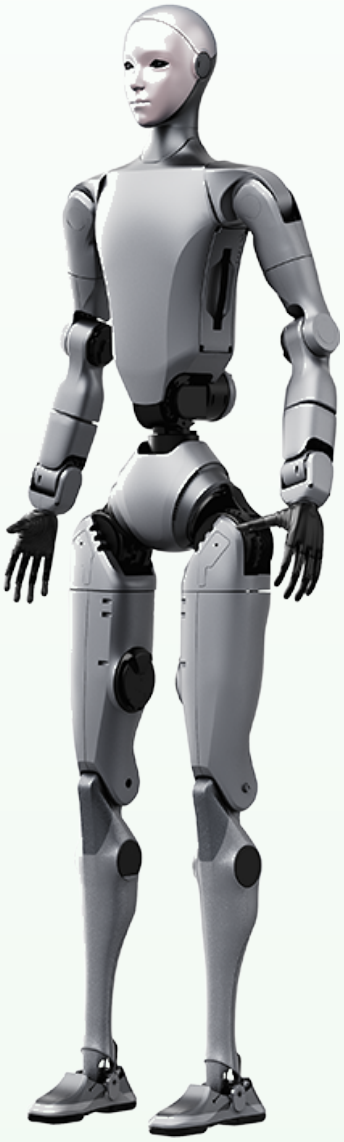
MYBOT SHOP

HUMANOID ROBOT

Unitree H2

User Manual



Unitree H2 · Biped · Model 2024
Webserver v9.8.4 · ROS 2 Humble

CONTACT
support@mybotshop.de
MYBOTSHOP GmbH

Contents

1 Attention 4

2 Disclaimer 5

3 About the H2 6
- 3.1 Key capabilities 6
- 3.2 Specifications 6

4 Startup & Shutdown Procedure 8
- 4.1 Pre-start Checks 8
- 4.2 Startup 8
- 4.3 Abnormal Situations (No Emergency Stop) 9
- 4.4 Shutdown 9
- 4.5 Quick Reference 10

5 Quick Start 11
- 5.1 The computers in the robot 11
- 5.2 Connecting your PC to the robot 11
- 5.3 Opening the Web Cockpit 12
- 5.4 Signing In 12
- 5.5 SSH access 12
- 5.6 Network Table 13

6 Dashboard 14
- 6.1 Health strip 14
- 6.2 Information cards & 3D model 14

7 The Cockpit 15
- 7.1 Robot Modules 15
- 7.2 Camera & 3D Views 16

8 Robot States (FSM) 17
- 8.1 Step-by-step: from Damp to driving 17
- 8.2 Driving the Robot 17
- 8.3 Upper-body (arm) control 18

9 Manipulation 19
- 9.1 Joint Mode 19
- 9.2 Cartesian Mode 19
- 9.3 Hands & Head 20
- 9.4 Transfer, Collision & Home 20
- 9.5 Missions 20

10 VR & Exoskeleton Teleoperation 22
- 10.1 Quest Teleop (WebXR) 22
- 10.2 Exoskeleton Leader-Arm 23

11 Remote Screen (VNC) 24

12 Navigation 25

13 SDK Development & Network Setup 26
- 13.1 Requirements 26
- 13.2 Install, build and the Python interface 26
- 13.3 Network configuration 27
- 13.4 Control interfaces 27

14 ROS 2 Package Reference 28
14.1 H2-Specific Packages 28
14.2 Shared Robot Packages 29
14.3 Building the Workspace 29
15 Fresh Installation 30
15.1 Prepare the workspace 30
15.2 Sync the source tree 30
15.3 Install dependencies and build 30
15.4 Install the autostart services 30
16 Safety Guidelines 31
16.1 Work-area safety 31
16.2 Electrical safety 31
16.3 Locomotion safety 31
16.4 Manipulation safety 31
16.5 Emergency response 32
16.6 Data security & privacy 32
16.7 Human-interaction safety 32
16.8 Residual risks 32
17 Getting Help & RMA 33
17.1 Raising an RMA case 33
17.2 Help resources 33
18 FAQ 34

1 | Attention

Read this document carefully. In case of doubt, it is essential to consult specialists, experts, or the manufacturers of the assemblies used. The robot must not be put into operation before clarification.

Contact information is available in the Getting Help & RMA chapter at the end of this document. The Unitree H2 is a full-size bipedal humanoid: it can fall, step, and strike or grasp with its arms. Every control, physical or in the web cockpit (see Chapter 5 for how to open it), commands real motion.

Important: Do not allow persons who are not familiar with robotics or with these instructions to operate the robot. Robots are dangerous in the hands of untrained users. The H2 has no emergency stop: always keep it secured on a crane or support frame whenever possible and know how to damp it and remove power (see Chapter 4).



2 | Disclaimer

The provided robot, supplied by MYBOTSHOP GmbH, is an R&D device. The H2 is controlled through Unitree's CycloneDDS-based stack (`unitree_sdk2` in C++ or Python), which runs with or without ROS 2; the MYBOTSHOP web cockpit documented here (see Chapter 5 for how to open it) layers on top of that stack. A working knowledge of ROS 2 (or the Unitree SDK), C++ or Python, and basic Linux networking is the practical prerequisite for this manual.

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

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



3 | About the H2

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 dual-computer onboard platform, it targets research, embodied-AI development, human-robot interaction, and advanced demonstration.

3.1 | Key capabilities

- **Expressive whole-body motion:** 31 DOF (6-DOF legs, 7-DOF arms, 3-DOF waist, 2-DOF head) with joint torque up to 360 N·m for walking, balancing, gesturing and dynamic actions.
- **Onboard compute:** PC1 (Intel Core i5, Unitree motion control) and PC2 (Intel Core i7, free for your own code), plus optional AI modules (PC3/PC4; up to 2070 TOPS with the optional Jetson AGX Thor), a binocular vision system, and joint-level sensing.
- **Payload and endurance:** arm payload of about 7 kg rated (about 15 kg peak) and about 3 hours of runtime.
- **Open control:** low-latency interfaces over `unitree_sdk2` (C++) and `unitree_sdk2_python`, plus the MYBOTSHOP web cockpit (Chapters 5–11).

3.2 | Specifications

3.2.1 Physical & mechanical

Parameter	Specification
Height (standing)	~182 cm (1820 × 456 × 218 mm)
Weight (with battery)	~70 kg
Total degrees of freedom	31
DOF per leg / per arm	6 / 7
Waist / head DOF	3 / 2
Peak joint torque	Up to 360 N·m (legs), 120 N·m (arms)
Joint actuators	Low-inertia, high-speed internal-rotor PMSM, industrial crossed-roller bearings



3.2.2 Performance, power & sensing

Parameter	Specification
Walking speed	Under 2 m/s
Arm payload (rated / peak)	~7 kg / ~15 kg (Unitree specification)
Battery	Lithium battery, **not hot-swappable** (power the robot down before changing it)
Runtime	~3 hours (configuration dependent)
Onboard computers	PC1 Intel Core i5 (Unitree motion control), PC2 Intel Core i7 (development); optional PC3 / PC4 AI modules, up to Jetson AGX Thor (2070 TOPS)
Vision	Humanoid binocular camera (wide FOV); software support not yet available, see Chapter 7
Proprioception	IMU + joint-level (dual-encoder) sensing
Tactile (optional)	Via dexterous hands

Specification source: values are drawn from Unitree's published material and may vary by model, firmware, configuration and operating conditions. Always cross-reference the official Unitree H2 Developer Guide before development or deployment.

Safety notice: the H2 is a full-size, high-power humanoid capable of forceful, dynamic motion, and it **cannot actively maintain balance until it has entered motion-control mode**. Keep it supported during startup and posture transitions and keep all personnel clear.

4 | Startup & Shutdown Procedure

The H2's physical power is provided by its battery; the web cockpit (see Chapter 5 for how to open it) does not gate power. Follow the sequence below before launching, or after stopping, the on-board software. Exact controller key combinations depend on the firmware shipped with your unit, so also consult the controller reference and the official Unitree H2 documentation for the current key map.

4.1 | Pre-start Checks

> **Warning:** the H2 cannot actively maintain balance until it has entered motion-control mode. Keep it supported and keep all personnel clear during startup and every posture transition.

1. Suspend the H2 from a crane or securely support it on a suitable support frame, with its front facing forward.
2. Clear the surrounding area of people and obstacles.
3. Arrange the arms and legs in a natural, untangled posture.
4. Make sure everyone present knows that there is no emergency stop and how to damp the robot and remove power (see Shutdown below).

