Right Turn: Push Joystick towards Right
- Left Turn: Push Joystick towards Left

Concept Descriptions

The B2 robot features several operational modes. Below is a detailed explanation of each mode:

Mode	Description
Locking the Standing Mode	When the robot dog is standing, lock the legs and do not respond to joystick commands.
Damping Mode	When the robot dog is standing, lock the legs and do not respond to joystick commands.
Walking Mode	When the remote control does not push the lever, the robot dog will remain stationary. After pushing the rocker, the robot dog will move in a stepping motion.
Continuous Walking Mode	The robot is always in a stepping state and responds to the joystick command to move.
Running Mode	When the remote control does not push the lever, the robot dog remains in place. When the lever is pushed, the robot dog will move and run.
Continuous Running Mode	The robot is always in a running state and responds to the joystick command to move at the same time.
Stair Climbing Mode	After entering this mode, regular operation control can be used to climb stairs (up and down).
Visual Mode	After entering this mode, AI operation control can perform visual walking, such as climbing stairs and walking on gravel roads.
Endurance Mode	After entering this mode in regular operation control, adjust the dogleg power mode appropriately to reduce power consumption.
Pose Mode	fter entering this mode, the regular operation control can swing the body according to the joystick command.
Crossing Mode	Supports 40cm cross-ditch and climbing (this mode is dangerous, no one can stand in front, and each execution requires App authorization)

Key Descriptions

The B2 offers two distinct operational modes: Mode 1 - Regular Motion Control and Mode 2 - AI Motion Control . Each mode is designed to optimize performance for different scenarios:

- Regular Motion Control (Mode 1): This mode relies on manual input and user decision-making, ensuring greater stability and reliability for tasks such as running and posing. It is ideal for controlled environments where precision and consistency are required.
- AI Motion Control (Mode 2): In this mode, the B2 autonomously adapts to its surroundings, making real-time decisions. This enables the robot to navigate complex terrain , climb stairs , and adjust its movement dynamically without direct manual intervention.

By default, the B2 starts in AI Motion Control Mode upon boot-up. You can switch between modes using the remote control, following the instructions provided in the table below.

B2 Regular Motion Control

Basic Movement

Gesture	Button Combination
Move back or forth	Push Left Rocker Forward/Backward
Sideways movement	Push Left Rocker Push Left/Right

Posture Switching

The B2 robot supports various postures for different applications. Use the table below for guidance:

Button	Function
START	Unlock/Lock Walk Mode
START(double click)	Continuous walking mode
SELECT	Make a pose
L2 (Long Press) + A	Switch between standing lock and lying down posture
L2 (Long Press) + B	Damping mode (soft emergency stop)
L2 (Long Press) + START	Running mode
L2 (Long Press) + START (Double Click)	Continuous running mode
RIGHT (Long Press) + START	Stair climbing mode: ascending stairs, descending stairs
L1 (Long Press) + SELECT	Endurance mode

L2 (Long Press) + X	Back to stand up after falling down
L1 + up/down arrow	Adjust leg lifting height (5-35cm)
Up/down arrow	Adjust body height (45-60cm)
L1 (Long Press) + L2 (Long Press) + R1 (Long Press)+ R2 (Long Press) + X (Click)	Select regular motion control
L1 (Long Press) + L2 (Long Press) + R1 (Long Press) + R2 (Long Press) + B (Click)	Select AI motion control

B2 AI Motion Control 2

Button	Function
START	Unlock/Lock Walk Mode
START(double click)	Continuous walking mode
L2 (Long Press) + A	Switch between standing lock and lying down posture
L2 (Long Press) + B	Damping mode (soft emergency stop)
L2 (Long Press) + X	Fall recovery standing
x	Visual assisted walking
B	walking on level ground
L2 (Double Click)	Low-speed mode
L1 (Double Click)	High-speed mode
L1 (Long Press) + L2 (Long Press) + R1 (Long Press)+ R2 (Long Press) + X (Click)	Select regular motion control
L1 (Long Press) + L2 (Long Press) + R1 (Long Press) + R2 (Long Press) + B (Click)	Select AI motion control

B2-W Remote Control Instructions

Button	Function
L2 (Long Press) + A	Switch between standing lock and lying down posture
L2 (Long Press) + B	Damping mode (soft emergency stop)
L2 (Long Press) + X	Fall recovery standing
L2 (Long Press) + R2 (Long Press) + X	Rotating Jumping (special configuration)
L2 (Long Press) + R2 (Long Press) + Y	A1 Mode(Slow movement Speed)

L2 (Long Press) + R2 (Long Press) + Start	Reguler Motion Mode(Fast movement Speed) Also used for exiting front leg stand mode
L1 (Long Press) + L2 (Long Press) + R1 (Long Press)+ R2 (Long Press) + X (Click)	Stand on front legs and balace it self

Disclaimer

The Unitree B2 is designed for controlled environments. Always adhere to operational guidelines to prevent instability or damage. Unauthorized usage or deviations from specified instructions may void warranty and risk safety.

6 ROS 2 Quick Start – Install, Autostart, Webserver, Teleop, Visualization

B2-W Robot

Pre-requisites

- Safety Guidelines (see B2 Manuals).
- Power On Powering-on .
- Network Interface Network Interface .
- Installation Installation .

Attention: Read this document carefully! In case of doubt, consult specialists, experts, or manufacturers of the assemblies used. The robot must not be operated before all uncertainties are clarified. For further assistance, refer to the guides below or contact a specialist. Important : Contact information is available in the RMA/ Support Section. Warning : Do not allow untrained individuals or those unfamiliar with robotics or these instructions to operate the robot. Improper use of robots is dangerous.

This guide will walk you through the process of setting up and configuring the Unitree B2 robot, including network configurations, SSH access, internet setup, software installation and various operational controls. Follow the steps below carefully to ensure proper operation.

Autostart

Humble

All ROS2 drivers for the B2W developed by MYBOTSHOP operate with the environment variable ROS_DOMAIN_ID=10 . The drivers start automatically on boot using robot_upstart services.

Note: Please ensure the emergency stop is released before powering on the robot. The B2 services will start automatically after boot (approximately 1-2 minutes).

```
sudo systemctl status b2-hardware
sudo systemctl status b2-webserver
```

List all B2 services:

```
systemctl list-units --type=service | grep b2
```

Service Name	Launch File	Auto-Start
b2-hardware	hardware.launch.py	Yes
b2-statepublisher	state_publisher.launch.py	Yes
b2-domain-bridge	bridge.launch.py	Yes
b2-front-video	front_video.launch.py	Yes
b2-rear-video	rear_video.launch.py	Yes
b2-twistmux	twistmux.launch.py	Yes
b2-webserver	webserver.launch.py	Yes
b2w-description	b2w_description.launch.py	Yes
b2-livox-mid360	livox_mid360.launch.py	Yes
b2-pan-tilt	dynamixel.launch.py	Yes

```
# Start/stop/restart service
sudo systemctl start b2-<service_name>
sudo systemctl stop b2-<service_name>
sudo systemctl restart b2-hardware

# Enable/disable service at boot
sudo systemctl enable b2-<service_name>
sudo systemctl disable b2-<service_name>

# Reload systemd after config changes
sudo systemctl daemon-reload
```

```
# View recent logs for a service
journalctl -u b2-hardware -n 50

# Follow logs in real-time
journalctl -u b2-webserver -f

# View logs since last boot
journalctl -u b2-hardware -b
```

Run the startup installer to configure all systemd services:

```
ros2 run b2_bringup startup_installer.py
```

This installs services using robot_upstart with:

- ROS Domain ID: 10
- RMW Implementation: CycloneDDS
- Workspace setup: /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash

Variable	Default	Description
B2_NS	b2_unit_001	Robot namespace
ROS_DOMAIN_ID	10	ROS 2 Domain ID
B2_DESC	1	Enable B2 description
B2_Z1	0	Enable Z1 arm in URDF

Note: Replace b2_unit_001 with your robot's namespace.

Foxy

Warning: ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W uses robot_upstart to manage ROS 2 nodes as systemd services.

Service Name	Description
b2-hardware	Core hardware interface (driver, camera, LiDAR, Dynamixel)
b2-webserver	Web interface for remote control

```
# Check service status
systemctl status b2-hardware
systemctl status b2-webserver

# View logs
journalctl -u b2-hardware -f
```

```
# Stop service
sudo systemctl stop b2-hardware

# Start service
sudo systemctl start b2-hardware

# Restart service
sudo systemctl restart b2-hardware
```

```
# Disable auto-start
sudo systemctl disable b2-hardware

# Enable auto-start
sudo systemctl enable b2-hardware
```

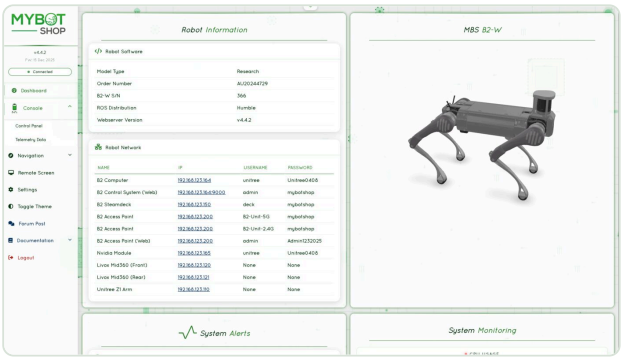
Use the robot_upstart package:

```
ros2 run robot_upstart install b2_bringup/launch/hardware.launch.py \
  --job b2-hardware \
  --rmw rmw_cyclonedds_cpp \
  --symlink
```

```
ros2 run robot_upstart uninstall b2-hardware
```

Webserver

Humble



B2 Webserver Dashboard

The webserver module comes pre-installed and should be accessible directly at https://192.168.123.164:9000/ or via the WiFi IP to which the robot is connected.

