RP1 White Paper

System Design and Engineering of the RP1 Humanoid Platform

RoboParty | October 2026



RP1 captured in motion during a studio photo shoot.

Abstract

RP1.3 is a modular humanoid platform for motion control and manipulation research. Its engineering contributions are a serviceable actuator-based body with separated hip axes and parallel ankles, segmented CAN-FD communication with a common joint-command interface, and an integrated workflow from model preparation to motion learning and deployment. Module-level records include 120 h of cumulative unloaded upper-limb motion and 10 min of single-arm dynamic operation with a 5.5 kg payload under the conditions in Section 6. These records do not establish whole-robot service life or a continuous payload rating. PartyOS remains under development. Model, interface, and engineering-source references support further investigation; each result is limited to its stated configuration.

Keywords: humanoid robotics; modular actuation; motion retargeting; sim-to-real; teleoperation; endurance testing.

Introduction

This white paper covers RP1.3, a comparatively stable third iteration of the RP1 platform. It is intended for readers interested in humanoid robots, including researchers and developers. Specific configurations and historical illustrations retain their individual identifiers.

Assessment uses task-specific quantities, including joint torque, range of motion, and operating duration, with the associated load and test conditions. Versatility, efficiency, and reliability remain design objectives unless supported by corresponding measurements.

Mechanical and electrical documentation, software interfaces, models, and test records connect platform design to evaluation. Mini Cheetah and ToddlerBot provide context for this engineering workflow [4, 5].

The paper proceeds from mechanical design (Section 1) through electrical architecture (Section 2), embedded software (Section 3), control and deployment (Section 4), PartyOS integration (Section 5), and testing (Section 6). Design values, calculations, simulations, RP1 measurements, upstream results, and planned functions are distinguished throughout.

Contents

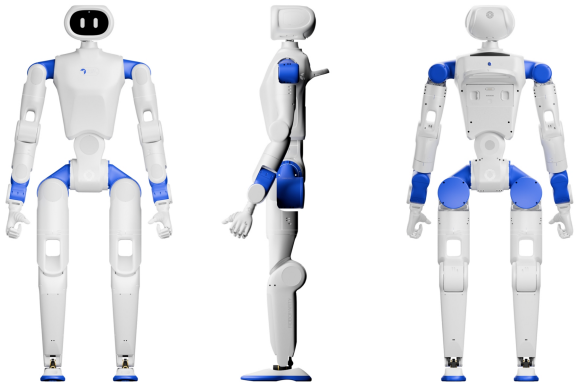
Introduction 1
1 Mechanical Structure 3
1.1 Overview and Design Principles 3
1.2 Platform Positioning 3
1.3 Human Correspondence and Motion Mapping 4
1.4 Joint Module Selection 6
1.5 Joint Axis Layout and Task Space 10
1.6 Manufacturing Assembly and Service 18
1.7 From Mechanical Assembly to URDF 21
2 Electrical System 24
2.1 System Architecture 24
2.2 Board Roles and Hardware Implementation 24
2.3 Bus Segmentation and Communication Layers 26
2.4 Hot Swapping Protection and Abnormal Response 27
3 Embedded Software 28
3.1 Architecture and Communication 28
3.2 Hardware Abstraction and System Support 30
3.3 Logging and Stress Testing 31
3.4 Interaction and Remote Control 32
4 Algorithms 32
4.1 hhtools Motion Retargeting 33
4.2 Training Framework 35
4.3 Deployment Framework 37
4.4 Whole Body Intelligence 39
5 PartyOS 41
6 System Testing 42
6.1 System Configuration and Validation Scope 42
6.2 Complementary Upper and Lower Limb Endurance Tests 43
6.3 Dynamic Single Arm Load Validation 44
6.4 Test Software and Evidence Management 47
7 Limitations and Future Work 48
8 Conclusion 48
Appendix A Test Methods and Planned Conditions 48
Appendix B Configuration Checks 48
Appendix C RP Hand 48
C.1 Configuration and Specifications 48
C.2 Dimensions and Electrical Interface 49
C.3 Software and Developer Interfaces 50
Appendix D References 50
Appendix E Humanoid Platform Index 52
Appendix F Figure and Table Index 55
Appendix G Engineering Evidence Index 57

1 Mechanical Structure

1.1 Overview and Design Principles

RP1.3 (internally designated ROBOTO 01) integrates actuators, computing, power, and sensors in a configurable research body. Published proportions, joint layouts, and interfaces support module replacement. CNC-machined parts and standard components suit small-batch production; independently removable sections provide access to wear components.

The mechanical platform provides a stable reference for changes in software and control. Accessible interfaces and independently replaceable modules allow different research directions to reuse the same physical platform as algorithms evolve.



The 24- and 28-DoF bodies share two 6-DoF legs and a 2-DoF waist (roll and yaw). Each arm has 5 or 7 DoF, respectively. A 2-DoF head brings the 28-DoF body to 30 active DoF; two 6-axis hands bring the total to 42. Passive finger motion is excluded. Head and hand modules are counted separately; RP Hand specifications appear in Appendix C.

Figure 1.1. Three views of RP1 in the white-and-blue color scheme. Source: RoboParty [21].

Table 1.1. RP1 body configurations.

Item	24-DoF body	28-DoF body	Notes
Active DoF per leg	6	6	Hip 3; knee 1; ankle 2
Active waist DoF	2	2	Roll and yaw
Active DoF per arm	5	7	Additional wrist orientation
URDF mass sum (kg)	39.1093	40.7264	URDF values based on physical weighing; includes structure, actuator modules, electrical components, and wiring

1.2 Platform Positioning

RP1 is a 135 cm humanoid for research, development, and education, within the small-to-medium range in Figure 1.2.

Manipulation. The upper limbs are intended to reach desks and common work surfaces with limited tiptoe standing or bending.

Motion. Lower body mass and leg inertia support faster stepping. Under comparable conditions, a lower center of mass reduces balancing torque, and fall energy is lower than for platforms taller than 1.6 m.

Interaction. Lower mass and end-effector inertia reduce collision energy. Required separation and protection still depend on task, speed, tooling, and risk assessment; size alone does not establish safe collaboration.

Cost. Under geometric scaling, joint torque demand grows approximately with the fourth power of body size. The 1.35 m body reduces actuator and reducer requirements. Design targets include single-person handling, short service cycles, and use in ordinary laboratories and teaching spaces.

The approximate fourth-power scaling of torque follows from geometric similarity: mass scales with the cube of characteristic length, while the gravitational moment arm scales linearly. Their product gives the fourth-power dependence of static torque. For motion with comparable linear acceleration, rotational inertia scales with the fifth power of length and angular acceleration inversely with length, giving the same torque scaling. This is a scaling estimate under these assumptions, rather than a sizing law for robots with different materials, proportions, or trajectories.

The design balances reach, motion, interaction, and manufacturing cost without replicating every anatomical joint.

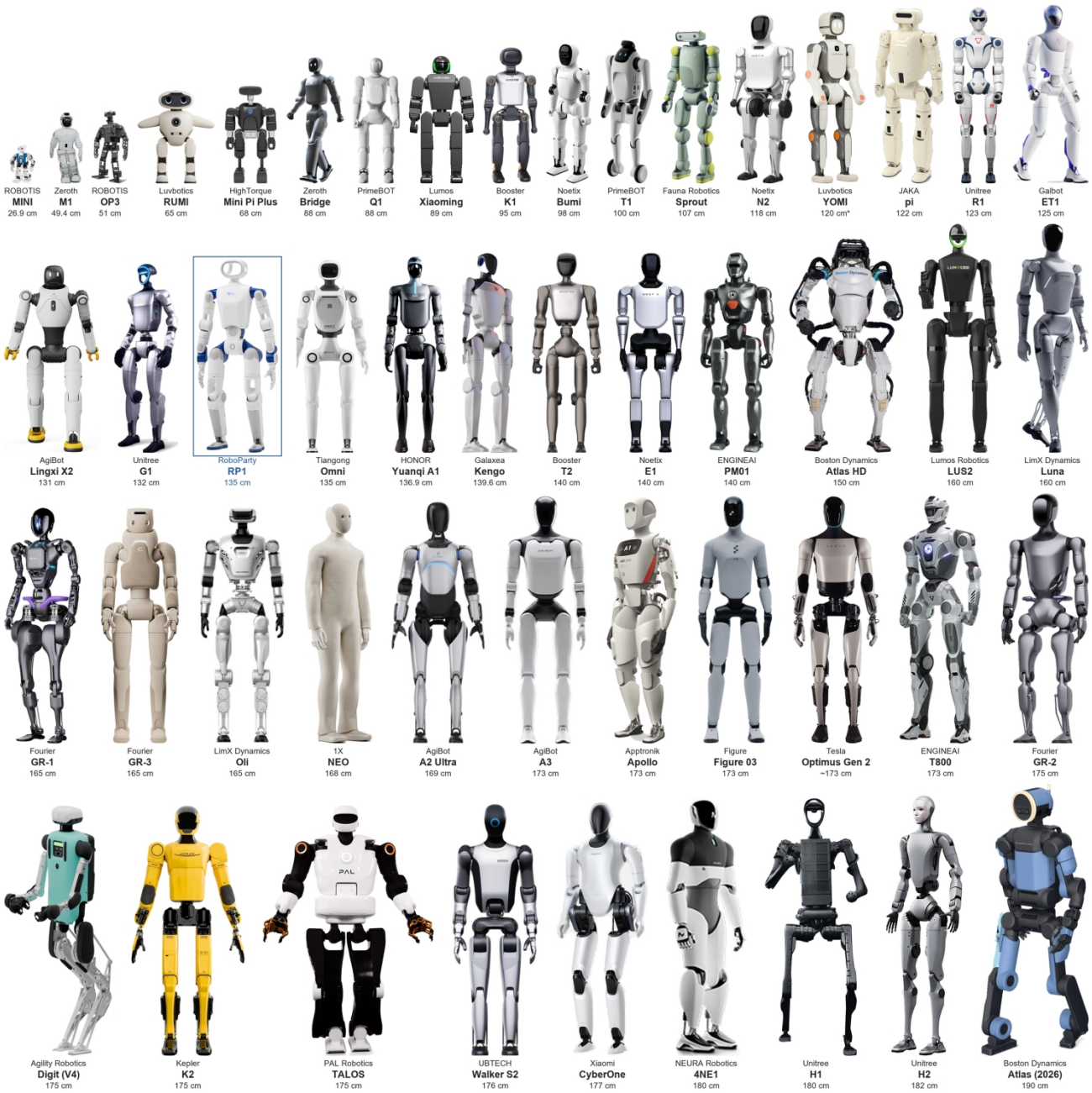


Figure 1.2. Humanoids by nominal height, 26.9-190 cm; RP1 is blue. Versions and sources: Appendix E. Heights do not share a measurement protocol.

1.3 Human Correspondence and Motion Mapping

1.3.1 Task Space Correspondence

RP1 uses human workspaces and tasks as the morphological reference. Limb workspaces cover common operating regions without requiring one robot axis for every anatomical DoF.

This task-space correspondence allows a common body to serve different tasks and reuse human motion data.

Forward kinematics maps joint angles to end-effector poses; joint limits and collisions constrain reachability. The Jacobian maps joint to end-effector velocities and identifies directional constraints and singularities.

1.3.2 From Human Motion to Robot Trajectories

Two-stage mapping

Anatomical mapping associates human hips, knees, ankles, shoulders, elbows, and wrists with robot joints using segment proportions, landmarks, and end-effector poses.

Constraint correction then adapts trajectories to robot joint limits, ground contact, and self-collision constraints.

Motion capture, simulation, and generated trajectories share the anatomical mapping, followed by platform-specific correction. Hardware changes need not rebuild the human-side workflow.

SMPL-X as the human reference

SMPL-X (SMPL eXpressive) [6] represents the body, hands, and face in one parametric mesh. Shape, pose, and expression parameters provide landmarks, segment orientations, and root poses for mapping motion data to RP1.

The source specification describes 54 SMPL-X joints, compared with 24 or 28 active RP1 body DoF. Human-model parameters and robot DoF are not equivalent counts.

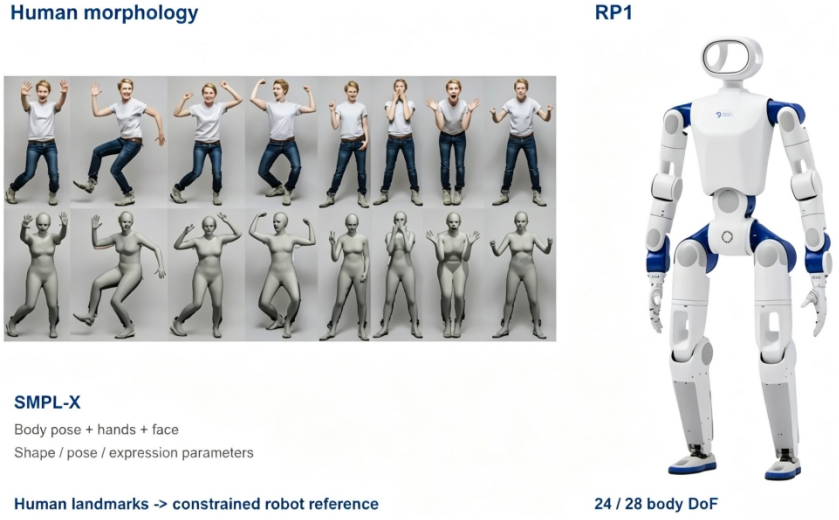


Figure 1.3. Human poses, SMPL-X meshes, and RP1 morphology. Sources: SMPL-X [6] and RoboParty [21].

Actuator groups and human joints

Figure 1.4 presents a schematic functional correspondence between 30 RP1 actuator modules in 14 groups and selected SMPL-X landmarks and anatomical regions. The accompanying table gives regional module and group counts.

The illustrated 30-DoF configuration comprises the 28-DoF body and a 2-DoF head. Reducing each wrist from three modules to one gives the 24-DoF body configuration.

Three human spine segments share one 2-module torso group. Finger joints are omitted and belong to independent hand modules.

1.3.3 Base DoF and Extension Modules

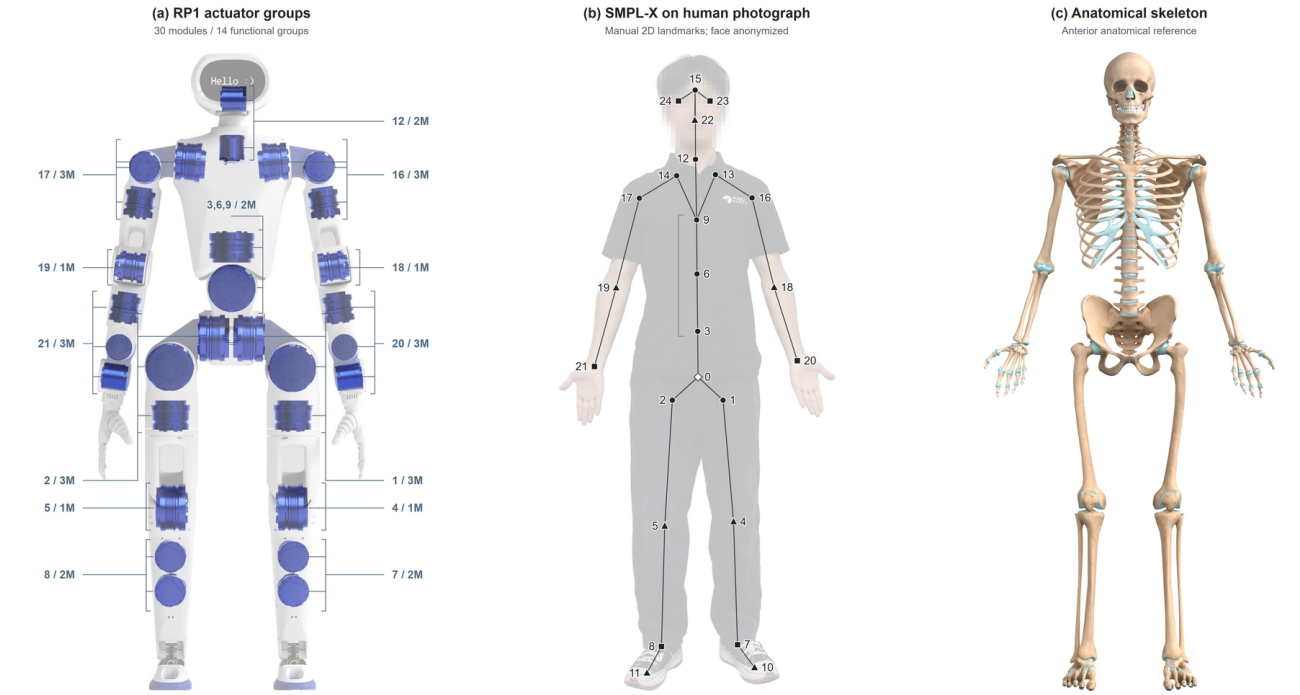
Locomotion and manipulation require different DoF allocations. RP1 separates a common base, optional upper limbs, and independent head and hand modules to avoid unnecessary hardware while retaining task-specific motion.

Configuration allocation

Common base: two 6-DoF legs and a 2-DoF waist, totaling 14 DoF.

Upper limbs: 5 DoF per arm in the 24-DoF body; 7 per arm in the 28-DoF body for additional manipulation redundancy.

Extensions: a 2-DoF head and two 6-axis hands raise the 28-DoF configuration to 42 active DoF.



ACTUATOR-GROUP CORRESPONDENCE

Region	SMPL-X joint IDs (L / R)	Modules / group	Groups	Total modules
Neck	12 neck	2	1	2
Shoulder	16 / 17 shoulder	3	2	6
Elbow	18 / 19 elbow	1	2	2
Wrist	20 / 21 wrist	3	2	6
Torso	3, 6, 9 spine1-3	2 (shared)	1	2
Hip	1 / 2 hip	3	2	6
Knee	4 / 5 knee	1	2	2
Ankle	7 / 8 ankle	2	2	4
Total			**14**	**30**

Heights normalized; manually estimated joints and simplified functional DOF. No calibrated kinematic fit.

READING THE CORRESPONDENCE

17 / 3M — SMPL-X joint ID / modules in one group

Local branches collect the modules of each group

M = actuator module; paired IDs follow anatomical L / R.

● 3 DOF ■ 2 DOF ▲ 1 DOF ◇ Root: 3R + 3T

Wrist: 3 robot modules; 2 DOF in the human schematic.

Torso: one shared 2-module group for spine1-3.

Ankle: two shank-mounted modules per ankle.

No separate module group assigned in this diagram:

0 pelvis; 10/11 foot; 13/14 collar; 15 head;
22 jaw; 23/24 eye_smplhf. Fingers are excluded.

Figure 1.4. RP1 actuator-group correspondence with SMPL-X human joints. Sources: SMPL-X [6] and RoboParty [21]. Anatomical illustration source: Anatomia, https://anatomia.com.mx/anatomia/lineaOsteo/MicroSitio/Esqueleto/index.html.

1.4 Joint Module Selection

Four sizes of integrated rotary ROMOMO modules meet joint-specific torque, speed, and packaging requirements. Clamp mounts and hollow cable passages provide assembly and service access while accommodating cooling and wiring.

1.4.1 Actuator Topology

Transmission type

RP1 uses rotary joints throughout the body. The alternative considered was linear actuation through a screw and linkage.

Rotary modules provide direct angular output, compact flange mounts, high bandwidth, and potentially continuous rotation; loaded holding consumes power and generates heat. Linear actuators convert motor rotation to thrust and then to joint torque. The source comparison reports 3-6 times the thrust of a comparable ball screw for the screw types considered. Self-locking and preload can improve holding and backlash, depending on mechanism design. Cited strokes of 35-70 mm limit travel, and torque follows $\tau = F r \sin(\text{theta})$. High reduction, self-locking, and linkage dynamics can reduce response and passive yielding.

Rotary joints suit large travel and rapid response; linear actuators can suit limited travel, high loads, and long holding periods. RP1 selects rotary modules for reproducibility, cost, and service access. The following tables retain qualitative comparisons from the source design records; they are not controlled, matched-condition benchmarks.

In a rotary module, the reducer converts motor speed directly into joint torque. A screw actuator first converts rotation into axial force and then uses a linkage moment arm to generate joint torque. Planetary roller screws

distribute load through multiple threaded roller contacts, in contrast to the localized contacts of ball screws. Stroke and linkage geometry restrict the achievable motion and make the force-to-torque relationship configuration-dependent. These considerations must be assessed together with impact loading, holding duty, backdrivability, and distal inertia; self-locking is not an inherent property of every roller-screw design.

Table 1.2. Transmission characteristics. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.

Property	Rotary joint module	Linear actuator considered in source
Output	Angle / torque	Displacement / thrust, followed by linkage conversion
Range	Up to continuous 360-degree rotation	Cited stroke 35-70 mm; angular range depends on linkage
Static holding	Powered holding; heat and energy use without a brake	Self-locking capability depends on screw design
Impact	Planetary transmission tolerates overload; harmonic flexsplines can be vulnerable	High impact capacity reported for the considered screw type; source text clipped
Backlash	Harmonic: nominal zero; planetary: 10-15 arcmin in source comparison	Preload can reduce backlash; remaining source specification is clipped
Dynamic bandwidth	High; low-inertia response	Lower with high reduction and additional linkage
Control	Direct angular output	Nonlinear torque: tau = F r sin(theta)
Cost	Low in the source comparison	Low in this source comparison; a different integration comparison below reports high cost

Table 1.3. Actuator integration comparison. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.

Property	Rotary joint	Linear actuator
Drive arrangement	Integrated motor and planetary reducer	Motor and screw with motion conversion
Complexity	Simpler	More complex
Cost	Lower	Higher
Mass	Lighter	Heavier
Transmission efficiency	Higher	Additional screw-transmission losses
Maintenance	Not visible in the supplied cropped screenshot	Not visible in the supplied cropped screenshot

Reducer selection

RP1 uses planetary reducers for stiffness, overload resistance, moderate cost, and reported single-stage efficiency above 0.97. A ratio near 20 balances response and oscillation. In the compared designs, RV reducers are heavier and more expensive; harmonic reducers have greater compliance and flexspline fatigue concerns.

RV reducers combine a cycloidal reduction stage with an additional gear stage, offering substantial torque capacity and precision at the cost of a larger mass and envelope in the source comparison. Harmonic reducers obtain large reduction ratios through the tooth-count difference between a flexspline and a circular spline; ratios above 100 are feasible. Flexspline compliance and losses at light load and high speed must be weighed against precision, compactness, and torque density. RP1 selects planetary reduction for the stiffness, efficiency, and cost priorities of this platform; the comparison does not imply that one reducer type is preferable for every humanoid joint.

Table 1.4. Reducer comparison. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.

Property	Planetary	Harmonic	RV
Meshing	Multiple contacts	Elastic flexspline deformation	Cycloidal gearing
Impact / overload	High	Low	High
Stiffness	High	Low	High
Efficiency	Single stage > 0.97	Low	Moderate
Cost	Moderate	Higher	Higher
Mass at comparable size	Relatively low	Low	High
Service life	Long in source comparison	Affected by flexspline deformation	Long in source comparison

The selected NGW arrangement uses shared planet gears with internal and external meshing. Typical single-stage ratios are 3-10; two stages can reach 40 or more. The alternative NW arrangement can be shorter axially but requires more complex machining and a larger radial envelope, limiting reuse in stacked assemblies.

Table 1.5. Planetary arrangements. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.

Property	NGW	NW
Arrangement	Internal/external mesh with common planet gears	Internal/external mesh with double-row planets
Complexity	Simple	More difficult machining
Axial size	Compact	More compact
Radial size	Smaller	Larger
Efficiency	High	Slightly lower
Reuse in stacked assemblies	Good	Limited

Rotor arrangement



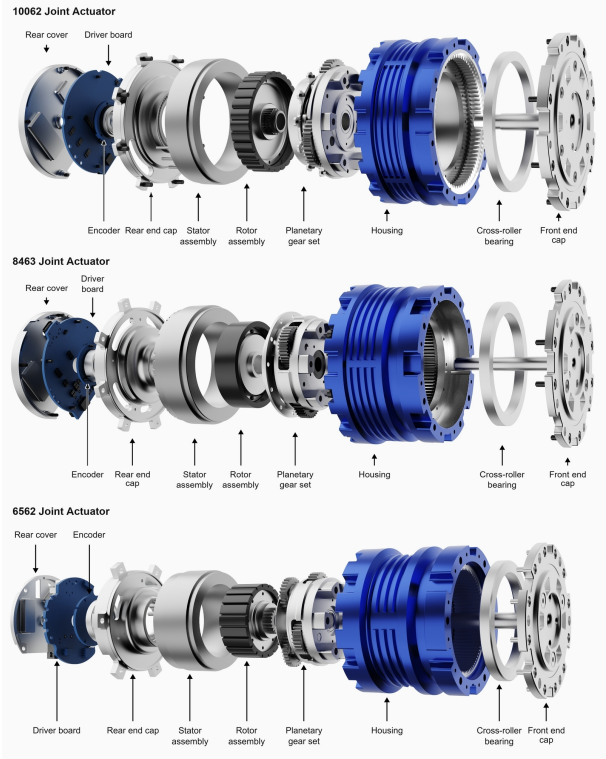


Figure 1.5. ROMOMO joint module series. Source: RoboParty [21].