4.2 | Startup

4.2.1 Install the battery

Insert the battery into its compartment until the latch engages with a clear **click**. Confirm the pack is fully seated and secured. Do not force it: if it does not seat smoothly, remove it and check the orientation. The battery is not hot-swappable: the robot must be powered down whenever the battery is installed or removed.

4.2.2 Power on

1. Briefly press the battery power button once.
2. Press and hold the button for more than **2 seconds** to turn the battery on.
3. Wait approximately **one minute** for the robot to initialise and enter its zero-torque / damping state. Keep the area unobstructed during this process.

4.2.3 Stand and operate

Using the remote controller (or the web cockpit, see Chapter 5), bring the H2 into its standing-preparation posture while firmly supporting it. Lower it until the feet make stable contact with the ground, then switch into motion-control mode. Verify the robot is balancing stably before releasing support or commanding motion. From the cockpit this corresponds to the state sequence Damp → Ready → Walk described step by step in Chapter 8.

> **Note:** once the on-board software is up, confirm it from your PC by opening the web cockpit (see Chapter 5); the Dashboard should show the robot `Online` with the expected Robot Modules `Active`.

4.3 | Abnormal Situations (No Emergency Stop)

Caution: the H2 has **no emergency stop**. Plan for abnormal behaviour before every session: keep the robot on a crane whenever possible, keep the area clear, and know how to damp it and remove power.

If the H2 enters an abnormal or unsafe state, switch it to Damp with the remote controller or from the web cockpit (see Chapter 8) and remove power as described in the Shutdown section below. In Damp the joints go soft: unless the robot hangs on a crane it will lower or collapse to the ground, so keep everybody clear. The cockpit also provides a software E-STOP / E-RELEASE (see Chapter 8) that halts commanded motion; it is a convenience, not a safety function.

4.4 | Shutdown

There are two ways to shut the H2 down, depending on whether a crane is available:

4.4.1 A. Robot hanging on a crane

1. Make sure the robot is hanging securely on the crane and the area is clear.
2. Remove power: briefly press the battery power button, then press and hold for more than 2 seconds.

4.4.2 B. No crane available

1. Put the robot into Ready mode (see Chapter 8).
2. Lay it down on the floor in an open, level area with adequate clearance.
3. Remove power: briefly press the battery power button, then press and hold for more than 2 seconds.

If the robot will be unused for an extended period, remove the battery pack (with the robot powered down) for safe storage.

Important: keep hands and clothing clear of the joints during all posture transitions to avoid pinch injuries. Handle the battery per lithium-battery safety guidelines.

4.5 | Quick Reference

Action	Control	Notes
Power on	Battery button: press once, then hold >2 s	Robot is supported; wait ~1 min for init
Enter motion control	Controller / cockpit (Damp → Ready → Walk)	Feet must touch the ground (crane or standing); see Chapter 8
Relax joints	Damp (controller / cockpit)	Robot collapses unless hanging on a crane
Shutdown, with crane	Remove power at the battery button	Robot hanging on the crane
Shutdown, no crane	Ready mode, lay on the floor, then remove power	Open, level area
Software stop	Cockpit E-STOP	Halts commanded motion only; not a safety function



5 | Quick Start

Important: complete the startup procedure (Chapter 4) before continuing. The robot must be supported (crane or frame) and everybody must know how to damp it and remove power.

5.1 | The computers in the robot

The H2 carries several on-board computers on its internal network:

- **PC1** (192.168.123.161): Unitree's **closed-source** proprietary motion controller lives here. It is not meant to be modified; PC2 communicates with it over DDS.
- **PC2** (192.168.123.162 internally, 192.168.124.162 on the operator link): completely **open for your own code**. The MYBOTSHOP web cockpit and its ROS 2 nodes run on PC2.
- **PC3 and PC4** (optional AI modules, depending on your configuration): also completely open for your own code.

5.2 | Connecting your PC to the robot

Connect your own PC with an Ethernet cable to the **left LAN port** on the back of the H2 (the two Gigabit Ethernet ports are marked [12] and [15] in the rear-panel schematic below, taken from the Unitree H2 documentation; the original is at `https://support.unitree.com/home/en/H2_developer`).

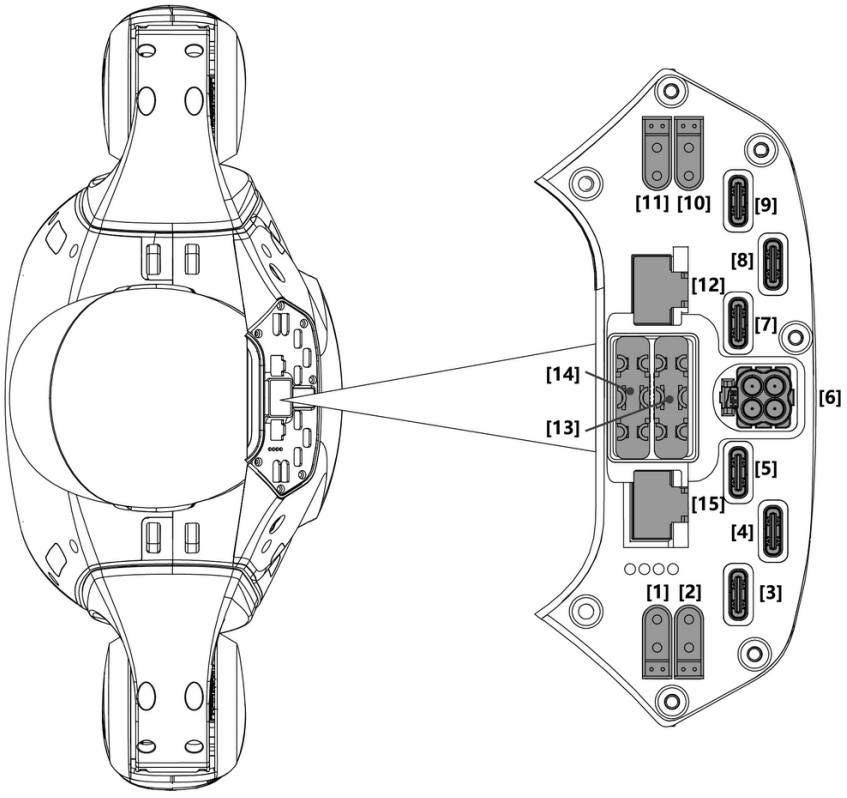


Figure 5.1. H2 rear panel connectors [1]–[15]; the Ethernet ports are [12] and [15]. Source: Unitree H2 developer documentation.

Then set a manual (static) IPv4 configuration on your PC's network adapter:

Setting	Value
IP address	`192.168.124.150`
Netmask	`255.255.255.0`

5.3 | Opening the Web Cockpit

The cockpit is served from PC2. Open one of the following in a modern browser (Chrome or Edge):

Address	Use	Notes
`https://192.168.124.162`	Recommended	Full features incl. 3D model and Quest teleop
`http://192.168.124.162:9000`	Direct / fallback	Cockpit + REST only; VR teleop not available

Why HTTPS matters: HTTPS is needed if the Meta Quest must access the browser, otherwise the connection gets rejected. The 3D robot model and Quest teleop use secure WebSockets; a browser will not open an insecure `ws://` connection from a secure page, so these only work over `https://` (terminated by nginx), not over the plain `:9000` address. The self-signed certificate triggers a one-time "Not secure" warning on the local robot network; proceed to the site.

5.4 | Signing In

Sign in with the cockpit credentials:

Field	Default
User	`admin`
Password	`mybotshop`

Change the default password before the robot joins any shared network, the cockpit can command motion, so treat access as privileged.

5.5 | SSH access

You can log in to PC2 over SSH:

```
ssh unitree@192.168.124.162
```

Default password: `Unitree0408` (on robots with early firmware versions, try `Unitree#24226`).

Note: the ROS 2 stack uses CycloneDDS on `ROS_DOMAIN_ID=0` (`RMW_IMPLEMENTATION=rmw_cyclonedds_cpp`), pinned to the robot's wired interface.

5.6 | Network Table

Endpoint	Device	Login	Notes
192.168.123.161	PC1, Unitree motion controller	–	Closed source; PC2 talks to it over DDS
192.168.123.162	PC2, development computer	unitree	Internal robot network
192.168.124.162	PC2, operator link	unitree	SSH; password Unitree0408 (early firmware: Unitree#24226)
...124.162 (https) / : 9000	Web cockpit	admin	Default password mybotshop
192.168.124.150	Your PC	–	Static, netmask 255.255.255.0

6 | Dashboard

The Dashboard is the cockpit's landing page. It summarises the robot's identity, software and network, shows live system health, and renders a real-time 3D model with every joint angle.