- Username: admin
- Password: mybotshop

- HTTP/HTTPS server with authentication
- VNC desktop streaming (1920x1080)
- ROS bridge WebSocket interface (port 9090)
- GPS waypoint management
- Service status monitoring
- Rosbag recording
- Dynamixel servo control sliders
- AI/LLM integration (optional)

- View IP Address of the B2W

- View System load
- Enable/Disable the B2W ROS2 Services
- Record System logs
- Battery status monitoring
- Pre-configured action buttons
- Web Joystick for teleoperation
- Online video stream (front and rear cameras)

- Access on-board screen of the B2W's computer
- VNC password: mybotshop
- VNC port: 6080

- Web-based terminal access
- Command execution interface

- Map visualization and navigation control
- Start/Stop SLAM
- Save maps
- Start/Stop Nav2

- GPS-based outdoor navigation
- Waypoint management
- No-Go zone configuration

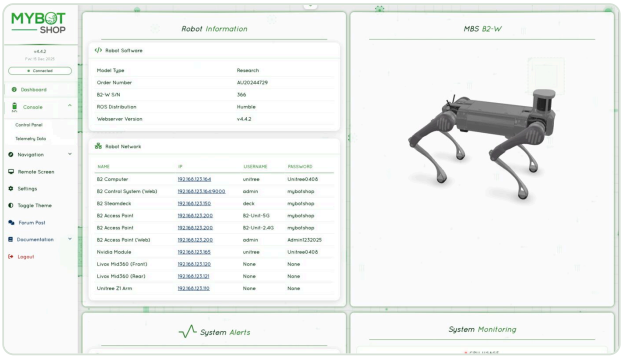
Configuration file: b2_webserver/config/robot_webserver.yaml

```
web_server_ip: "0.0.0.0"
web_server_port: 9000
web_user: "admin"
web_password: "mybotshop"
robot_cmd_vel: "webserver/cmd_vel"
robot_max_linear_velocity: 0.1
robot_max_angular_velocity: 0.1
robot_services:
  - "b2-twistmux"
  - "b2-hardware"
  - "b2-webserver"
  - "b2-rear-video"
  - "b2-front-video"
  - "b2w-description"
  - "b2-livox-mid360"
  - "b2-statepublisher"
  - "b2-domain-bridge"
  - "b2-pan-tilt"
```

Foxy

Warning: ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W includes a built-in web interface for remote control and monitoring.



B2 Webserver Dashboard

Open a browser and navigate to:

```
http://192.168.123.164:8080
```

- Camera Streams: Live video feeds from onboard cameras
- Joystick Control: Virtual joystick for manual navigation
- Mode Selection: Robot mode controls (stand up, sit down, gaits)
- Servo Control: Dynamixel pan-tilt camera sliders
- Status Display: Real-time robot status information

The webserver is configured via the launch file parameters:

```
# In b2_webserver/launch/webserver.launch.py
webserver_node = Node(
    package='b2_webserver',
    executable='webserver',
    parameters=[{
        'port': 8080,
        'host': '0.0.0.0',
    }],
)
```

Cannot access webserver:

1. Verify the service is running: systemctl status b2-webserver
2. Check network connectivity: ping 192 .168.123.164
3. Verify port is open: ss -tlnp | grep 8080

Teleoperation

Humble

Command Line Interface

This requires the installation of the ROS2 Modules on B2. To teleoperate the B2W, run the following command:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  cmd_vel:=/b2_unit_001/hardware/cmd_vel
```

Note: Replace b2_unit_001 with your robot's namespace. Reduce speed by pressing z until 0.1 for safe indoor operation.

The twist multiplexer arbitrates between multiple velocity command sources:

Source	Priority	Timeout
E-Stop Lock	255	None
Steamdeck Joy	20	0.5s
Logitech Joy	15	0.5s
Webserver	10	0.5s
Interactive Marker	7	0.5s
External cmd_vel	5	0.5s
Autonomous High	3	0.5s
Autonomous Mid	2	0.5s
Autonomous Low	1	0.5s

```
ros2 launch b2_control teleop.launch.py
```

Configuration file: b2_control/config/teleop_ps4.yaml

The b2_steamdeck package enables wireless teleoperation using a Steam Deck as a portable controller with audio feedback.

Movement (requires L1 held):

Input	Action
L1 + Left Stick	Linear X/Y movement
L1 + Right Stick X	Angular rotation
R1 + Sticks	Turbo mode (faster speeds)

Robot Commands:

Button Combo	Action
L2 + D-Pad Up	Stand up
L2 + D-Pad Down	Sit down

Speed Configuration:

Mode	Linear X	Linear Y	Angular
Normal	0.7 m/s	0.6 m/s	0.8 rad/s
Turbo	1.0 m/s	1.0 m/s	0.8 rad/s

Launch Steam Deck teleop:

```
ros2 launch b2_steamdeck system.launch.py
```

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W supports multiple teleoperation methods.

The default controller is mapped for Xbox-compatible gamepads:

Input	Action
Left Stick	Linear velocity (forward/backward, strafe)
Right Stick	Angular velocity (rotation)
A Button	Stand up
B Button	Sit down
X Button	Recovery mode
Y Button	Toggle gait mode

Launch teleop node:

```
ros2 launch b2_bringup teleop.launch.py
```

For keyboard teleoperation:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  --ros-args --remap cmd_vel:=/b2_unit_001/cmd_vel
```

Use the built-in webserver joystick at:

```
http://192.168.123.164:8080
```

The B2W accepts standard Twist messages:

```
ros2 topic pub /b2_unit_001/cmd_vel geometry_msgs/msg/Twist \
  "{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.3}}"
```

Visualization

Humble

RViz2 Visualization

To visualize the robot state in RViz2:

```
ros2 launch b2_viz view_robot.launch.py
```

Standard B2 (legged):

```
export B2_DESC=1
ros2 launch b2_description b2_description.launch.py
```

Wheeled B2W:

```
ros2 launch b2_description b2w_description.launch.py
```

```
export B2_DESC=1
ros2 run xacro xacro \
  /opt/mybotshop/src/mybotshop/b2_description/xacro/robot.xacro \
  > /opt/mybotshop/src/mybotshop/b2_description/xacro/b2.urdf
```

To debug the TF tree:

```
ros2 run rqt_gui rqt_gui --ros-args \
  --remap tf:=/b2_unit_001/tf \
  tf_static:=/b2_unit_001/tf_static
```

To view the robot's URDF model:

```
ros2 launch b2_description b2w_description.launch.py
```

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

Launch the visualization tools for the B2W.

View the robot URDF model:

```
ros2 launch b2_description display.launch.py
```

With Z1 arm enabled:

```
export B2_Z1=1
ros2 launch b2_description display.launch.py
```

Launch complete visualization with sensor data:

```
ros2 launch b2_bringup rviz.launch.py
```

The default RViz configuration includes:

- Robot model visualization
- TF tree display
- LaserScan from LiDAR
- PointCloud2 visualization
- Camera image displays
- Odometry arrows
- Path visualization

Custom configurations are stored in:

```
b2_bringup/rviz/
```

View the transform tree:

```
ros2 run tf2_tools view_frames

# Or real-time in terminal
ros2 run tf2_ros tf2_echo base_link odom
```

7 ROS 2 Reference – Rigs, Manipulation, Navigation, Sensors, Simulation, Packages

Rigs

Humble

The B2-W features specialized wheeled feet, combining speed, adaptability, and cutting-edge design. Engineered to navigate various terrains and tackle complex tasks with ease, it pushes the boundaries of robotic mobility, offering enhanced efficiency and versatility in motion.

- Maximum speed: 50rad/s wheel velocity
- Weight (With Battery): About 85kg
- Standing Height: 758mm
- Folded Height: 450mm

The B2W can switch between various gaits suitable for wheeled locomotion:

```
# Set trotting gait for wheel movement
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'gait_trot'}"

# Set flat terrain walking mode
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'gait_flatwalk'}"
```

The B2-W supports various payload configurations:

- Maximum Standing Load: About 40kg
- Continuous Walking Load: About 120kg

Additional hardware can be mounted on the robot's back including:

- Unitree Z1 robotic arm
- Pan-tilt camera systems
- Sensor arrays
- Custom payloads

The default rig configuration includes a pan-tilt camera system controlled via Dynamixel servos.

Servo Configuration:

- Servo 0 (Pan): Range 1150-3200, Default 2170
- Servo 1 (Tilt): Range 230-900, Default 565

Control via webserver sliders or ROS2 topics:

```
# Center pan servo
ros2 topic pub --once /dynamixel/servo_0/command \
  std_msgs/msg/Float64 "data: 2170"

# Center tilt servo
ros2 topic pub --once /dynamixel/servo_1/command \
  std_msgs/msg/Float64 "data: 565"
```

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2-W features specialized wheeled feet, combining speed, adaptability, and cutting-edge design.

- Maximum speed: 50rad/s wheel velocity
- Weight (With Battery): About 85kg
- Standing Height: 758mm
- Folded Height: 450mm

The B2W can switch between various gaits suitable for wheeled locomotion:

```
# Set trotting gait for wheel movement
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'gait_trot'}"

# Set flat terrain walking mode
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'gait_flatwalk'}"
```

The B2-W supports various payload configurations:

- Maximum Standing Load: About 40kg
- Continuous Walking Load: About 120kg

Additional hardware can be mounted including:

- Unitree Z1 robotic arm
- Pan-tilt camera systems
- Sensor arrays
- Custom payloads

The default rig configuration includes a pan-tilt camera system controlled via Dynamixel servos.

Servo Configuration:

- Servo 0 (Pan): Range 1150-3200, Default 2170

- Servo 1 (Tilt): Range 230-900, Default 565

Control via webserver sliders or ROS 2 topics:

```
# Center pan servo
ros2 topic pub --once /dynamixel/servo_0/command \
  std_msgs/msg/Float64 "data: 2170"

# Center tilt servo
ros2 topic pub --once /dynamixel/servo_1/command \
  std_msgs/msg/Float64 "data: 565"
```

Manipulation

Humble

Important: EXPERIMENTAL: The Z1 driver is under active development.

If your B2W is equipped with the Unitree Z1 arm:

Enable the arm:

```
ros2 service call /b2_unit_001/z1/enable std_srvs/srv/Trigger
```

Send trajectory command:

```
ros2 topic pub --once /b2_unit_001/z1/joint_trajectory \
  trajectory_msgs/msg/JointTrajectory "{
    joint_names: ['joint1', 'joint2', 'joint3',
      'joint4', 'joint5', 'joint6', 'jointGripper'],
    points: [{
      positions: [0.0, 1.57, -1.57, 0.0, 0.0, 0.0, 0.0],
      time_from_start: {sec: 3, nanosec: 0}
    }]
  }"
```

Disable the arm:

```
ros2 service call /b2_unit_001/z1/disable std_srvs/srv/Trigger
```

Name	Type	Description
z1/joint_states	sensor_msgs/JointState	Joint positions, velocities, efforts
z1/joint_trajectory	trajectory_msgs/JointTrajectory	Trajectory commands
z1/enable	std_srvs/Trigger	Enable arm control
z1/disable	std_srvs/Trigger	Disable arm

For detailed Z1 arm specifications, refer to the Unitree Z1 documentation .

Enable Z1 in the robot description:

```
export B2_Z1=1
ros2 launch b2_description display.launch.py
```

Foxy

Warning: ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

Important: EXPERIMENTAL: The Z1 driver is under active development.

