ORBITUS ROBOTICS / WHITE PAPER

From a single-arm X-Hub host to a dual-arm development system

A read-only AMBER B1 case study in preservation, runtime compatibility and continuous control

OR-WP-2026-004 | Version 1.0 | November 5, 2026

Orbitus Robotics · Project lead: Rodolfo Aramayo

Abstract

An older robot controller may be useful as a spare without being a drop-in clone. This case study compares a B1 V1 X-Hub/EtherCAT installation with ROB's current dual-arm host. The audit identifies a 32 GB versus 64 GB disk mismatch, an incompatible C runtime, single-arm launch assumptions and conflicting model limits. A short CPU sample cautions against copying the current workload onto the smaller processor. The study preserves 393 selected files with matching source hashes and proposes separate replacement media, independent arm ownership, transport identification and explicit commissioning gates. Two powered X-Hubs were disconnected during inspection; firmware identity, electrical compatibility and physical retargeting remain unverified.



RESEARCH RECORD

A read-only audit of an older AMBER B1 host identifies storage, runtime, timing and model constraints, then defines a staged path to two-arm control.

Evidence originates in the 2026-11-05 report. Technical white paper · Not peer reviewed.

ROB holding a can during the October 29, 2026 session, photographed by Kierie. This earlier scene provides project context; it does not show the audited spare or a commissioned X-Hub conversion.

Keywords: AMBER B1, X-Hub, EtherCAT, dual-arm control, runtime compatibility, robot recovery, VR teleoperation

1. ROB and the purpose of the spare

ROB is Orbitus Robotics' experimental tracked humanoid for robotics education, human-robot interaction and supervised manipulation. Its two seven-joint arms share a moving torso and mobile base. Cerebro coordinates perception and command admission; phone and visionOS interfaces provide operator input. A second arm-control computer could support recovery, development and a later replacement controller, provided its software, transport and timing are qualified. [6,7,8]

This study asks whether an older AMBER B1 computer can become a reproducible two-arm host. A November 5, 2026 read-only audit compared its installed stack with the working dual-arm master. The contribution is a measured compatibility assessment and a staged conversion plan. It is not a completed clone, an X-Hub firmware identification or a physical control trial. [1]

Evidence class	Method and limit
Live host observations	Read-only OS, storage, interface, process, socket and configuration snapshots. The current master kept running.
Static software inspection	Launch files, package metadata, ELF version requirements, symbols, disassembly, headers and robot models. No vendor launcher executed.
Historical record	An October 20, 2024 ROS log documents an earlier EtherCAT connection. It is not a measurement of current hubs.
Preservation	393 selected robot-stack files copied and individually hash-compared with source reads. No whole-disk image or boot restore was performed.
Operator context	Two X-Hubs were reported powered on, with neither connected by Ethernet. No physical label, EEPROM identity or firmware image was obtained.

The public evidence supplement retains the numerical findings and binary identities while omitting host addresses, adapter serials, access details, operator messages and vendor payloads. Readers can reproduce the publication from its committed inputs; independently reproducing the hardware observations requires matching equipment and a separately controlled inspection. [1]

2. What is installed, and what remains unknown

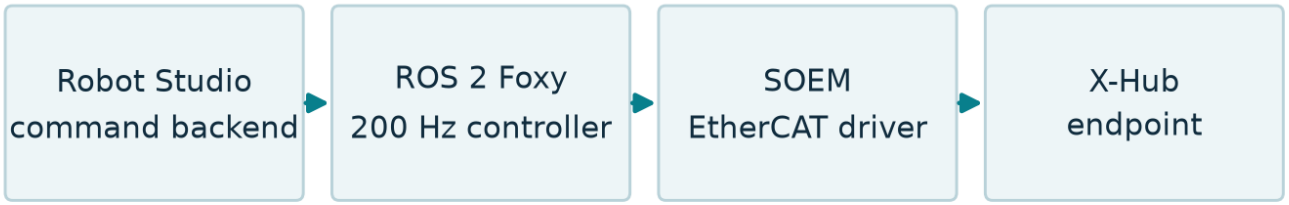


Figure 1. Installed single-arm architecture inferred from launch files and binary inspection; no active X-Hub exchange was performed.