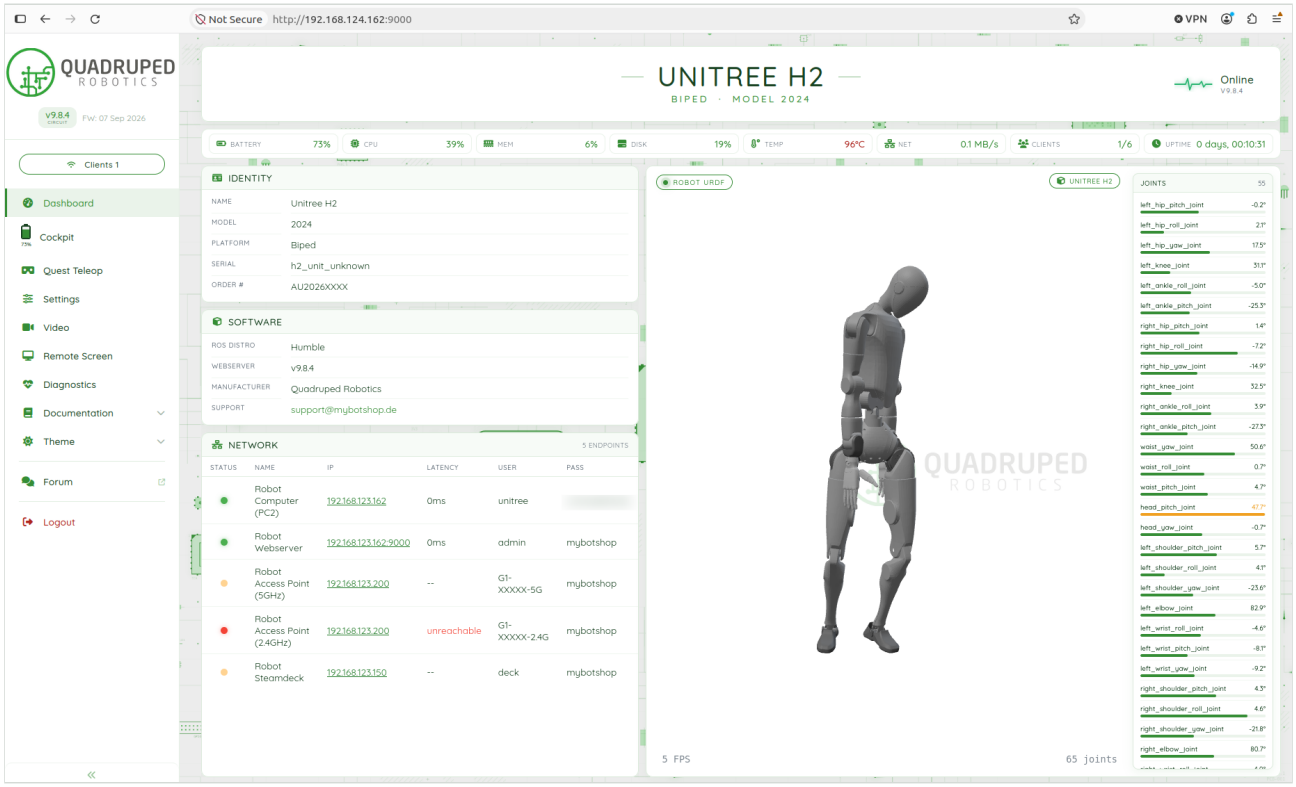


Figure 6.1. The Dashboard: health strip, Identity/Software/Network cards, live URDF viewer and joint list.

6.1 | Health strip

The bar across the top reports, left to right: Battery, CPU, Memory, Disk, Temperature, Network throughput, connected Clients, and Uptime. Values turn amber/red near their limits.

> **Temperature:** investigate sustained high readings before long operation, overheating can throttle or shut down the on-board computer. Keep vents clear.

6.2 | Information cards & 3D model

- **Identity:** name, model, platform, serial, order number.
- **Software:** ROS distribution, webserver version, manufacturer, support contact.
- **Network:** on-board computers, access points and operator devices, with status, IP and latency.

The centre panel renders the robot's live pose from its URDF; the right-hand Joints list reports each joint angle in degrees and updates in real time. Confirm the model matches the robot's real posture before commanding motion.

7 | The Cockpit

The Cockpit (pilot page) is the main operating screen: watch the robot, toggle robot modules, change states (Chapter 8), drive (Chapter 8) and control the arms (Chapter 9).

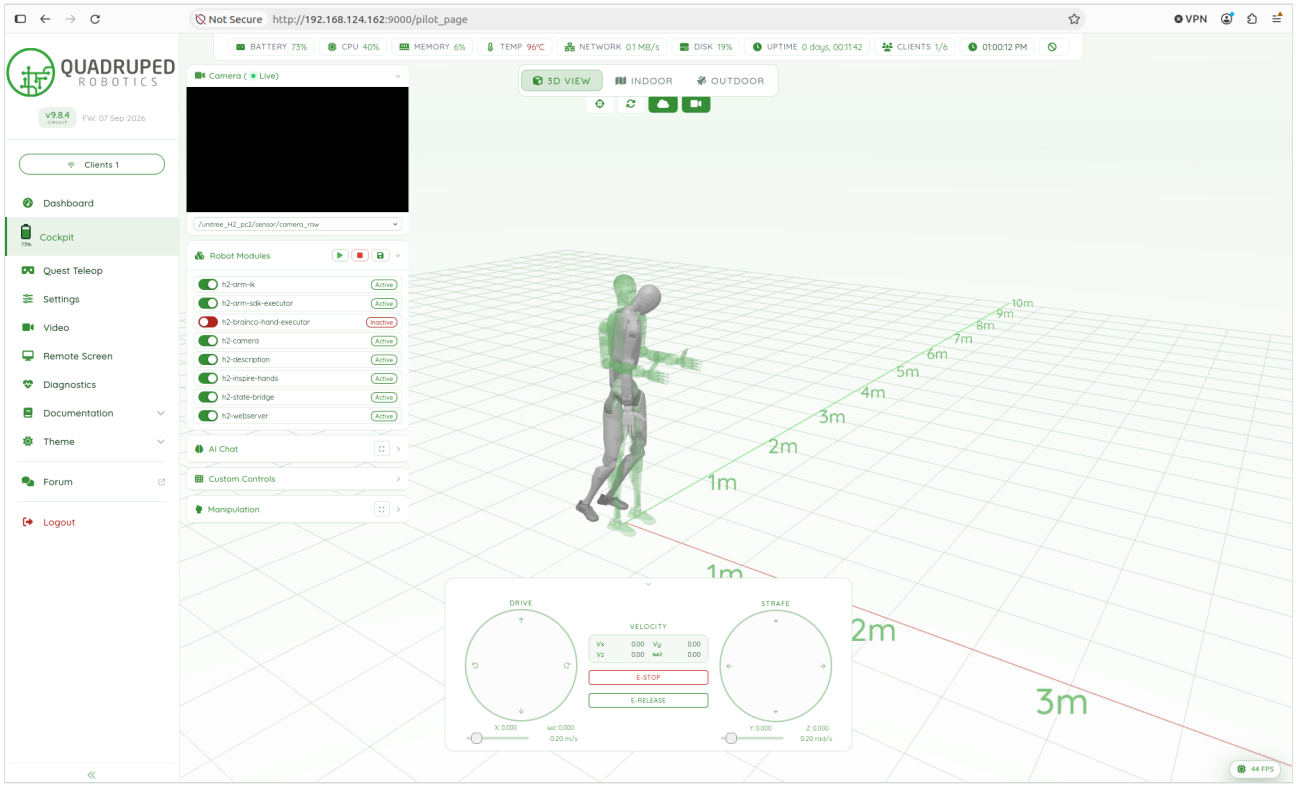


Figure 7.1. The Cockpit: camera panel (left, black while no camera software is available), 3D view (centre), drive controls (bottom).

The left column stacks collapsible panels, Camera, Robot Modules, AI Chat, Custom Controls and Manipulation. The centre shows the 3D scene. The drive controls dock at the bottom.

7.1 | Robot Modules

The Robot Modules panel lists the robot's ROS 2 nodes, each with a live state badge and an on/off switch, so you can start or stop a subsystem without a terminal.