If your B2W is equipped with the Unitree Z1 arm:

Enable the arm:

```
ros2 service call /b2_unit_001/z1/enable std_srvs/srv/Trigger
```

Send trajectory command:

```
ros2 topic pub --once /b2_unit_001/z1/joint_trajectory \
  trajectory_msgs/msg/JointTrajectory "{
    joint_names: ['joint1', 'joint2', 'joint3',
      'joint4', 'joint5', 'joint6', 'jointGripper'],
    points: [{
      positions: [0.0, 1.57, -1.57, 0.0, 0.0, 0.0, 0.0],
      time_from_start: {sec: 3, nanosec: 0}
    }]
  }"
```

Disable the arm:

```
ros2 service call /b2_unit_001/z1/disable std_srvs/srv/Trigger
```

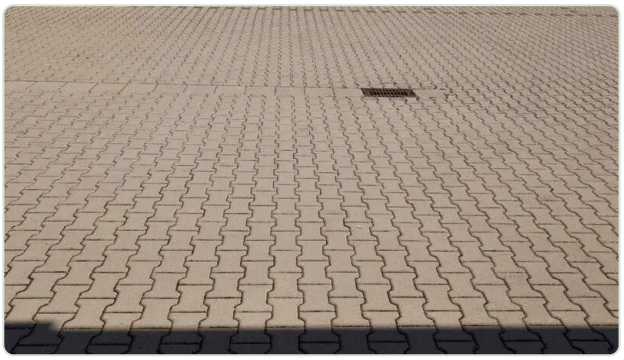
Name	Type	Description
z1/joint_states	sensor_msgs/JointState	Joint positions, velocities, efforts
z1/joint_trajectory	trajectory_msgs/JointTrajectory	Trajectory commands
z1/enable	std_srvs/Trigger	Enable arm control
z1/disable	std_srvs/Trigger	Disable arm

Enable Z1 in the robot description:

```
export B2_Z1=1
ros2 launch b2_description display.launch.py
```

Navigation

Humble



B2 Navigation Demo

The B2W uses the Nav2 navigation stack for autonomous navigation.

Run SLAM to create a 2D occupancy grid map using the LiDAR sensor:

```
ros2 launch b2_nav2 slam.launch.py
```

Drive the robot slowly (recommended 0.2 m/s) using the joystick controller or keyboard teleop to build the map:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  cmd_vel:=/b2_unit_001/controls/cmd_vel
```

Once satisfied with the map, save it using:

```
ros2 run nav2_map_server map_saver_cli -f \
  /opt/mybotshop/src/mybotshop/b2_nav2/maps/custom_map \
  --ros-args --remap map:=/b2_unit_001/map
```

Note: Replace b2_unit_001 with your robot's namespace. Maps are saved to the b2_nav2/maps/ directory. After saving, rebuild the workspace with colcon build --symlink-install so the new map can be found.

SLAM Configuration:

Configuration file: b2_nav2/param/nav2_slam.yaml

```
slam_toolbox:
  ros__parameters:
    solver_plugin: solver_plugins::CeresSolver
```

```
    resolution: 0.05
    max_laser_range: 40.0
    mode: mapping
```

Run odometry-based navigation without a pre-built map. This mode navigates in the global map frame using only odometry for localization:

```
ros2 launch b2_nav2 odom_navi.launch.py
```

This is useful for relative navigation where you don't have a map of the environment yet.

Use a pre-built 2D map for autonomous navigation with AMCL localization:

```
ros2 launch b2_nav2 map_navi.launch.py
```

This launches the full Nav2 stack including:

- Map Server - loads pre-built occupancy grid maps
- AMCL - Adaptive Monte Carlo Localization
- Controller Server - path following (DWB controller)
- Planner Server - global path planning (NavFn)
- Behavior Server - recovery behaviors
- Waypoint Follower - for multi-point navigation

For outdoor GPS-based navigation:

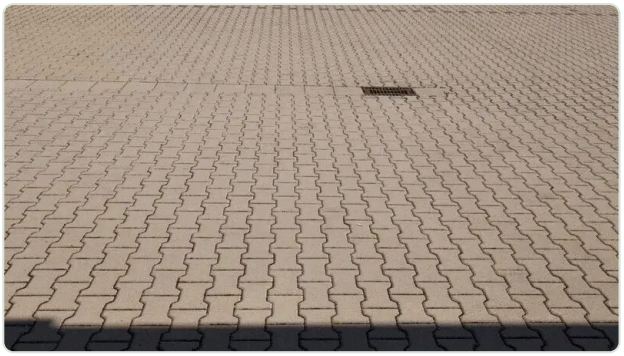
```
ros2 launch b2_nav2 gps_navi.launch.py
```

Launch File	Description
slam.launch.py	2D SLAM mapping mode
odom_navi.launch.py	Odometric navigation (no map)
map_navi.launch.py	Full autonomous navigation with map
gps_navi.launch.py	GPS-based outdoor navigation

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W supports autonomous navigation using Nav2.



B2 Navigation Demo

Create a map using SLAM:

```
ros2 launch b2_navigation slam.launch.py
```

Save the map when complete:

```
ros2 run nav2_map_server map_saver_cli -f ~/maps/my_map
```

Localize on a saved map:

```
ros2 launch b2_navigation localization.launch.py \
  map:=~/maps/my_map.yaml
```

Launch full navigation stack:

```
ros2 launch b2_navigation navigation.launch.py \
  map:=~/maps/my_map.yaml
```

Via RViz2:

Use the “2D Goal Pose” tool to send navigation goals.

Via Command Line:

```
ros2 action send_goal /navigate_to_pose \
  nav2_msgs/action/NavigateToPose \
  "{pose: {header: {frame_id: 'map'}, pose: {position: {x: 1.0, y: 2.0, z: 0.0},
orientation: {w: 1.0}}}}"
```

The B2W provides multiple odometry sources:

- /b2_unit_001/odom - Leg odometry
- /b2_unit_001/odom_imu - IMU-fused odometry
- /odometry/filtered - EKF filtered odometry (if configured)

Sensors

Humble

The B2W is equipped with dual Livox Mid360 LiDAR sensors for 3D perception.

Network Configuration:

Sensor	IP Address
Livox Mid360 (Front)	192.168.123.120
Livox Mid360 (Rear)	192.168.123.121

Launch Options:

```
# Launch both LiDARs
ros2 launch b2_lidars livox_mid360.launch.py lidar_position:=both

# Launch front only
ros2 launch b2_lidars livox_mid360.launch.py lidar_position:=front

# Launch rear only
ros2 launch b2_lidars livox_mid360.launch.py lidar_position:=rear
```

Launch Arguments:

Argument	Default	Description
lidar_position	both	front, rear, or both
xfer_format	0	Data transfer format
publish_freq	10.0	Publishing frequency (Hz)
namespace		Topic namespace

Published Topics:

Topic	Type
livox/front/lidar	sensor_msgs/PointCloud2
livox/rear/lidar	sensor_msgs/PointCloud2
livox/front/imu	sensor_msgs/Imu
livox/rear/imu	sensor_msgs/Imu
scan	sensor_msgs/LaserScan

The B2W can be equipped with Intel RealSense D435i depth cameras.

For RealSense documentation, see: https://github.com/IntelRealSense/realsense-ros

The b2_dynamixel package controls Dynamixel servos for pan-tilt mechanisms using Protocol 2.0.

Hardware Setup:

- Supported Servos: XM, XH, XC, XL430, XW series
- Interface: U2D2 USB adapter
- Serial Port: /dev/b2/dynamixel
- Baud Rate: 57600

Configuration:

Configuration file: b2_dynamixel/config/dynamixel.yaml

```
dynamixel_driver:
  ros__parameters:
    port: '/dev/b2/dynamixel'
    baudrate: 57600
    servo_ids: [0, 1]
    position_min: [1150, 230]
    position_max: [3200, 900]
    profile_velocity: 100
    profile_acceleration: 50
    publish_rate: 20.0
    position_topic_prefix: 'dynamixel/servo'
    feedback_topic: 'dynamixel/joint_states'
```

Topics and Services:

Topic/Service	Type	Description
dynamixel/servo_0/command	Float64	Pan (1150-3200)
dynamixel/servo_1/command	Float64	Tilt (230-900)
dynamixel/joint_states	JointState	Positions (rad)
dynamixel/reboot	Trigger	Clear errors

Usage:

```
# Launch driver
ros2 launch b2_dynamixel dynamixel.launch.py

# Center pan servo (default: 2170)
ros2 topic pub --once /dynamixel/servo_0/command \
  std_msgs/msg/Float64 "data: 2170"

# Center tilt servo (default: 565)
ros2 topic pub --once /dynamixel/servo_1/command \
  std_msgs/msg/Float64 "data: 565"

# Read joint states
ros2 topic echo /dynamixel/joint_states
```

```
# Reboot servos (clear hardware errors)
ros2 service call /dynamixel/reboot std_srvs/srv/Trigger
```

Udev Rules:

To create persistent device symlinks:

```
echo 'SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", \
  ATTRS{idProduct}=="6014", SYMLINK+="b2/dynamixel", \
  MODE="0666"' | sudo tee /etc/udev/rules.d/99-dynamixel.rules
sudo udevadm control --reload-rules
sudo udevadm trigger
```

Important: EXPERIMENTAL: The Z1 driver is under active development.

The b2_z1 package provides a ROS 2 driver for the Unitree Z1 robotic arm.

Quick Start:

```
# Launch the driver
ros2 launch b2_z1 z1.launch.py

# Enable the arm (required before sending commands)
ros2 service call /b2_unit_001/z1/enable std_srvs/srv/Trigger

# Send trajectory command
ros2 topic pub --once /b2_unit_001/z1/joint_trajectory \
  trajectory_msgs/msg/JointTrajectory "{
    joint_names: ['joint1', 'joint2', 'joint3',
      'joint4', 'joint5', 'joint6', 'jointGripper'],
    points: [{
      positions: [0.0, 1.57, -1.57, 0.0, 0.0, 0.0, 0.0],
      time_from_start: {sec: 3, nanosec: 0}
    }]
  }"

# Disable the arm
ros2 service call /b2_unit_001/z1/disable std_srvs/srv/Trigger
```

Topics and Services:

Name	Type	Description
z1/joint_states	sensor_msgs/JointState	Joint positions
z1/joint_trajectory	trajectory_msgs/JointTrajectory	Commands
z1/enable	std_srvs/Trigger	Enable arm
z1/disable	std_srvs/Trigger	Disable arm

URDF Integration:

Enable Z1 in the robot description:

```
export B2_Z1=1
ros2 launch b2_description display.launch.py
```

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W platform includes multiple sensor systems.

The robot is equipped with front and rear Livox Mid360 LiDAR units.

Network Configuration:

- Front LiDAR: 192.168.123.120
- Rear LiDAR: 192.168.123.121

Topics:

Topic	Type
/livox/front/lidar	sensor_msgs/PointCloud2
/livox/rear/lidar	sensor_msgs/PointCloud2
/livox/front/imu	sensor_msgs/Imu
/livox/rear/imu	sensor_msgs/Imu

Depth camera for close-range perception.

Topics:

Topic	Type
/camera/color/image_raw	sensor_msgs/Image
/camera/depth/image_rect_raw	sensor_msgs/Image
/camera/depth/color/points	sensor_msgs/PointCloud2
/camera/imu	sensor_msgs/Imu

Launch camera node:

```
ros2 launch realsense2_camera rs_launch.py
```

Pan-tilt camera mount using Dynamixel servos.

Configuration:

- Servo 0 (Pan): Range 1150-3200, Default 2170
- Servo 1 (Tilt): Range 230-900, Default 565

Control Topics:

```
# Set pan position
ros2 topic pub --once /dynamixel/servo_0/command \
  std_msgs/msg/Float64 "data: 2170"

# Set tilt position
ros2 topic pub --once /dynamixel/servo_1/command \
  std_msgs/msg/Float64 "data: 565"
```

Joint States:

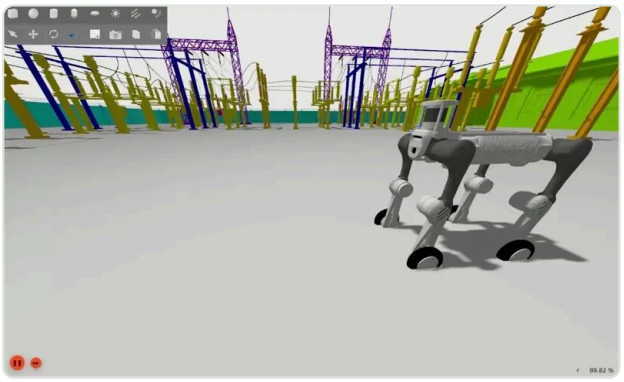
```
ros2 topic echo /b2_unit_001/joint_states
```

IMU Data:

```
ros2 topic echo /b2_unit_001/imu
```

Simulation

Humble



B2 Gazebo Simulation

The B2W simulation package uses ROS2 Humble with Gazebo Fortress, featuring the complete B2W platform with wheeled quadruped locomotion.

- ROS 2 Humble
- Gazebo Fortress
- gz_ros2_control

```
ros2 launch b2_gazebo b2_fortress_simulation.launch.py
```

```
ros2 launch b2_gazebo b2w_fortress_simulation.launch.py
```

To kill Gazebo zombie processes:

```
ros2 run b2_gazebo kill_gz.sh
```

Configure the simulation using environment variables before launch:

Variable	Default	Description
B2_DESC	1	Enable B2 robot description
B2_Z1	0	Include Z1 robotic arm

To visualize the robot in RViz2:

```
ros2 launch b2_viz view_robot.launch.py
```

To teleoperate the B2W using your keyboard:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  cmd_vel:=/b2_unit_001/hardware/cmd_vel
```

Note: Reduce speed by pressing z until 0.1 for safe indoor operation.

```
ros2 action send_goal \
  /joint_effort_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory -f "{
    trajectory: {
      joint_names: [
        'FL_hip_joint', 'FL_thigh_joint', 'FL_calf_joint',
        'FR_hip_joint', 'FR_thigh_joint', 'FR_calf_joint',
        'RL_hip_joint', 'RL_thigh_joint', 'RL_calf_joint',
        'RR_hip_joint', 'RR_thigh_joint', 'RR_calf_joint'
      ],
      points: [{
        positions: [0.0, 0.9, -1.5, 0.0, 0.9, -1.5,
          0.0, 0.9, -1.5, 0.0, 0.9, -1.5],
        time_from_start: {sec: 2, nanosec: 0}
      }]
    }
  }"
```

```
# Forward
ros2 topic pub /velocity_controller/commands \
  std_msgs/msg/Float64MultiArray \
  "{data: [-1.0, -1.0, -1.0, -1.0]}"
```

```
# Backward
ros2 topic pub /velocity_controller/commands \
  std_msgs/msg/Float64MultiArray \
  "{data: [1.0, 1.0, 1.0, 1.0]}"

# Stop
ros2 topic pub /velocity_controller/commands \
  std_msgs/msg/Float64MultiArray \
  "{data: [0.0, 0.0, 0.0, 0.0]}"
```

If controllers are not starting, manually activate them:

```
ros2 control set_controller_state \
  -c /b2_unit_001/controller_manager \
  joint_state_broadcaster configure

ros2 control set_controller_state \
  -c /b2_unit_001/controller_manager \
  joint_state_broadcaster activate
```

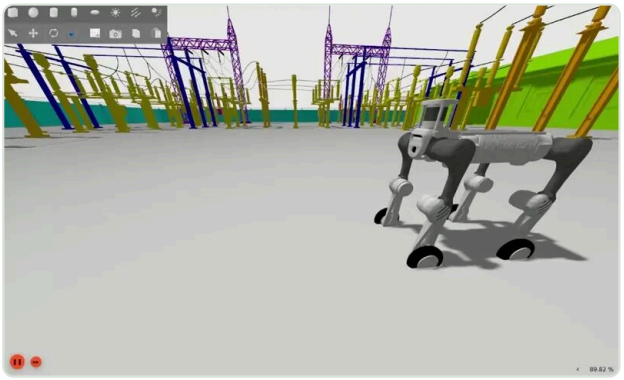
Check hardware interfaces:

```
ros2 control list_hardware_interfaces \
  -c /b2_unit_001/controller_manager
```

Foxy

Warning: ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W can be simulated using Gazebo.



B2 Gazebo Simulation

Note: ROS 2 Foxy uses Gazebo 11 (classic). For Ignition Gazebo Fortress, please use ROS 2 Humble.

Launch the B2W in Gazebo 11:

```
ros2 launch b2_gazebo spawn.launch.py
```

With Z1 arm:

```
export B2_Z1=1
ros2 launch b2_gazebo spawn.launch.py
```

Specify a custom world:

```
ros2 launch b2_gazebo spawn.launch.py \
  world:=warehouse.world
```

The simulated robot responds to the same topics as the real hardware:

```
# Velocity commands
ros2 topic pub /b2_unit_001/cmd_vel geometry_msgs/msg/Twist \
  "{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"

# Mode commands
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'stand_up'}"
```

Simulated sensors publish to the same topics as the real robot:

- LiDAR point clouds
- Camera images
- IMU data
- Joint states

Packages

Humble

The QRE B2W workspace is organized into two main categories: mybotshop (custom MBS packages) and third_party (external dependencies).

Located under: /opt/mybotshop/src/mybotshop/

Package	Version	Description
b2_platform	1.0.0	Unitree SDK hardware interface, odometry, video streaming
b2_interface	1.0.0	Custom ROS 2 services (B2Modes.srv, B2Light.srv)
b2_dynamixel	1.0.0	Dynamixel servo control for pan-tilt mechanism
b2_bringup	1.0.0	System startup and service installer
b2_description	1.0.0	URDF/Xacro robot descriptions

b2_z1	1.0.0	Unitree Z1 arm driver

Package	Version	Description
b2_lidars	1.2.0	Livox Mid360 dual-LiDAR integration
b2_depth_camera	1.2.0	RealSense D435i depth camera
b2_usbcam	1.2.0	USB camera streaming
b2_gps	1.2.0	GPS/GNSS integration (Fixposition, Drotek)
b2_vision_action	1.0.0	Computer vision and object detection

Package	Version	Description
b2_control	1.2.0	Twist mux, teleop, EKF localization
b2_nav2	1.0.0	Nav2 navigation stack, SLAM
b2_webserver	2.0.0	Web-based remote operation interface
b2_viz	1.0.0	RViz visualization configurations
b2_gazebo	1.0.0	Gazebo Fortress simulation
b2_steamdeck	1.0.0	Steam Deck wireless teleoperation

The b2_platform package provides ROS 2 integration with the Unitree SDK 2.0.

Key Launch Files:

- hardware.launch.py - Main hardware control
- state_publisher.launch.py - Robot state publishing
- front_video.launch.py - Front camera streaming
- rear_video.launch.py - Rear camera streaming
- bridge.launch.py - ROS domain bridging

Configuration:

Configuration file: b2_platform/config/b2_platform.yaml

```
robot_odom_topic: "platform/odom"
robot_odom_publisher_on: true
transform_odom_frame: "odom"
transform_base_frame: "base_link"
front_camera_port: 1720
rear_camera_port: 1721
```

Custom ROS2 message and service definitions for B2W packages.

Key Services:

- B2Modes.srv - Robot mode control (stand_up, stand_down, gaits, etc.)
- B2Light.srv - LED light control

Control system package for robot base. Provides twist mux, teleop configuration, and EKF localization.

Key Launch Files:

- twistmux.launch.py - Twist mux configuration
- control.launch.py - Full control stack
- teleop.launch.py - Teleop configuration
- ekf_localization.launch.py - EKF localization

EKF Localization:

```
ros2 launch b2_control ekf_localization.launch.py
```

Configuration file: b2_control/config/localization.yaml

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

Overview of the B2W ROS 2 packages.

