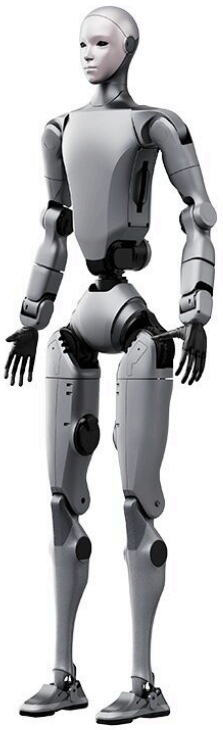
MYBOTSHOP

QUADRUPED ROBOTICS

Unitree H2

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 clarification. Contact information is available in Chapter 7, Getting Help & RMA.

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

The H2 is a full-size, high-power humanoid capable of forceful, dynamic motion. Maintain adequate clearance during operation, keep the emergency stop accessible at all times, and never allow untrained personnel to operate the robot. The H2 cannot actively maintain balance until it has entered motion-control mode — keep it supported and keep all personnel clear during startup and any posture transition.

Caution: Keep hands and clothing clear of the joints during all posture transitions to avoid pinch injuries. This product is intended exclusively for lawful civilian applications — hazardous modifications and deployment in dangerous contexts are strictly prohibited.

2 Disclaimer

The H2 robot documented here is sold and supported by *MYBOTSHOP / QUADRUPED Robotics* as the German distributor for Unitree Robotics. The H2 is controlled through Unitree's CycloneDDS-based communication stack (`unitree_sdk2` in C++ or Python), which operates with or without ROS 2 while remaining ROS 2 compatible via the separate `unitree_ros2` package — a working knowledge of C++ or Python and basic Linux networking is the practical prerequisite for the material in this manual.

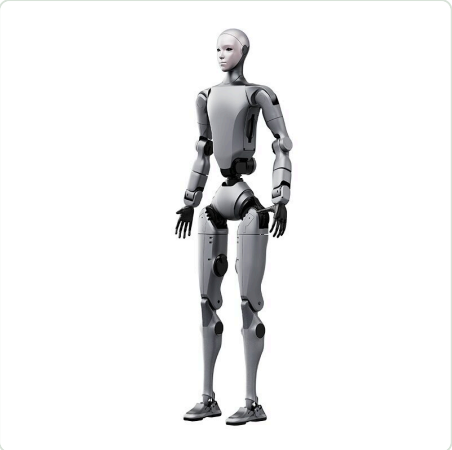
The client acknowledges and agrees that any information or materials provided by MYBOTSHOP / QUADRUPED Robotics are for R&D and development purposes only. Any services are provided "AS IS" and without any representation or warranty of any kind, express or implied, including but not limited to any warranty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

This product is intended for civilian use only. Users should avoid making dangerous modifications or using the robot in hazardous ways. Values summarised in this manual are drawn from Unitree's published material and may vary by model, firmware version, configuration, and operating conditions — some figures (e.g. exact battery capacity) have not been officially disclosed. Always cross-reference the official Unitree H2 Developer Guide before development or deployment. For terms, policies and compliance with local laws and regulations, refer to the Unitree Robotics website and the RMA/support process in Chapter 7.

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 H2

Unitree H2



Unitree H2 Full-Size Humanoid Robot

The Unitree H2 is a full-size, bionic humanoid robot built for natural, expressive whole-body motion and real-world task execution. With 31 degrees of freedom , high-torque joints, and a high-performance onboard AI compute platform, the H2 targets research, embodied-AI development, human–robot interaction, and advanced demonstration scenarios.

Expressive Whole-Body Motion

The H2 features 31 degrees of freedom , including 6-DOF legs , 7-DOF arms , a 3-DOF waist , and a 2-DOF head . Combined with joint torques up to 360 N·m , this kinematic configuration enables fluid, human-like movement — walking, balancing, gesturing, and dynamic full-body actions.

Onboard Intelligence

A high-performance AI compute platform rated at up to 2070 TOPS supports onboard perception, planning, and multimodal model inference. Paired with a binocular vision system and joint-level sensing, the H2 can perceive its environment and execute perception-driven behaviours in real time.

Practical Payload and Endurance

The H2 carries a 7 kg continuous payload, with short-duration capacity up to 21 kg , supported by a quick-release smart battery for fast swaps during extended operation.