Each module combines an inner-rotor motor and an NGW planetary reducer. Low rotor inertia supports rapid response, and stator windings near the housing conduct heat outward. The compared outer-rotor design has higher inertia and slower response.

Table 1.6. Nominal specifications of the ROMOMO modular actuator series applied on RP1.3

Parameter	5552	6562	8463	10062
Gear ratio	36:1	19.5:1	20:1	22:1
Rated torque / N·m	9	10	20	44
Peak torque / N·m	30	35	80	160
Rated speed / rpm	130	60	60	80
Max no-load speed / rpm	160	200	155	156
Outer dia. × height / mm	55 × 52	65 × 62	84 × 63	100 × 62
Hollow bore dia. / mm	4.5	4.5	4.5	7.5
Mass / kg	0.38	0.53	0.78	1.196

1.4.2 Parameters and Sizing

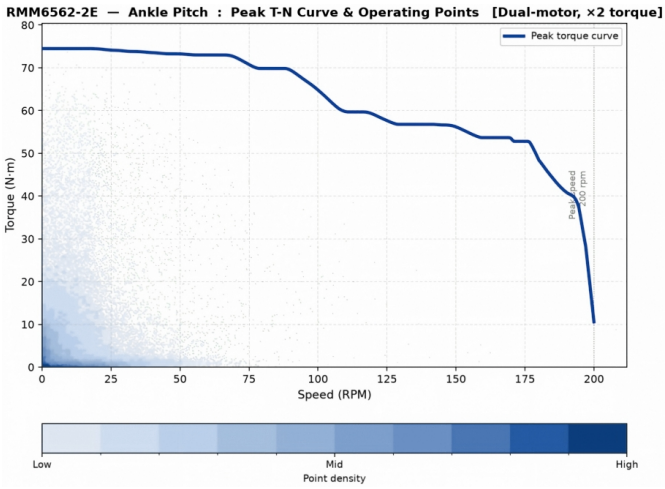


Figure 1.6. RMM6562-2E operating points for five representative motion conditions. Source: RoboParty [21].

Actuator sizing considers torque, speed, and rotor inertia.

Proximal joints are sized primarily for support torque; distal joints also prioritize size and swing inertia. Two methods define the selection envelope.

Simulation starts with an approximate physical model and exports its geometry and mass parameters to the simulation environment. Joint torque and speed are sampled under representative motions and payloads, then compared with the corresponding module torque-speed curves (Figures 1.6-1.8). The resulting operating-point distributions guide module selection across the required motion envelope.

Industry reference ranges for the 1.3-1.4 m class, including the G1 and PM01 examples in the source design comparison, provide an initial envelope of approximately 120-140 N m peak hip/knee torque and output speeds generally below 200 rpm. These ranges are preliminary design references, not measurements of RP1. Final module selection combines them with the simulated joint operating points.

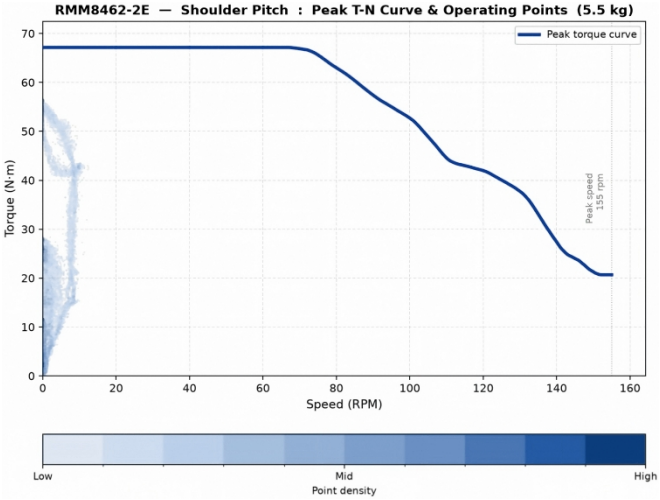


Figure 1.7. RMM8463-2E operating points for combined lateral raises and elbow lifts with a 5.5 kg end load. Source: RoboParty [21].

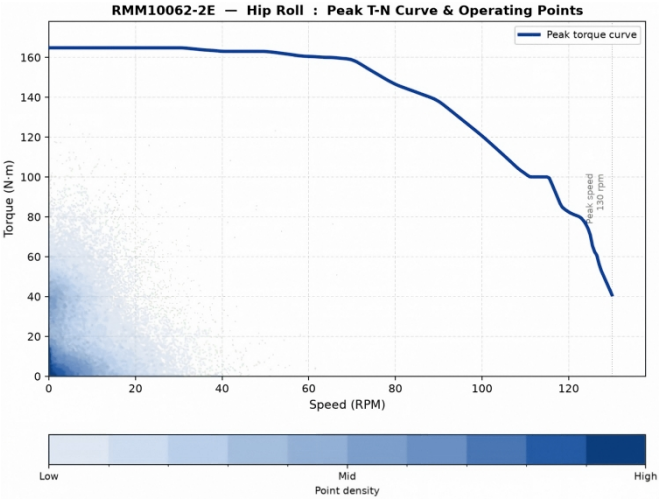


Figure 1.8. RMM10062-2E operating points for five representative motion conditions. Source: RoboParty [21].

1.4.3 Mechanical Integration

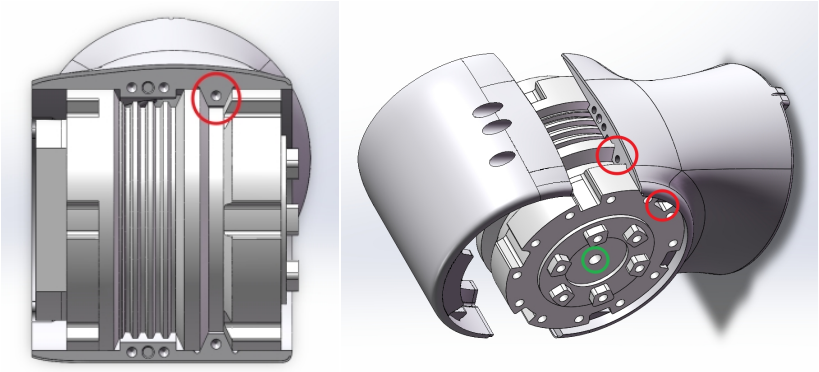


Figure 1.9. Clamp interface and split-clamp removal. Source: RoboParty [21].

Clamp mounts and locating grooves constrain axial and radial motion. They provide compact centering and allow removal without dismantling adjacent flanges.

A hollow motor passage routes cables internally to reduce swinging, wrapping, wear, and joint interference.

A five-minute replacement target is reported by manufacturing personnel; no measured procedure or acceptance record is included in the present evidence.

1.5 Joint Axis Layout and Task Space

Reachability depends on hand orientation, elbow position, and whole-limb clearance. Stepping and squatting similarly require feasible leg paths and ground approach.

Joint-axis layout defines the mechanism. Task space specifies required hand or foot poses; workspace describes reachable poses [1]. Travel and collision checks must cover the entire trajectory.

RP1 combines 5- or 7-axis arms, separate waist and hip joints, and parallel ankles. Shell clearance accommodates motion; clamps and tool interfaces provide local service access.

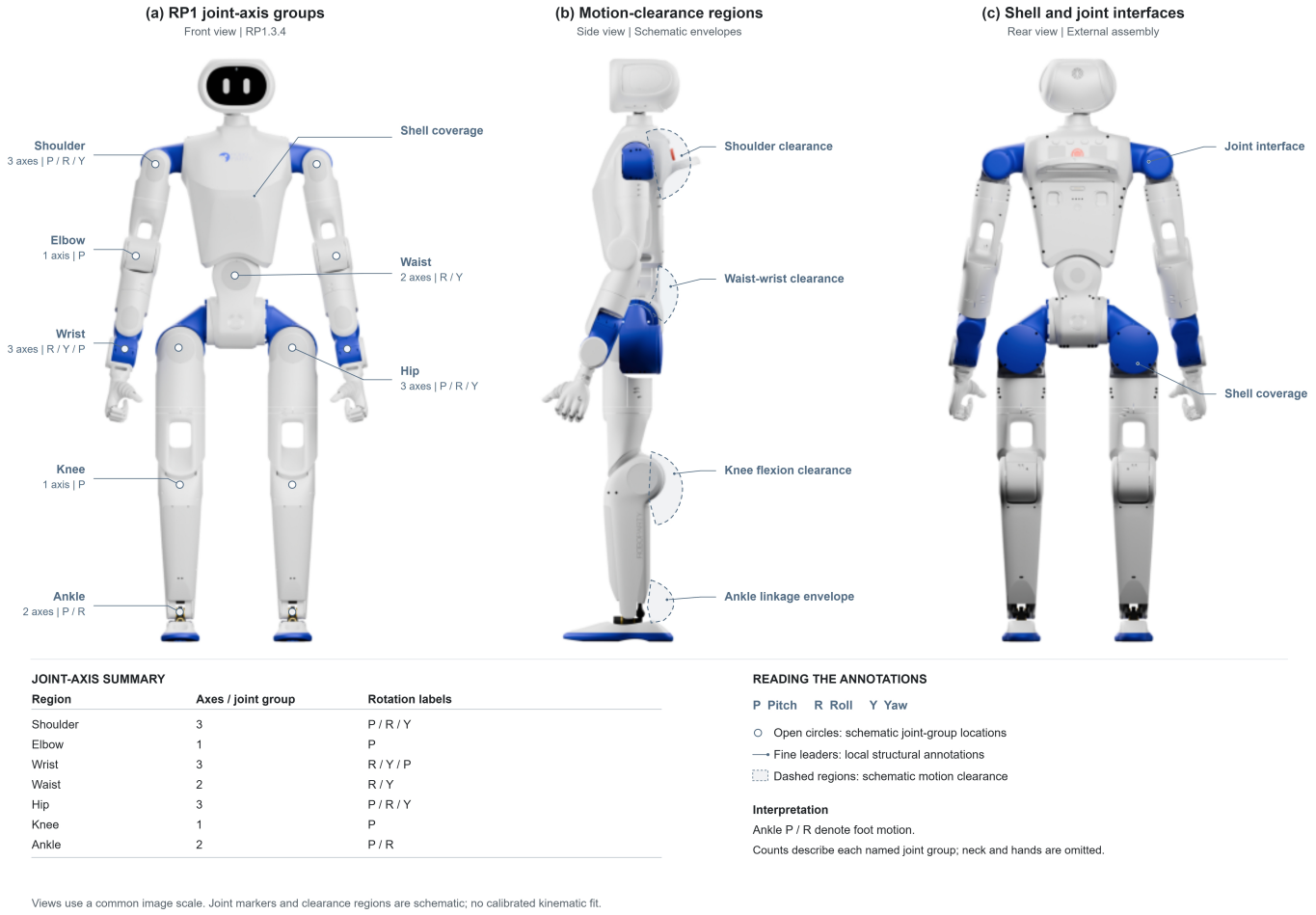


Figure 1.10. RP1.3.4 joint-axis groups, shell layout, and motion-clearance regions. P: pitch; R: roll; Y: yaw. Axis locations and clearance regions are schematic. Source: RoboParty [21].

Table 1.7. RP1 joint limits and human reference ranges. Robot zero positions and signs differ from anatomical conventions; human sources are identified below by the table-local labels [ROM-1] and [ROM-2].

RP1 and Human Joint Range of Motion						
Region	**RP1 joint**	**RP1 + limit (°)**	**RP1 - limit (°)**	**RP1 total (°)**	**Human normal ROM (°) [ROM-1, ROM-2]**	**Human total (°)**
Head / neck	Pitch	+45	-30	75	Extension 0–45; flexion 0–45	90
	Yaw	+30	-30	60	Rotation: left 0–60; right 0–60	120
Waist	Yaw	TBD	TBD	TBD	Rotation: left 0–35; right 0–35	70
	Roll	TBD	TBD	TBD	Lateral flexion: left 0–45; right 0–45	90
Upper limb	J1 Shoulder pitch	+135	-195	330	Flexion 0–180; extension 0–60	240
	J2 Shoulder roll	+171	-27	198	Abduction 0–180; adduction back to neutral 0*	180*
	J3 Shoulder yaw	+165	-165	330	Internal rotation 0–70; external rotation 0–90	160
	J4 Elbow	+71	-91	162	Flexion 0–150; extension to 0	150
	J5 Wrist roll	+147	-147	294	Pronation 0–80; supination 0–80	160
	J6 Wrist yaw	+85	-85	170	Radial deviation 0–20; ulnar deviation 0–30	50

RP1 and Human Joint Range of Motion						
Region	**RP1 joint**	**RP1 + limit (°)**	**RP1 - limit (°)**	**RP1 total (°)**	**Human normal ROM (°) [ROM-1, ROM-2]**	**Human total (°)**
	J7 Wrist pitch	+70	-70	140	Flexion 0–80; extension 0–70	150
Lower limb	hip_pitch	+164	-164	328	Flexion 0–120; extension 0–30	150
	hip_roll	+149	-42.5	191.5	Abduction 0–45; adduction 0–30	75
	hip_yaw	+158.5	-158.5	317	Internal rotation 0–45; external rotation 0–45	90
	knee	+4	-162.5	166.5	Flexion 0–150; extension to 0	150
	ankle_pitch	+35	-45	80	Dorsiflexion 0–20; plantarflexion 0–50	70
	ankle_roll	+15	-15	30	Inversion 0–35; eversion 0–15	50
Basis: RP1 values are supplied limits; TBD = to be defined. Human values are AAOS clinical references [ROM-1, ROM-2], not population intervals or individual maxima; age, sex and active/passive testing are not distinguished here.						
Comparison: Robot signs/zeros differ from anatomical neutral. Totals span endpoints of one motion axis. J1-J7 follow the existing RP1 model order; J5 corresponds functionally to forearm rotation.						
Anatomy: Shoulder elevation includes the shoulder girdle; waist values cover the thoracolumbar region; inversion/eversion covers the ankle-foot complex. Elbow/knee extension is 0 degrees; individual hyperextension is excluded.						
* J2 total covers abduction to neutral only. The cited AAOS tables do not specify additional cross-body adduction under this convention; 0 denotes neutral, not absence of adduction.						
[ROM-1] Skinner SB, McVey C. Pocket Notes for the Physical Therapist Assistant. 2nd ed. Jones & Bartlett Learning; 2013:11,13. Tables 1.4 and 1.6 (AAOS values). Publisher source						
[ROM-2] Norkin & White's Measurement of Joint Motion: A Guide to Goniometry. 6th ed. Appendix A, Table A.1 (AAOS column). Publisher source						

1.5.1 Upper Limbs

Selecting for hand position and orientation

A side grasp and a palm-down approach can require different wrist and elbow configurations at the same hand position. Arm selection therefore depends on target poses.

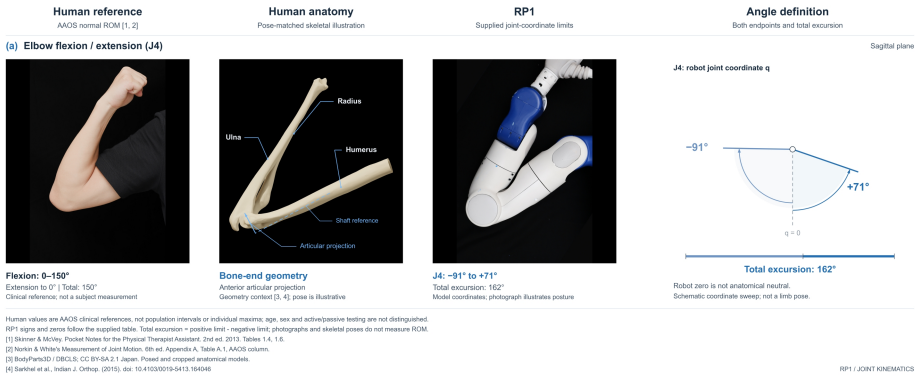
The 7-axis arm has three shoulder, one elbow, and three wrist axes. The 5-axis arm has fewer wrist adjustments; feasible end-effector orientations depend on the target position and coordinated shoulder, elbow, and torso motion.

Shoulder axes establish the upper-arm direction, elbow flexion adjusts reach and the forearm path, and wrist axes orient the end effector. These roles are coupled during a task: a desired hand pose may require coordinated motion across the arm, while the redundant axes of the 7-axis configuration allow alternative elbow and wrist configurations.

Checking the complete arm path

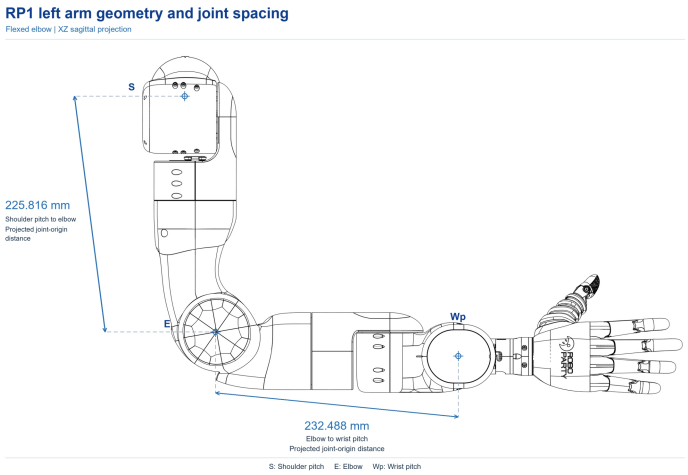
Checks include palm orientation and elbow position during reaching, forearm-to-torso clearance during flexion, and hand-to-hip clearance at rest. Bimanual checks also include the held object.

Clearance must hold throughout reaching, grasping, and withdrawal, not only at the final pose.



RP1 elbow flexion: 162 degrees, with extension at zero. Human ranges in this and the following comparisons are literature references, not measurements of the photographed person; in-figure citations apply locally.

Figure 1.11. Human and RP1 elbow flexion. Human reference: [42]; RP1 limits and photographs: [21].



The supplied left-arm drawing gives projected joint-origin distances of 225.816 mm from shoulder pitch to elbow and 232.488 mm from elbow to wrist pitch. Figure 1.12 documents the XZ sagittal projection in a flexed-elbow pose; these values describe the depicted geometry and should not be substituted for configuration-specific three-dimensional URDF transforms.

Figure 1.12. Left-arm geometry and projected joint-origin spacing. S: shoulder pitch; E: elbow; Wp: wrist pitch. Dimensions follow the supplied CAD drawing and illustrated pose. Source: RoboParty [21].

1.5.2 Waist and Hips

Adapting human motion to robot joints

RP1 has waist roll and yaw but no independent waist pitch. Retargeting must distribute human forward bending across hips, knees, and whole-body posture, accounting for axis locations and segment proportions [2].

Figure 1.13 compares one-, two-, and three-axis waist configurations. RP1 roll and yaw support lateral trunk tilt and heading adjustment; forward bending requires lower-body coordination. The G1 EDU example denotes the optional three-axis configuration. Axis count alone does not establish range, torque capacity, or task performance.

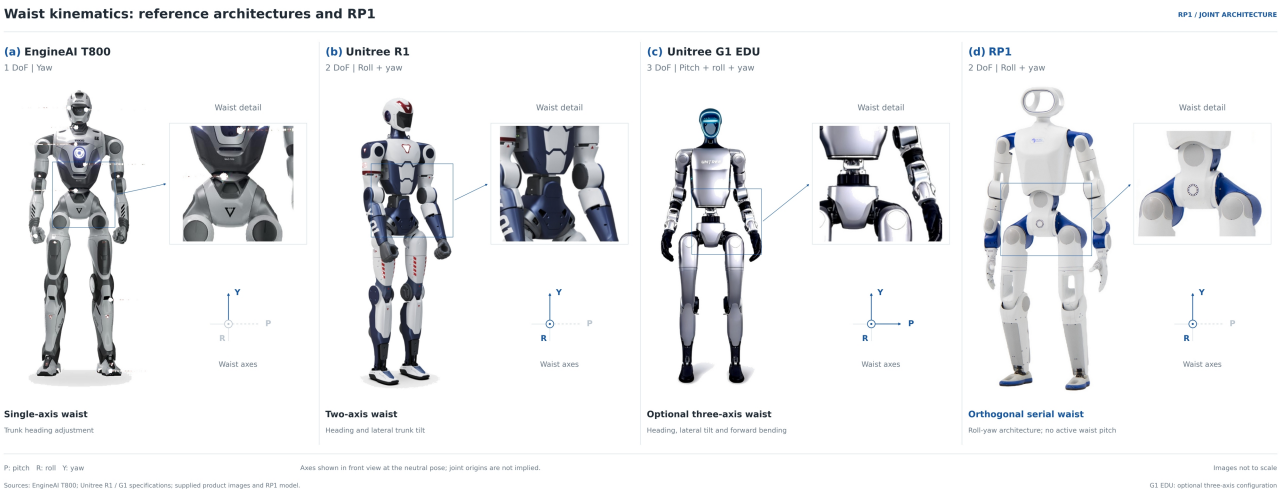


Figure 1.13. Waist architectures: EngineAI T800, Unitree R1, Unitree G1 EDU, and RP1. P: pitch; R: roll; Y: yaw. Model variants are as labeled; images are not to scale. Source: supplied comparison artwork; manufacturer sources in Appendix E; RP1 [7, 21].

Serial hip actuation

The human hip combines flexion, abduction, and axial rotation in a ball-and-socket joint. RP1 uses three serial joints with separated origins, producing a different motion path.

FAR order and RP1 layout

FAR denotes flexion-abduction-rotation, or pitch-roll-yaw, from pelvis to thigh. Here, R means axial rotation (yaw), not roll. The order describes connectivity, not isolated motion during operation.

FAR and ARF move different downstream structures. The comparison holds the pelvis and other joint angles fixed while moving one joint.

Table 1.8. Hip serial order.

Comparison	FAR: flexion-abduction-rotation	ARF: abduction-rotation-flexion
Pelvis-to-thigh order	Pitch -> Roll -> Yaw	Roll -> Yaw -> Pitch
Pitch alone	Moves the downstream roll/yaw structures and leg.	Last hip joint; moves the downstream leg. Roll and yaw are upstream.
Roll alone	Moves the downstream yaw structure and leg.	Moves the downstream yaw/pitch structures and leg.

Selection depends on module mass, dimensions, axis spacing, clearance, and target motions. Step length, energy use, and stability cannot be inferred from axis order alone.

The reviewed RP1 URDF uses pitch-roll-yaw (FAR). Joint origins are separated, with roll below pitch at zero pose. Concurrency, inclination, and spacing require the actual model.

Lowering the roll module frees space above the hip and beside a hanging arm. The changed thigh path requires clearance checks for forward and lateral steps, adduction, and deep squats.

Figure 1.14 compares anatomical and robotic hip architectures. Lowering the RP1 roll axis changes thigh clearance and swept volume. The illustrated 61 mm offset applies to the supplied CAD configuration; other revisions require verification. These geometric differences motivate collision and motion checks, rather than a performance ranking.