Package	Description
b2_driver	Hardware interface to Unitree B2 SDK
b2_interface	Custom message and service definitions
b2_description	URDF/Xacro robot description
b2_bringup	Launch files and configuration

Package	Description
livox_ros_driver2	Livox Mid360 LiDAR driver
realsense2_camera	Intel RealSense camera driver
dynamixel_driver	Dynamixel servo control

Package	Description
b2_navigation	Nav2 configuration and launch files
b2_slam	SLAM configuration

b2_interface/srv/B2Modes:

```
string request_data
---
bool success
string message
```

Common mode values:

- stand_up , stand_down , damp , recovery
- gait_idle , gait_trot , gait_flatwalk
- speed_low , speed_high
- body_height_low , body_height_mid , body_height_high

Debugging

Humble

This section covers debugging tools and techniques for the B2W ROS2 system.

```
# Verify ROS_DOMAIN_ID
echo $ROS_DOMAIN_ID  # Should be 10

# Check network
ping 192.168.123.165
```

```
# Check USB device
ls -la /dev/b2/dynamixel

# Reboot servos
ros2 service call /dynamixel/reboot std_srvs/srv/Trigger
```

```
# Verify network
ping 192.168.123.120  # Front
ping 192.168.123.121  # Rear
```

```
# Check what environment the service sees
sudo systemctl show b2-hardware \
  --property=Environment

# Verify startup.bash sources correctly
cat /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

Warning: Do not run duplicate nodes! If a service is running (e.g., b2-hardware is active), the corresponding launch file is already running. Manually launching the same file will create duplicate nodes, causing conflicts.

Check Before Manual Launch:

Always run this before manually launching any ROS2 nodes:

```
# Quick check - shows all active b2 services
systemctl list-units --type=service --state=active | grep b2

# Or check specific service
systemctl is-active b2-hardware && \
  echo "SERVICE RUNNING - DO NOT LAUNCH MANUALLY"
```

If you accidentally ran duplicate nodes:

```
# Kill all ROS2 nodes in namespace
pkill -f "ros2.*b2_unit_001"

# Restart the service
sudo systemctl restart b2-hardware
```

- Red marker - The startup job has failed
- Green marker - Everything is working correctly
- Grey marker - The service has not started yet

```
# Restart B2 services
sudo systemctl restart b2-hardware
sudo systemctl restart b2-webserver
```

```
# View recent logs for a service
journalctl -u b2-hardware -n 50

# Follow logs in real-time
journalctl -u b2-webserver -f

# View logs since last boot
journalctl -u b2-hardware -b
```

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

This section covers debugging tools and techniques for the B2W ROS 2 system.

```
# Verify ROS_DOMAIN_ID
echo $ROS_DOMAIN_ID  # Should be 10

# Check network
ping 192.168.123.165
```

```
# Check USB device
ls -la /dev/b2/dynamixel

# Reboot servos
ros2 service call /dynamixel/reboot std_srvs/srv/Trigger
```

```
# Verify network
ping 192.168.123.120  # Front
ping 192.168.123.121  # Rear
```

```
# Check what environment the service sees
sudo systemctl show b2-hardware --property=Environment

# Verify startup.bash sources correctly
cat /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

Warning: Do not run duplicate nodes! If a service is running (e.g., b2-hardware is active), the corresponding launch file is already running.

Check Before Manual Launch:

```
# Quick check - shows all active b2 services
systemctl list-units --type=service --state=active | grep b2

# Or check specific service
systemctl is-active b2-hardware && \
  echo "SERVICE RUNNING - DO NOT LAUNCH MANUALLY"
```

If you accidentally ran duplicate nodes:

```
# Kill all ROS2 nodes in namespace
pkill -f "ros2.*b2_unit_001"

# Restart the service
sudo systemctl restart b2-hardware
```

```
sudo systemctl restart b2-hardware
sudo systemctl restart b2-webserver
```

```
# View recent logs
journalctl -u b2-hardware -n 50

# Follow logs in real-time
journalctl -u b2-webserver -f
```

Miscellanious

Humble

The B2W supports various operational modes controlled via ROS 2 services.

Available Modes:

Mode	Description
damp	Damping mode (safe state)
stand_up	Stand up from lying position
stand_down	Lie down safely
recovery	Recovery mode for fault conditions
stop_move	Stop all movement
gait_idle	Idle gait (standing)
gait_trot	Trotting gait
gait_trot_running	Running trot gait
gait_visualwalk	Visual walking mode
gait_flatwalk	Flat terrain walking
speed_low / speed_high	Speed settings
body_height_low/mid/high	Body height settings

Mode Control Examples:

```
# Stand up
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'stand_up'}"

# Stand down (safe shutdown position)
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'stand_down'}"

# Enter damping mode
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'damp'}"

# Set trotting gait
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'gait_trot'}"

# Recovery mode
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'recovery'}"
```

Speed Settings:

```
# Set low speed
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'speed_low'}"

# Set high speed
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'speed_high'}"
```

Body Height Settings:

```
# Low body height
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'body_height_low'}"

# Medium body height
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'body_height_mid'}"

# High body height
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'body_height_high'}"
```

The low-level control for the B2 can be directly accessed via Unitree's provided examples in their documentation . Running the provided qre_b2 driver with the low-level commands from Unitree examples should work concurrently.

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

The B2W supports various operational modes controlled via ROS 2 services.

Available Modes:

Mode	Description
damp	Damping mode (safe state)
stand_up	Stand up from lying position
stand_down	Lie down safely
recovery	Recovery mode for fault conditions
stop_move	Stop all movement
gait_idle	Idle gait (standing)
gait_trot	Trotting gait

gait_trot_running	Running trot gait
gait_flatwalk	Flat terrain walking
speed_low / speed_high	Speed settings
body_height_low/mid/high	Body height settings

Mode Control Examples:

```
# Stand up
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'stand_up'}"

# Stand down
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'stand_down'}"

# Enter damping mode
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'damp'}"
```

Speed Settings:

```
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'speed_low'}"

ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'speed_high'}"
```

Body Height Settings:

```
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'body_height_low'}"

ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'body_height_mid'}"

ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'body_height_high'}"
```

The low-level control for the B2 can be directly accessed via Unitree's provided examples in their documentation .

Low-Level Control

The low-level control for the B2 can be directly accessed via Unitree's provided examples in their documentation . Running the provided qre_b2 driver with the low-level commands from Unitree examples should work concurrently.

Installation

Humble

This section covers the installation procedures for the B2W system.

> **Note:** This should already be configured by the MYBOTSHOP team.

1. Create workspace:

```
sudo hostnamectl set-hostname b2-unit-366
sudo mkdir /opt/mybotshop
sudo chown -R unitree:unitree /opt/mybotshop
```

1. Set timezone:

```
sudo timedatectl set-timezone Europe/Berlin
sudo date -s "$(wget --method=HEAD -qSO- \
  --max-redirect=0 google.com 2>&1 | \
  sed -n 's/^ *Date: *//p')"
```

1. Run installer:

```
cd /opt/mybotshop/src/mybotshop
./b2_install.bash
```

1. Build workspace:

```
cd /opt/mybotshop
colcon build --symlink-install
source install/setup.bash
```

1. Add to .bashrc :

```
source /opt/ros/humble/setup.bash
source /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

1. Install services:

```
ros2 run b2_bringup startup_installer.py
```

1. Configure secondary IP for camera stream:

```
sudo nano /etc/netplan/01-network-manager-all.yaml
```

Add the following configuration:

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    eth0:
      dhcp4: false
      addresses: [192.168.123.164/24,192.168.123.170/24]
      nameservers:
        addresses: [114.114.114.114,8.8.8.8]
```

Apply the configuration:

```
sudo netplan apply
```

1. Source ROS and build:

```
source /opt/ros/humble/setup.bash
colcon build --symlink-install
```

1. Configure network:

Connect to 192.168.123.x network with static IP:

- Address: 192.168.123.51
- Netmask: 24

1. Source setup:

```
source /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

1. Verify connection:

```
ping 192.168.123.164
ssh -X unitree@192.168.123.164
# Password: Unitree0408
```

To sync your development workspace to the robot:

```
rsync -avP -t --delete -e ssh src \
  unitree@192.168.123.164:/opt/mybotshop
```

Foxy

> **Warning:** ROS 2 Foxy Fitzroy reached end-of-life in June 2023. We recommend upgrading to ROS 2 Humble for continued support.

This section covers the installation procedures for the B2W system.

> **Note:** This should already be configured by the MYBOTSHOP team.

1. Create workspace:

```
sudo hostnamectl set-hostname b2-unit-366
sudo mkdir /opt/mybotshop
sudo chown -R unitree:unitree /opt/mybotshop
```

1. Set timezone:

```
sudo timedatectl set-timezone Europe/Berlin
```

1. Run installer:

```
cd /opt/mybotshop/src/mybotshop
./b2_install.bash
```

1. Build workspace:

```
cd /opt/mybotshop
colcon build --symlink-install
source install/setup.bash
```

1. Add to .bashrc :

```
source /opt/ros/foxy/setup.bash
source /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

1. Install services:

```
ros2 run b2_bringup startup_installer.py
```

1. Source ROS and build:

```
source /opt/ros/foxy/setup.bash
colcon build --symlink-install
```

1. Configure network: Address: 192.168.123.51 Netmask: 24
2. Verify connection:

```
ping 192.168.123.164
ssh -X unitree@192.168.123.164
```

```
rsync -avP -t --delete -e ssh src \
```

```
unitree@192.168.123.164:/opt/mybotshop
```

8 B2-W OTT – Mobile Manipulator Build

B2-W OTT

QRE Advanced B2-W Manipulator

The QRE Advanced B2-W Manipulator is a fully integrated mobile manipulation platform combining the Unitree B2-W wheeled quadruped robot with the Unitree Z1 robotic arm. This system is designed for advanced research and development applications requiring both mobility and manipulation capabilities.

System Overview

The B2-W OTT platform integrates:

- Unitree B2-W Wheeled Quadruped: High-speed wheeled locomotion with 50rad/s wheel velocity
- Unitree Z1 Robotic Arm: 6-DOF manipulator with gripper for object manipulation
- Dual Livox Mid360 LiDAR: Front and rear 3D perception for navigation and mapping
- Intel RealSense D435i: Depth camera for close-range perception and manipulation
- Pan-Tilt Camera System: Dynamixel-actuated camera mount for enhanced field of view

Specifications

Physical Dimensions:

- Standing Height: 758mm
- Folded Height: 450mm
- Weight (with battery): About 85kg

Mobility:

- Maximum Wheel Velocity: 50rad/s
- Maximum Standing Load: About 40kg
- Continuous Walking Load: About 120kg

Payload Configuration:

The robot supports various payload configurations including the Z1 arm, pan-tilt camera systems, sensor arrays, and custom payloads.

Quick Start Guide

1. Power On:

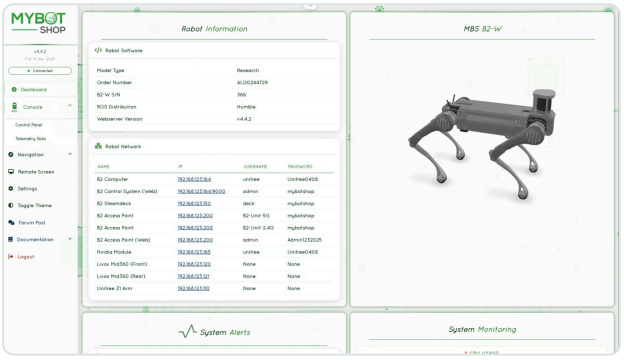
Press the battery power button once, then hold for 3 seconds. The robot will initialize and the ROS2 services will start automatically.

2. Verify Services:

```
systemctl status b2-hardware
systemctl status b2-webserver
```

3. Access Webserver:

Navigate to http://192.168.123.164:9000 in a web browser for the control interface.