Module	Function
`h2-webserver`	The cockpit itself
`h2-state-bridge`	Robot telemetry / state feedback
`h2-description`	URDF robot description for the 3D viewer
`h2-camera`	Camera stream (not yet available, waiting on Unitree for a software update)
`h2-arm-ik`	Arm inverse-kinematics / target streaming
`h2-arm-sdk-executor`	Sends arm commands to the robot (arm_sdk)
`h2-inspire-hands`	Inspire dexterous hands driver
`h2-brainco-hand-executor`	BrainCo hand driver (alternative hand)

A badge reads Active or Inactive. The play / stop / save controls at the top start, stop or persist the whole module set.

> **Note:** arm and hand teleoperation need `h2-arm-ik` and `h2-arm-sdk-executor` active, plus the matching hand driver (`h2-inspire-hands` or `h2-brainco-hand-executor`).

7.2 | Camera & 3D Views

> **Camera not yet available:** the H2's camera is not yet usable, we are waiting on Unitree for a software update. The Camera panel therefore shows a black frame.

Once available, the Camera panel will show the selected video topic live. The centre scene offers the 3D View (the robot on a measured grid); the solid model is the current pose and the translucent green overlay previews the commanded pose. The Indoor / Outdoor map tabs belong to navigation, which the H2 does not support at the moment (see Chapter 12).



8 | Robot States (FSM)

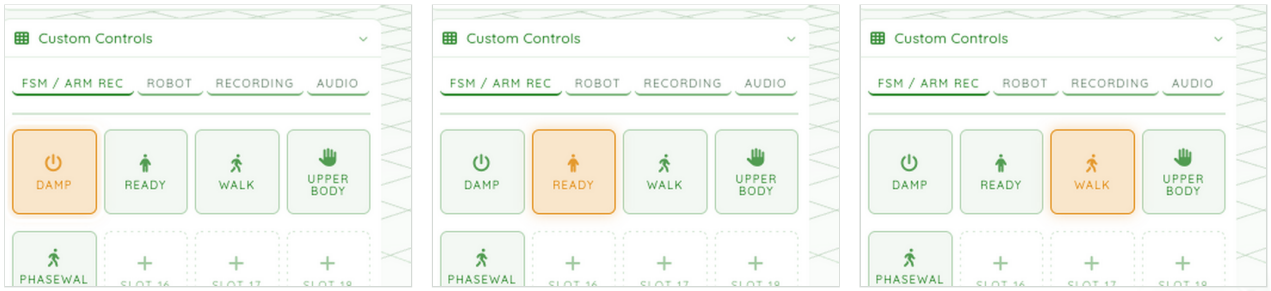
The robot runs a state machine (FSM). The robot can only walk while it is in the Walk state, so understand the states first. Switch states from the Custom Controls panel on the Cockpit, under the **FSM / ARM REC** tab. The active state is highlighted in orange.

State	What it does	When to use
Damp	Relaxes joints to a soft, compliant hold	Start state; safe parking. Robot must hang on a crane or lie on the floor
Ready	Stands and holds position (balanced)	Starting point before walking
Walk	Enables the walking controller	Driving with the joysticks
PhaseWalk	Phase-based walking gait	Required for upper-body (arm) control
Upper Body	Enables upper-body / arm control	Only works in PhaseWalk
Slot 16–18	User-assignable preset slots	Custom states / recorded poses

Warning: switching to Damp while the robot stands unsupported on the ground will let it collapse. Only damp the robot when it hangs on a crane or lies on the floor.

8.1 | Step-by-step: from Damp to driving

1. **Damp.** The robot starts in Damp mode. It must hang on a crane or lie on the floor.
2. **Ready.** Switch to Ready. Now lift the robot up until it is standing; if it is on a crane, lower it until its feet touch the ground.
3. **Walk.** Only now switch to Walk. The joysticks work from this point on.



1 · Damp
On the crane or on the floor

2 · Ready
Lift to stand / feet on the ground

3 · Walk
Joysticks now work

Figures 8.1–8.3. Custom Controls, FSM / ARM REC tab, with the active state highlighted.

8.2 | Driving the Robot

Two circular joysticks dock at the bottom of the Cockpit:

- **Drive:** push up/down to walk forward/back; left/right to turn.
- **Strafe:** push left/right to side-step; the second axis trims vertical motion.

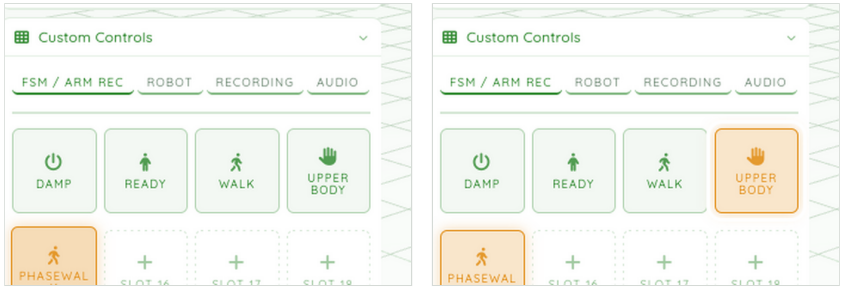
The Velocity readout shows the commanded Vx, Vy, Vz and ωz. The sliders under each stick cap the linear speed (m/s) and turn rate (rad/s), start low.

The robot only drives in Walk. Velocity commands move the robot only in the Walk (or PhaseWalk) state. In other states the sticks do not move the legs.

E-STOP / E-RELEASE: E-STOP halts commanded motion immediately; E-RELEASE clears it. These are software controls only. The H2 has no hardware emergency stop and the robot does not hold itself up through this button.

8.3 | Upper-body (arm) control

Upper-body control only works in **PhaseWalk**. If the robot is in Ready or Walk, first switch to PhaseWalk, then activate Upper Body.



4 · PhaseWalk
From Ready or Walk

5 · Upper Body
Activate while in PhaseWalk

Figures 8.4–8.5. PhaseWalk selected, then Upper Body activated.

The remaining tabs, Robot, Recording and Audio, hold robot-specific actions, motion-recording shortcuts, and audio/voice playback. State requests are issued by the cockpit to the robot's command service (the `h2_control` stack), which republishes them to the Unitree controller.

Typical sequence: Damp → Ready (stand / feet on the ground) → Walk → drive. For arm work: PhaseWalk → Upper Body.

9 | Manipulation

The Manipulation panel controls the arms, hands, head and waist in two modes, Joint and Cartesian, plus mission recording. Motion previews on the 3D model first and reaches the real robot only when you arm it (Transfer to real robot, below).

> **Prerequisite:** arm control only works with the robot in PhaseWalk and Upper Body activated (see Chapter 8).

9.1 | Joint Mode

In Joint mode you set each joint directly. Choose Left Arm, Right Arm or Waist, then drag the sliders, shoulder (×3), elbow, and wrist roll/pitch/yaw, reading the live angle beside each.