The installation belongs to the B1 V1 X-Hub/EtherCAT software family. Its launch scripts select amber_b1_hw_gui_ethercat.launch.py with hardware and EtherCAT enabled. The AmberB1Hardware plugin imports SOEM functions and uses a fixed network interface. The web/API configuration reports version 1.1; this value does not identify the circuit board or hub firmware. [1,2]

Component	Recorded finding
Platform	Ubuntu 20.04.3; ROS 2 Foxy; glibc 2.31.
ROS packages	controller_manager 0.9.0; joint_trajectory_controller 0.7.0; rclcpp 2.4.1; February 2022 package builds.
Vendor layer	Hardware, GUI, interfaces and bringup packages 0.0.0; description 0.0.1; soem_ros2 1.4.0.
Control configuration	200 Hz; open_loop_control=true; position and velocity command interfaces.
Available materials	Installed headers, binaries, launch files and models; original workspace source tree absent.

The EtherCAT-facing port had no carrier, matching the reported disconnected cables. The driver, GUI control node and command backend were inactive. The historical log records one slave, 92 input bytes, 56 output bytes and expected working counter 3; later entries report packet errors and a lost slave. These describe a past session, not a diagnosis of the powered but disconnected hubs. [1]

The pinned public vendor tree exposes no C/C++ source tree; archived release contents were not exhaustively searched. Installed headers contain CAN error counters, per-joint CAN status and message proxy fields. This supports investigating a CAN bridge, but proves neither electrical compatibility nor interchangeable motor commands. [1,2]

3. Why a direct copy is insufficient

Observed item	Older host	Current dual-arm master
Processor	Celeron N2840; 2 cores	Celeron J4125; 4 cores
Memory / swap	1.96 GB RAM; no swap	3.92 GB RAM; 3.99 GB swap
Disk capacity	32,017,047,552 bytes	64,023,257,088 bytes
Boot layout	UEFI; DOS/MBR; EFI + ext4	UEFI; GPT; EFI + boot + LVM
OS / C runtime	Ubuntu 20.04.3 / glibc 2.31	Ubuntu 22.04.3 / glibc 2.35
Installed transport	Single-arm SOEM EtherCAT	Two USB-CAN adapters

A raw image of the current master cannot fit the smaller disk unchanged. The preservation-first path is separate replacement media, with exact capacity and connector compatibility verified before a restore. A compact fresh installation is another option; neither requires overwriting the original V1 system. [1]

Both current arm executables require GLIBC_2.34, which the old glibc 2.31 installation cannot provide. Ubuntu 22.04 is the initial compatibility baseline, with all other libraries, ABI and CPU instruction requirements still subject to verification. Foxy targets Ubuntu 20.04; its installed binaries should not be casually mixed into another ROS distribution. Canonical lists standard maintenance for Ubuntu 22.04 through May 2027, so a supported successor is a separate validation task. [1,3,4]

A 2.025-second sample of the current master measured 194.1% and 187.1% CPU for the two vendor processes, with 19 threads each: about 3.81 logical CPUs combined. Percentages use one fully occupied logical CPU as 100%. This short snapshot is not a benchmark on the N2840 or a worst-case timing study. It warns against copying the workload unchanged; it does not show that an efficient two-arm controller cannot run on two cores. [1]

The older launchers also assume one arm: broad process matching during restart, shared controller names, unprefixed joints, a fixed command endpoint and a fixed interface. Starting the same launcher twice does not establish isolated two-arm ownership. [1,2]

4. Models, transport and continuous retargeting

Model inspected	Recorded values	Implication
Old standalone URDF	J7: -314 to +314 rad; all joint velocities: 10.	Not a measured limit profile.
Old launched Xacro	J7: +/-3.05 rad; J1-4 velocities: 3.1415; J5-7: 5.23.	Geometry and origins also differ from the standalone model.
Current solver files	One 7-joint chain per arm; limits +/-2.09; effort 100000; velocity 1000000.	Large values cannot be treated as commissioned actuator ratings.