B2 Webserver Dashboard

4. Stand Up:

```
ros2 service call /b2_unit_001/hardware/modes \
  b2_interface/srv/B2Modes "{request_data: 'stand_up'}"
```

5. Control Movement:

```
ros2 topic pub /b2_unit_001/cmd_vel geometry_msgs/msg/Twist \
  "{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

Robot Modes

The B2-W supports various operational modes:

Mode	Description

damp	Damping mode (safe state)
stand_up	Stand up from lying position
stand_down	Lie down safely
recovery	Recovery mode for fault conditions
gait_idle	Idle gait (standing)
gait_trot	Trotting gait
gait_trot_running	Running trot gait
gait_flatwalk	Flat terrain walking
speed_low / speed_high	Speed settings
body_height_low/mid/high	Body height settings

Z1 Arm Control

Important: EXPERIMENTAL: The Z1 driver is under active development.

Enable the arm:

```
ros2 service call /b2_unit_001/z1/enable std_srvs/srv/Trigger
```

Send trajectory command:

```
ros2 topic pub --once /b2_unit_001/z1/joint_trajectory \
  trajectory_msgs/msg/JointTrajectory "{
    joint_names: ['joint1', 'joint2', 'joint3',
      'joint4', 'joint5', 'joint6', 'jointGripper'],
    points: [{
      positions: [0.0, 1.57, -1.57, 0.0, 0.0, 0.0, 0.0],
      time_from_start: {sec: 3, nanosec: 0}
    }]
  }"
```

Disable the arm:

```
ros2 service call /b2_unit_001/z1/disable std_srvs/srv/Trigger
```

Pan-Tilt Camera System

The pan-tilt camera mount uses Dynamixel servos:

- Servo 0 (Pan): Range 1150-3200, Default 2170
- Servo 1 (Tilt): Range 230-900, Default 565

Control via webserver sliders or ROS2 topics:

```
# Center pan servo
ros2 topic pub --once /dynamixel/servo_0/command \
  std_msgs/msg/Float64 "data: 2170"

# Center tilt servo
ros2 topic pub --once /dynamixel/servo_1/command \
  std_msgs/msg/Float64 "data: 565"
```

Sensor Systems

Livox Mid360 LiDAR:

- Front LiDAR: 192.168.123.120
- Rear LiDAR: 192.168.123.121

Topics:

- /livox/front/lidar - Front point cloud
- /livox/rear/lidar - Rear point cloud
- /livox/front/imu - Front IMU
- /livox/rear/imu - Rear IMU

Intel RealSense D435i:

Topics:

- /camera/color/image_raw - Color image
- /camera/depth/image_rect_raw - Depth image
- /camera/depth/color/points - Point cloud
- /camera/imu - Camera IMU

Navigation

SLAM Mapping:

```
ros2 launch b2_navigation slam.launch.py
```

Autonomous Navigation:

```
ros2 launch b2_navigation navigation.launch.py \
  map:=~/maps/my_map.yaml
```

Simulation

Launch the B2-W in Gazebo Fortress:

```
ros2 launch b2_gazebo spawn.launch.py
```

With Z1 arm:

```
export B2_Z1=1
ros2 launch b2_gazebo spawn.launch.py
```

Network Configuration

IP Address	Device	Username	Password
192.168.123.161	B2 MCU	x	x
192.168.123.164	B2 Nvidia Orin NX	unitree	Unitree0408
192.168.123.164:8080	B2 Webserver	admin	mybotshop
192.168.123.120	Livox Mid360 Front	x	x
192.168.123.121	Livox Mid360 Rear	x	x

ROS_DOMAIN_ID is set to 10 .

Debugging

Check Services:

```
systemctl list-units --type=service --state=active | grep b2
```

View Logs:

```
journalctl -u b2-hardware -f
```

Restart Services:

```
sudo systemctl restart b2-hardware
sudo systemctl restart b2-webserver
```

Safety Guidelines

Warning: Always maintain a safe distance from the robot during operation. The robot can move quickly and unexpectedly.

- Keep the emergency stop accessible at all times
- Ensure the operating area is clear of obstacles and personnel

- Monitor battery levels to prevent sudden shutdowns
- Do not exceed the maximum payload capacity
- Perform regular maintenance checks on mechanical components

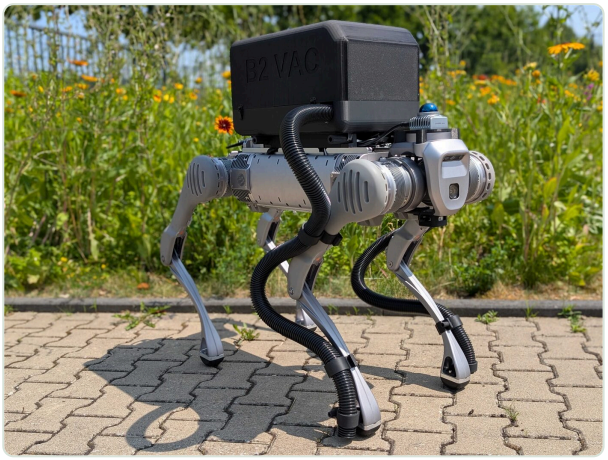
For detailed safety information, refer to the safety guidelines section.

Resources

- Unitree B2 Documentation
- Unitree Z1 Arm Documentation
- B2-W RnD User Manual

9 B2 Vacc – Autonomous Vacuum Demo Build

B2 Vacc



B2 Vacc Robot

The B2 Vacc is an autonomous vacuum demonstration platform based on the Unitree B2 quadruped robot. This system showcases advanced vision-based navigation and autonomous cleaning capabilities.

System Overview

The B2 Vacc platform integrates:

- Unitree B2 Quadruped: High-performance legged locomotion
- Robosense LiDAR: 3D perception for navigation and mapping
- Intel RealSense D435i: Depth camera for close-range perception
- Vision-based Detection: AI-powered mission detection system
- Webserver Interface: Remote monitoring and control

B2 Vacc Robot

Network Configuration

IP Addresses:

Name	IP Address	Username	Password
B2 MCU	192.168.123.161	x	x
B2 Robosense Lidar	192.168.123.162	x	x
B2 Onboard PC	192.168.123.164	unitree	Unitree0408

IP Reference:

- 192.168.123.161 - B2 MCU
- 192.168.123.164 - B2 Auxiliary PC (LAN only)

Note: This guide describes how to interface with the B2 robot using Ubuntu 20.04 and ROS2 Humble .

Important: Make sure B2 Edu is fully set up, paired, and verified through the app.

Static Network Connection

To configure the network initially:

1. Connect your PC to the B2 using a LAN cable .
2. In Ubuntu Settings → Network , click + to add a new connection.
3. In IPv4 settings , set the method to Manual : - Address: 192.168.123.51 - Netmask: 24
4. Save and restart your network.

To verify the connection:

```
ifconfig
ping 192.168.123.164
```

To SSH into the B2:

```
ssh -X unitree@192.168.123.164
# Password: Unitree0408
```

Note: It is recommended to disable all other networks while connecting to B2 to avoid interference.

Quick Start

Drivers Startup:

Note: Drivers should already be configured by the MBS-Team.

To launch RViz:

```
ros2 launch b2_viz view_robot.launch.py
```

Teleoperation:

```
ROS_DOMAIN_ID=10 ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  cmd_vel:=/b2_unit_001/hardware/cmd_vel

ROS_DOMAIN_ID=10 ros2 run teleop_twist_keyboard teleop_twist_keyboard \
  cmd_vel:=/b2_unit_001/controls/cmd_vel
```

Robot Modes

Warning: At startup, only the webserver is active. Please click Reactivate Services to enable all functionalities. Once reactivated, the robot will sit and then stand up - transitioning from AI Mode to Sports Mode . Be cautious: Sports Mode is very fast and responsive.

Available Modes:

- damp - Damping mode (safe state)
- stand_up - Stand up from lying position
- stand_down - Lie down safely
- recovery - Recovery mode for fault conditions
- stop_move - Stop all movement
- gait_idle - Idle gait (standing)
- gait_trot - Trotting gait
- gait_trot_running - Running trot gait
- gait_visualwalk - Visual walking mode
- gait_flatwalk - Flat terrain walking
- speed_low / speed_high - Speed settings
- body_height_low / body_height_mid / body_height_high - Body height settings

Modes are defined in: b2_platform > src > b2_highroscontrol.cpp

Mode Control:

```
# Stand up
ROS_DOMAIN_ID=10 ros2 service call /b2_unit_001/hardware/modes \
  b2_srvs/srv/B2Modes "{request_data: 'stand_up'}"

# Stand down
```

```
ROS_DOMAIN_ID=10 ros2 service call /b2_unit_001/hardware/modes \
  b2_srvs/srv/B2Modes "{request_data: 'stand_down'}"
```

Note: stand_up and stand_down have been verified. Please refer to Unitree documentation for details on other modes.

Sensors

Realsense D435i:

- The D435i must be manually connected from the main PC to PC4.
- It is off by default .

```
ros2 launch b2_depth_camera realsense_d435i.launch.py
```

Note: The Realsense D405 is also off by default.

Realsense Configuration:

- The launch file enables continuous depth streaming without lag.
- To modify parameters, update: b2_depth_camera/launch/realsense_d4XX.launch.py

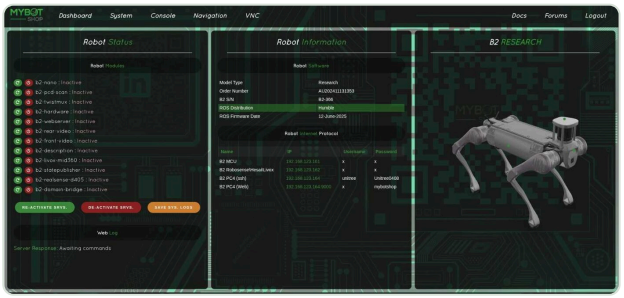
Webserver

The B2 web server can be accessed by connecting to the B2's Wi-Fi network:

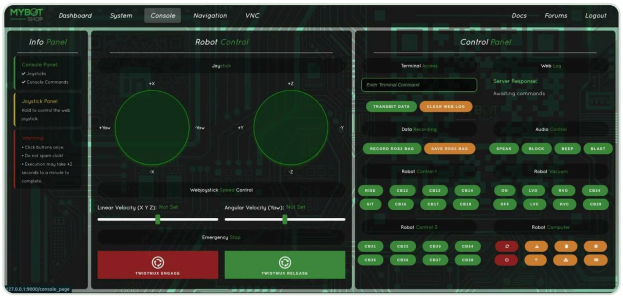
```
http://<robot_ip>:9000
```

Replace <robot_ip> with the actual IP address of the B2 (e.g., 192.168.123.164).

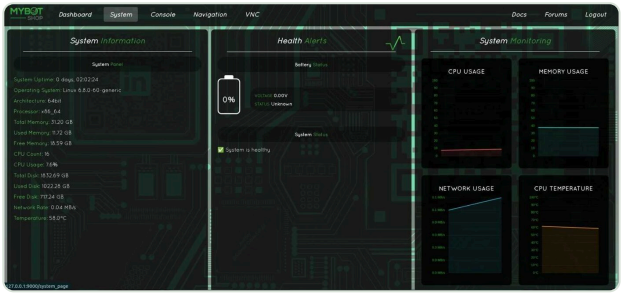
Note: The webserver provides useful control panels and monitoring features. Ensure your PC is connected directly to the B2 network.



B2 Webserver



B2 Webserver



B2 Webserver

Navigation

Odometric Navigation:

```
ros2 launch b2_navigation odom_navi.launch.py
```

Map Navigation:

Ensure that a map has been generated and is available in the ROS package directory.