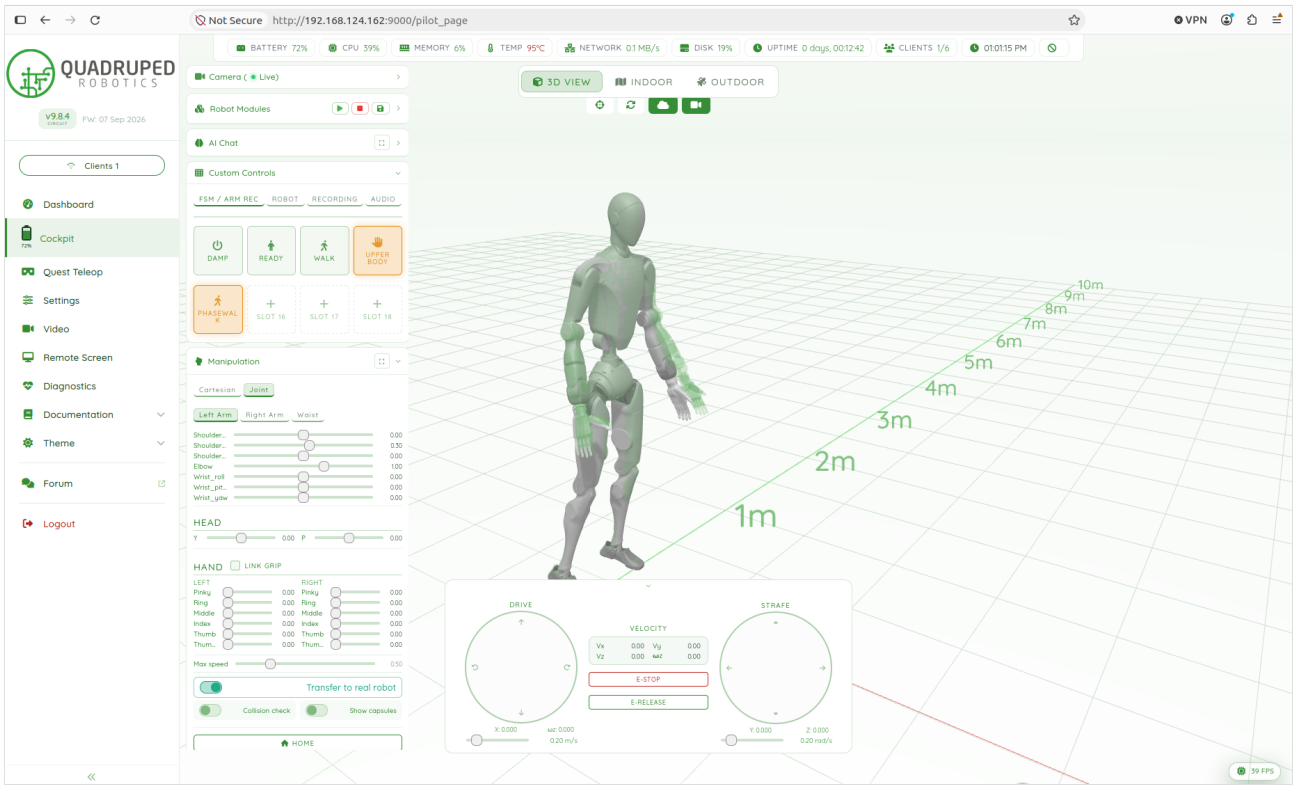


Figure 9.1. Joint mode: per-joint sliders for the selected arm, plus head and hand controls.

9.2 | Cartesian Mode

In Cartesian mode you move the hand to a point in space and the cockpit solves the joint angles (inverse kinematics). Choose Dual Arm, Left Arm or Right Arm, then set position (X Y Z) and orientation (R P Y). The blue hand markers in the 3D view are the end-effector targets.

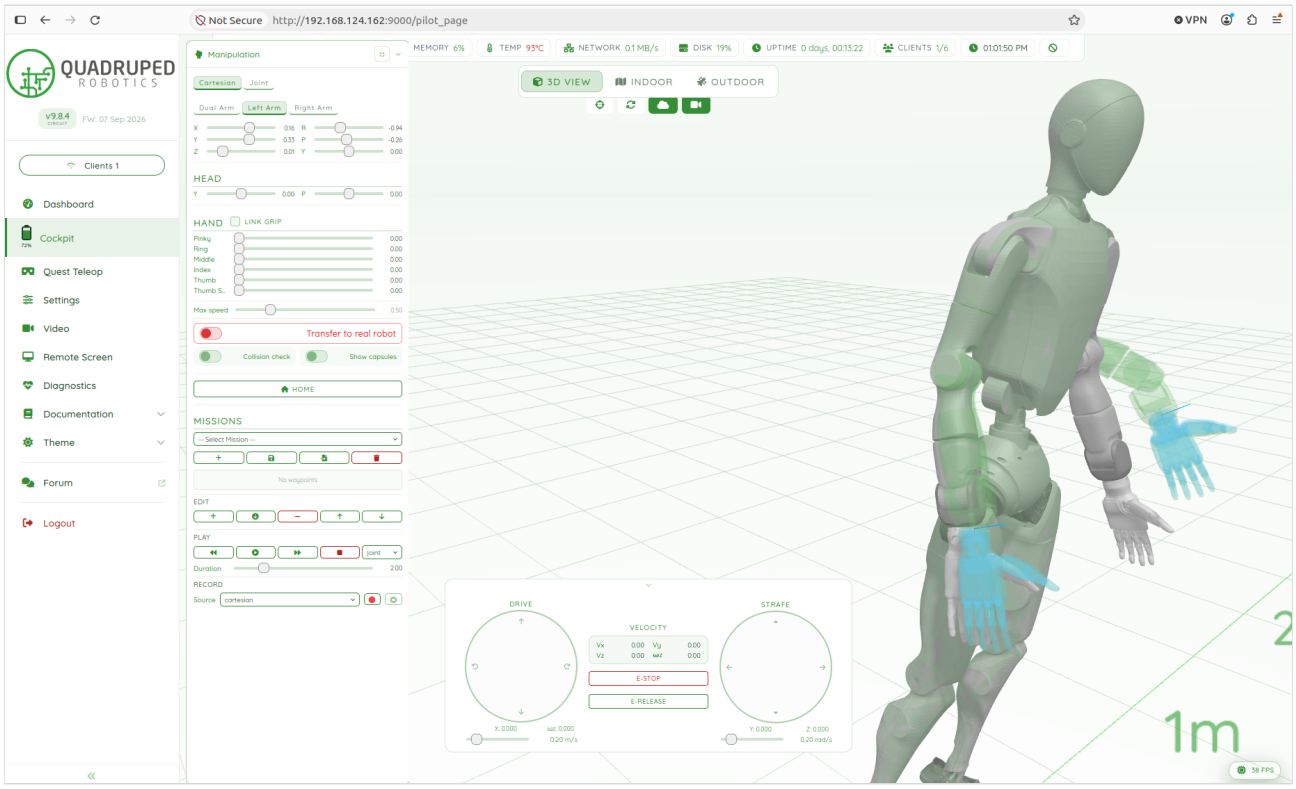


Figure 9.2. Cartesian mode: X/Y/Z + roll/pitch/yaw targets; the blue hands mark the IK goals.

9.3 | Hands & Head

The Head sliders set yaw (Y) and pitch (P). The Hand section gives a slider per finger, pinky, ring, middle, index, thumb and thumb opposition, for the left and right hands (Inspire or BrainCo). Tick Link Grip to drive all fingers together as a single open/close grip. Max speed caps how fast the hands move.

9.4 | Transfer, Collision & Home

- **Transfer to real robot:** the master safety gate. *Off:* poses preview on the model only. *On:* the arms move on the real robot (turning it on auto-enables the arm command path, `arm_sdk`).
- **Collision check / Show capsules:** enable self-collision checking and visualise the collision capsules on the model.
- **Home:** smoothly returns the arms to the home pose (a gentle glide, not a jump).

Arm the transfer deliberately. With Transfer to real robot on, slider and Cartesian changes move the physical arms immediately. Keep clear of the arms, start slow, and keep the robot secured on the crane or support frame.

9.5 | Missions

The Missions section records and replays arm motions as sequences of waypoints.

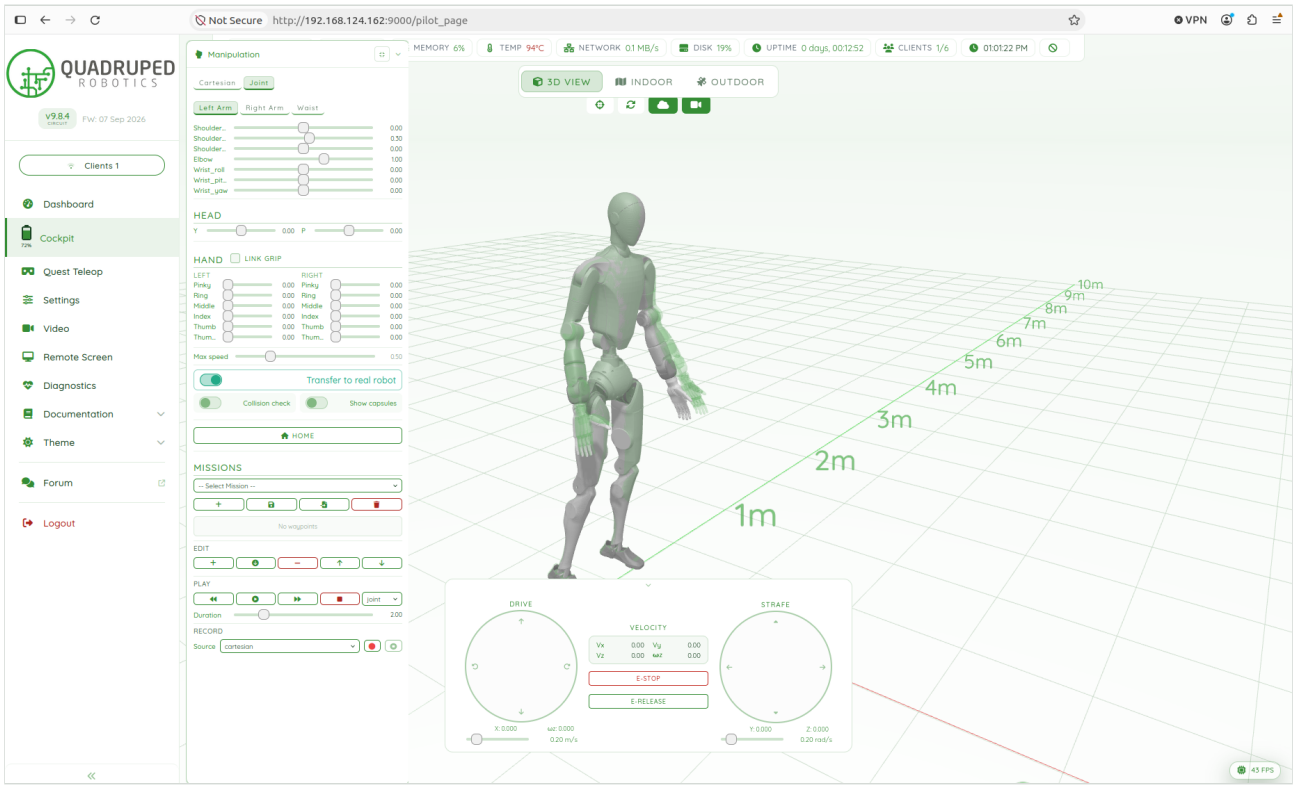


Figure 9.3. Missions: build a waypoint list, then play it back at a set duration in joint or Cartesian space.