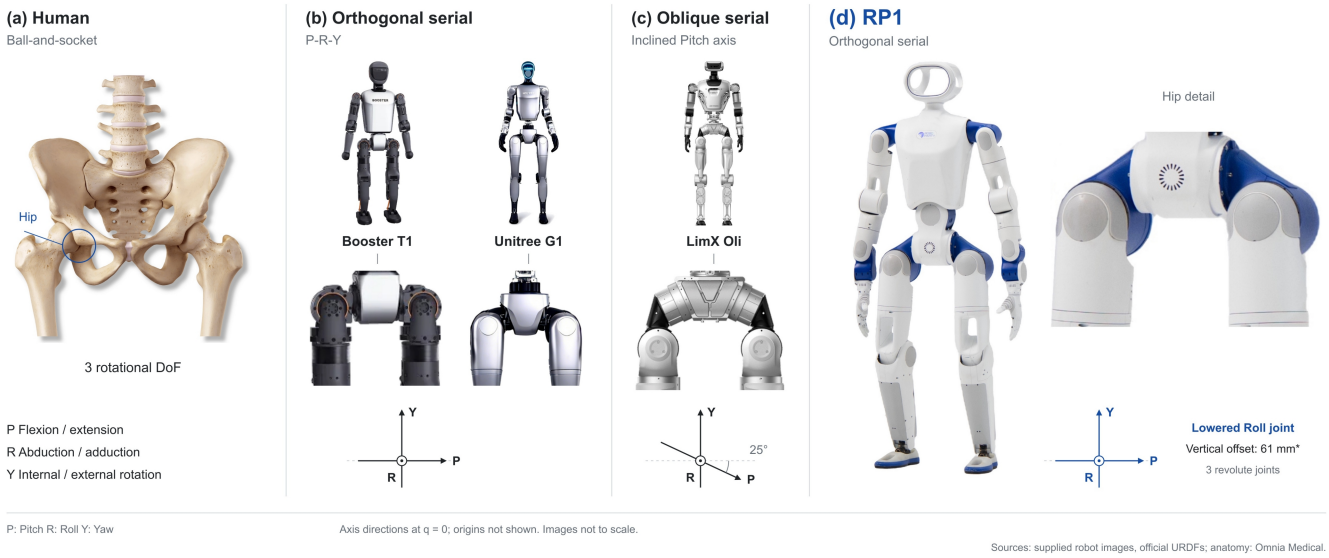
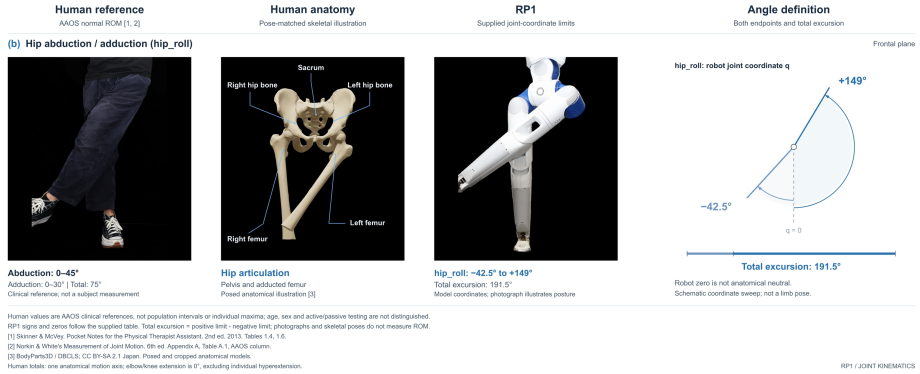


Figure 1.14. Hip reference architectures and RP1 joint layout. Axis sketches indicate neutral-pose directions, not coincident origins. Source: supplied comparison artwork and labeled robot sources; anatomy credited to Omnia Medical; RP1 [7, 21].



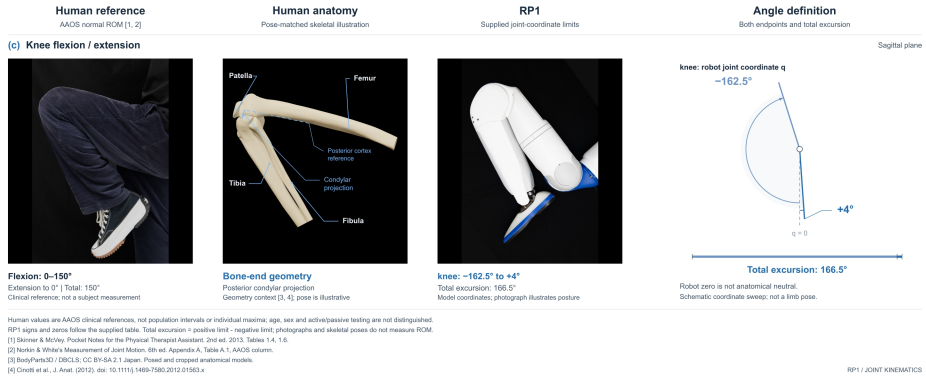
RP1 hip adduction: 42.5 degrees from the pelvis-fixed neutral pose. Human reference: [43].

Figure 1.15. Human and RP1 hip adduction. Human reference: [43]; RP1 limit and photographs: [21].

1.5.3 Lower Limbs

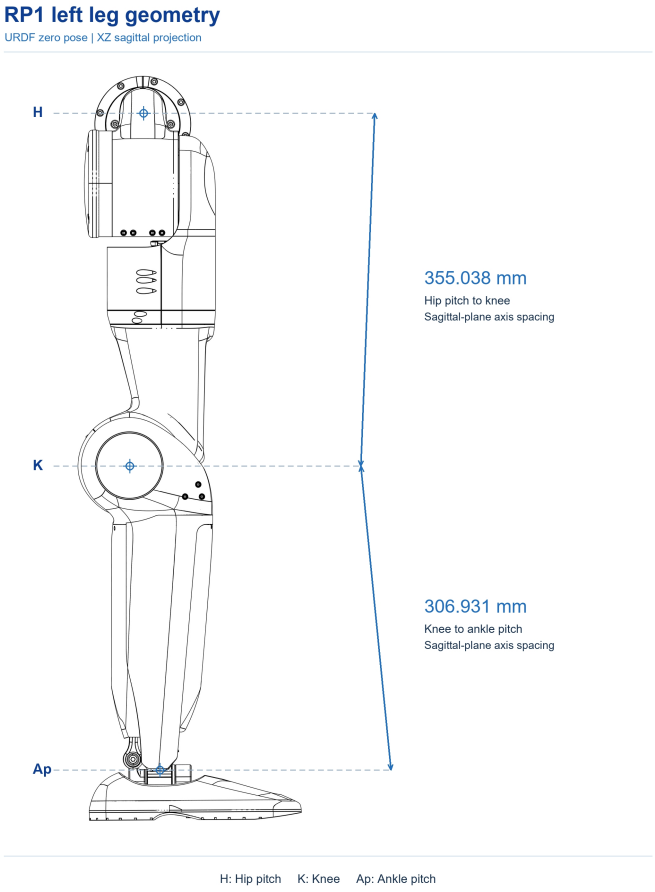
Coordinated hip, knee, and ankle motion

Each leg has three hip, one knee, and two ankle axes. Coordinated hip, knee, and ankle motion controls thigh orientation, leg extension, foot height, and placement.



RP1 knee flexion: 166.5 degrees, with full extension at zero. Human reference: [42].

Figure 1.16. Human and RP1 knee flexion. Human reference: [42]; RP1 limits and photographs: [21].

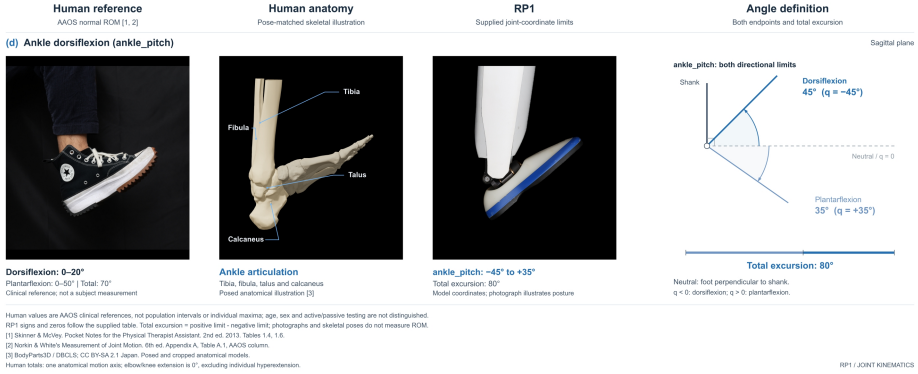


At the supplied URDF zero pose, the left-leg sagittal projection gives 355.038 mm from hip pitch to knee and 306.931 mm from knee to the functional ankle-pitch axis (Figure 1.17). The ankle reference represents the equivalent foot-pitch motion of the parallel mechanism. These dimensions apply to the illustrated configuration and projection.

Figure 1.17. Left-leg geometry and sagittal-plane axis spacing at the URDF zero pose. H: hip pitch; K: knee; Ap: functional ankle-pitch reference. Source: RoboParty [21].

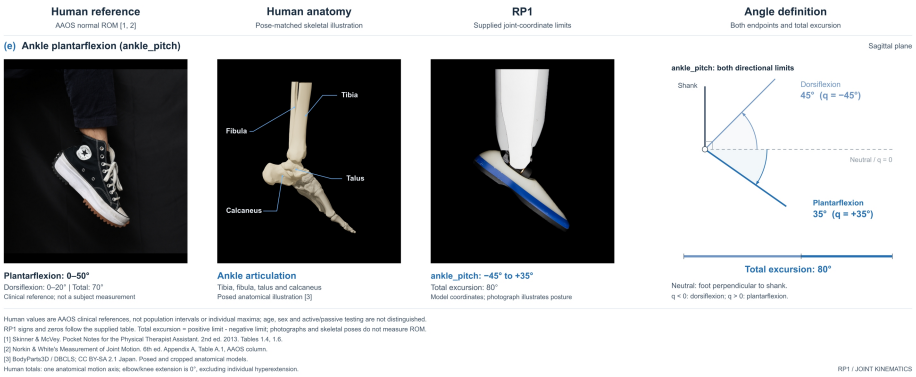
The RP1.3.4 parallel ankle connects two motors to the foot. Both motors contribute to pitch and roll; neither controls one equivalent axis alone.

Checks include combined pitch-roll motion, link swept volumes, remaining articulation, and shank-to-foot clearance.



RP1 ankle dorsiflexion: +45 degrees from a foot perpendicular to the shank. Human reference: [42].

Figure 1.18. Human and RP1 ankle dorsiflexion. Human reference: [42]; RP1 limits and photographs: [21].



RP1 ankle plantarflexion: -35 degrees from the same neutral pose. Human reference: [42].

Figure 1.19. Human and RP1 ankle plantarflexion. Human reference: [42]; RP1 limits and photographs: [21].

Motion clearance and service access

Clearance checks cover toes during steps, knee backs and ankle links during squats, and legs and hip shells during lateral motion. Local fasteners must remain accessible.

Clamp extraction paths are checked for obstructing covers and neighboring components.

1.5.4 Feet

Support area and swing clearance

Larger foot plates enlarge the geometric support region but increase collision and scuffing risks. RP1 uses AL7075 plates, sized with ankle positions and foot-placement paths.

Plan and side views check foot-edge separation and toe/heel height throughout swing. Ankle position changes these trajectories.

Actual contact area depends on surface shape, foot pose, and ground conditions; it is not necessarily the projected plate area.

Figure 1.20 separates the foot/ankle-pitch assembly in side view from the isolated foot in front view. Ground-reaction components act at C; equivalent section reactions and moments act at the ankle references. The illustrated reference heights are 63.92 mm for ankle pitch and 45.00 mm for ankle roll. Horizontal moment arms depend on the contact location and operating condition, so the diagram defines the load geometry without assigning a universal ankle load.

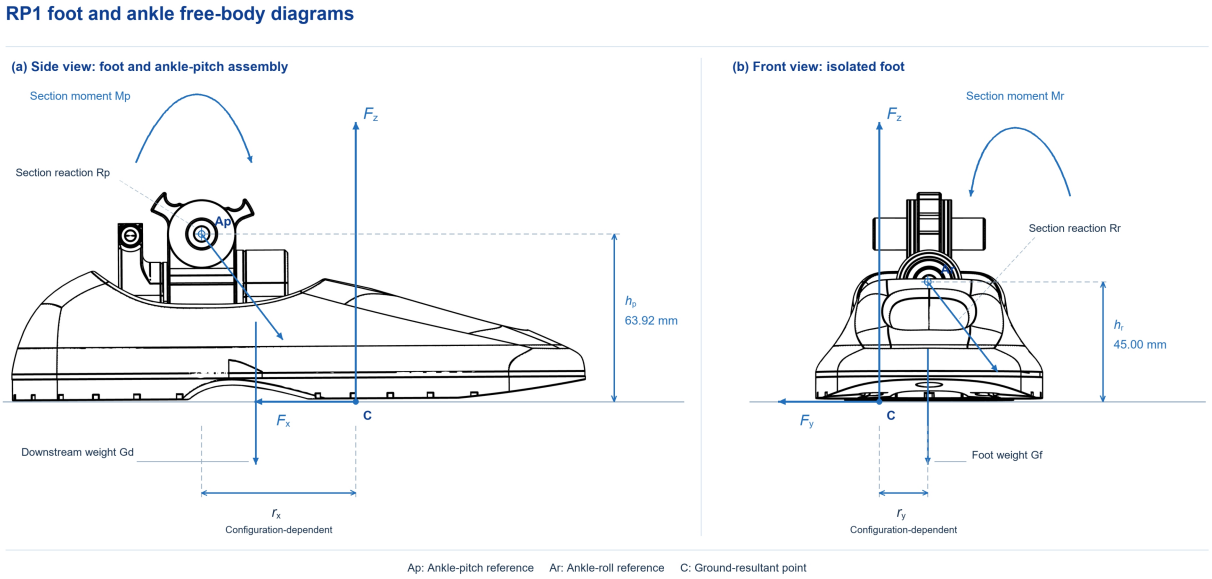


Figure 1.20. Foot and ankle free-body diagrams. (a) Side view of the foot and ankle-pitch assembly. (b) Front view of the isolated foot. Weight terms follow the corresponding free-body boundary; moment arms depend on the operating condition. Source: RoboParty [21].

1.6 Manufacturing Assembly and Service

Dense joints, wiring, and mixed components make assembly and service sensitive to dimensional errors and interface inconsistencies. Poor access increases commissioning time and fault-isolation effort.

RP1 combines design for manufacturability (DFM), design for assembly (DFA), and design for serviceability (DFS), using material tiers, actuator families, standardized interfaces, and planned removal paths.

DFM addresses cost and machining; DFA addresses assembly and connections; DFS addresses access, removal, and diagnosis. These decisions support the shared goals of versatility, efficiency, and reliability.

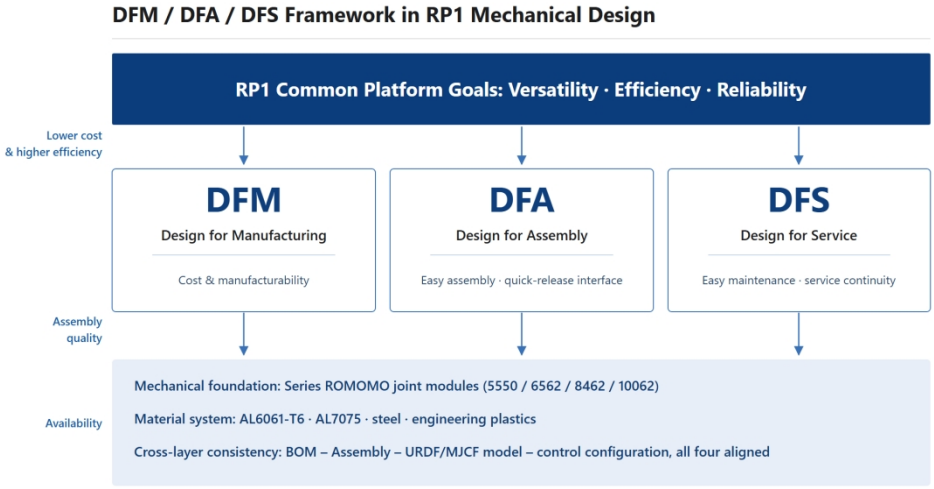


Figure 1.21. DFM, DFA, and DFS framework. Source: RoboParty [21].

1.6.1 Design for Manufacturability

DFM defines materials, machining datums, critical tolerances, and standard parts before assembly. Structural and lifetime requirements constrain reductions in cost and machining complexity.

Table 1.9. Material selection.

Material	Suggested use	Engineering controls	RP1 application
AL6061-T6	Primary structures and routine brackets	Check thin-wall distortion, temper, and critical hole tolerances; balance machining, cost, and strength.	Torso and limb frames
AL7075	Parts constrained by high load, strength, or mass	Select for load, fatigue, and environment; review supply condition, finish, joints, and machining cost.	Foot plates
Steel	Supports, stops, pins, and wear-loaded interfaces	Check contact stress, wear, corrosion, and galvanic compatibility with aluminum.	Joint stops and universal-joint parts
Engineering polymers	Covers, harness supports, insulation and separation	Select for temperature, insulation, flammability, creep, wear, and fastening.	Shells and electrical protection

Each component class has a default material. Upgrades require calculation, simulation, or test evidence and consideration of machining, finishing, fastening, and supply.

ROMOMO sizes 5757, 6562, 8462, and 10062 cover peak torques of 25-140 N m. All use 48 V power and CAN-FD; size follows joint location and load. The 75, 67, and 165 N m points read from the supplied motor T-n test curves are interpreted as follows: 75 N m is the doubled effective peak torque of the two-motor 6562 ankle parallel mechanism, whereas 67 N m and 165 N m are single-motor/equivalent-point values for 8462 and 10062, respectively.

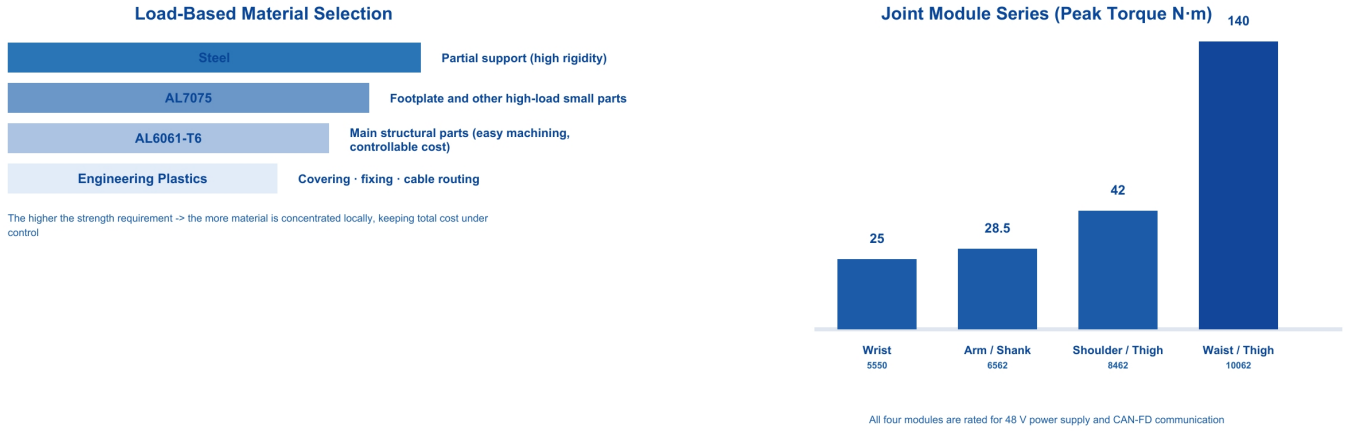


Figure 1.22. Material tiers and joint module families. Source: RoboParty [21].

Material tiers and module families reduce overspecification and repeated design. Cost benefits require current machining, material, component-count, and procurement data.

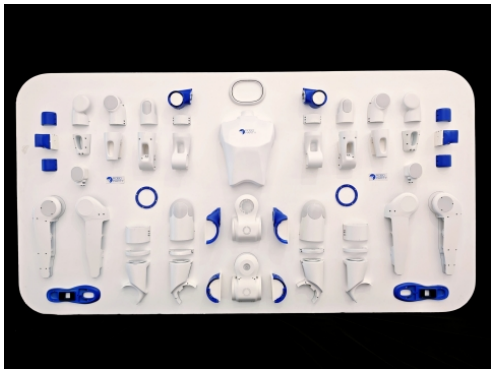


Figure 1.23. RP1 body components. Source: RoboParty [21].

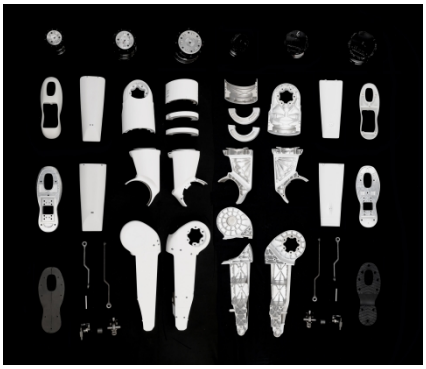


Figure 1.24. RP1 core structural components. Source: RoboParty [21].



Figure 1.25. Details of core components. Source: RoboParty [21].

1.6.2 Design for Assembly

Mechanical assembly

Shoulder and thigh roll interfaces combine a load-bearing body and a fastened split clamp. Defined support, clamp, and fastener interfaces make installation and removal paths explicit.

Harness integration

Hollow internal passages and separate limb harness zones limit exposed wiring. Defined paths, connectors, and fixation points reduce interference and establish service boundaries.

Harness planning supports consistent assembly, cable protection, and fault isolation.

Reducing unnecessary parts and connections, defining module mounting interfaces, and standardizing harness interfaces reduce reliance on temporary adjustment and operator experience. Assembly planning, therefore, treats structural modularity and cable routing as related design decisions.

DFA evaluation criteria include part and step counts, assembly time, connection errors, and rework; quantified RP1 results for these criteria are not reported here.

1.6.3 Design for Serviceability

Service boundaries follow end effectors, the head, and limb interfaces to limit disassembly.

The wrist accepts interchangeable tools (Figure 1.26). Tool changes also require updated mass properties and control settings.



Figure 1.26. RP1 wrist interface and three interchangeable end effectors. Source: RoboParty [21].



Figure 1.27. Removal of the RP1 head perception module. Source: RoboParty [21].

The removable head and perception module support alternative configurations (Figures 1.27-1.28). Custom heads require mechanical, electrical, communication, and control compatibility.

Shoulder and leg roll clamps permit limb removal after releasing fasteners and disconnecting interfaces (Figure 1.29).



Figure 1.28. Perception, data-collection, and motion-control head configurations. Source: RoboParty [21].

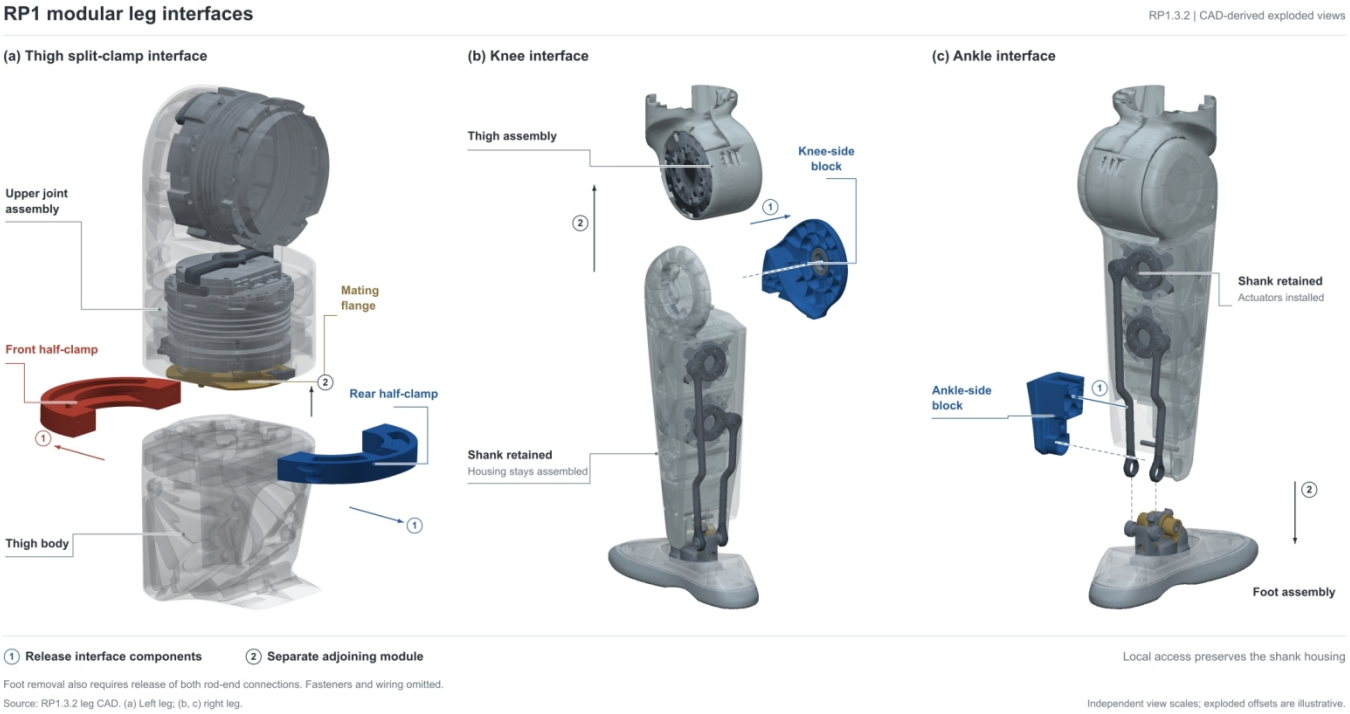
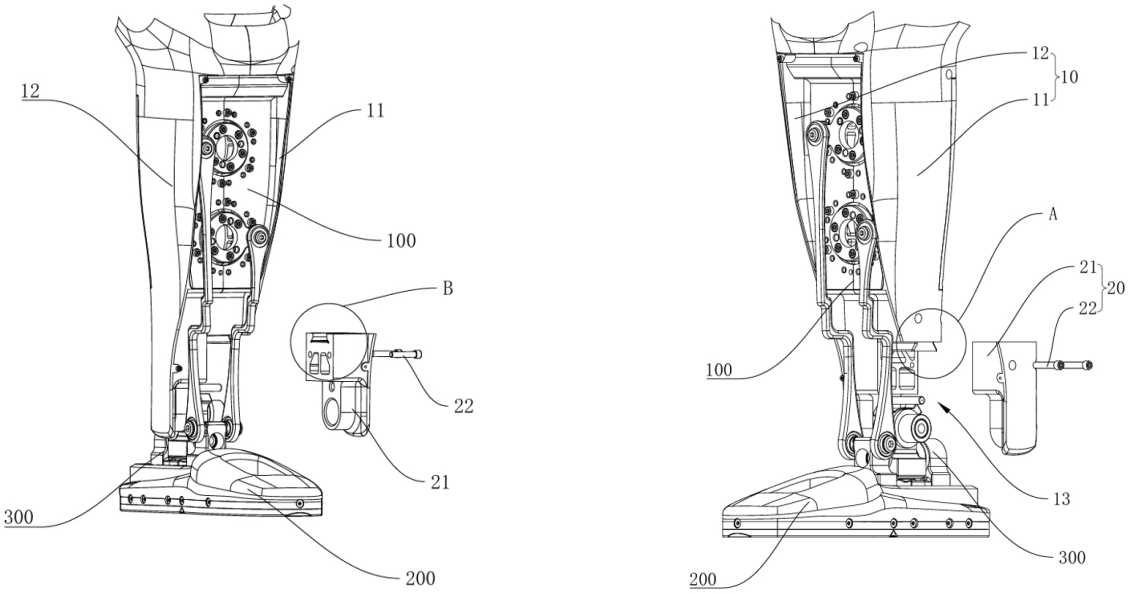


Figure 1.29. Selected quick-release interfaces in RP1 leg modules. Source: RoboParty [21].