```
ros2 launch b2_navigation map_navi.launch.py
```

SLAM (Simultaneous Localization and Mapping):

Ensure b2_bringup is already running.

```
ros2 launch b2_navigation slam.launch.py
```

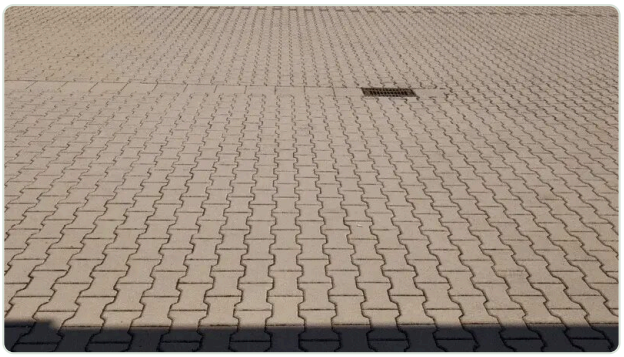
Use teleoperation (e.g., at 0.2 m/s) to map the area. Once complete, save the map:

```
ROS_DOMAIN_ID=10 ros2 run nav2_map_server map_saver_cli \
  -f /opt/mybotshop/src/mybotshop/b2_nav2/maps/custom_map \
  --ros-args --remap map:=/b2_unit_001/map
```

After saving the map, rebuild your workspace:

```
colcon build --symlink-install
source /opt/mybotshop/install/setup.bash
```

Autonomous Vacuum Demo



B2 Vacuum Demo

Launch mission detection and movement:

```
ros2 launch b2_vision_action mission.launch.py
```

Save training images for RoboFlow dataset:

```
ros2 run b2_vision_action libimagecapture.py
```

Train the vision model:

```
python3 train.py
```

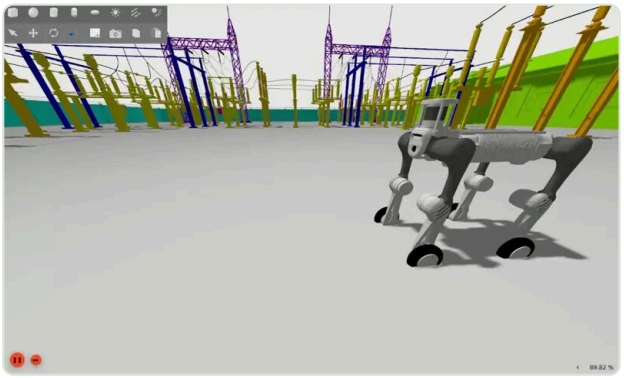
Motion behavior configuration:

Modify: b2_vision_action/b2_vision_action/librosnode.py

Webserver control:

```
http://192.168.68.168:9000/console_page
```

Simulation



B2 Gazebo Simulation

B2 Fortress Simulation:

```
ros2 launch b2_gazebo b2_fortress_simulation.launch.py
```

B2W Fortress Simulation:

```
ros2 launch b2_gazebo b2w_fortress_simulation.launch.py
```

Cleanup After Simulation:

```
ros2 run b2_gazebo kill_gz.sh
```

Joint Control

Effort Trajectory Control:

```
ros2 action send_goal /joint_effort_controller/follow_joint_trajectory \
  control_msgs/action/FollowJointTrajectory -f "{
  trajectory: {
    joint_names: [
      'FL_hip_joint', 'FL_thigh_joint', 'FL_calf_joint',
      'FR_hip_joint', 'FR_thigh_joint', 'FR_calf_joint',
      'RL_hip_joint', 'RL_thigh_joint', 'RL_calf_joint',
      'RR_hip_joint', 'RR_thigh_joint', 'RR_calf_joint'
    ],
    points: [
      {
        positions: [0.0, 0.9, -1.5, 0.0, 0.9, -1.5,
                    0.0, 0.9, -1.5, 0.0, 0.9, -1.5],
        time_from_start: {sec: 2, nanosec: 0}
      }
    ]
  }
}"
```

B2W Wheeled Movement:

```
# Move Forward
ros2 topic pub /velocity_controller/commands \
  std_msgs/msg/Float64MultiArray "{data: [-1.0, -1.0, -1.0, -1.0]}"

# Move Backward
ros2 topic pub /velocity_controller/commands \
  std_msgs/msg/Float64MultiArray "{data: [10.0, 10.0, 10.0, 10.0]}"

# Stop Movement
ros2 topic pub /velocity_controller/commands \
  std_msgs/msg/Float64MultiArray "{data: [0.0, 0.0, 0.0, 0.0]}"
```

Auto-Startup Services

The B2 does not automatically start ROS jobs unless explicitly configured. By default, only the system.launch.py in b2_bringup is launched.

Check startup status:

```
sudo service b2-<service_name> status
```

Status Indicators:

- Red marker - Service failed
- Green marker - Service running correctly
- Grey marker - Service not started

Restart a service:

```
sudo service b2-<service_name> restart
```

Install new startup configuration:

Modify b2_bringup/system.launch.py , then:

```
ros2 run b2_bringup startup_installer.py
```

Camera Streaming

Front Camera:

```
gst-launch-1.0 udpsrc address=230.1.1.1 port=1720 multicast-iface=eth0 \
  ! application/x-rtp, media=video, encoding-name=H264 \
  ! rtph264depay ! h264parse ! avdec_h264 ! videoconvert ! autovideosink
```

Rear Camera:

```
gst-launch-1.0 udpsrc address=230.1.1.1 port=1721 multicast-iface=eth0 \
  ! application/x-rtp, media=video, encoding-name=H264 \
  ! rtph264depay ! h264parse ! avdec_h264 ! videoconvert ! autovideosink
```

Installation

B2 Nvidia Board:

1. Set hostname and directory:

```
sudo hostnamectl set-hostname B2-366
sudo mkdir /opt/mybotshop && sudo chown -R unitree:unitree /opt/mybotshop
```

1. Sync system time:

```
sudo timedatectl set-timezone Europe/Berlin
sudo date -s "$(wget --method=HEAD -qSO- --max-redirect=0 google.com 2>&1 | sed -n 's/^
*Date: *//p')"
```

1. Run installer:

```
cd /opt/mybotshop/src/mybotshop
./b2_install.bash
```

1. Build ROS2 workspace:

```
cd /opt/mybotshop
colcon build --symlink-install
source install/setup.bash
```

1. Update .bashrc :

```
source /opt/ros/humble/setup.bash
source /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

Host PC (Ubuntu 20.04 with ROS2 Humble):

1. Ensure ROS2 Humble is installed but not sourced before first run.
2. Install dependencies:

```
./b2_install.bash
```

1. Create workspace and clone qre_b2 to /opt/mybotshop
2. Build Cyclone DDS:

```
colcon build --packages-select cyclonedds
```

1. Source workspace:

```
source install/setup.bash
source /opt/ros/humble/setup.bash
```

1. Build repository twice :

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

1. Source the B2 bringup config:

```
source /opt/mybotshop/install/setup.bash
source /opt/mybotshop/src/mybotshop/b2_bringup/config/setup.bash
```

Miscellaneous

Sync Host to Unitree PC:

```
rsync -avP -t --delete -e ssh src unitree@192.168.123.164://opt/mybotshop
```

SSH Key Setup:

```
ssh-copy-id unitree@192.168.123.164
# Password: Unitree0408
```

Xacro to URDF:

```
export B2_DESC=1
ros2 run xacro xacro \
  /opt/mybotshop/src/mybotshop/b2_description/xacro/robot.xacro \
  > /opt/mybotshop/src/mybotshop/b2_description/xacro/b2.urdf
```

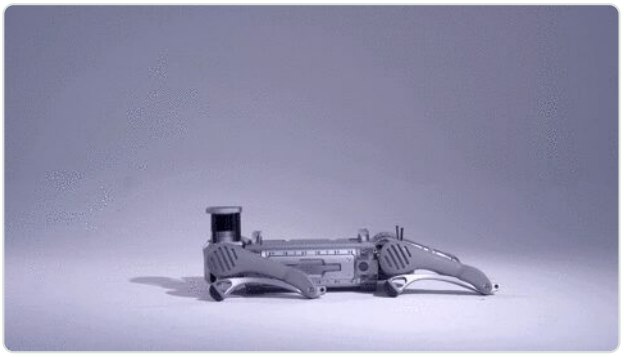
Software Updates

The latest software updates for QRE B2 can be pulled from the qre_b2 GitHub repository.

> **Note:** If you do not have access to the repository, contact support@mybotshop.de with your GitHub Username and Purchase ID .