Joint axes, signs, zero offsets, geometry and limits must be reconciled against the identified arm revision and fresh feedback. The selected solver models do not describe the entire mobile body, torso, open grippers, payload and observed table/floor. Collision planning must include these frames and swept paths, especially when the base approaches a table. Model edits alone cannot serialize overlapping commands. [1,5]

Two transport paths remain candidates. Direct, isolated USB-CAN would resemble the current master, but requires verified pinout, power separation, termination, bitrate, motor IDs, units, modes and watchdogs. Retaining the two X-Hubs requires a tested multi-slave EtherCAT topology or supported separate interface paths. With one of the older host's two NICs used for LAN, interface allocation matters. Neither a conventional switch nor a second copy of the single-arm driver establishes the needed topology. [1]

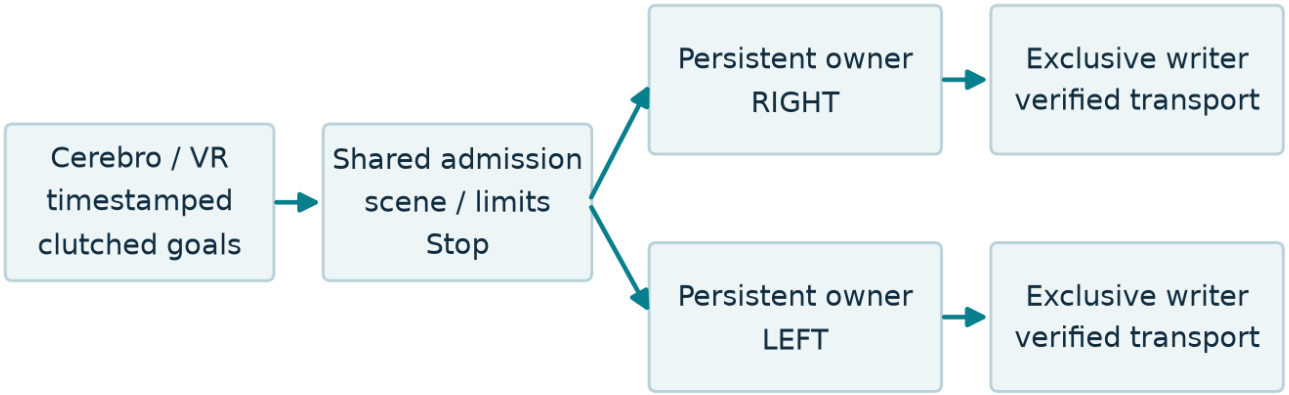


Figure 2. Proposed two-arm ownership boundary. The transport and physical stop behavior still require commissioning.

Each arm should have one persistent trajectory owner and a bounded latest-target mailbox. A new goal updates a coherent position, velocity and acceleration state while limits and Stop remain under the same owner. VR input adds relative clutching, timestamp checks, quaternion interpolation, continuous-branch inverse kinematics and tracking-loss behavior. These are proposed integration requirements. The old GUI was not shown to contain the same detached-command-worker defect documented for the newer core. [1,5]

5. A conversion sequence with reversible gates

Stage	Work and completion evidence
1. Preserve originals	Keep the V1 disk intact. Create coordinated offline images, independent copies and a verified restore target. File hashes alone do not demonstrate a bootable recovery.
2. Prepare separate media	Use a compatible SSD or compact clean installation. Assign a unique machine identity and independent credentials. Disable motor startup before first boot with motor links disconnected.
3. Reproduce the runtime	Verify libraries, ABI and instruction requirements. Separate per-arm models and profiles from host configuration. Install and test deliberate startup/recovery ownership inhibition.
4. Qualify this CPU offline	Exercise both inert owners and simulated transports under realistic load. Record timing, queue age, RAM, temperature and faults; investigate busy polling. Use a stronger host if margin is inadequate.
5. Resolve the transport	Identify both hubs and arm revisions. Verify topology, electrical interfaces, feedback codec, PDO layout where applicable, signs, zeros, modes and drive-side watchdog behavior.
6. Commission incrementally	Under separate physical-test authorization: one arm, tiny bounded move and Stop, then moving retargets; repeat for the other arm, then paired motion and VR.

> First implementation milestone: a bootable development clone with a unique identity, motor startup disabled, two independent simulated arm owners and a reproducible deployment manifest. This milestone does not require a motor adapter change.