Patent drawings from the mechanical technical disclosure entitled "A Separable Fixation Method for the Shank and Ankle of a Humanoid Robot."

1.7 From Mechanical Assembly to URDF

1.7.1 Model Overview

RP1_description contains URDF, MJCF, and meshes for kinematics, dynamics, simulation, and control [7]. URDF records links, joints, frames, geometry, mass, and inertia.

Links, joint axes, signs, limits, and zero poses follow the physical mechanism. Common datums keep CAD, URDF, and hardware definitions consistent.

1.7.2 Kinematic Structure and Joint Configuration

Links group rigidly connected parts; joints define relative position, motion direction, and range.

The waist-hip base branches to the torso, head, arms, and legs. Fixed interfaces use fixed joints; moving interfaces use physical axes and limits. Configurations reuse names and hierarchy where possible, changing only affected modules.

A useful simulation model requires both physical consistency and a manageable level of geometric detail. CAD describes parts, machining features, and assembly; URDF organizes rigid-body motion, joint constraints, and inertial properties for software use. The conversion therefore requires explicit modeling decisions before export.

1.7.3 SolidWorks to URDF

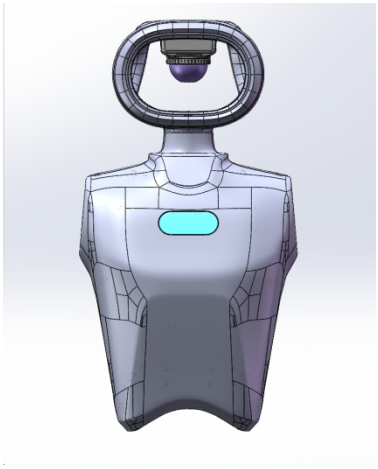


Figure 1.30. Head-torso link. Source: RoboParty [21].

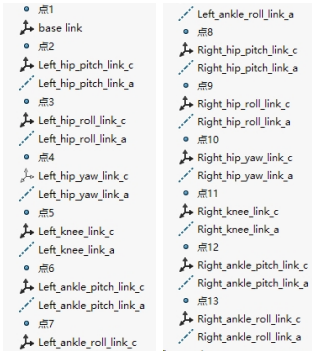


Figure 1.31. Link hierarchy. Source: RoboParty [21].

Before export, group SolidWorks parts by rigid-body motion. Shells, batteries, and fasteners can share a link if they remain fixed relative to one another. Figure 1.30 shows the head-torso example.

Reference frames define poses and datum axes define rotation. Left and right references must follow installation directions to preserve joint signs after mirroring.

The parent-child tree defines each joint's type, axis, and limits (Figure 1.31). Fixed parts retain fixed connections.

SolidWorks URDF Exporter [8, 19] converts link groups, frames, axes, and mass properties into URDF and exports visual and collision meshes.

The export sequence is: organize the SolidWorks assembly; define rigid links, joint frames, and axes; run SolidWorks URDF Exporter; generate URDF and accompanying meshes; then inspect the model in visualization, simulation, and control tools. An unprepared full CAD assembly is not automatically a valid robot model [8, 19].

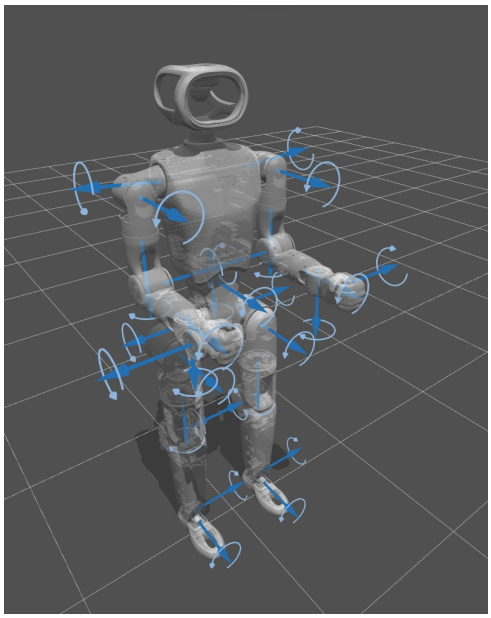


Figure 1.32. Principal joint axes and rotation directions. Red arrows indicate axis directions; green arrows indicate the corresponding rotations. The illustration supports checking joint conventions against the mechanical assembly. Source: RoboParty [21].

For a joint with a fixed flange and a rotating housing, the positive joint direction must describe the housing motion relative to the flange. Under the source workflow, the local joint Z axis is chosen accordingly, rather than copied from the motor output-shaft convention. Left and right joints require separate checks against the actual installation, zero position, and intended positive motion. Figure 1.32 visualizes these axis and rotation directions.

1.7.4 Model Fidelity and CAD Preparation

Use stable reference geometry for joint centers, axes, and interfaces so local CAD edits do not invalidate frames. Group rigid parts while retaining every relative-motion interface.

The RP1 URDF mass values are based on physical weighing. Record the weighed configuration, included components, measurement method, and uncertainty alongside the model. Wiring, connectors, and other installed components must be accounted for consistently. Link mass, center of mass, and inertia must describe the same configuration; measured mass alone does not establish the center of mass or inertia tensor.

Visual meshes can retain detail. Collision meshes should preserve contact boundaries while omitting internal parts, fasteners, holes, and small features that increase simulation cost.

1.7.5 Collision Mesh Simplification

Collision meshes are used for contact and self-collision calculations. Detailed CAD exports can contain tens or hundreds of thousands of triangles in a single link, increasing model-loading time, memory use, and collision-processing cost. Some simulation back ends also impose mesh-size limits. The source workflow uses two complementary simplification stages.

Simplification before CAD export

Before export, suppress motor and reducer internals, internal slots and holes, threads, small chamfers, fasteners, and parts fully enclosed by a shell when they do not affect external contact. An envelope or simple geometric proxy can represent an internal component that does not participate in collision. The chest-shell example in Figure 1.33 removes small surface features while retaining the main structural envelope.

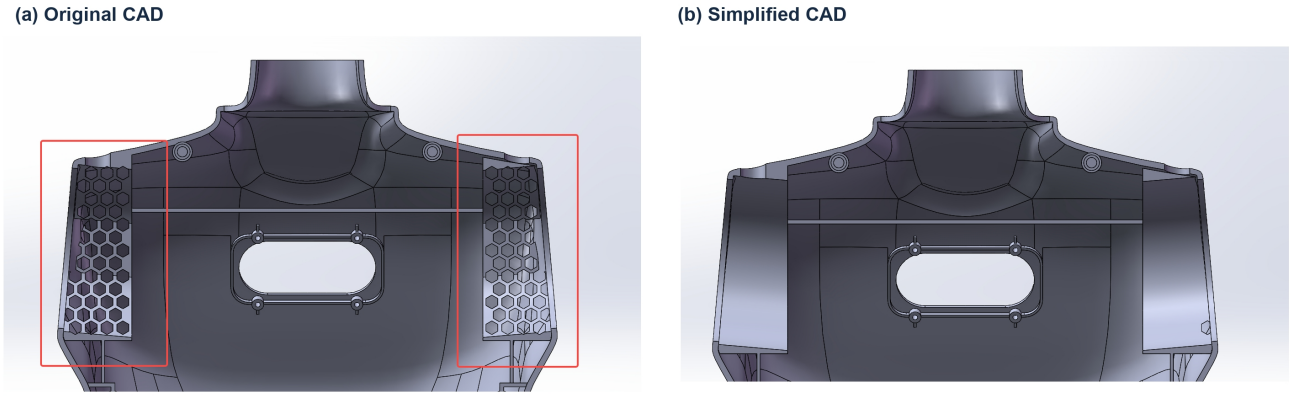


Figure 1.33. Chest-shell geometry before and after CAD simplification. (a) Original model, with detailed regions highlighted. (b) Simplified model for export. Source: RoboParty [21].

CAD-level simplification allows important contours and contact regions to be preserved selectively. It requires additional model preparation, however, and the simplified assembly must be maintained when the mechanical design changes.

Decimation after mesh export

For an existing URDF, STL or OBJ meshes can be simplified without changing the original CAD. Reduction may be specified by triangle-count ratio, target face count, or geometric error. Edge-collapse methods and quadric error metrics (QEM) reduce vertices and faces while approximating the original surface; convex hulls or simple primitives may be suitable for geometrically regular components.

Mesh-level processing is convenient when CAD editing is unavailable, but a uniform reduction ratio does not distinguish critical contact regions from unimportant details. Foot edges, joint clearances, and boundaries between adjacent links need local protection. Select target face counts or error limits separately for each link, according to geometric complexity and collision relevance.

Excessive reduction can cause surface collapse, distorted edges, holes, open or non-manifold geometry, and inconsistent normals. Figure 1.34 shows visible surface damage in the head-torso region after aggressive decimation. Inspect these defects and the retained contact envelope before using the mesh for collision calculations. Simplification should remove unnecessary detail while preserving the main dimensions and motion relationships.

1.7.6 Synchronized CAD and URDF Maintenance

CAD and URDF must refer to the same mechanical configuration. A change in link dimensions requires a new check of joint origins, frames, and collision boundaries; a change in material, wall thickness, or internal components requires updated link mass, center of mass, and inertia. A change in installation direction also requires checking joint signs, zero poses, and limits.

Replacement heads, hands, and computing modules can change mass and envelope even when their mounts remain unchanged. Update the visual and collision meshes, inertial properties, affected links and joints,

device configuration, and control interfaces together. When a module adds or removes a degree of freedom, revise the kinematic tree and control mapping accordingly.

1.7.7 Modular Configurations

Both body configurations retain two 6-DoF legs and a 2-DoF waist. Arms have 5 or 7 DoF; hand DoF remain separate.

Adding or removing modules requires synchronized links, joints, meshes, and dynamics to keep the model consistent with hardware.

Across configurations, reuse link and joint names and parent-child relationships wherever the hardware is unchanged. The resulting URDF and meshes retain the structural correspondence with CAD and the physical robot while supporting model visualization, kinematic calculations, and collision checking [7].

2 Electrical System

RP1 distributes power and communication across actuator, computing, and sensor modules. Protection and backup paths define behavior during battery replacement and abnormal conditions.

2.1 System Architecture

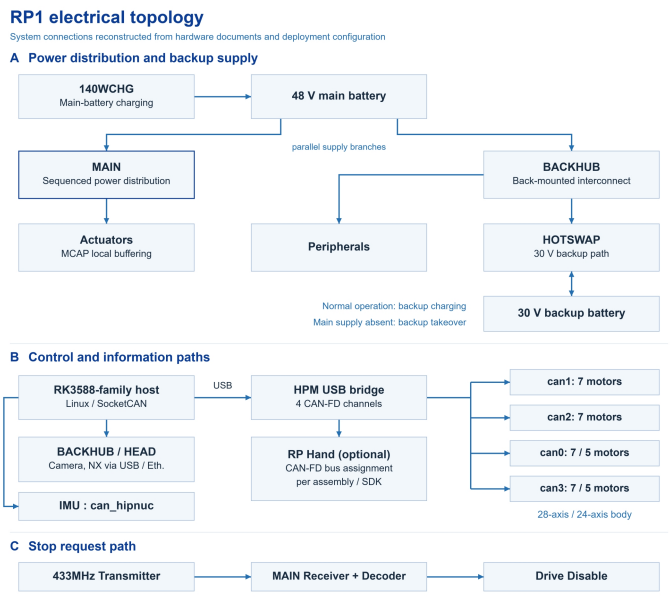


Figure 2.1. RP1 electrical topology: power, communication, and stop paths. Source: RoboParty [21].

Power paths connect the battery, protection, distribution, computers, and actuators. Signal paths connect the host, bridges, sensors, and drives.

Separating power and signal paths limits coupling through shared grounding and supply transients and supports independent current, thermal, impedance, and timing checks. Both paths must still be evaluated under the same task loads.

The battery feeds parallel MAIN and BACKHUB branches. MAIN distributes actuator and peripheral power; BACKHUB connects the rear modules and HOTSWAP backup. Four USB-to-CAN-FD segments serve the actuators. USB/FPC and Ethernet connect the controller and backplane; cameras, hands, and extension computers vary by configuration. Figure 2.1 separates power, communication, and stop paths.

2.2 Board Roles and Hardware Implementation

Power and communication are distributed among independently manufacturable and replaceable boards (Table 2.1).

Separate boards simplify high-current and signal routing, manufacturing, validation, replacement, and 24/28-axis configuration.

Table 2.1. Functions of the principal boards.

Module	System role
MAIN / BACKHUB	Application computing, distribution, and rear expansion
HOTSWAP	Backup power and transfer
MCAP	Local actuator energy storage
FAN	Fan power and control

Module	System role
IMU / HEAD	Orientation sensing and head peripherals
HANDPlusPro / FPC	Hand connection and controller-backplane interconnect
433TX / indicators	Operator inputs and status
MINGMEI-BC / 140WCHG	Battery connection and charging

Figures 2.2-2.3 show MAIN, BACKHUB, HOTSWAP, HANDPlusPro, and MCAP layouts. Bottom views are mirrored; colors denote fabrication layers.

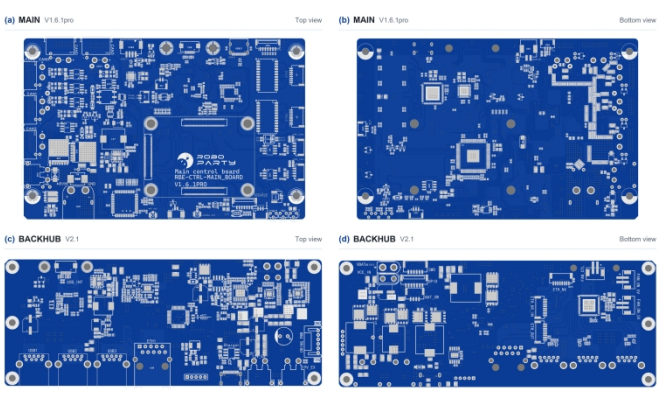


Figure 2.2. Top and bottom Gerber renderings of MAIN and BACKHUB, not to a common scale. Source: RoboParty [21].

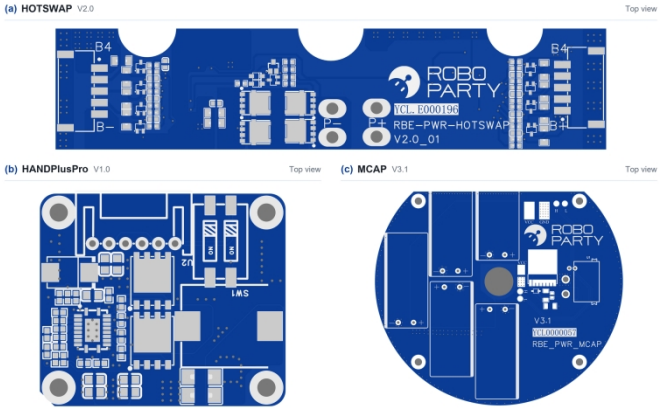


Figure 2.3. Top-layer Gerber renderings of HOTSWAP, HANDPlusPro, and MCAP. Source: RoboParty [21].

2.2.1 PCB Power and Signal Layers

MAIN and BACKHUB use six layers, HOTSWAP and HANDPlusPro four, and MCAP two. Power layers carry current and spread heat; signal layers meet impedance targets. For the nominal 50-ohm single-ended trace, the design calculation uses FR4 epsilon_r approximately 4.2, a 0.2 mm dielectric thickness, a 0.3 mm trace width, and 0.035 mm copper thickness, giving approximately 54 ohms. The approximately 8% difference from the 50-ohm target reflects the microstrip approximation; final compliance remains subject to simulation and impedance testing.

Current capacity and signal quality are estimated during design. Using IPC-2221 calculations [18] with a 10-degree-Celsius allowable temperature rise, a 20 mil external trace is estimated at approximately 1.46 A, and a 55 mil trace at 3.05 A externally or 1.52 A internally. These estimates identify rails requiring wider traces or additional copper. A microstrip approximation gives approximately 54 ohms for a nominal 50-ohm single-ended trace on FR4. Stack-up, width, and spacing are adjusted for high-speed interfaces and then checked by simulation and impedance testing. Target values are 120 ohms differential for CAN-FD, 90 ohms for USB, 100 ohms for Ethernet, and 50 ohms for critical single-ended signals. These are design targets for controlling reflections and crosstalk, not substitutes for measurement.

Table 2.2. Target impedances.

Interface	Target impedance	Application
CAN-FD differential pair	120 ohms differential	Actuator segments, including termination
USB 2.0/3.0 pair	90 ohms differential	Host-bridge and expansion
Gigabit Ethernet pair	100 ohms differential	ETH1/ETH2, 1000BASE-T
Critical single-ended signals	50 ohms	Clock and debug signals

2.2.2 Computing and Perception

An RK3588-series host with 8 GB RAM and 32 GB eMMC provides the base runtime on MAIN. It aligns 24/28-axis feedback, IMU, and camera data, runs actuator middleware, sends segmented commands, and supports network debugging and logging.

The host handles motion control; extension computers handle demanding vision and point-cloud workloads. Upgrades must meet mounting, power, and airflow constraints.

Body and camera IMUs have separate extrinsics and time sources. High-rate orientation and angular velocity support balance; configuration-dependent cameras support environmental and manipulation tasks. Bandwidth, power, and data age are budgeted with actuator traffic.

2.2.3 Energy Storage Across Operating Conditions

Running, flips, and loaded manipulation have different power demands. Time-resolved whole-robot current samples form power distributions from which peak power, duration, and energy deficits inform local storage and protection. Each leg has 5,000 microfarads of MCAP capacitance, totaling 10,000 microfarads. At 48 V, stored energy is approximately 11.5 J; allowing the bus to fall to 40 V yields approximately 3.5 J of usable energy. This corresponds to approximately 7 ms for a 500 W transient deficit. The MCAP hold-time, inrush, and soft-start values in this subsection are calculated values; no measured waveforms or test conditions are included in the present evidence.

Local capacitance buffers short actuator peaks and reduces upstream transients. Actual bus-voltage margins require measurement under the stated load and voltage conditions.

Inrush is a second concern. With an initially discharged capacitor and a 50 milliohm loop resistance, a simplified 48 V calculation gives 960 A. A soft-start circuit therefore controls the voltage rise through a MOSFET-gate RC network. An equivalent resistance of approximately 198 kilohms and capacitance of 100 nF give a time constant near 20 ms, spreading the rise over tens of milliseconds and limiting inrush.

2.3 Bus Segmentation and Communication Layers

Four CAN-FD segments serve the actuators. CAN-FD defines link transmission [16], vendor protocols define commands and feedback, and SocketCAN provides Linux access. Middleware maps these layers to common joint semantics.

The 28-axis configuration connects seven motors to each of CAN1, CAN2, CAN0, and CAN3. The 24-axis configuration uses 7/7/5/5 motors.

At 1 kHz, illustrative 8-byte commands and 8-byte feedback per motor produce about 0.9 Mbit/s for seven motors and 3.6 Mbit/s for 28. These are payload rates, not full-link occupancy. Arbitration runs at 1 Mbit/s and data at 5 Mbit/s; framing and phase timing must also be counted. Segmentation reduces traffic and localizes faults, but whole-body motion still requires coordinated fault handling. Actual margin depends on scheduling and captured traffic.

Table 2.3 lists nominal interface rates, data, and constraints. Nominal bandwidth is not measured utilization.

Table 2.3. Interfaces and nominal rates.

Interface	Protocol	Nominal rate	Traffic / use
CAN-FD, CAN0-CAN3	ISO 11898-1 CAN-FD	Arbitration 1 Mbit/s; data 5 Mbit/s in the current configuration (CAN-FD maximum 8 Mbit/s)	Actuator commands and feedback, 24/28 axes
USB host-bridge/peripherals	USB 2.0 / 3.0	480 Mbit/s / 5 Gbit/s	Bridge uplink, cameras, and debugging
ETH1/ETH2	1000BASE-T	1 Gbit/s	Remote debugging, compute expansion, and data
Type-C charging/debugging	USB 2.0 / PD	Data per protocol; 140 W charger rating	Charging and SoC debugging
Controller-backplane FPC/USB	Parallel signals / USB	Interface-dependent	Board interconnect and rear expansion
433 MHz stop path	433 MHz RF	kbit/s range	Stop requests and remote confirmation
Wi-Fi	802.11ax, AP6275S	Nominal 1,200 Mbit/s, 2T2R	Wireless debugging and data
Bluetooth	BT 5.3	Approximately 2 Mbit/s, LE	Peripheral connection

Figure 2.4 identifies external power, Type-C charging, two Ethernet ports, and three USB ports.

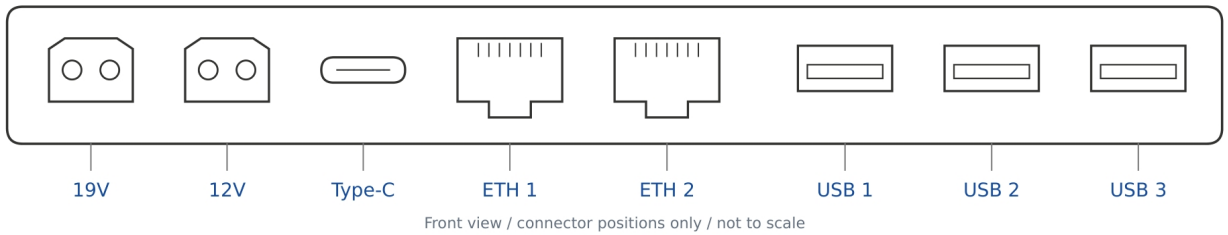


Figure 2.4. Front-view line drawing of RP1 external interfaces. Source: RoboParty [21].

2.4 Hot Swapping Protection and Abnormal Response

2.4.1 Hot-Swap Logic

The main battery is a 13S2P ternary-lithium pack with a nominal voltage of 46.8 V, a 54.6 V charge limit, a 39 V discharge cutoff, a typical capacity of 10 Ah (minimum 9.8 Ah), and approximately 500 Wh rated energy. Continuous discharge is 30 A at 25 deg C; peak discharge is 80 A for no more than 1 s, with overcurrent thresholds of 50 A/3 s, 80 A/1 s, and 140 A/250 ms. The 30 V backup is an 8S1P pouch pack with an expected 30.4 V long-term voltage and approximately 450 mAh (approximately 15 Wh). HOTSWAP and BACKHUB retain critical circuits during battery exchange and provide the backup path.

Reconnection is intended to restore main power without a new soft-start sequence or load-visible interruption. This requires validation at the applicable load.

The rear battery can be released, removed, inserted, and reseated while the robot stands (Figure 2.5); the rear handle provides support.

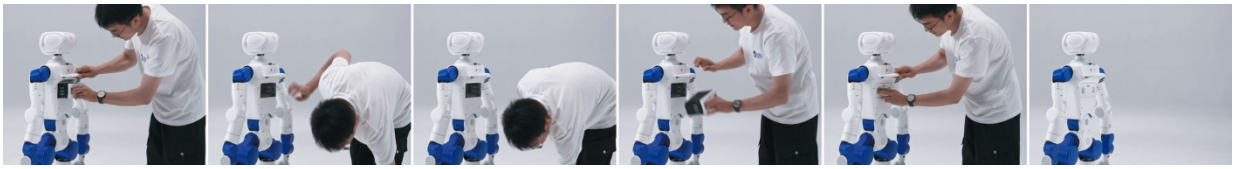


Figure 2.5. RP1 main-battery removal and replacement. Source: RoboParty [21].