Open Control and Developer-Friendly Design

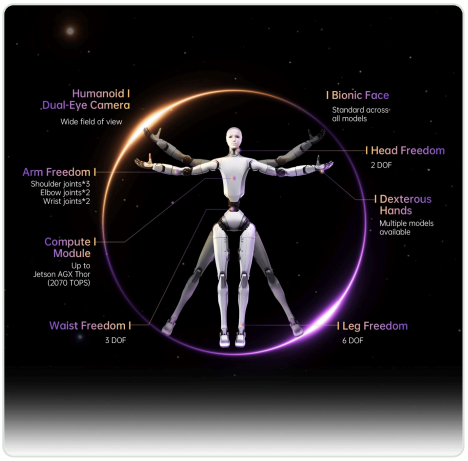
The H2 exposes fully open, low-latency control interfaces over Unitree's unitree_sdk2 (C++), unitree_sdk2_python (Python), and unitree_ros2 packages, communicating via DDS. It is suitable

for:

- Humanoid locomotion and whole-body control research
- Embodied AI and reinforcement learning
- Human–robot interaction experiments
- Manipulation with optional dexterous hands

> **Note:** For SDK setup, network configuration, and example commands, see the SDK Development section. The official developer reference is maintained by Unitree at the H2 Developer Guide .

Specifications



Unitree H2 parameter specification diagram

Official Unitree H2 parameter diagram

Physical & Mechanical

Parameter	Specification
Height (standing)	~180–182 cm
Weight (with battery)	~70 kg
Total Degrees of Freedom	31
Single Leg DOF	6
Single Arm DOF	7
Waist DOF	3
Head DOF	2
Peak Joint Torque	Up to 360 N·m

Performance

Parameter	Specification
Walking Speed	Under 2 m/s
Payload (continuous)	7 kg
Payload (short duration)	Up to 21 kg

Power

Parameter	Specification
Battery	Quick-release smart battery
Runtime	~3 hours (configuration dependent)
Hot Swap	Quick-release for fast exchange

Sensing & Compute

Parameter	Specification
AI Compute	Up to 2070 TOPS
Vision	Humanoid binocular camera (wide FOV)
Proprioception	IMU + joint-level sensing
Tactile (optional)	Via dexterous hands

Warning: Safety Notice: The H2 is a full-size, high-power humanoid capable of forceful, dynamic motion. Maintain adequate clearance during operation, keep the emergency stop accessible, and never allow untrained personnel to operate the robot. Civilian Use Only: This product is intended exclusively for lawful civilian applications. Hazardous modifications and deployment in dangerous contexts are strictly prohibited.

Note: Specification Source: Values summarised here are drawn from Unitree's published material and may vary by model, firmware version, configuration, and operating conditions. Some figures (e.g. exact battery capacity) have not been officially disclosed. Always cross-reference the official Unitree H2 Developer Guide before development or deployment.

4 Quick Start – Startup & Shutdown Procedure

Quick Start

Startup Procedure

Step 1: Pre-start Checks

Before powering on, suspend or securely support the H2 using a suitable support frame, with its front facing forward. Ensure the surrounding area is clear of people and obstacles, and arrange the arms and legs in a natural, untangled posture.

> **Warning:** The H2 is a full-size humanoid and cannot actively maintain balance until it has entered motion-control mode. Keep it supported and keep all personnel clear during startup.

Step 2: Install the Battery

Insert the smart battery into its compartment until the latch engages with a clear click . Confirm the pack is fully seated and secured. Do not force the battery — if it does not seat smoothly, remove it and check the orientation.

Step 3: Power On

Power on the battery using the standard Unitree sequence:

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

Wait approximately one minute for the robot to complete initialization and enter its zero-torque / damping state. Keep the environment unobstructed during this process.

Step 4: Stand and Operate

Using the remote controller, bring the H2 into its standing-preparation posture while firmly supporting it. Lower it until its feet make stable contact with the ground, then switch into motion-control mode. Verify the robot is balancing stably before releasing support or issuing movement commands.

> **Note:** Exact controller key combinations depend on the firmware build shipped with your unit. Refer to the controller reference supplied with the robot and the official Unitree H2 documentation for the current key map.

Emergency Stop

If the H2 enters an abnormal or unsafe state, trigger the emergency stop — either via the physical E-stop or the controller stop combination. The robot will switch to a damping/protective mode and lower itself toward the ground in a controlled manner.

Important: Always know the location of the emergency stop before operating the robot, and keep it within reach throughout operation.