Physical-side routing must be explicit. The current installation's internal L/R names are reversed relative to physical sides, so a clone must bind verified adapter identities and profiles to the intended arm rather than infer side from directory names or enumeration order. New hardware needs its own mapping. [1]

The source-owned controller candidate remains physically inert in the cited commissioning record. Prior software and virtual-CAN tests support continued development; they do not implement or authorize live streaming on either of the proposed transports. [6]

6. Acceptance criteria and limits of the evidence

Gate	Required evidence before promotion
Restore and routing	Boot the spare with motor links detached; verify rollback, unique per-arm endpoints and rejection of missing or swapped adapters.
Exclusive ownership	A second writer cannot acquire the arm. Legacy startup and recovery cannot silently restore a competing controller.
Retarget continuity	Forward/reverse goals and input floods preserve bounded position, velocity, acceleration and jerk, stable worker count and latest-target replacement.
Feedback and Stop	Stale feedback, packet loss, partial writes, queue backlog, disconnect and restart produce defined fault behavior without automatic resumption.
Timing at 200 Hz	A 5 ms period sets the scheduling budget. Measure worst-case and high-percentile latency, misses and queue age under load; declare release thresholds before testing.
World and VR	Validate measured model transforms and full swept paths. Exercise clutch release, tracking loss, stale timestamps, orientation equivalence and IK branch changes.

The audit preserved 393 regular files in a 15,203,162-byte archive and compared every archived file SHA-256 with fresh source reads, with zero mismatches. This establishes selected-file integrity at the sampled times. It is not a whole-system backup, a reproducible vendor build or a tested disk restore. The archive and private report are not redistributed; the supplement provides curated findings and identity hashes. [1]

No X-Hub firmware, EEPROM identity or physical board label was obtained. No electrical compatibility, two-hub topology, motor tracking, real-time deadline margin, whole-disk recovery or physical stop behavior was validated. SMART tooling was absent, and physical power stability was not tested. These are open work items, not negative test results. [1]

> Conclusion: use the old computer first as a preserved development and recovery candidate. A dual-arm conversion becomes credible when media recovery, runtime compatibility, timing, transport semantics and exclusive motion ownership each have their own evidence. The plan does not require an arm to settle before every future goal; it requires a commissioned controller that can retarget continuously.

References and supporting records

[1] Curated November 5 host-audit evidence (this publication)

https://www.orbitusrobotics.com/whitepapers/amber-xhub-dual-arm-conversion.evidence.json

[2] AMBER B1 vendor launcher, pinned revision 4074511

https://github.com/MrAsana/AMBER_B1_ROS2/blob/407451125c2adb2dbac8a9c823b54f7cfee4a71a/Scripts/b1.sh

[3] ROS 2 Foxy release documentation: supported platforms

https://raw.githubusercontent.com/ros2/ros2_documentation/foxy/source/Releases/Release-Foxy-Fitzroy.rst

[4] Canonical Ubuntu release cycle, checked November 5, 2026

https://ubuntu.com/about/release-cycle

[5] OR-WP-2026-001: Motion retargeting and the persistent owner

https://www.orbitusrobotics.com/whitepapers/arm-transition-fix-2026-11-01/

[6] OR-WP-2026-003: Recovery-first CAN takeover plan

https://www.orbitusrobotics.com/whitepapers/amber-can-takeover-and-recovery-plan/

[7] ROB project chronology and public codebase map

https://www.orbitusrobotics.com/whitepapers/rob-project-background.json

[8] ROB software platform and operator interfaces

https://www.orbitusrobotics.com/products/

Web edition and supplements: https://www.orbitusrobotics.com/whitepapers/amber-xhub-dual-arm-conversion/

Version 1.0 adapts the November 5 read-only system audit. Conversion, two-hub identification and physical commissioning remain pending.

Prepared with AI assistance from a project audit, installed-file inspection and recorded evidence. No new physical robot trial was performed for this publication. Private machine identifiers, access details, operator messages and vendor binary payloads are omitted.

Photography: Kierie, October 29, 2026. Diagrams: Orbitus Robotics. Correspondence: orbitus@orbitusrobotics.com