During normal operation, the main battery supplies MAIN and BACKHUB while HOTSWAP monitors the bus voltage. When the bus remains below threshold for t_debounce after battery removal, the 30 V backup connects through BACKHUB to maintain temporary whole-body supply. After a new battery is inserted and confirmed, the system returns to the main battery and charges the backup. The ideal-diode arrangement prevents reverse feed; measured observation showed almost no reverse current when the large battery was inserted.

The 120 s value is a design target and budget value; the available test record reports more than 120 s. At an estimated 100 W critical standing load, 120 s requires 12,000 J of usable backup energy, equivalent to approximately 0.111 Ah at 30 V before conversion losses. Design margin, actual load, and available backup capacity are established through whole-robot measurements.

2.4.2 Stop Inputs and State Feedback

Stop functions operate through three paths: wired and 433 MHz paths handled by the MCU, and software stop confirmation at the remote-control interface. All three paths ultimately assert Drive Disable, which cuts motor-drive enable; trigger, transport, and recovery behavior are evaluated separately.

Software, 433 MHz, and wired paths address normal operation, remote operation, and host/communication failure. Redundancy requires fault testing. Record request receipt, drive-enable change, and mechanical stop separately to identify delay or loss.

Figure 2.6 shows the body button, remote control, and software interface. The example panel displays stop/reset state and motor temperatures from device feedback.



Figure 2.6. RP1 stop inputs and example software interface. Source: RoboParty [21].

2.4.3 Stopping Distance and Braking Budget

Stopping distance combines travel during response latency with braking distance. At an initial 1.0 m/s, 0.1 s response time, and 2.0 m/s squared deceleration, the illustrative distance is 0.35 m; at 0.3 s response time it is 0.55 m. The response budget separates signal transport, 433 MHz module processing and enable removal, drive/brake response, and mechanical play. The wireless path includes RF transport and module processing and is treated as the longer-path estimate; the wired response is expected to be shorter.

Actual stopping depends on friction, support state, and available torque. These illustrative calculations are not measured stopping performance and do not establish a validated separation distance.

3 Embedded Software

Embedded software maps distributed devices to common motion interfaces through protocol layers, device abstractions, and operating records.

3.1 Architecture and Communication

3.1.1 Execution Architecture

Four layers connect policy to hardware: applications generate joint targets; drivers encode protocols; Linux uses SocketCAN; USB bridges forward CAN-FD traffic. Mapped feedback returns as policy observations.

The packaged configuration uses 250 Hz control and 50 Hz policy updates. Real-time scheduling and CPU isolation reduce interference. These configured rates are distinct from the illustrative 500 Hz communication budget below and the 1 kHz payload calculation in Section 2.3; bandwidth and latency require separate assessment.

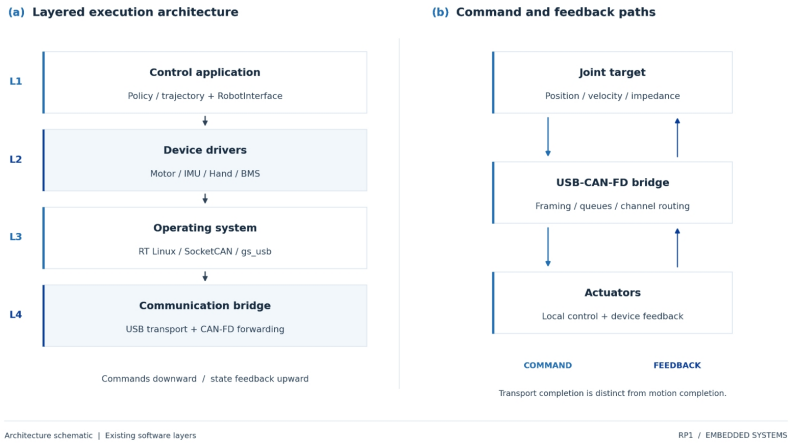


Figure 3.1. Embedded execution architecture. (a) Control, driver, operating-system, and bridge layers. (b) Joint-command and actuator-feedback paths. Source: RoboParty [21].

3.1.2 Communication Budget

Communication is budgeted separately for the control cycle, CAN bus, and USB transport. Consider seven motors on one bus at 500 Hz, giving a 2 ms period. A batched command occupies one 64-byte frame per cycle, and each motor returns one 12-byte feedback frame. This gives 500 command and 3,500 feedback frames per second, with an application-payload rate of

$$B_{CAN} = (64 + 7 \times 12) \times 500 = 74{,}000\, B/s = 74\, kB/s$$

This rate includes application data only. CAN-FD separates arbitration and data phases, configured here at 1 and 5 Mbit/s. At 5 Mbit/s, the payload bits of a 64-byte command take 102.4 microseconds and those of a 12-byte feedback frame take 19.2 microseconds. Payload occupancy per cycle is therefore

$$T_{payload} = 102.4 + 7 \times 19.2 = 236.8\, \mu s$$

$$U_{payload} = 236.8 \,/\, 2{,}000 = 11.84\%$$

11.84% is a payload-only lower bound. Full occupancy includes arbitration, control, checksums, bit stuffing, gaps, other devices, and retransmissions at their respective rates. Acceptance requires actual frame formats and captures.

USB accounting uses encapsulated transfer frames. The current implementation allocates a 12-byte header and 64-byte data region per CAN-FD frame, so even 12-byte feedback uses 76 bytes. Each cycle contains one command, seven feedback frames, and one command echo, giving

$$B_{USB} = (1 + 7 + 1) \times 76 \times 500 = 342{,}000\, B/s = 342\, kB/s$$

The USB payload rate is 2.736 Mbit/s, about 0.57% of the nominal 480 Mbit/s USB High-Speed [17], before transaction overhead. Sum USB traffic across buses; assess CAN occupancy per bus. Echo releases a transmission context but does not confirm motion completion.

Beyond bandwidth, the control period constrains scheduling, queuing, forwarding, and device response. For serial processing stages, the time budget is

$$T_{sched} + T_{queue} + T_{USB} + T_{bridge} + T_{CAN} + T_{device} + T_{sw} \leq T$$

Budget terms follow the actual command-feedback path. Buffered commands also age: 100 commands at 500 Hz represent 200 ms of history. Measure cycle distributions, backlog, and feedback age rather than treating queue capacity as margin.

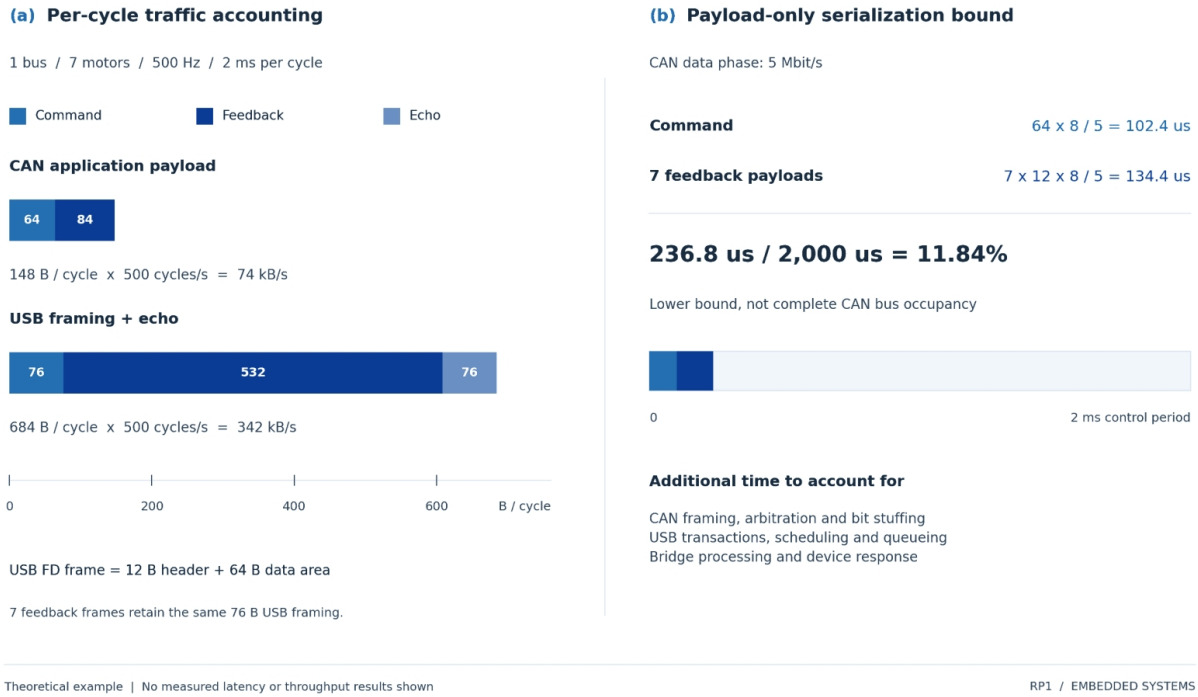


Figure 3.2. Theoretical CAN/USB traffic and payload timing. The 11.84% value excludes framing and other delays; the unused area is not a validated margin. Source: RoboParty [21].

3.2 Hardware Abstraction and System Support

3.2.1 Hardware Abstraction Layer

Common interfaces and factories select motors, IMUs, BMS devices, and hands. Transport, device protocol, and robot mapping remain separate.

Table 3.1. Hardware abstractions.

Layer	Implementation	Provided capability
Device	MotorDriver, IMUDriver, BmsDriver, HandDriver	Motor commands/feedback, inertial measurements, battery state, and hands
Communication	CAN / CAN-FD transport abstractions	Frame transmission, receive callbacks, and backend selection
Mechanism	Decouple and mechanism-specific implementations	Forward and inverse mappings for parallel-mechanism position, velocity, and torque
Robot	RobotInterface	Joint order, direction, zero offsets, IMU extrinsics, and mechanism composition

IMU factories support HIPNUC and MCT7123; BMS factories support several protocols. Parallel mechanisms use a common interface with mechanism-specific solvers. Device changes affect backends and settings; body changes also affect joint sets, models, and policies.

3.2.2 Robot Board Support Package

The Armbian-based BSP configures a real-time kernel, CPU isolation, interrupt affinity, and scheduling permissions, together with drivers, device trees, and startup services.

BSP patches include gs_usb receive-request resubmission after recoverable errors and exposed error counters. Driver and kernel changes share the system-image build.

Board families include RoboPi2, RDK X5, Orange Pi, Radxa, and NanoPi. Configurations, SoC rules, and patches capture board differences; services and packages share delivery infrastructure.

3.2.3 Open Builds and OTA Delivery

GitHub Actions performs version checks and Debian-package builds. Artifacts pass through Releases and repository ingestion to APT delivery; robopi-config provides an operating interface. Drivers, tools, and applications can update separately.

Published source, build scripts, logs, and artifacts allow version comparison and rebuilding. Image, firmware, package, and robot-configuration versions are recorded together.

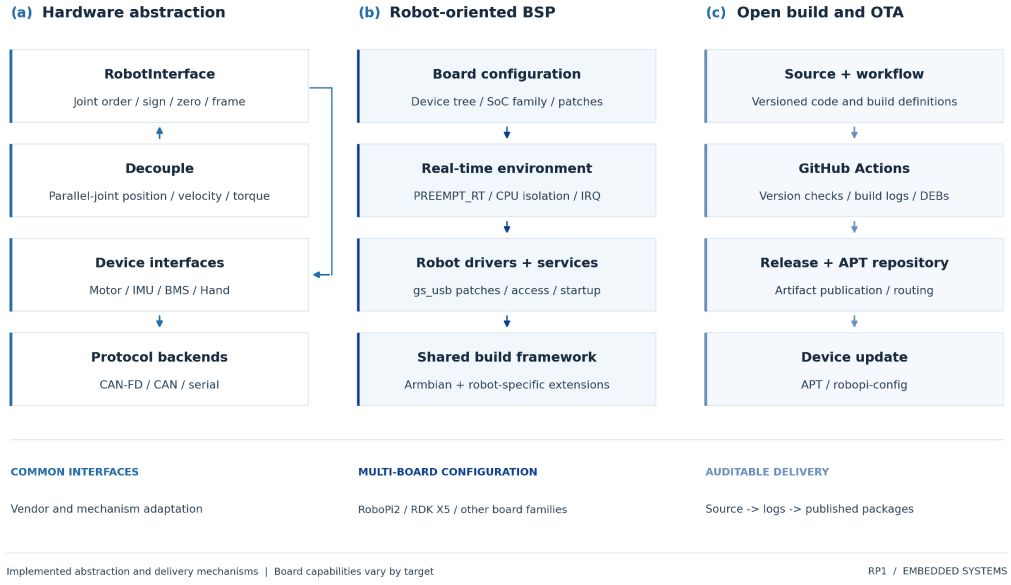


Figure 3.3. Hardware interfaces and parallel-joint transforms; robot BSP configuration; GitHub Actions and APT delivery. Source: RoboParty [21].

3.3 Logging and Stress Testing

3.3.1 Multidimensional Logs and Fault Isolation

robopi-analyze collects eight record classes: power and BMS, CAN interface state, kernel logs, bridge firmware logs, raw USB traffic, CAN frames, inference output, and temperature/thermal state. Rotation and snapshots retain operating windows. Offline tools generate timelines and analyze CAN periods, USB completion, echo pairing, and error intervals.

CAN state locates bus errors; USB captures locate stalls; firmware logs record queues and drops; power and thermal logs explain restarts and performance changes. Correlate these records by time.

Session export packages raw data, metadata, and analysis with file and archive SHA-256 checksums for offline reproduction and file-integrity verification; these checksums do not by themselves validate measurement accuracy.

CAN records retain raw and ASC formats. ASC tools summarize periods, jitter, and gaps by channel and message ID; DBC-compatible tools decode signals to relate timing to commands and errors.

3.3.2 Existing Checks and Test Scope

robopi-config exposes communication stress and hardware checks for bus traffic, motor commands, USB re-enumeration, and peripherals. Targets and criteria appear below.

Table 3.2. Tests and checks.

Test	Method	Records and checks
CAN communication stress	Continuous CAN/CAN-FD transmission on a paired bench; default 4 h	Send success/failure, receive counts, interface errors, bus-off and recovery; feedback gaps and backlog as load rises
Motor communication stress	Repeated batched MIT commands; default 4 h	Submission, feedback, errors, and interface state; sustained communication, not mechanical endurance
Motor chain and loaded operation	Sequential or continuous batched commands on a fixed bench with specified loads	Actual frame rate, communication, and loaded behavior; separately record load, pose, and duration
USB recovery	Repeated hub reset; default 100 cycles; observe re-enumeration	Device counts, enumeration and kernel errors; validate service recovery through device communication
BMS communication	Check packages, settings, ports, and state after service restart	Startup, successful reads, and timeouts; no battery-control or parameter-write commands
Peripherals	Head queries, expansion indicators, GPIO/SIG checks	Responses, outputs, and wiring; functional checks do not replace stress tests
Drivers and virtual bus	Mock transport and vcan for filtering, callbacks, shutdown, and errors	Receive ownership, resource release, error handling, and concurrency; not physical bandwidth

Default durations and loop counts are script settings, not completed test results. Current CAN receive counts do not establish message integrity. Motor scripts include zero calibration, and error aggregation and shutdown cleanup need improvement; they are not yet routine acceptance procedures. Unified tests require sequence/content checks, separated calibration, and defined final device states.

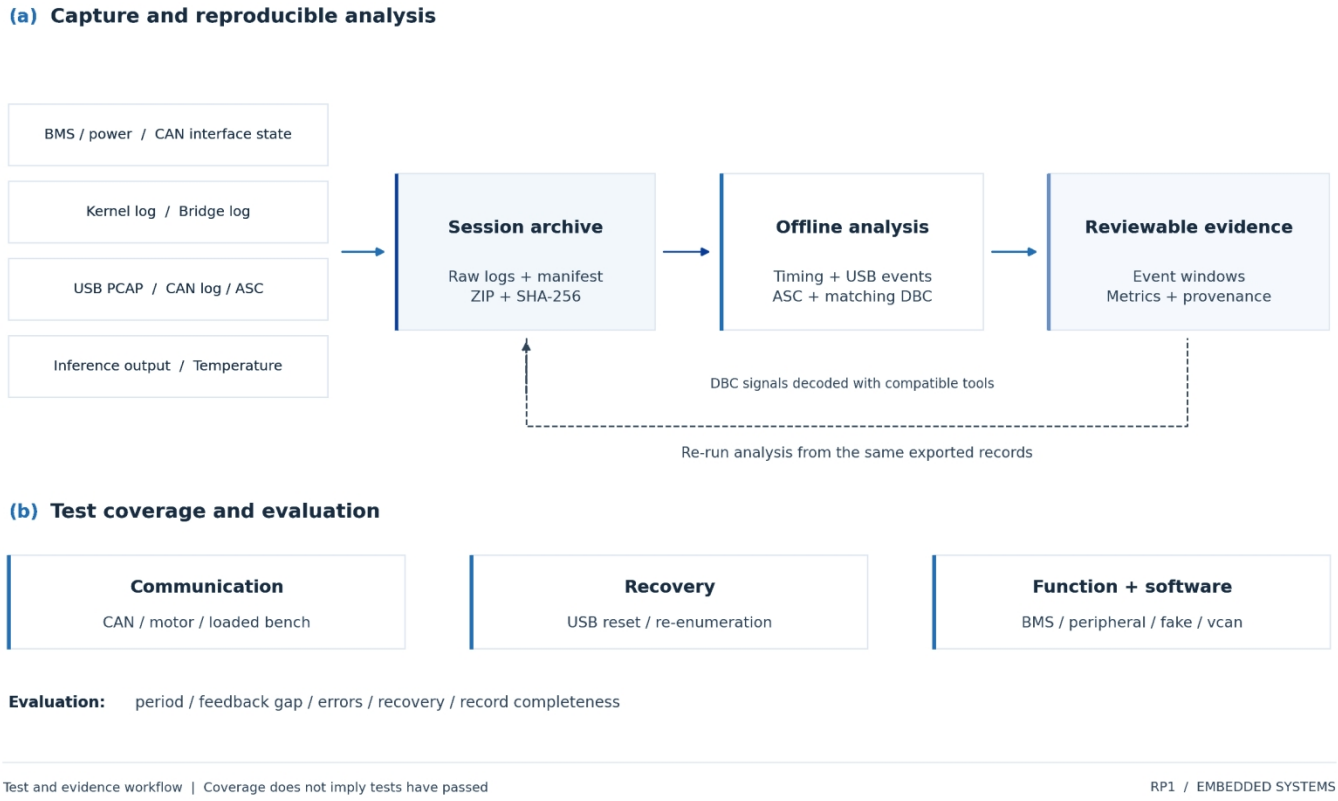


Figure 3.4. Session archives, checksums, and offline analysis; communication, recovery, and functional checks. Dashed paths denote reanalysis. Coverage does not imply acceptance. Source: RoboParty [21].

3.3.3 Concurrent Loads and Metrics

System stress tests combine inference, cameras, logging, and storage, comparing control periods, feedback age, CPU use, and temperature under idle and concurrent load. Scheduling tests measure thread wake-up latency; end-to-end tests track command submission, bus transport, and feedback. These are distinct evaluations.

Fault cases include missing feedback, USB disconnection and re-enumeration, bus errors, and service restarts. Detection, stopping, and communication recovery times are recorded separately, together with device mapping and re-enable conditions. Restored communication alone does not establish readiness to resume motion.

Record devices, wiring, load, frequency, software version, actual cycles, 99th-percentile latency, longest stall, deadline misses, and error/drop counts. Incomplete records do not pass. Logging tools alone do not establish concurrent-load or fault acceptance.

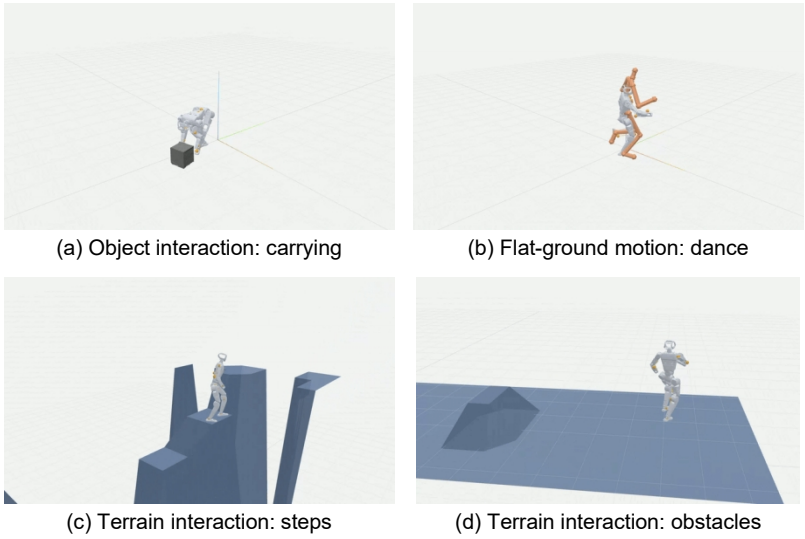
3.4 Interaction and Remote Control

Remote controls, robot services, and configuration tools submit commands, query states, and run maintenance. Receipt and execution results are reported separately, with device status and errors.

4 Algorithms

Section 4.1 describes the human-humanoid-tools workflow [9]; Sections 4.2-4.4 address training, deployment, and whole-body integration. Default settings are implementation-specific; RP1 tuning examples derive from development experiments recorded with that implementation.

4.1 hhtools Motion Retargeting



(a) Object interaction: carrying (b) Flat-ground motion: dance

(c) Terrain interaction: steps (d) Terrain interaction: obstacles

Figure 4.1. hhtools retargeting: carrying, dance, steps, and obstacles. Source: hhtools [9, 38].

Retargeting adapts human motion to RP1 proportions, waist roll/yaw, separated hip axes, and parallel ankles. The base 24-axis configuration excludes dexterous hands. These differences determine the references available for learning.

New robots require mappings and weights; new datasets require frame and zero-pose conversion. Interaction tasks often need separate tools, with quality assessed manually.

hhtools [9] combines robot preparation, source-pose calibration, retargeting, and review in one workflow. Setup and solve time depend on the robot, clip, backend, and hardware.

4.1.1 Scope

hhtools integrates solvers based on NVIDIA SOMA-Retargeter and Amazon holosoma [22, 23]. Like GMR [2], they use constrained inverse kinematics and trajectory optimization. The contribution is a common data-production workflow across robots and datasets.

The workflow imports a motion and target robot, aligns their reference poses, selects a solver, previews the result, and exports a trajectory. Object and terrain geometry remain attached to interaction sequences. This section describes the implementation at commit 6e6588c; the project page and installation guide provide the corresponding entry points [9].

Table 4.1. Retargeting workflow comparison.

Capability	Common retargeting workflow	hhtools
New robot	Manual mappings and weights	URDF and meshes generate the initial configuration
Zero alignment	Hard-coded source-pose assumptions	One interactive calibration per robot/source pair
Scenarios	Flat-ground IK or interaction meshes	Two integrated solvers selected by motion type
Inputs	One or two formats or datasets	More than ten sources, video-to-motion, and robot transfer
Quality	User-written checks	Plausibility scores, behavior labels, and subset recommendations
Interfaces	Primarily scripts	CLI, WebUI, desktop, and agent interfaces

4.1.2 Robot URDF Integration

Users import a robot URDF and meshes through the WebUI. The import workflow prepares a preset; the generated mappings and joint limits still require review against the actual RP1 configuration.

URDF repair. Mesh paths, unnecessary world or base_footprint roots, and invalid inertial properties are checked and corrected.

Joint mapping. Keywords and edit distance first associate 17 standard human joints with robot links: pelvis, spine, chest, neck, head, and bilateral shoulders, elbows, wrists, hips, knees, and ankles. Parent-child topology then corrects mismatches caused by inconsistent naming.

Validation. Check required pelvis, hip, knee, ankle, and wrist mappings, left/right assignment, and actuated-joint order. Use the preset validator and a short preview before processing a dataset; name matching alone does not establish anatomical correctness.

Geometry estimation. Robot height and joint-scale baselines are calculated from the meshes.

Import and confirmation produce an editable robot.yaml containing joint order, link mappings, tracking weights, and feet. Changes apply on the next run. RP1 uses RP1_description [7]; its 24- and 28-axis variants require their own presets. See the URDF import demonstration and preset-generation implementation [9, 38].

4.1.3 Interactive Calibration and Automatic Alignment

SMPL uses a T-pose; RP1 zero angles correspond to arms-down standing. Users match the robot to the reference skeleton in the WebUI and save the calibration to remove these offsets.