Shutdown Procedure

1. Move the H2 to an open, level area with adequate clearance, keeping it supported.
2. Command the robot into its standing-preparation / resting posture using the controller.
3. Engage zero-torque / damping mode so the joints relax safely.
4. Briefly press the battery power button, then press and hold for more than 2 seconds to power down.

After shutdown, if the robot will be unused for an extended period, remove the battery pack for safe storage.

Attention: Keep hands and clothing clear of the joints during all posture transitions to avoid pinch injuries.

5 SDK Development & Network Setup

SDK Development

This section summarises how to set up and develop applications for the Unitree H2 using the official SDKs. It is based on Unitree's H2 Developer Guide and the public Unitree repositories. Always treat the official guide as the authoritative reference for your firmware version.

Overview

The H2 is controlled through Unitree's CycloneDDS-based communication stack, which operates with or without ROS 2 while remaining ROS 2 compatible. Three primary SDKs are available:

- unitree_sdk2 — the C++ SDK for high- and low-level control, state subscription, and sensor access. Repository: unitree_sdk2 .
- unitree_sdk2_python — the Python interface (pybind11 wrapper) for rapid prototyping. Repository: unitree_sdk2_python .
- unitree_ros2 — ROS 2 packages that bridge the same DDS interface into ROS 2 topics and messages. Repository: unitree_ros2 .

All Unitree open-source SDKs are published under the Unitree Robotics GitHub organisation .

Requirements

Item	Requirement
Operating System	Ubuntu 20.04 LTS
Compiler	GCC 9.4.0 (C++17)
Build System	CMake 3.10+
Architecture	x86_64 or aarch64
Transport	DDS (CycloneDDS)
Python (optional)	Python 3.8+

Installing Dependencies

Install the build toolchain and required libraries:

```
sudo apt update
sudo apt install -y cmake g++ build-essential \
    libyaml-cpp-dev libeigen3-dev libboost-all-dev \
    libspdlog-dev libfmt-dev
```

Building and Installing unitree_sdk2

Clone and build the C++ SDK:

```
git clone https://github.com/unitreerobotics/unitree_sdk2.git
cd unitree_sdk2
mkdir build && cd build
cmake ..
make
```

To install the SDK system-wide (recommended for use in your own CMake projects):

```
cd unitree_sdk2/build
cmake .. -DCMAKE_INSTALL_PREFIX=/opt/unitree_robotics
sudo make install
```

Installing the Python Interface

For Python development, install unitree_sdk2_python :

```
git clone https://github.com/unitreerobotics/unitree_sdk2_python.git
cd unitree_sdk2_python
pip3 install -e .
```

Network Configuration

The H2 communicates with a development PC over Gigabit Ethernet using CycloneDDS. Connect the robot and your computer with an Ethernet cable, then identify the interface that links to the robot:

```
# List network interfaces and find the one connected to the H2
ifconfig
```

Unitree robots use the 192.168.123.x subnet by convention. Ensure your development machine is on the same subnet as the robot's onboard computer. Configure your PC's wired interface to a static address on that subnet:

```
# Example: set a static IP on the wired interface (replace eth0 with your NIC)
sudo ip addr add 192.168.123.222/24 dev eth0
sudo ip link set eth0 up

# Confirm the robot is reachable (use the robot's documented address)
ping 192.168.123.<robot>
```

Important: The exact network setup and any robot-specific addresses are documented in the official H2 Developer Guide . Confirm the values for your unit before configuring your PC.

Specifying the Network Interface

unitree_sdk2 examples and applications must be told which network interface is connected to the robot. The interface name (for example eth0) is passed as a command-line argument when launching an example:

```
# Run a built example, passing the network interface name
./example_program eth0
```

In C++, the interface is supplied when initialising the DDS channel factory:

```
#include <unitree/robot/channel/channel_factory.hpp>

int main(int argc, char** argv)
{
    // argv[1] is the network interface connected to the H2 (e.g. "eth0")
    unitree::robot::ChannelFactory::Instance()->Init(0, argv[1]);
    // ... create publishers / subscribers and control the robot ...
    return 0;
}
```

The equivalent in Python:

```
import sys
from unitree_sdk2py.core.channel import ChannelFactoryInitialize

# sys.argv[1] is the network interface connected to the H2 (e.g. "eth0")
ChannelFactoryInitialize(0, sys.argv[1])
# ... create publishers / subscribers and control the robot ...
```

Control Interfaces

The SDK exposes both high-level and low-level control:

- High-level control issues task-space commands (locomotion modes, posture, velocity) and is the safe default for most applications.
- Low-level control commands individual joint targets (position, velocity, torque, stiffness, and damping). Low-level commands bypass built-in balancing safeguards and must remain within the documented joint limits.