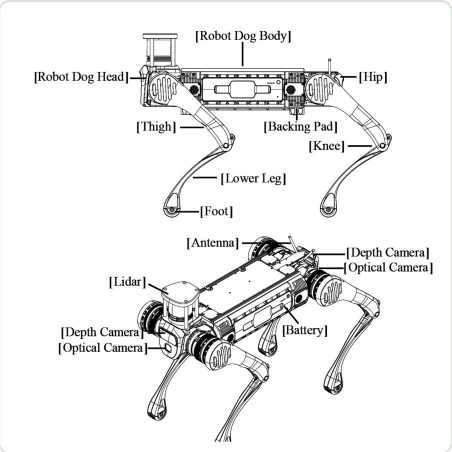
10 Hardware Overview & Specification

B2 Overview



B2 Robot

The Unitree B2 is a next-generation quadruped robot designed for advanced industrial, research, and commercial applications. As one of the most powerful quadruped robots developed by Unitree, the B2 combines high mobility, intelligent perception, and exceptional load-bearing capabilities, making it an ideal solution for a wide range of use cases, including industrial inspection, autonomous exploration, logistics support, and security surveillance. Equipped with high-torque joint actuators and precision control algorithms, the B2 is capable of navigating complex terrains with agility and stability. Its robust legged design allows it to traverse uneven surfaces, climb stairs, and adapt to dynamic environments with ease. The robot is powered by an advanced AI-driven control system, enabling autonomous decision-making and real-time obstacle avoidance. The Unitree B2 features a modular and expandable architecture, allowing users to integrate additional hardware such as robotic arms, LiDAR sensors, cameras, and custom payloads to tailor the robot to specific tasks. With a significant payload capacity, it can carry and manipulate objects efficiently, making it suitable for heavy-duty operations. Built with industrial-grade durability, the B2 can operate in harsh conditions while maintaining high performance and reliability. Its long battery life ensures extended operation times, and its wireless connectivity options allow seamless integration with remote control systems and cloud-based data processing. With its blend of power, adaptability, and intelligence, the Unitree B2 sets a new benchmark in quadruped robotics, providing industries, research institutions, and businesses with a highly capable robotic platform to enhance automation, efficiency, and innovation.



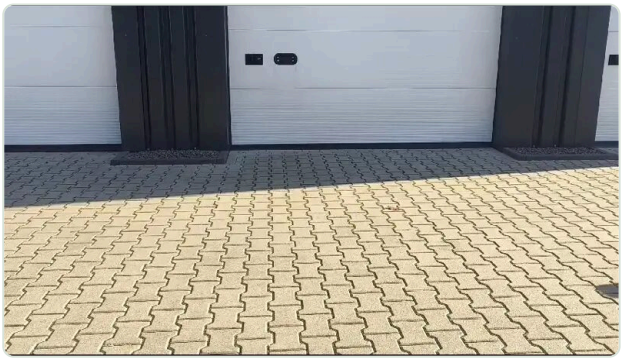
H1 Robot

B2-W Overview

As a wheeled variant of the Unitree B2, the B2-W combines the stability of legged locomotion with the speed and efficiency of wheels, making it a highly versatile platform for dynamic and high-speed operations.

With a modular and expandable architecture, the B2-W robust payload capacity and power efficiency make it suitable for heavy-duty applications, including logistics automation, security patrol, and industrial inspection.

Engineered for durability, the B2-W features weather-resistant construction, long battery life, and wireless connectivity options like 5G and Wi-Fi, ensuring remote operation, real-time data transmission, and cloud-based processing. Whether deployed in warehouse automation, autonomous exploration, or collaborative robotic tasks, the Unitree B2-W sets a new standard for efficiency, versatility, and intelligence in modern robotics.



H1 Robot

B2 Radar and Camera FOV

The B2 robot is equipped with advanced sensory systems, including the RS-Helios 5515 laser radar,D435i depth camera x2 and Optical Camera ×2 , which work in unison to provide superior environmental perception and spatial awareness.

RS-Helios 5515 Laser Radar

The RS-Helios 5515 laser radar is seamlessly integrated into the B2's head, providing outstanding environmental sensing capabilities. Utilizing omnidirectional and full-angle scanning technology, it achieves a horizontal field of view (FOV) of up to 360° and a maximum vertical angle of 59°. This advanced lidar system enables the B2 to capture highly precise, real-time environmental data, generating detailed, high-resolution point cloud information. By rapidly detecting and measuring surrounding objects, the B2 ensures accurate and comprehensive spatial awareness, enhancing its ability to navigate and interact with complex environments efficiently.

D435i Depth Camera

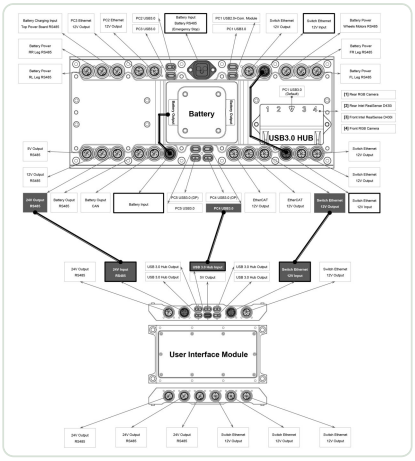
The B2 is also equipped with two Intel RealSense D435i depth cameras, one positioned at the front and the other at the rear, further enhancing its visual perception capabilities. These cameras enable the robot to accurately perceive and interpret its surroundings, providing depth information for precise spatial awareness and obstacle detection. With this advanced vision system, the B2 can intelligently navigate complex environments, adapt to dynamic scenarios, and interact more effectively with its surroundings, ensuring optimal performance across a wide range of applications.

Combined FOV of RS-Helios 5515 and D435i

The integration of the RS-Helios 5515 laser radar and the D435i depth camera provides a merged field of view, enabling the B2 robot to achieve unparalleled environmental perception and adaptability. Together, these sensors ensure the B2 can operate effectively across diverse tasks and dynamic environments.

B2 User Interface Module

The top of the B2 is equipped with a variety of User interfaces, allowing seamless integration with body joint motors, sensor peripherals, network ports, and other expansion modules. This well-structured interface design enhances ease of use, facilitating efficient debugging, troubleshooting, and secondary development. By providing a flexible and user-friendly platform, the B2 enables developers to customize and extend its functionalities, making it suitable for a wide range of applications in research, industry, and robotics innovation.



Top View of B2 Electrical Interface

Top View of B2 Electrical Interface

Below is a description of the connectors and their respective interface specifications:

No	Connector	Interface Description (Short)	Interface Specification
2	4x RS485	24V (2+2) Power Line Interface	24V/10A power output
3	Network	12V (8+2) Signal Line Interface	12V/10A power output Gigabit Ethernet (GbE)
6	4x Type-C	Type-C	Supports USB3.0 host, 5V/1.5A power output
10	5577	I/O OUT	5V/3A power output

B2 On-Board Computer

The B2 comes with a standard configuration featuring an Intel Core i5 processor for platform functions and an Intel Core i7 for user development. For enhanced performance, an optional configuration is available, offering an Intel Core i7 alongside up to three Nvidia Jetson Orin NX units, providing powerful computing capabilities for advanced applications and AI-driven tasks..

Development Computing Unit (PC 3)

Parameter	Specification
Model	Jetson Orin NX
CPU	Arm® Cortex®-A78AE
Number of Cores	8
Number of Threads	8
Maximum Clock Speed	2 GHz
Graphics Memory	16 GB
Total Memory	16 GB

Cache	2 MB L2 + 4 MB L3
Storage	2 TB
Intel® Image Processing Unit	Not Present
GPU	1024 NVIDIA Ampere Architecture GPUs with 32 Tensor Cores
Maximum GPU Frequency	918 MHz
Gaussian and Neural Accelerator	3.0
Intel® Deep Learning Boost	Yes
Intel® Adaptix™ Technology	Yes
Intel® Hyper-Threading Technology	Yes
Instruction Set	64-bit
OpenGL	4.6
OpenCL	3.0
DirectX	12.1
IP Address	192.168.123.

Attention: The Operation and Control Computing Unit is solely designated for the Unitree motion control program and is not available for public access. Developers are only authorized to use the Development Computing Unit for secondary development. To obtain the initial user password, please refer to the FAQ. In the table, the PC2 [Development Computing Unit] is assigned the IP address 192.168.123.. Additionally, the CPU modules may be shipped with a more advanced version, ensuring performance that meets or exceeds the listed specifications.

B2 Joint Motor

The B2 joint motor utilizes a self-developed Unitree motor that demonstrates exceptional performance and characteristics. With a maximum torque of 120 N·m, the motor features a hollow axis design, contributing to a more compact and lightweight structure. Additionally, the motor is equipped with dual encoders, enabling precise position and velocity feedback, which is essential for high-precision control requirements.

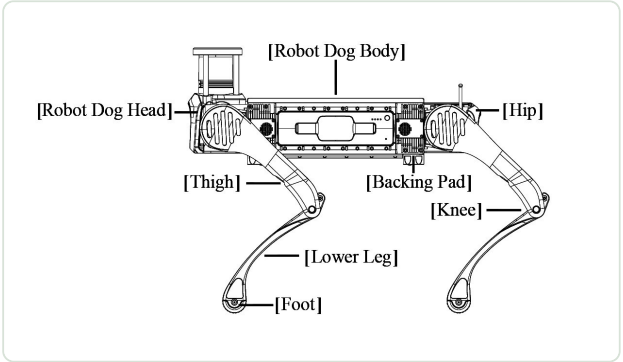
Joint Serial Number and Limits

The following table provides the joint indices, names, and respective joint limits in radians:

Index	Joint Name	Limit (rad)
1	FL_Hip	-0.87 ~ 0.87
2	FL_Thigh	-0.94 ~ 4.69

3	FL_Calf	-2.82 ~ 0.43
4	FR_Hip	-0.87 ~ 0.87
5	FR_Thigh	-0.94 ~ 4.69
6	FR_Calf	-2.82 ~ 0.43
7	RL_Hip	-0.87 ~ 0.87
8	RL_Thigh	-0.94 ~ 4.69
9	RL_Calf	-2.82 ~ 0.43
10	RR_Hip	-0.87 ~ 0.87
11	RR_Thigh	-0.94 ~ 4.69
12	RR_Calf	-2.82 ~ 0.43

The image below displays the Unitree B2 robot with all joints labeled. It serves as a visual reference, correlating the joint listed in the table to their positions on the robot.

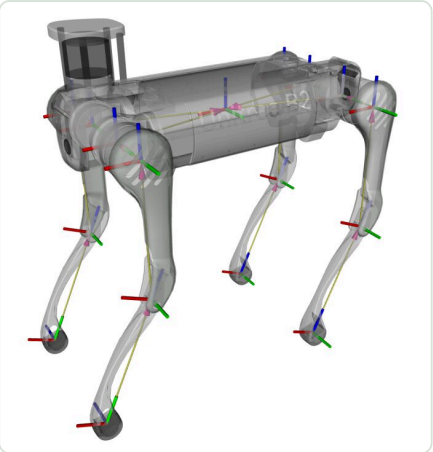


H1 Robot

Reference Frame, Joint Axes, and Zero Position

When all joints are in their zero positions, the coordinate systems for the joints are as follows:

- The X-axis is represented in red .
- The Y-axis is represented in green .
- The Z-axis is represented in blue .



Top View of B2 Electrical Interface

11 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 .

12 Getting Involved

Getting Involved

The software provided are Opensource, so contributions from community are welcome on our Github repository

13 FAQ

Frequently Asked Questions

(Coming Soon)