One-time calibration. Eight reference conventions cover SMPL, SMPL-X, GVHMR, SOMA BVH, LAFAN BVH, mocap BVH, Xsens, and GLB. The saved calibration binds robot identity, reference name, and calibrated joint angles. GLB uses the loaded clip's reference skeleton; calibration must be reassessed when the robot or reference changes.

Automatic alignment. Forward kinematics at the calibrated pose provides closed-form per-link scale, rotation, and translation offsets. These are derived from the saved pose at runtime; the implementation's sub-millisecond calculation note refers to this derivation, not the complete import, calibration, or retargeting workflow.

Normalization. Human height is normalized to 1.65 m while retaining individual segment proportions.

Reference-specific files use retarget_calibration_<reference>.yaml, with user overlays where needed. Preview limb direction, symmetry, trunk attitude, and feet before reusing the calibration. See the first-calibration demonstration and calibration implementation [9, 38].

4.1.4 Solvers for Motion and Interaction

Separate solvers address flat-ground motion and object/terrain interaction.

Mimic motion. Newton and NVIDIA Warp [24, 25] provide inverse kinematics for skeleton-only clips. CUDA Graph replay reduces iteration overhead on supported NVIDIA GPUs, with a CPU path when GPU execution is unavailable. Position and orientation objectives track mapped body links; joint bounds and soft joint filters regularize poorly constrained shoulder and hip axes. Ground-contact stabilization must be configured for the motion type, particularly for rolls and flips.

Interaction motion. The Interaction-Mesh backend handles objects and terrain using body targets, scene anchors, and optional Laplacian shape regularization. MuJoCo supplies forward kinematics, Jacobians, and collision distances; sequential quadratic programming (SQP) solves local QP subproblems, with OSQP enforcing linearized collision and contact constraints. The inspected configuration uses a 2 mm penetration tolerance and an MPC horizon of one frame; both are tunable. The Laplacian weight is zero by default in this revision. These numerical settings are not hardware clearance guarantees.

Backend selection follows whether the input contains only motion, object interaction, or terrain interaction. The fast-retargeting demonstration shows the workflow; the method overview and interaction configuration specify the implementation and its optional stabilization settings [9, 38].

4.1.5 Motion Data and Input Sources

Adapters convert inputs to metric units, Z up, and +X forward, with quaternion order made explicit at interchange. Users obtain data and required body-model weights under upstream licenses. Entries in Table 4.2 link to source pages; the cited papers are listed in Appendix D.

Table 4.2. Motion data and input sources.

Category	Datasets / sources	Workflow role
Optical motion capture	AMASS [26], LAFAN1 [27], 100STYLE [28], SOMA/SEED [29]	Whole-body, gait, and stylized motion
Curated motion	Motion-X [30], PHUMA [31]	Dynamic, martial-arts, and physically filtered motion
Monocular video	GVHMR [32]	Video-derived references
Human-object interaction	OMOMO [33], OmniContact [34]	Carrying, pushing, and contact tasks

Category	Datasets / sources	Workflow role
Human-terrain interaction	holosoma [23, 35], PARC MS [36]	Parkour, climbing, and obstacles
Robot trajectories	MotionDecode [37]	G1-to-RP1 transfer

BVH, GLB, and unified NPZ support file-based input. GVHMR [32] derives motion from video through a separately installed CUDA environment and model weights. Robot-to-robot transfer reconstructs source link motion by forward kinematics before fitting the target robot, using a source-target calibration. For MotionDecode G1 files without time metadata, the documented source rate is 120 Hz; it must be supplied explicitly. See motion-input preview and robot-to-robot transfer [9, 38].

The analysis panel scores physical plausibility out of 100 using floating, penetration, slip, jitter, and self-collision penalties. Complexity labels and clustering suggest diverse training subsets. Sequences are flagged rather than deleted because airborne phases may represent valid dynamic motion.

Inspect contact consistency and joint continuity before accepting a training subset. Quality scores support screening but do not establish dynamic feasibility. See the dataset-analysis demonstration [38].

4.1.6 Human and Agent Interfaces

One solver core serves the WebUI, desktop packages, CLI, and JSON CLI/MCP adapters. The WebUI provides single-clip and queued export; CLI scripts support resumable, isolated batch runs. Current Agent contracts cover human-to-robot retargeting, scene-free robot-to-robot transfer, pair calibration, and ordered batches. Video-to-motion, dataset analysis, scene-bearing robot-to-robot transfer, and deployment are outside the Agent interface. See the Agent interface documentation and batch-retargeting demonstration [9, 38].

Export contract. Robot CSV output contains time; root_x, root_y, root_z; root_qx, root_qy, root_qz, root_qw; and named dof_* columns. Header metadata identifies the robot, sample rate, and DoF count. Scene bundles also retain object trajectories and terrain assets. CSV uses xyzw quaternions, while the PKL bundle declares wxyz: consumers must honor each format's schema. The CSV specification and bundle exporter define these contracts [9].

RP1 integration. Match the exported joint names, ordering, units, sample rate, and root-frame convention to the training loader for BeyondMimic, MimicLite, or UFO [11, 12, 13]. Table 4.2 describes supported inputs and project sources; it does not assert that every listed dataset was used to train a policy. The 24- and 28-axis versions require separate calibration and export with their own URDFs. Preserve the source clip identifier, robot assets, calibration, solver settings, software commit, and review outcome with each accepted sequence so it can be reproduced.

Replay exports against the same robot model before dataset release. Geometric references still require slip, contact-force, actuator-limit, and tracking-stability evaluation during training and deployment in Section 4.2 and Section 4.3.

4.2 Training Framework

Isaac Lab 3.0 [10] provides task configuration, parallel training, replay, and export. Task objectives, robot assets, and physics backends are configured separately.

4.2.1 Tasks and Backends

Registered tasks link environments and algorithm settings. Manager-based environments compose observations, actions, rewards, events, and curricula; Direct environments implement state updates and batch computation in one class. Both use the same training interface.

PhysX and Newton/MJWarp presets are independent of environment type. Adapted assets can switch backend through configuration without changing task objectives or algorithm settings.

Training can run headlessly; checks and replay use visualization. Newton tasks without Kit dependencies support Kit-less execution.

4.2.2 Experiments and Parallel Execution

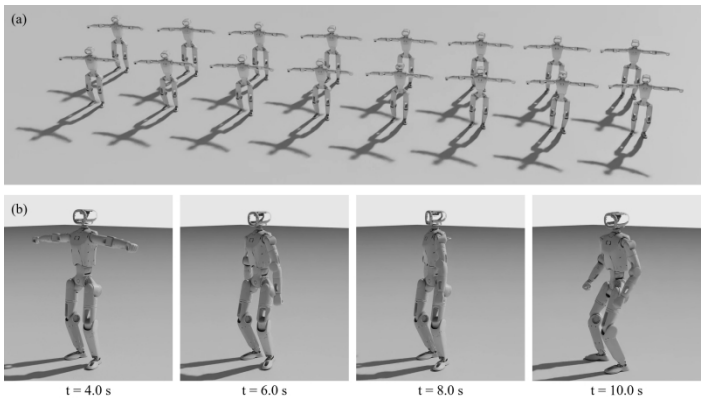
Registered settings serve training, replay, and export. Environment count and device settings scale a task from small checks to GPU training without rewriting its logic.

Each run saves resolved environment/algorithm settings and checkpoints for resumption and review.

In RP1 velocity tracking, policy and value networks use different observations. The policy receives three frames of proprioceptive state, velocity commands, and previous actions, with noise added to selected proprioceptive inputs during training. The value network additionally receives contact and dynamics information and, on complex terrain, terrain scans. Histories describe recent state, while privileged simulation information is restricted to training-time value estimation. Denoting the two inputs by H_t^{π} and H_t^{V}, the deterministic policy and value estimate are

$$a_t = f_\theta(N_\pi(H_t^{\pi})), \qquad V_t = V_\phi(N_V(H_t^{V})), \qquad N(x) = \frac{x-\mu}{\sigma+\varepsilon}. \tag{1}$$

ONNX export includes the policy and observation normalization, excluding the value network and privileged inputs. Deployment must reproduce the training observation definition.



GPU simulation samples thousands of environments. Multi-GPU training synchronizes advantage statistics, mean KL divergence, and gradients for normalization, learning-rate adjustment, and updates. Parallel training and single-environment replay share observation, action, and robot definitions (Figure 4.2).

Figure 4.2. RP1 reference-motion tracking. (a) Sixteen parallel environments. (b) Single-environment policy replay. Source: RoboParty [21].

Published Isaac Lab 3.0-EA humanoid benchmarks at 8,192 environments report Newton/MJWarp mean training throughput of about 2.24 times PhysX for Direct tasks and 2.08 times for Manager-based tasks (Figure 4.3a) [10]. These are upstream results, not RP1 measurements.

At 4,096 Newton environments, Warp reduces step time relative to Torch by 22.63% for Direct and 29.16% for Manager-based tasks (Figure 4.3b) [10]. Unlike the throughput benchmark, this measures environment stepping without policy updates.

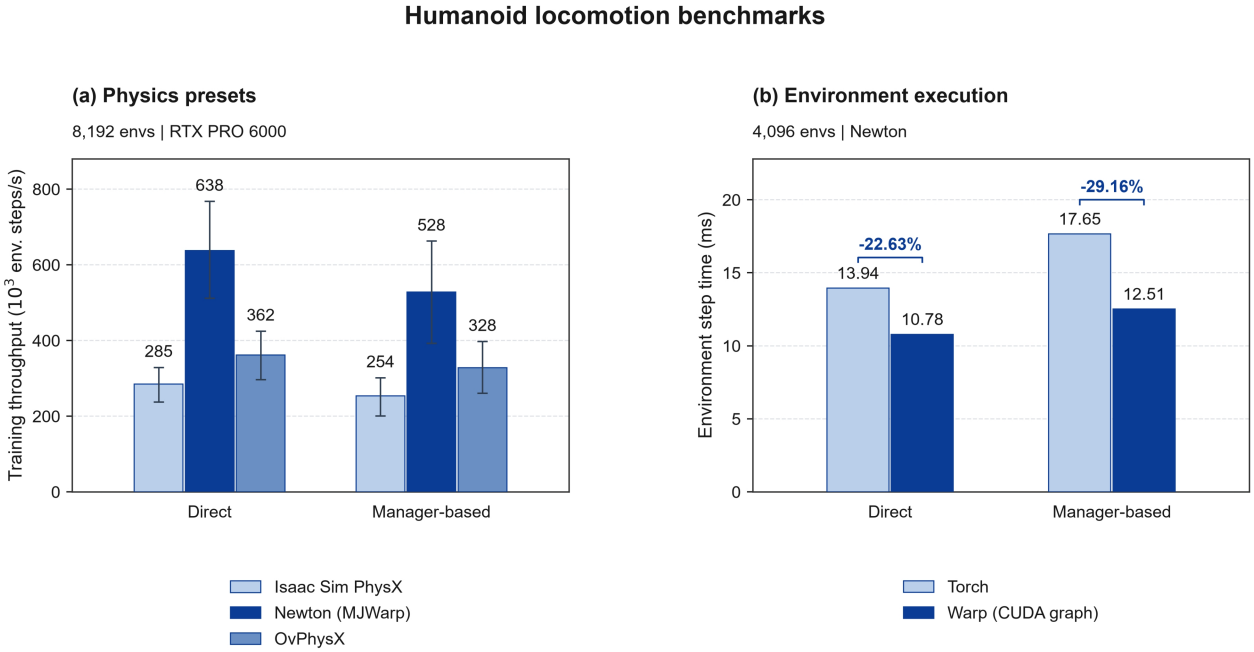
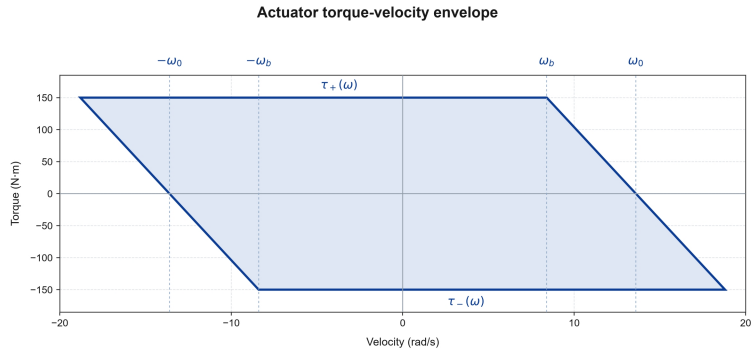


Figure 4.3. Published upstream humanoid benchmarks, not RP1 hardware measurements [10]. (a) Training throughput by physics preset. (b) Torch and Warp environment-step times.

4.2.3 Dynamics and Parameter Identification

Training requires actuator output limits in addition to rigid-body and contact dynamics. RP1 constrains PD output with a speed-dependent four-quadrant torque envelope, representing peak torque, high-speed derating, and braking limits (Figure 4.4). Let peak torque be τ_p, no-load speed ω_0, and maximum speed at peak torque ω_k. Define $\tau_s = \tau_p/(1-\omega_k/\omega_0)$ and $\omega_c = \omega_0(1+\tau_p/\tau_s)$, and clip current speed to $\tilde{\omega} = \text{clip}\,(\omega, -\omega_c, \omega_c)$. The upper and lower torque limits are

$$\tau_+(\omega) = \min\{\tau_s(1-\tilde{\omega}/\omega_0), \tau_p\}, \qquad \tau_-(\omega) = \max\{\tau_s(-1-\tilde{\omega}/\omega_0), -\tau_p\}. \tag{2}$$



The dynamics model also includes command delay. Tasks share actuator parameters; complex-terrain tasks randomize contact and robot dynamics and adjust terrain difficulty using travel distance and velocity commands.

Figure 4.4. Torque-speed envelope for RP1 waist, hip, and knee actuators.
Source: RoboParty [21].

PACE identifies actuator parameters through parallel simulation. It replays commands with a fixed base and disabled contacts, evaluating one parameter candidate per environment. CMA-ES updates the distribution from joint-position error over the complete trajectory. For n joints, ζ contains per-joint added inertia, viscous friction, friction coefficient, and encoder offset, plus a shared discrete delay, totaling parameters. The objective is

$$\zeta^* = \arg\min_{\zeta\in\mathcal{B}} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=1}^{n}\left(q_{t,j}^{\text{sim}}(\zeta) - b_j - q_{t,j}^{\text{enc}}\right)^2. \tag{3}$$

Here, b_j is the true-angle offset relative to the encoder reading, and $\mathcal{B}$ is the search domain. The identification model applies delay after PD computation and torque clipping to fit response timing. Commands and feedback recorded during deployment can directly provide PACE fitting data.

4.3 Deployment Framework

Configuration links policies and the Python SDK to common joint-state and command interfaces. The robot interface handles communication and coordinate transforms.

4.3.1 Configuration-Driven Policy Adaptation

Body settings define devices, mappings, mechanisms, and closed-chain indices. Control settings define gains, nominal pose, and period. Policy settings define models, observations, and action conversion. These settings can be edited separately but must remain mutually compatible with the selected robot model and policy.

For the 24-DoF RP1, velocity tracking uses velocity commands and observation history, while reference tracking adds reference joint positions and velocities (Table 4.3). Although their inputs and histories differ, both output 24-dimensional actions converted to joint targets by the same inference program.

Table 4.3. Example configurations for two RP1 policies.

Policy task	Observations	Model interface
Velocity tracking	81 values/frame; 3-frame obs-major history	243 inputs; 24 outputs
Reference tracking	Single frame including 48 reference-joint values	126 inputs; 24 outputs

Observation settings specify field order, scale, clipping, history length, and frame offsets. New observation types require module implementation. Action settings define joint order, scale, and nominal pose.

Each input component requires a defined physical meaning, order, and sample time. Let $x_t^{(j)}$ denote observation vector j, with $h_0 > \cdots > h_{K-1} = 0$ ordered from older to newer offsets. When fields share offsets, two common layouts are

$$o_t^{\text{obs}} = \text{concat}_{j=1}^{M}\text{concat}_{k=0}^{K-1}x_{t-h_k}^{(j)}, \qquad o_t^{\text{frame}} = \text{concat}_{k=0}^{K-1}\text{concat}_{j=1}^{M}x_{t-h_k}^{(j)}. \tag{4}$$

Obs-major groups each field's history; frame-major groups complete observations by time (Figure 4.5). Deployment ordering must match training.

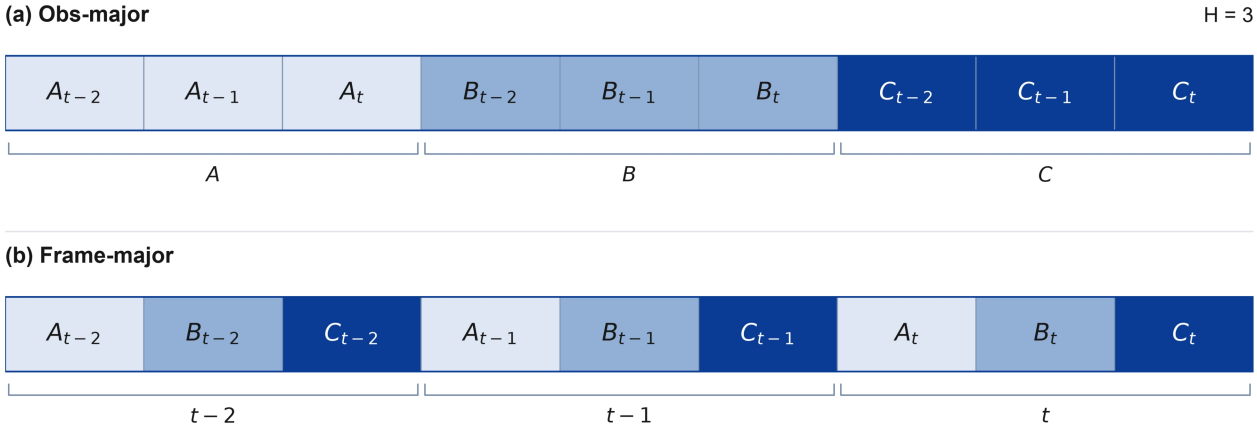


Figure 4.5. Three-frame observation history in (a) obs-major and (b) frame-major order. Source: RoboParty [21].

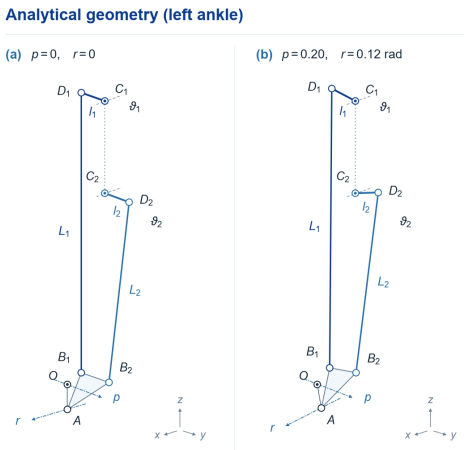
4.3.2 Python SDK and Robot Integration

Python bindings expose C++ state and command interfaces. The SDK sets references, gains, and experiment parameters; the robot interface handles motor mapping, sensors, and ankle transforms. Position, velocity, gains, and feedforward torque share one joint order.

Existing devices use parameter and mapping configuration; new types require module extensions. Joint interfaces support control, while motor and IMU interfaces support device tests.

The response recorder logs targets, feedback, timestamps, sample rate, sweep range, and amplitude for PACE identification. Testing and deployment share joint definitions and transforms.

4.3.3 Parallel Ankle Kinematics



Equivalent ankle motion is expressed as pitch and roll. The robot interface transforms positions, velocities, and torques between equivalent joint and motor coordinates, while policies and the SDK operate in equivalent joint space. Figure 4.6 shows two branches comprising fixed motor centers C_i, crank endpoints D_i, and foot attachments B_i. The pitch origin O and roll center A have a fixed offset.

Figure 4.6. Analytical left-ankle geometry. (a) Zero pose. (b) Combined pitch and roll. Source: RoboParty [21].

Let $q = [p, r]^T$ be equivalent joint coordinates and $\vartheta = [\vartheta_1, \vartheta_2]^T$ motor coordinates. The zero-pose roll center is a; b_i^0 and c_i denote zero-pose foot attachments and fixed motor centers. Crank and rod lengths are l_i and L_i. With motor zero offsets ϑ_{0i} and $\alpha_i = \vartheta_i + \vartheta_{0i}$, the closed-loop geometry is

$$b_i(q) = R_y(p)[a + R_x(r)(b_i^0 - a)], \qquad d_i(\vartheta_i) = c_i + (0, l_i\cos\alpha_i, -l_i\sin\alpha_i)^T. \tag{5}$$

$$F_i(q, \vartheta_i) = \frac{1}{2}(\|b_i(q) - d_i(\vartheta_i)\|^2 - L_i^2) = 0, \qquad i = 1,2. \tag{6}$$

Equation (5) explicitly retains the offset between joint axes. Equation (6) connects joint and motor coordinates through two constant-length constraints. Let $\delta_i = c_i - b_i(q)$. Expanding the constraints gives $A_i \cos\alpha_i + B_i \sin\alpha_i + C_i^* = 0$, where $A_i = 2l_i\delta_{iy}$, $B_i = -2l_i\delta_{iz}$, and $C_i^* = \|\delta_i\|^2 + l_i^2 - L_i^2$. Define $\rho_i = (A_i^2 + B_i^2)^{1/2}$ and $\phi_i = atan2(B_i, A_i)$. The analytical motor-angle candidates are

$$\vartheta_i = \phi_i \pm \arccos\left(-\frac{C_i^*}{\rho_i}\right) - \vartheta_{0i}, \qquad \rho_i > 0, \quad |C_i^*| \le \rho_i. \tag{7}$$

For each branch, the solver selects the candidate with smaller absolute motor angle, defining the inverse map $\vartheta(q)$. Let $v_i = b_i - d_i$ and $u_i = d_i - c_i$, with motor-axis direction $s = (-1,0,0)^T$. Differentiating (6) gives $G_q\dot{q} = G_m\dot{\vartheta}$, where $(G_q)_{i,:} = v_i^T \partial b_i/\partial q$ and $G_m = \mathrm{diag}\{v_i^T(s \times u_i)\}$. Where both matrices are nonsingular, the local velocity mapping is

$$J_{m\to q} = G_q^{-1}G_m, \qquad J_{q\to m} = G_m^{-1}G_q, \qquad \dot{q} = J_{m\to q}\dot{\vartheta}. \tag{8}$$

Virtual work, $\tau_q^T\dot{q} = \tau_m^T\dot{\vartheta}$, gives the dual generalized-force transformation:

$$\tau_m = J_{m\to q}^T\tau_q, \qquad \tau_q = J_{q\to m}^T\tau_m. \tag{9}$$

State reading uses numerical forward kinematics to recover equivalent joint angles and the analytical Jacobian to transform velocities. Starting from the previous successful solution, forward kinematics corrects the motor-angle residual:

$$q^{(n+1)} = q^{(n)} + \eta J_{m\to q}(q^{(n)})[\vartheta_{\mathrm{meas}} - \vartheta(q^{(n)})], \qquad \eta = 0.5. \tag{10}$$

Each iteration evaluates the analytical inverse solution, forms the motor-angle residual, and uses the Jacobian for the joint-angle correction. Both ankles share the solution form, with separate attachment geometry and motor-zero offsets.

The controller first calculates PD torque in joint space, then maps it to motor space using the Jacobian at the measured configuration:

$$\tau_q^* = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + \tau_{\mathrm{ff}}, \qquad \tau_m^* = J_{m\to q}^T\tau_q^*. \tag{11}$$

The resulting τ_m^* contains the two motor torque targets. Higher-level policies define joint-space goals; the robot interface performs the motor transformation. Feedback and commands use the same mechanism, model, and coordinates.

4.4 Whole-Body Intelligence

RP1 combines VR or inertial teleoperation, whole-body control (WBC), RP Hand mapping, and synchronized recording for WBI demonstrations.

4.4.1 Whole-Body Control and Manipulation

WBC follows human references using VR or Noitom Pnlink-FB-G inertial capture. Inertial capture supplies fuller body references than five-point VR tracking.

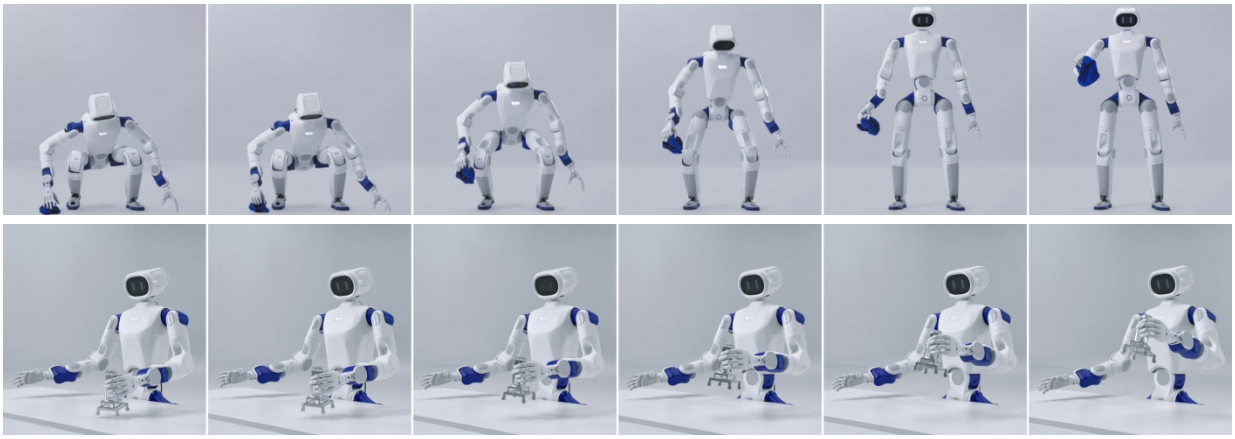


Figure 4.7. Whole-body posture tracking and manipulation demonstrations. Source: RoboParty [21].