The H2 joint indices, names, and motion limits used by the low-level interface are defined in the SDK headers and the official H2 Developer Guide . Always validate commanded joint motions against the documented limits before sending them.

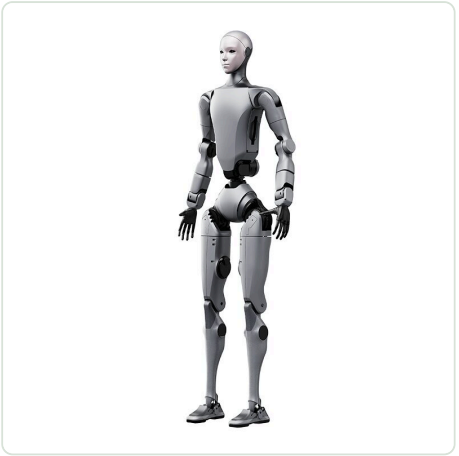
Warning: Custom low-level control code can command the robot into unsafe states. Test new control software with the H2 safely supported or in a clear, open area, keep the emergency stop within reach, and validate joint commands against the documented limits before deployment.

Further Resources

- H2 Developer Guide (official)
- unitree_sdk2 (C++ SDK)
- unitree_sdk2_python (Python interface)
- unitree_ros2 (ROS 2 packages)
- Unitree Robotics on GitHub
- Unitree Document Center

6 Hardware Overview & Specification

Hardware



Unitree H2 Full-Size Humanoid Robot

Component Overview

The Unitree H2 is a full-size bionic humanoid composed of an upper body and a lower body with a highly articulated, modular structure. It provides a total of 31 degrees of freedom , enabling expressive, human-like whole-body motion for locomotion, balancing, gesturing, and manipulation.

Each leg provides six degrees of freedom (hip, thigh, knee, and ankle joints) for stable, dynamic locomotion. Each arm provides seven degrees of freedom , supporting a wide manipulation workspace. A 3-DOF waist and 2-DOF head enhance balance, motion coordination, and environmental perception.

Degrees of Freedom

Category	Specification
Total DOF	31
Single Leg DOF	6
Single Arm DOF	7
Waist DOF	3
Head DOF	2

Joint Actuators

The H2 uses Unitree's self-developed electric joint actuators, engineered for high torque density, dynamic response, and precise closed-loop control. Selected joints deliver peak torque up to 360 N·m , and dual encoders provide high-resolution position and velocity feedback for accurate state estimation and whole-body coordination.

Parameter	Specification
Actuator Type	High-performance electric (PMSM)
Peak Joint Torque	Up to 360 N·m
Feedback	Dual encoder (position + velocity)

Note: Joint ordering, limits, and reference frames used by the control SDK are defined in the official H2 Developer Guide . Always confirm joint indices against the SDK header definitions before sending low-level commands.

Perception

Vision

The H2 integrates a humanoid binocular camera with a wide field of view, providing visual and depth perception for navigation, interaction, and manipulation. The forward-facing configuration supports perception-driven behaviours and human–robot interaction.

Proprioception

An onboard IMU combined with per-joint dual encoders provides the proprioceptive sensing required for balance, contact estimation, and stable whole-body locomotion control.

Tactile (Optional)

When equipped with optional dexterous hands , the H2 gains tactile sensing for fine manipulation and controlled object interaction.

Onboard Computer

The H2 ships with a high-performance AI compute platform rated at up to 2070 TOPS , providing substantial headroom for onboard perception, planning, and multimodal model inference.

Parameter	Specification
AI Compute	Up to 2070 TOPS
Workloads	Perception, planning, multimodal inference

Important: For security reasons, change any default login credentials after first access and follow proper shutdown procedures before performing hardware or software modifications.

Power System

The H2 is powered by a quick-release smart battery , enabling fast swaps to extend operating sessions. Runtime is approximately 3 hours , depending on workload and configuration.

Parameter	Specification
Battery	Quick-release smart battery
Runtime	~3 hours (configuration dependent)
Swap	Quick-release for fast exchange

Warning: Always verify that the battery is fully latched before operation. If the H2 will be stored for an extended period, remove the battery pack. Handle batteries according to lithium-battery safety guidelines.

7 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 .

8 Getting Involved

Getting Involved

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

9 FAQ

Frequently Asked Questions

(Coming Soon)