1. Select or create a mission from the Select Mission dropdown (add / save / load / delete).
2. Pose the arms, then use Edit (add, insert, remove, reorder) to capture waypoints.
3. Choose the Record source (joint or Cartesian) to capture live motion.
4. Set Duration and use the Play controls (rewind, play, fast-forward, stop) to replay.

Caution: playback moves the real arms when Transfer to real robot is on. Rehearse new missions with transfer off first, and keep the workspace clear during playback.

10 | VR & Exoskeleton Teleoperation

Beyond the on-screen controls, the H2 arms and hands can be teleoperated two ways: from a Meta Quest headset (WebXR) and from a physical exoskeleton leader-arm. Both are independent input paths; only one should be operated at a time, and both honour the cockpit's Transfer to real robot gate.

10.1 | Quest Teleop (WebXR)

The Quest Teleop page maps the headset's controllers to the robot's hands.

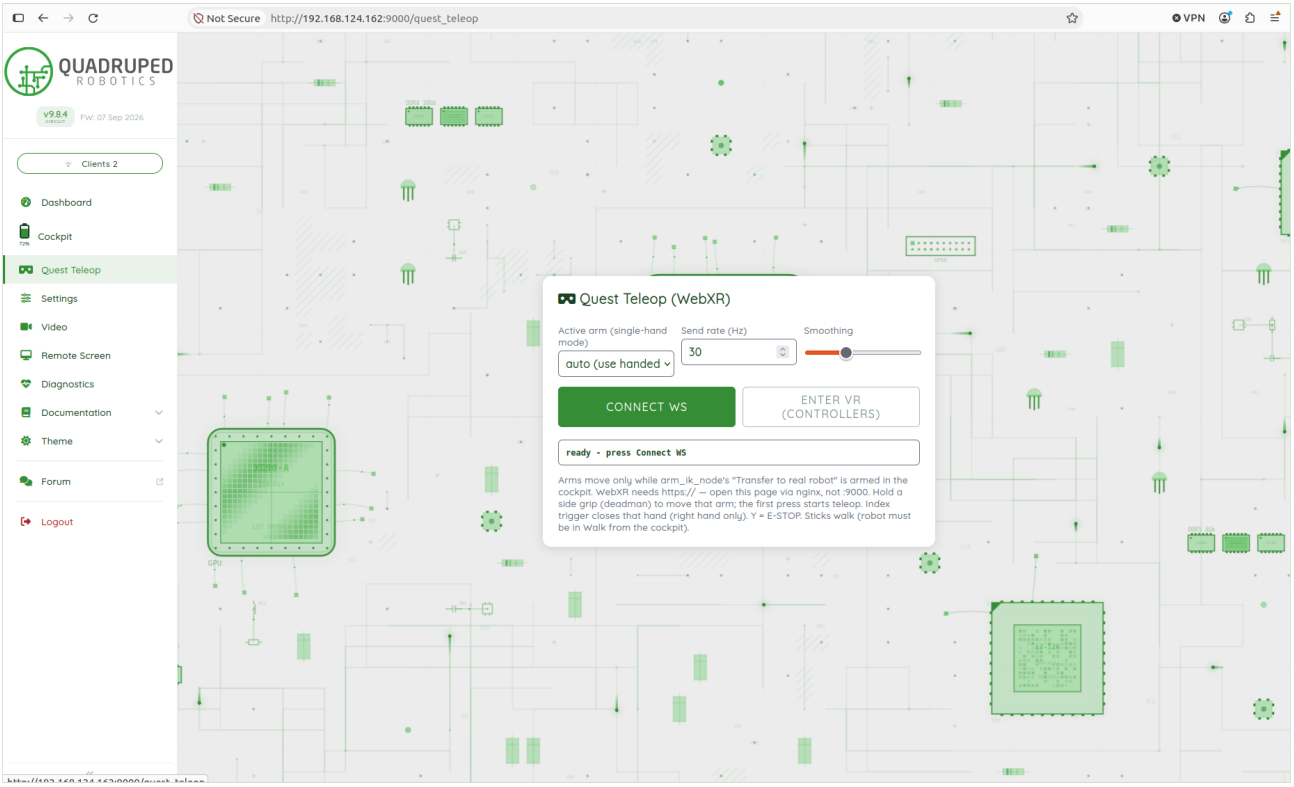


Figure 10.1. Quest Teleop: choose the active arm, connect, then enter VR.

1. Open the page over HTTPS (`https://192.168.124.162/quest_teleop`) in the headset's browser, WebXR needs a secure context, so the plain :9000 address will not work.
2. Choose the Active arm (auto / left / right), a Send rate (Hz) and a Smoothing level.
3. Press Connect WS; the status should read connected.
4. Press Enter VR (Controllers) and put on the headset.

Input	Action
Side grip (deadman)	Hold to move that arm; first press starts teleop
Index trigger	Close that hand (right hand)
Thumbsticks	Walk the robot (must be in Walk)
Y button	Software E-STOP (halts commanded motion only)

Arm the cockpit first. Arms move in VR only while Transfer to real robot is armed, the robot is in PhaseWalk with Upper Body active (Chapter 8), and the legs walk only in Walk. Release the deadman grip to stop the arm instantly.

10.2 | Exoskeleton Leader-Arm

The optional Foxtec exoskeleton leader-arm drives both H2 arms in joint mode from your PC. It runs from the separate `foxtec_leaderarm` repository and connects directly to the arm-IK WebSocket on PC2:

```
cd ~/foxtec_leaderarm/exo_teleop_h2
H2_HOST=192.168.124.162 ./run_h2.sh
```

On launch it checks the exo serial port, calibration, and the robot services, then streams both arms in joint mode. The deadman is button C and go-home is button D. As with the Quest, the arms move on the robot only while Transfer to real robot is armed.

More resources: further material for the H2, in particular the bash script for the Foxtec exoskeleton arms, is available at **https://docs.mybotshop.de/h2.html**.

Warning: operate the exoskeleton supervised, with the robot secured. Start with the robot in PhaseWalk with Upper Body active, confirm the arm tracks 1:1 with no jump on the first grip, and keep the workspace clear.

11 | Remote Screen (VNC)

The Remote Screen page embeds PC2's desktop in the browser over a secure connection, no SSH client required.