Visual recognition maps VR hand poses to RP hand targets. Hand commands and body state share recording and replay.

Figure 4.8 shows a 5 kg bucket curl with hand grasping; the payload excludes the fixture. The motion is produced by VR or inertial teleoperation, and latency depends on the user's network conditions rather than a single fixed value. Hand interfaces and tools are described in Appendix C.

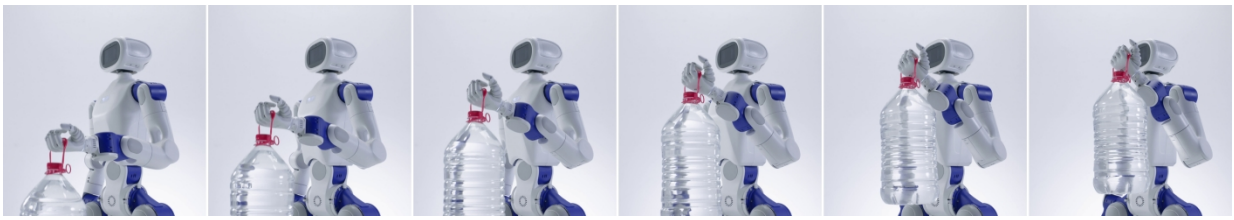


Figure 4.8. Loaded upper-limb manipulation. Source: RoboParty [21].

4.4.2 Data Collection and Conversion

Teleoperation records stereo egocentric video, joint states, IMU data, whole-body references, and hand commands for conversion and WBI development (Table 4.4).

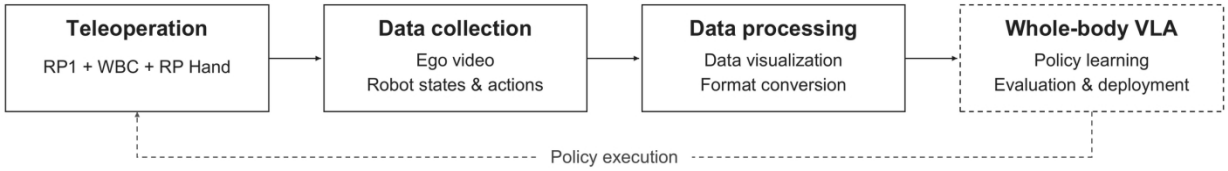


Figure 4.9. WBI development workflow. Source: RoboParty [21].

A standalone HTML viewer synchronizes stereo video, signal curves, and values. It supports timeline navigation, frame stepping, playback speed, signal selection, and zoom (Figure 4.10).

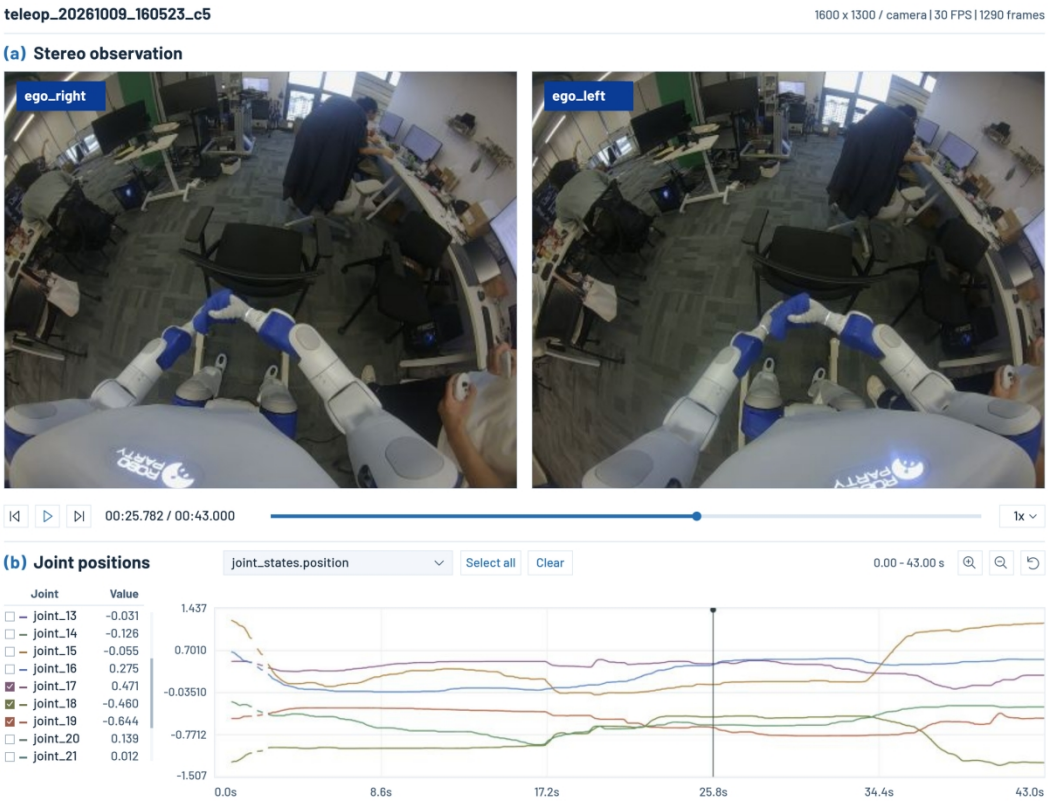


Figure 4.10. Synchronized video and joint-state replay. Source: RoboParty [21].

Table 4.4. Data in the example recording.

Data	Recorded content	Dimensions
Egocentric video	Task scene and hand operation	Left and right, each 1600 x 1300 pixels
Joint state	Position, velocity, torque	28 values for each quantity
Motion reference	Joint targets and whole-body tracking	28 joint targets; 35 whole-body values
Inertial data	Body orientation/angular velocity/linear acceleration; ego angular velocity/linear acceleration	Body orientation: 4; each angular-velocity and acceleration stream: 3
Hand data	Left/right states and actions	6 values per hand

Conversion aligns coordinates, time, and joint semantics with downstream data formats and policy interfaces.

5 PartyOS

PartyOS is under development; complete methods and evaluation are planned for a separate report.

PartyOS combines retargeting, BFM motion control, deployment, and automated evaluation. Planned extensions cover scene/object interaction, VLA-based manipulation, and agentic humanoids.

Three algorithm layers share the infrastructure below.

System0: general whole-body control from reference motion, target poses, rewards, or teleoperation.

System1: a planned noncollapsed intent representation aligned with action semantics and composable motion primitives.

System2: large-model orchestration of motion control and VLA into reusable skills.

Infrastructure: retargeting, scene generation, automated data collection, evaluation, and a research workbench.

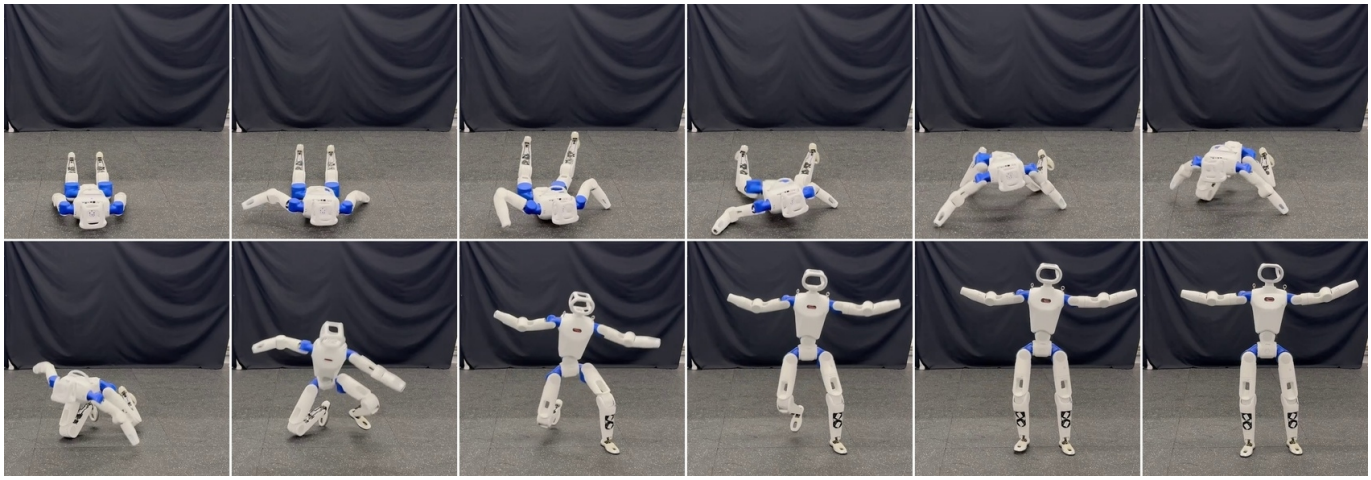
System0 and infrastructure are partly open-sourced. System1 and System2 remain research topics.

Released components

hhtools [9] retargets human motion to supported humanoids, with approximately 30 s reported for the cited workflow. It includes RP1 and supports transfer of G1 motion data.

MimicLite [12] provides efficient general motion tracking, with a reported training configuration of eight RTX 4090 GPUs and approximately 3 h.

UFO [13] is an unsupervised BFM framework for disturbance recovery and getting up after falls. The shared code supports six humanoids, including RP1.



gpt-6-astra-real2sim-workflow [39] generates editable, interactive simulation scenes from a single photograph.

LabWeft [40] records experimental lineage and evidence and supports agent-assisted algorithm iteration.

Interim results

Motion control. The source draft reports MimicLite v1.1 ranked first among 15 policies on a third-party G1 tracking leaderboard [41], with approximately 1/420 of SONIC's training cost. The timing and comparison

conditions are documented in the open MimicLite technical report. It reports UFO reducing unsupervised-BFM training from 45 to 6 h and supporting teleoperation or autonomous recovery under strong disturbances. These time- and benchmark-specific claims require the corresponding evaluation snapshot and configuration for reproduction [12, 13, 21].

Interaction. The source draft reports a UFO-based interaction BFM that carries, pushes, and kicks boxes with one policy, with 89.3% box-carrying success after falls compared with 1.3% for the cited planning approach. The underlying evaluation record is internal and has not been supplied as a public paper [14, 21].

Real-sim-real. The reported workflow generates a high-fidelity multi-room scene in approximately 2 min, compared with approximately 3 h for the referenced open alternative. Automated expert collection has been demonstrated on RPW and G1; the evaluation platform supports RP1 and RP1-WBC. These figures describe the source project's reported configurations [21, 39-41].

Planned work will reproduce perception and interaction BFMs on RP1 and connect scene/task generation, collection, fine-tuning, and evaluation.

6 System Testing

Sustained-operation validation of RP1 addresses actuators, structural connections, and cable harnesses. Test design therefore considers motion characteristics, loading methods, and fault inspection together. Modular benches and layered test software provide complementary conditions for evaluating cyclic motion, sustained loading, and connection integrity under vibration, while linking physical records to their execution configurations. This section describes the test system and representative validation results [21].

6.1 System Configuration and Validation Scope

Table 6.1. Test conditions and representative physical records. Source: RoboParty [21].

Validation item	Test conditions	Recorded test outcomes
Periodic upper-limb swing	Unloaded; poses from simulated running at 3 m/s; 8 h per session	120 h cumulative operation without a recorded fault
Repeated leg squatting	60 kg load; approximately three times the upper-body mass	8 h continuous stable operation; more than 100 h of cumulative testing
Continuous single-arm loading	2.5, 3.5, and 5.5 kg; payload-specific task trajectories executed at 1.5 times the reference speed	2.5 kg: 125.7 min; 3.5 kg: 120.0 min; 5.5 kg: 10 min of short-duration dynamic loading

Validation covers single-arm, upper-body, single-leg, and lower-body modules. Recorded operating time summed across these four module categories is approximately 900 h: approximately 205, 251, 230, and 217 h, respectively. This aggregate is not a continuous whole-robot operating record and excludes parallel runs, pauses, cooling, maintenance, and prototype replacement. The records include unloaded cycles, loaded motion, and impact tests. This report presents representative engineering verification rather than a cross-prototype or cross-version statistical comparison; trial identifiers, version lists, and complete raw records remain traceability materials. Table 6.1 lists representative physical records with explicitly defined conditions.

Repeated changeover and inspection require the benches to separate a common load-bearing structure from product-specific fixtures. Quick-release and locating interfaces accommodate different components (Figure 6.1). The lower-limb guide frame and fall-arrest structure constrain loaded motion, while drag chains organize cable routing. The power architecture separates actuator power from control power: four branches are independently switched, protected, and measured, while sharing the voltage setting and total current limit. The 24 V control supply is connected upstream of the main contactor KM0, retaining PLC power and state acquisition after actuator power is disconnected.

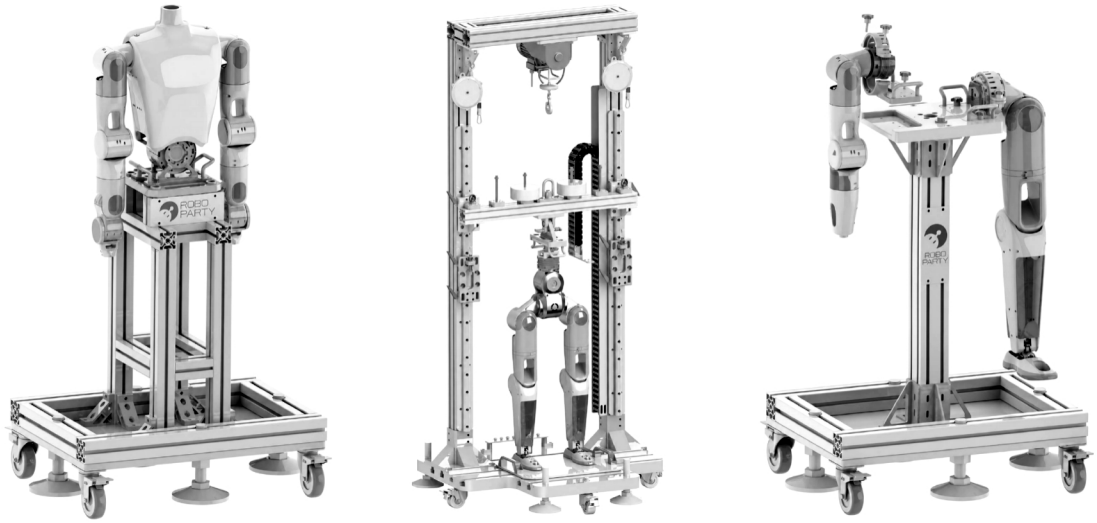


Figure 6.1. Modular test benches and quick-release fixtures. From left to right: upper-body bench, loaded lower-body bench, and single-arm bench with a clamp fixture. Design renderings. Source: RoboParty [21].

Each trial associates the prototype configuration, load, trajectory, and software version with its execution record and scheduled inspections. Battery replacement, cooling, and inspection periods are logged separately; accumulated durations count as effective execution time.

6.2 Complementary Upper and Lower Limb Endurance Tests

Upper-limb endurance testing examines mechanisms and cable harnesses under repeated motion. Periodic arm-swing trajectories are extracted from simulated running poses at 3 m/s; coordinated joint motion and repeated reversals create continuous cycles. Unloaded operation retains the motion range and cable-bending characteristics needed to inspect structural interference, routing clearance, and cyclic wear. Loaded operation additionally evaluates actuator load capacity and temperature rise.

Lower-limb loading and mechanical connections experience different stress conditions and are evaluated through squatting and stepping, respectively (Figure 6.2). Repeated flexion and extension under load assess joint support, actuator temperature rise, and sustained operation. Alternating stepping produces repeated vibration that perturbs structural connections and fasteners; subsequent inspections check for loosening, fastening failures, and local damage. These complementary conditions provide separate evidence for actuator performance and connection integrity.

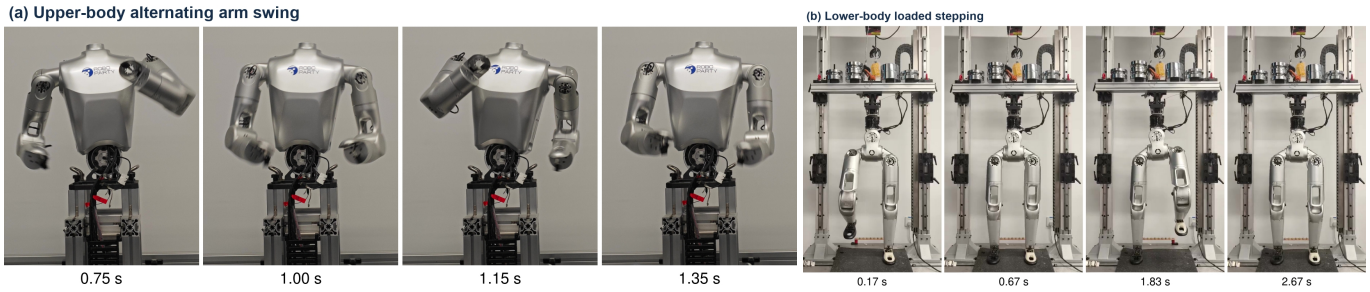


Figure 6.2. Physical endurance motions. (a) Alternating upper-body arm swing. (b) Loaded lower-body stepping. Each panel contains four frames with timestamps relative to its own source video; the two panels do not share a common video clock. Source: RoboParty [21].

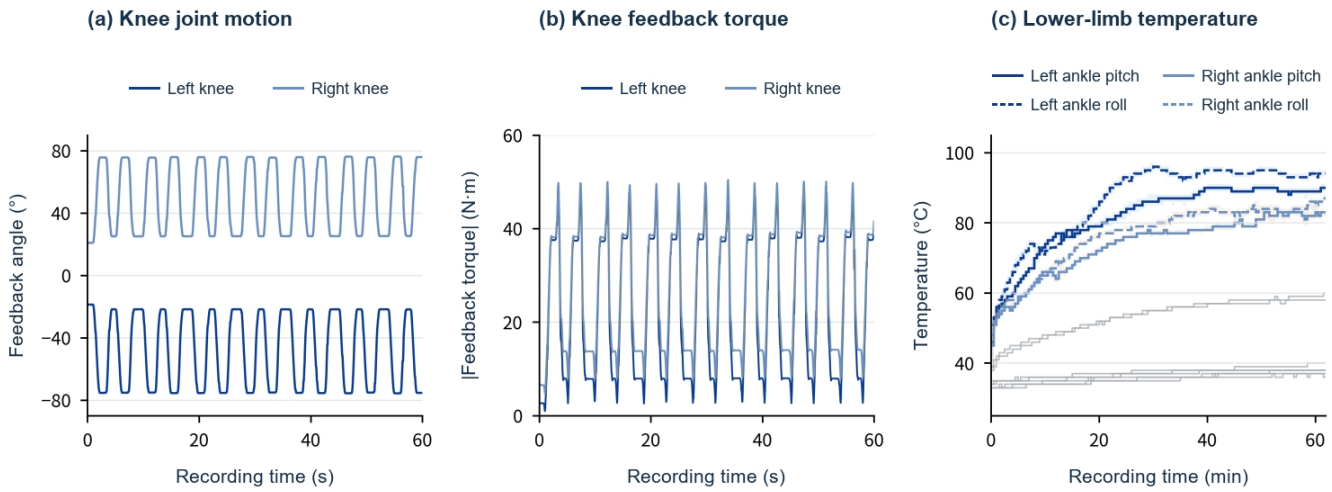


Figure 6.3. Joint motion, torque, and temperature during squatting with a 50 kg external load shared by both legs. (a, b) Bilateral knee feedback angles and absolute feedback torque over the first 60 s; angles follow each joint coordinate convention. (c) Temperatures of all 12 lower-limb axes over the complete 61.94 min record, summarized by 30 s window medians with the final window retained. The four ankle joints are colored; pale bands show their within-window ranges, and gray traces show the other eight axes. The record includes motion and holding phases. This 50 kg, 61.94 min record is separate from the 60 kg endurance summary in Table 6.1, which reports 8 h continuous and more than 100 h cumulative testing. Source: RoboParty [21].

6.3 Dynamic Single Arm Load Validation

Single-arm loaded trajectories are designed around representative task motions, with task coverage selected for each payload. The 2.5 kg condition covers eight task classes: workstation transfer, array pick-and-place, high/middle/low shelf handling, machine loading and unloading, approximately 90-degree reorientation, multi-face inspection, horizontal transfer, and instrument insertion and removal. The 3.5 kg and 5.5 kg conditions each cover four classes: workstation transfer, array pick-and-place, low/middle shelf handling, and machine loading and unloading (Figure 6.4).

The trajectories incorporate transfers at different heights and in different directions, orientation changes, and insertion and withdrawal motions. Continuous shoulder, elbow, and wrist coordination with repeated reversals establishes dynamic loading. Simulation checks joint range, velocity, acceleration, and motion transitions; bench measurements record temperature, feedback torque, and joint motion. Associating task classes, payloads, and operating responses allows sustained load performance to be assessed for the covered task motions.

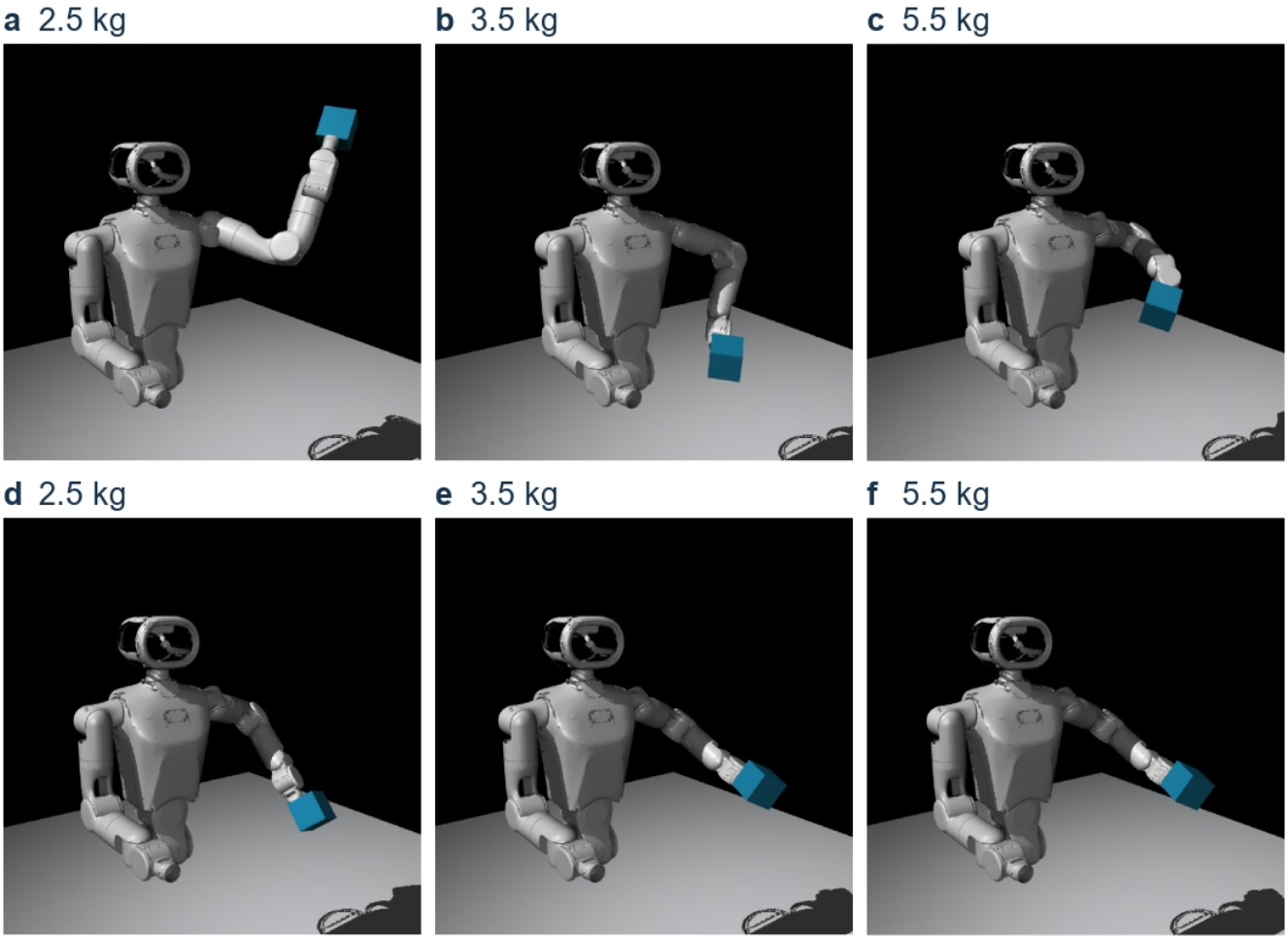


Figure 6.4. Simulated design of continuous single-arm loaded motions. Each column shows two poses at one payload: (a,d) 2.5 kg; (b,e) 3.5 kg; and (c,f) 5.5 kg. The left arm has seven degrees of freedom; the upper body and right arm are fixed. The payload center of mass is 11 cm from the output axis of joint 7. Source: RoboParty [21].