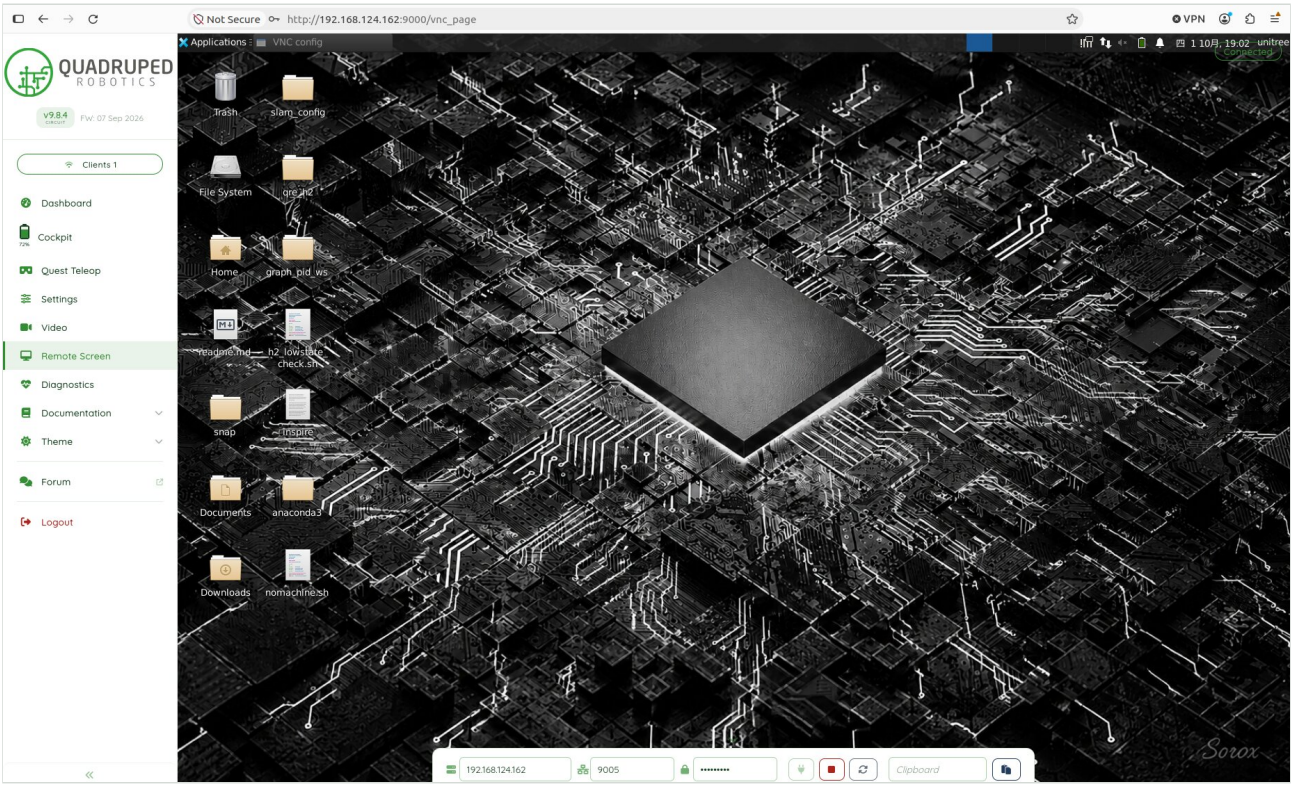


Figure 11.1. Remote Screen: the robot computer's desktop, embedded in the cockpit.

1. Open Remote Screen from the left navigation.
2. The bottom bar holds the host, port (`9005`) and password; the defaults connect to the robot desktop.
3. Use connect / disconnect / refresh, and the clipboard box to paste text into the session.

Note: Remote Screen is served as a secure WebSocket (`wss://<robot-ip>/vncws` , bridged to VNC on display :1 / port 5901) and therefore needs the HTTPS address. Use it for maintenance; avoid heavy desktop apps while teleoperating.

12 | Navigation

Not supported: the H2 does not support navigation at the moment, since no LiDAR is mounted. Mapping (SLAM), map navigation and obstacle avoidance are therefore not available.



13 | SDK Development & Network Setup

Below the MYBOTSHOP web cockpit, the H2 is controlled through Unitree's CycloneDDS-based communication stack, which runs with or without ROS 2. This chapter summarises developing directly against the SDK; always treat the official Unitree H2 Developer Guide as the authoritative reference for your firmware. PC1 (Unitree motion controller) is closed source; your own code runs on PC2 (or PC3 / PC4) and talks to PC1 over DDS.

SDK	Use
`unitree_sdk2`	C++ SDK for high- and low-level control, state subscription and sensor access.
`unitree_sdk2_python`	Python interface for rapid prototyping.
`unitree_ros2`	Unitree's ROS 2 bridge to the same DDS interface. Unitree's README lists Go2, B2 and H1; check Unitree's H2 documentation before relying on it for the H2.

13.1 | Requirements

Item	Requirement (per Unitree's SDK README)
Operating system	Ubuntu 20.04 LTS (documented by Unitree; PC2 runs ROS 2 Humble)
Compiler	GCC 9.4.0
Build system	CMake 3.10+
Architecture	x86_64 or aarch64
Transport	DDS (CycloneDDS)
Python (optional)	Python 3.8+, `cyclonedds==0.10.2`, numpy, opencv-python

13.2 | Install, build and the Python interface

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

git clone https://github.com/unitreerobotics/unitree_sdk2.git
cd unitree_sdk2 && mkdir build && cd build
cmake .. -DCMAKE_INSTALL_PREFIX=/opt/unitree_robotics
make && sudo make install            # system-wide install (optional)

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

If the Python install fails on `cyclonedds`, compile CycloneDDS separately and set `CYCLONEDDS_HOME` before retrying.

13.3 | Network configuration

Unitree robots use the 192.168.123.x subnet by convention; the H2's rear panel has two independent Gigabit Ethernet segments (.123.x and .124.x). Two ways to run SDK code:

- **On the robot (recommended):** log in to PC2 (see Chapter 5) and run your code there. Pass the name of the network interface on PC2 that carries the 192.168.123.x network (find it with `ip -br a`).
- **From an external PC:** Unitree's convention is a static address in the robot's subnet, for example `192.168.123.99` with netmask `255.255.255.0`, on the interface connected to that segment. Do not reuse the robot's own addresses (.161, .162). The operator link used for the cockpit is the separate .124.x segment; check the Unitree H2 Developer Guide for the DDS interface on your firmware.

```
ip -br a                                  # find the NIC connected to the H2
ping 192.168.123.161                      # confirm reachability (PC1)
```

SDK examples must be told which interface reaches the robot; the interface name is passed as an argument (and to the DDS channel factory):

```
./example_program eth0      # C++:    ChannelFactory::Instance()->Init(0, argv[1])
python3 example.py eth0     # Python: ChannelFactoryInitialize(0, sys.argv[1])
```

13.4 | Control interfaces

- 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, damping). It bypasses the built-in balancing safeguards and must stay within the documented joint limits.

Warning: custom low-level code can command the robot into unsafe states. Test new control software with the H2 hanging securely on a crane, keep the area clear, and validate joint commands against the documented limits before deployment.

Further resources: the official H2 Developer Guide, `unitree_sdk2` and `unitree_sdk2_python` on the Unitree Robotics GitHub, and the Unitree Document Center. Additional MYBOTSHOP resources for the H2 are available at https://docs.mybotshop.de/h2.html.

14 | ROS 2 Package Reference

The H2 software stack lives under `/opt/mybotshop/src/mybotshop/`. Packages split into two groups: robot-specific packages prefixed `h2_` (bring-up, description, hardware bridge, control) and shared packages prefixed `robot_` (arm IK, autostart, interfaces, and the operator webserver) reused across MYBOTSHOP platforms.

Note: for the exact topics, services and parameters a node publishes, consult the `launch/` and `config/` folders inside each package at `/opt/mybotshop/src/mybotshop/<package>/`.

14.1 | H2-Specific Packages

Package	Purpose
`h2_description`	URDF, Xacro, meshes and joint definitions for the H2 humanoid; feeds the cockpit 3D viewer and TF via `robot_state_publisher`.
`h2_platform`	Hardware-platform bring-up for the H2, the bridge to the Unitree control stack (PC1): publishes joint states, IMU and odometry, and forwards commands.
`h2_control`	High-level locomotion / state command service. Translates cockpit state requests (Damp, Ready, Walk, PhaseWalk, ...) and velocity into the Unitree controller.
`h2_interface`	H2-specific message, service and action definitions shared across the H2 packages (build-time only; no nodes).
`h2_teleop`	Teleoperation inputs (gamepad / joystick) and command translation.

Not supported on the H2: the packages `h2_navigation`, `h2_lidar`, `h2_depth_camera`, `h2_obstacle_avoidance` and `robot_navigation` are in the source tree but are not usable at the moment (no LiDAR is mounted and the camera software is not yet available). `h2_viz` (RViz configuration) has not been tested on the H2.

Warning: `h2_platform` is the sole bridge to the Unitree control stack. Do not run a second instance in parallel, duplicate publishers fight for joint control and can cause unsafe motion.

14.2 | Shared Robot Packages

Package	Purpose
`robot_webserver`	The operator web cockpit (Flask + Waitress on 9000, nginx HTTPS, rosbridge, noVNC, and optional LLM / voice). Covered in Chapters 5–11.
`robot_arm_ik`	Arm inverse-kinematics (`arm_ik_node`), the arm-command executor (`h2_arm_sdk_executor` → `arm_sdk`), and the hand executors (Inspire / BrainCo). Handles joint and Cartesian targets, slew limiting, collision veto and the transfer gate.
`robot_autostart`	Installs and manages the systemd autostart services for the whole stack (via `robot_upstart`), setting the correct `ROS_DOMAIN_ID` , RMW and run-as user. Launch copies live in the job folders under `/etc/ros/humble/*.d/` .
`robot_interface`	Common message / service / action types shared across the generic `robot_*` stack.

14.2.1 Service status, logs and manual control

```
sudo systemctl status h2-webserver.service
sudo systemctl restart h2-arm-ik.service
journalctl -u h2-webserver.service -b    # since last boot
```

14.3 | Building the Workspace

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

To build only one package and its dependencies:

```
colcon build --symlink-install --packages-up-to robot_webserver
```

Note: `--symlink-install` lets you edit Python and launch files without rebuilding; C++ packages still need a rebuild. After changing a service's launch file, restart the matching systemd unit (e.g. `sudo systemctl restart h2-webserver.service`), a rebuild does not re-launch running services.



15 | Fresh Installation

This procedure restores the MYBOTSHOP H2 software stack from a clean state onto PC2.

> **Warning:** a fresh installation modifies system services, the apt sources, and the autostart configuration. Run only on a robot you are authorised to reconfigure.

15.1 | Prepare the workspace

```
sudo mkdir -p /opt/mybotshop
sudo chown -R unitree:unitree /opt/mybotshop
```

15.2 | Sync the source tree

Push the source from your PC to PC2:

```
rsync -avP --delete -e ssh src unitree@192.168.124.162:/opt/mybotshop/
```

15.3 | Install dependencies and build

```
cd /opt/mybotshop/src/mybotshop
./g1_install.sh                  # apt + Python + ROS 2 dependencies
cd robot_webserver && ./webserver_installer.sh   # nginx, TigerVNC, websockify, TLS,
requirements
cd /opt/mybotshop
colcon build --symlink-install && source install/setup.bash
```

15.4 | Install the autostart services

```
ros2 run robot_autostart startup_installer.py                  # all services
ros2 run robot_autostart startup_installer.py robot-webserver  # one service
```

Then open `https://192.168.124.162` to verify the cockpit is reachable.

> **Provisioning notes:** on a stock image, refresh the ROS 2 archive key and switch apt to official mirrors before installing, the shipped key can be expired. Ensure `xacro` and `robot_upstart` are present before building, and pin `openai<2` for the LLM chat tab.

16 | Safety Guidelines

The Unitree H2 is a bipedal humanoid that combines a walking platform with two articulated arms. The hazards of both apply simultaneously: the robot can fall, walk into people, and grasp or strike with the same arm motion.

Important: a humanoid is far less stable than a wheeled robot. Treat any state change, low battery, or Damp / power-off event as a potential fall and apply the support procedure in Chapter 4.

16.1 | Work-area safety

- Keep a clean, well-lit, uncluttered area; poor lighting and reflective surfaces degrade perception.
- Reserve a footing area free of cables, low obstacles and trip hazards.
- Do not operate in explosive atmospheres or near flammable liquids, gases or dust.
- Keep bystanders clear of the arm sweep envelope and the walking path.
- When using a gantry or crane, ensure the suspension point is rated for the robot's full weight plus a safety factor.

16.2 | Electrical safety

- Inspect power cables and connectors before power-on; replace damaged cable immediately.
- Protect the robot from rain, splashing and condensation, it is for indoor use.
- The battery is not hot-swappable: power the robot down before installing or removing it.
- Do not operate at low battery; voltage sag during standing or arm acceleration can trigger an unsafe joint release.

16.3 | Locomotion safety

- Define and enforce safety zones; use markings or barriers around non-operators.
- Start with low speed limits on the joysticks and increase gradually.
- Never relax the legs (Damp) while the robot is standing unsupported on the ground (Chapter 8).

16.4 | Manipulation safety

- Keep collision checking enabled and respect the joint limits; do not edit limits without review.
- Keep the arm workspace clear, anything in reach must be part of the task or a known obstacle.
- Reduce arm speed when teaching trajectories or working near people.
- Remember arm motion and balance are coupled: a sudden arm acceleration can disturb stance.

16.5 | Emergency response

- The H2 has no emergency stop. Whenever possible, operate with the robot hanging on a crane.
- Know in advance how to damp the robot (controller or cockpit) and how to remove power (Chapter 4). Damping or removing power disables joint holding torque, so support is mandatory because the robot will sag or collapse otherwise.
- The cockpit software E-STOP halts commanded motion only; do not rely on it as a safety function.
- Mark the crane support point and the power button at every workstation and brief every operator.
- Run regular drills for both fall (legs) and pinch/strike (arms) events.
- After any abnormal event, verify joint health before resuming motion.

16.6 | Data security & privacy

- Apply least-privilege networking; change the default SSH and cockpit (`admin` / `mybotshop`) credentials before joining any shared network.
- Keep DDS traffic on the robot's wired network.
- Audit and update apt sources, ROS 2 keys and dependencies as part of maintenance.
- Comply with privacy regulations when capturing camera, audio or map data.

16.7 | Human-interaction safety

- Communicate operational state to nearby people; establish who holds motion authority and what triggers a stop.
- Keep hands clear of the arm sweep and the foot path, legs can move quickly during balance recovery.
- Where collaboration is unavoidable, run at reduced velocity with a second operator ready to damp the robot.

16.8 | Residual risks

- Impaired sensing from dust, occlusion, glare or hardware failure.
- Loss of balance and falls, especially during state transitions, low battery, or after damping the robot.
- Collisions in crowded or dynamic environments.
- Pinch and strike hazards during dual-arm manipulation, especially when the second arm is treated as static.
- Cybersecurity vulnerabilities in the operator network or on-robot services.

Operators must complete the documented procedures before each session. Skipping the support checklist or the prescribed shutdown sequence is the most common cause of damage and injury.

17 | Getting Help & RMA

If you encounter a problem, follow the steps below to raise a Return Merchandise Authorization (RMA) case.

17.1 | Raising an RMA case

1. **Write an e-mail to support.** Send a detailed description of the problem and the troubleshooting already tried to **support@mybotshop.de**, and always mention your order number.
2. **Request the RMA.** If an RMA is needed, fill out the form at **https://www.mybotshop.de/RMA** or call **+49 (0) 2271 588 93 20**. Support will evaluate the case and, if necessary, issue an RMA number with a return address or shipping label.

Shipping: do not ship goods without an RMA number, parcels sent without one will be declined and returned. Return all goods to: QUADRUPED Robotics GmbH c/o MYBOTSHOP GmbH, Willy-Messerschmitt-Strasse 12, 50126 Bergheim, Germany. Do not ship to the Leverkusen office.

17.2 | Help resources

Channel	Detail
Support e-mail	support@mybotshop.de (always mention your order number)
RMA form	https://www.mybotshop.de/RMA
Phone	+49 (0) 2271 588 93 20
More resources	https://docs.mybotshop.de/h2.html (e.g. the bash script for the Foxtec exoskeleton arms)



18 | FAQ

18.0.1 The cockpit will not load over https.

nginx may not be running. Use `http://192.168.124.162:9000` as a fallback and restart the web proxy. HTTPS is required for the 3D model and Quest teleop. Also check that your PC is set to 192.168.124.150 / 255.255.255.0 and connected to the correct LAN port (Chapter 5).

18.0.2 The robot will not drive from the joysticks.

Velocity commands only move the robot in the Walk state: follow Damp → Ready → Walk with the feet on the ground (Chapter 8), and clear any active E-STOP with E-RELEASE.

18.0.3 The arms do not move (cockpit or Quest).

The robot must be in PhaseWalk with Upper Body activated (Chapter 8), Transfer to real robot must be armed (Chapter 9), and `h2-arm-ik` / `h2-arm-sdk-executor` must be active.

18.0.4 The camera shows a black frame.

The H2 camera is not yet available; we are waiting on Unitree for a software update.

18.0.5 How long does the battery last, and can it be hot-swapped?

Runtime is approximately 3 hours depending on workload. The battery is **not hot-swappable**: power the robot down before removing or installing it. Remove it for long-term storage and handle it per lithium-battery safety guidelines.

18.0.6 Where is the authoritative low-level / SDK reference?

The official Unitree H2 Developer Guide and the `unitree_sdk2` headers define joint indices, limits and reference frames; always validate against them before sending low-level commands.

© MYBOTSHOP GmbH. Unitree H2 humanoid with MYBOTSHOP web cockpit (webserver v9.8.4, ROS 2 Humble). Provided for R&D use; features and figures may change by firmware and configuration. Cross-reference the official Unitree H2 Developer Guide for hardware control locations and mechanical specifications.