6.3.1 Physical Loaded Motion

The bench secures a payload to the arm end and executes the task trajectory corresponding to that payload. Figure 6.5 shows representative physical poses at all three loads, illustrating changes in payload position and orientation during coordinated shoulder, elbow, and wrist motion.

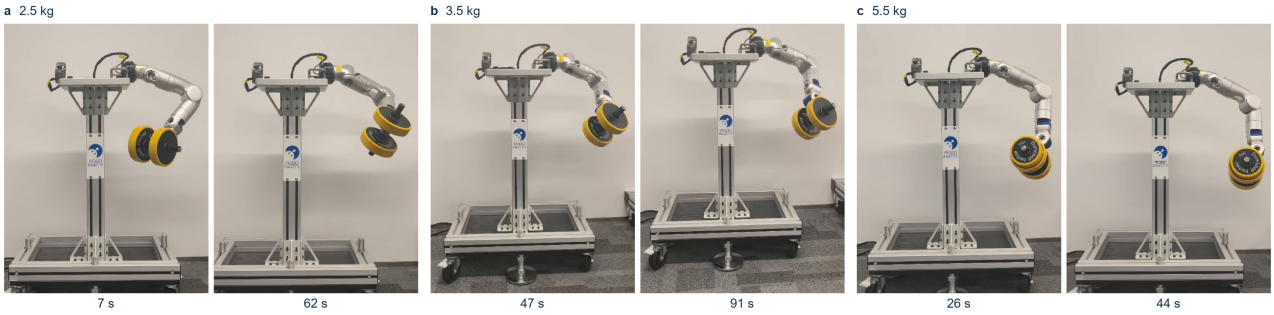


Figure 6.5. Physical single-arm motions at three payloads. From left to right, the three adjacent frame pairs (a-c) correspond to 2.5, 3.5, and 5.5 kg. Within each pair, frames from the corresponding test video are ordered chronologically from left to right. Timestamps are relative to each video. Source: RoboParty [21].

Payload trajectories ran at 1.5 times reference speed; angle amplitude was unchanged, with ideal velocity and acceleration scaling of 1.5 and 2.25 times. Continuous records were 125.7 min at 2.5 kg and 120.0 min at 3.5 kg. The 5.5 kg test was a 10 min short-duration validation, not a rated continuous load. No fault code was reported for seven analyzed joints; failed records enter internal statistics but are not expanded here.

6.3.2 Joint Responses and Recorded Performance

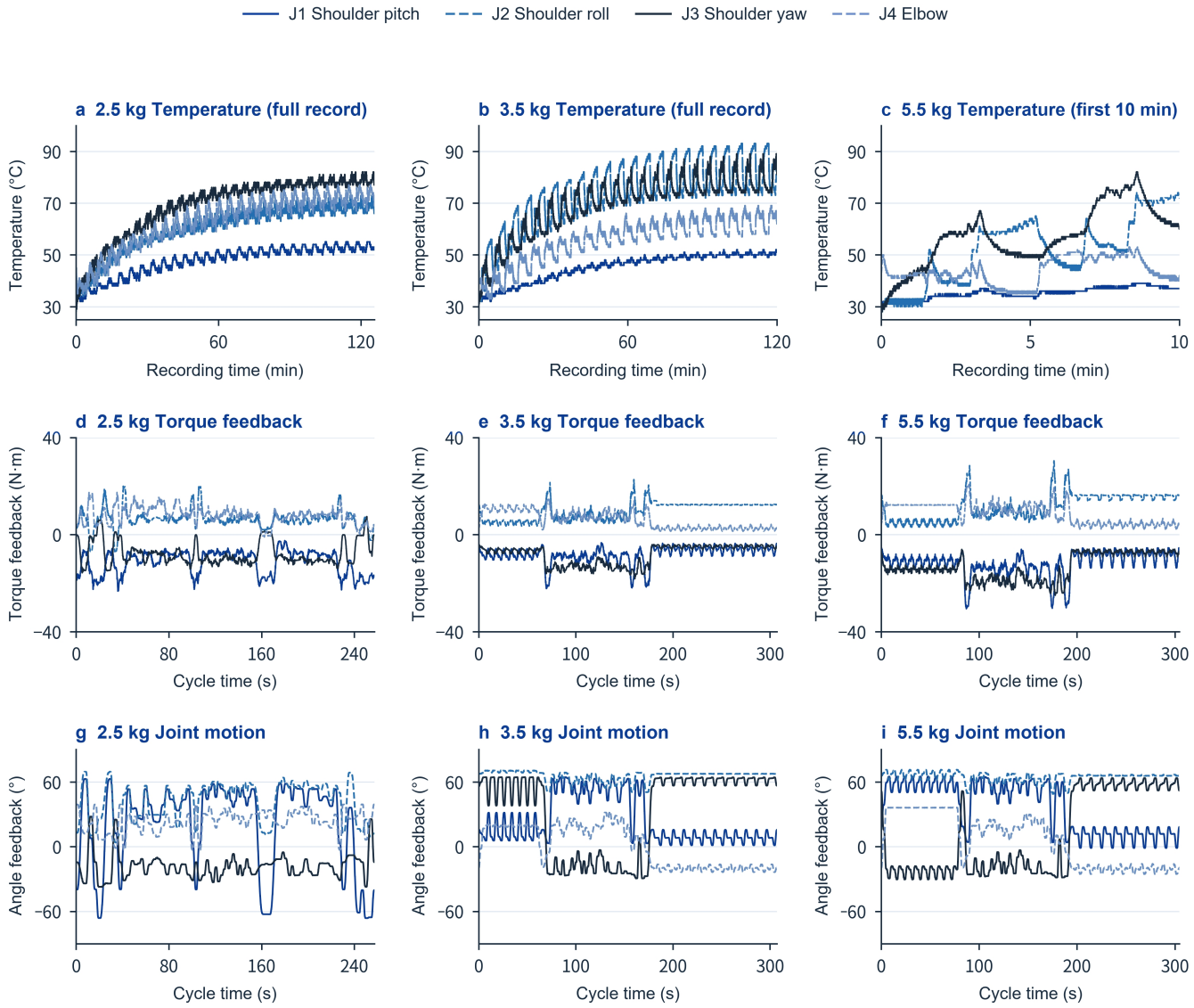


Figure 6.6. Key joint responses during continuous single-arm loading. Columns correspond to 2.5, 3.5, and 5.5 kg, with one trial per payload. The three shoulder joints and elbow are shown as J1-J4. (a-c) Device-feedback temperatures: complete records at 2.5 and 3.5 kg and the first 10 min at 5.5 kg. (d-f) Feedback torque and (g-i) joint angles during the first complete working cycle, with time measured from the cycle start. The curves retain all samples within the displayed intervals; torque denotes drive-feedback torque from the actuator channels. Source: RoboParty [21].

Under the tested task trajectories, the single arm completed sustained dynamic operation at 2.5 kg and 3.5 kg. A 10 min dynamic loading record demonstrates short-duration operation at a 5.5 kg task payload; it does not establish a continuous-duty payload rating.

6.4 Test Software and Evidence Management

The available records do not include a separate complete report on temperature-rise limits, alarms, loosening, wear, or harness inspection; some temperature rises are small and are retained in the plotted channels where legible. Continuous tests require device execution to remain independent, while archiving and querying must accommodate network interruptions and large time-series datasets. The software therefore separates motion control, local acquisition, and centralized evidence management (Figure 6.7). A desktop client accesses device services through a local gateway. The gateway and shared services coordinate access to PLC and CAN resources; uploads, file processing, and curve queries are scheduled separately.

Preservation of original evidence does not depend on network availability. Each workstation first retains raw CSV files and the execution configuration. After an operator verifies the test conditions and explicitly submits the record, the dedicated reliability_v1 interface archives the evidence. SQLite persists verification snapshots and upload status. Resumable chunked transfer and receipt reconciliation support recovery after interruption, allowing remote archiving to continue from existing local records.

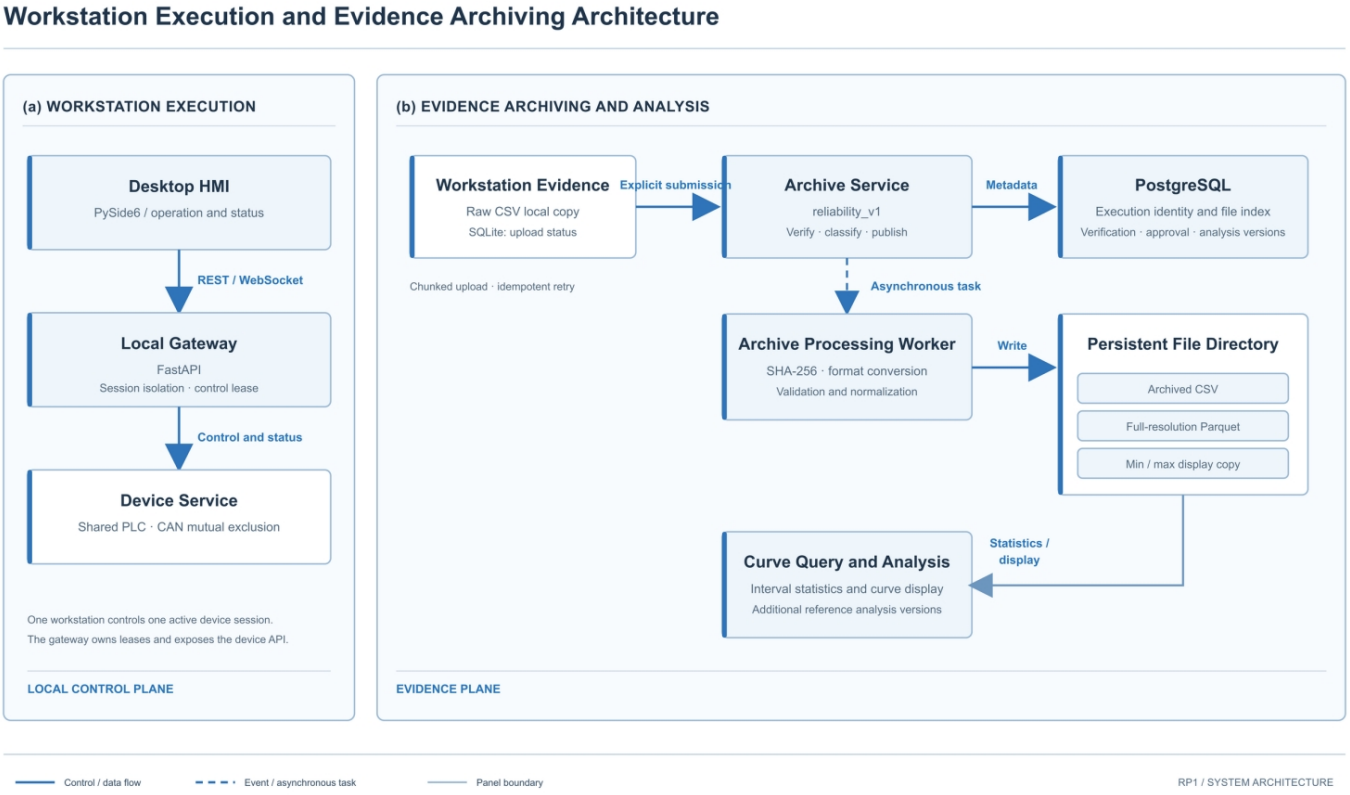


Figure 6.7. Test software and evidence-data architecture. Workstations manage device execution and local evidence; the platform manages file archiving, version associations, review, and queries. Solid arrows denote control or data flow, and dashed arrows denote asynchronous tasks. Source: RoboParty [21].

Test-condition verification and analysis may be updated after a trial, while measured facts must remain traceable. The data layer, therefore, separates original files from interpretation versions. PostgreSQL stores execution identities, file indexes, evidence hashes, and verification, approval, and analysis versions. Raw CSV files and derived query files reside in a separate persistent directory. Original measurements remain read-only; corrections and reanalysis append versions. Review freezes the relevant snapshot, and approval publishes the execution, result, and review records together.

Display efficiency and statistical precision are handled separately for long recordings. Background processing creates Min/Max display copies for viewport browsing, whereas interval statistics, sample queries, and exports read full-resolution data. Official statistics include only approved test results. Pending or rejected records, and "other" tests whose formal context is incomplete, retain their original evidence. Curve inspection, result review, and formal release thus each have an explicit data basis.

7 Limitations and Future Work

The evidence combines CAD and URDF values, configured software rates, calculations, simulation records, and physical tests. A complete fault-injection and acceptance matrix is outside the scope of this report; its absence does not change the interpretation of the measured loaded curves. Nominal communication budgets do not establish end-to-end latency, and bench hours alone do not establish field reliability or MTBF. Results remain conditional on the stated hardware, load, controller, and environment.

Future work should retain revision-controlled trial records and extend repeatable checks of combined joint motion, thermal endurance, power transitions, and stopping response. Perception and interaction policies also require RP1-specific evaluation. These are planned extensions, not additional completed tests.

8 Conclusion

RP1 integrates modular mechanics, power distribution, device abstraction, and a motion-learning workflow in a humanoid research platform. Its 24- and 28-DoF body configurations share the lower-body architecture and support alternative arm and perception modules. The documented workflow connects these configurations to model preparation, motion learning, deployment, and condition-specific module testing. Reproducible use requires the delivered configuration to match the model and each result to retain its settings and evidence. The reported records do not establish unrestricted payload capability or field reliability.

Appendices

Appendix A Test Methods and Planned Conditions

This protocol outline summarizes Section 6 and identifies conditions to retain for subsequent evaluations. It is not a complete reproducible test procedure and introduces no new results.

Endurance: fix the module, fixture, load, trajectory, controller, and duration. Record temperature, feedback torque, tracking error, faults, and post-test inspection; log cooling and battery changes separately from effective operating time.

Power and stopping: identify the supply configuration, initial state, trigger, and recovery sequence. Capture electrical and motion response with synchronized timestamps and define acceptance limits before testing.

Task evaluation: separate physical trials from simulated sessions. Retain initial conditions, load, policy version, success criteria, and independent trial count. Repeated cycles within one session are not independent experiments.

Appendix B Configuration Checks

Match body DoF, arm option, end effectors, and perception modules to the CAD/BOM and URDF or MJCF revision. Verify joint names, order, direction, zero positions, limits, and the parallel-ankle transformation.

Record firmware, host image, bus assignments, control rates, policy checkpoint, and protection settings. Logs must identify the actual configuration; check stop and recovery paths under the applicable test procedure.

Appendix C RP Hand

C.1 Configuration and Specifications

RP Hand has 11 mechanical DoF: six active and five coupled. The thumb has active abduction and flexion; the other fingers each have active flexion. Proprietary brushless coreless modules, linkages, miniature encoders, and a multi-axis driver fit inside the hand. It mounts at the RP1 wrist; specifications are in Table C.1 [20].



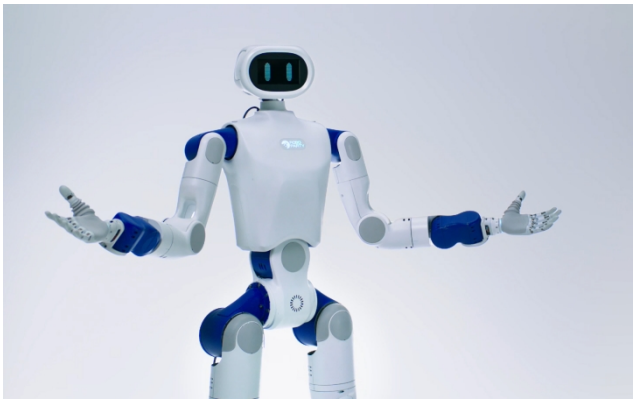


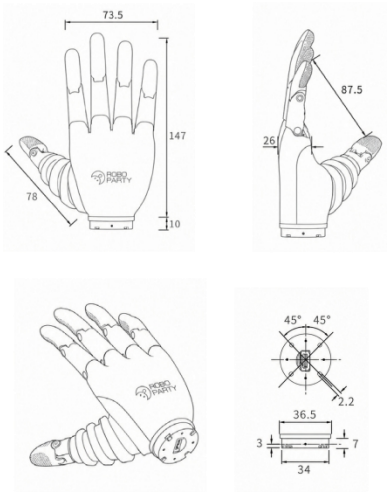
Figure C.1. RP Hand integration with RP1. Source: RP Hand [20].

Table C.1. RP Hand structural and electrical specifications.

Parameter	Specification
Dimensions	147.54 x 73.53 x 26.35 mm
Mass	379 +/- 10 g
Mechanical DoF and active axes	11 mechanical DoF; 6 active axes and 5 passive/coupled DoF
Transmission	Worm gear
Thumb range	Abduction: 0-87.5 degrees; flexion: 0-65 degrees
Other finger flexion	0-72 degrees
Angular resolution	1 degree
Supply input	12 V (-10%) to 48 V (+10%)
Quiescent / maximum current	<= 0.1 A / <= 2.5 A
Minimum opening/closing time	<= 0.7 s

C.2 Dimensions and Electrical Interface

RP Hand connects through CAN-FD with internal routing through the robot. The six-pin wrist connector uses pins 1 and 2 for VDC, 5 and 6 for GND, and 3 and 4 for CANH and CANL. Figure C.2 shows dimensions and pin assignments.



Power and CANFD Connector Pinout

Connector pin assignment and signal direction

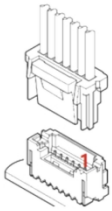
Illustration	Pin	Signal	I/O	Name
	1	VDC	Input	Power input, 12 (-10%) to 48 V (+10%)
	2	VDC	Input	
	5	GND	Output	
	6	GND	Output	
	3	CANH	Input / Output	CANFD
	4	CANL	Input / Output	

Figure C.2. RP Hand dimensions and connector pin assignments. Source: RP Hand [20].

C.3 Software and Developer Interfaces

Host debugging software, a vision-based gesture demonstration, and C++/Python SDKs support commissioning and integration.

The host application displays joint states, configures parameters, and commands axes numerically or by slider. A 3D model follows joint position. Teaching queues can be recorded, imported, exported, and replayed.

The vision-based gesture demonstration is available as applications for Windows and Linux x86-64.

roboparty_dexhand provides C++/Python commands for six-axis position/angle, speed, and current limits, with feedback for position, current, status, and alarms. It handles CAN-FD on Linux x86-64 and ARM64 and ships in RoboParty images. ROS 2 nodes support dual-hand control.

Left/right CAD, URDF, and MJCF support fit checks and model integration (Table C.2).

Table C.2. RP Hand software and models.

Software / model	Function	Platform / format
Host debugging tool	Communication, joint control, parameters, and state	Windows
Vision gesture demo	Vision-based gesture recognition	Windows / Linux
SDK	Six-axis control and feedback	Linux x86-64 / ARM64; C++ / Python
3D structures	Left/right geometry and mounting	STL
URDF / MJCF	Robot-model loading	URDF / MJCF

The source draft describes a linkage transmission in prose and a worm-gear transmission in the specification table. Both descriptions are retained pending confirmation against the delivered RP Hand revision.

Appendix D References

Numbered references identify publications, standards, software, and engineering records. The recorded default access date for verified online sources is October 17, 2026, unless an individual entry specifies another date. Unpublished or unavailable sources are marked; source attribution does not independently validate reported performance.

[1] K. M. Lynch and F. C. Park, Modern Robotics: Mechanics, Planning, and Control. Cambridge, UK: Cambridge University Press, 2017. modernrobotics.northwestern.edu/nu-gm-book-resource/.

[2] J. P. Araujo, Y. Ze, P. Xu, J. Wu, C. K. Liu, "Retargeting Matters: General Motion Retargeting for Humanoid Motion Tracking," arXiv:2510.02252, 2025. doi: 10.48550/arXiv.2510.02252. arxiv.org/abs/2510.02252.

[3] RoboParty, "RP1 hardware: mechanical source files, drawings, and configuration bills of materials," engineering repository, 2026. Revision not pinned in the supplied manuscript. github.com/Roboparty/rp1_hardware.

[4] B. Katz, J. Di Carlo, and S. Kim, "Mini Cheetah: A platform for pushing the limits of dynamic quadruped control," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), 2019, pp. 6295-6301. doi: 10.1109/ICRA.2019.8793865. doi.org/10.1109/ICRA.2019.8793865.

[5] H. Shi, W. Wang, S. Song, C. K. Liu, "ToddlerBot: Open-Source ML-Compatible Humanoid Platform for Loco-Manipulation," arXiv:2502.00893, 2025. doi: 10.48550/arXiv.2502.00893. arxiv.org/abs/2502.00893.

[6] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10975-10985. smpl-x.is.tue.mpg.de/.

[7] RoboParty, "RP1_description: URDF, MJCF, and mesh models," software repository, 2026. Commit not pinned in the source draft. github.com/Roboparty/RP1_description.

[8] Y. Zhao, "Detailed workflow for exporting URDF from SolidWorks," ver. 1.0, RoboParty internal workflow document, 2026. Unpublished; internal access required.

[9] RoboParty, "human-humanoid-tools (hhtools)," software and project documentation, 2026. Implementation inspected at commit 6e6588c (16 Oct. 2026); accessed 19 Oct. 2026. Repository and README | Project page | Agent interfaces. Parameters and performance are version-specific.

[10] Isaac Lab contributors, "Isaac Lab 3.0.0-EA documentation: environment and humanoid benchmarks; Warp environment performance," online technical documentation, 2026. Upstream benchmark evidence; not an RP1 hardware benchmark. isaac-sim.github.io/IsaacLab/v3.0.0-EA/source/setup/environments.html#benchmarks | isaac-sim.github.io/IsaacLab/v3.0.0-EA/source/concepts/warp_environments.html#performance-comparison.

[11] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, C. K. Liu, "BeyondMimic: From Motion Tracking to Versatile Humanoid Control via Guided Diffusion," arXiv:2508.08241, 2025. doi: 10.48550/arXiv.2508.08241. arxiv.org/abs/2508.08241.

[12] RoboParty, "MimicLite," motion-tracking software repository, 2026. The source report cites v1.1; benchmark settings and cost denominators require the corresponding release records. github.com/Roboparty/MimicLite.

[13] RoboParty, "UFO," unsupervised body foundation model software repository, 2026. Reported RP1 results refer to project configurations; commit and full trial records were not supplied. github.com/Roboparty/UFO.

[14] RoboParty, "P-BFM and I-BFM perception and interaction development materials," unpublished project records, 2026. Title supplied by the working manuscript; authors, version, and public publication identifier are not available.

[15] RoboParty, "INTACT intent and task-organization materials," unpublished project records, 2026. Title supplied by the working manuscript; authors, version, and public publication identifier are not available.

[16] International Organization for Standardization, ISO 11898-1:2015, Road Vehicles - Controller Area Network (CAN) - Part 1: Data Link Layer and Physical Signalling, 2015. This is the edition cited by the engineering source, not a claim that it is the latest edition. www.iso.org/standard/63648.html.

[17] USB Implementers Forum, Universal Serial Bus Specification, Revision 2.0, 2000; and Universal Serial Bus 3.0 Specification, Revision 1.0, 2008. Protocol nominal rates are distinguished from measured application throughput. www.usb.org/document-library/usb-20-specification | www.usb.org/document-library/usb-30-specification.

[18] IPC, IPC-2221, Generic Standard on Printed Board Design. The engineering manuscript does not identify the edition used for its current-capacity calculation; edition confirmation remains required. www.ipc.org/ipc-design-standards.

[19] ROS contributors, "URDF XML documentation" and "SolidWorks to URDF exporter," software documentation. wiki.ros.org/urdf/XML | github.com/ros/solidworks_urdf_exporter.

[20] RoboParty, "RP Hand product specifications, mechanical models, URDF/MJCF, and SDK materials," internal product documentation, 2026. Dimensions and ratings in Appendix C are supplier specifications; no external test certificate or document revision was supplied.

[21] RoboParty engineering teams, "RP1 electrical, embedded, algorithm, and reliability evidence package," unpublished engineering records, 2026. Includes RP1.3.2/RP1.3.4 CAD, BOMs, PCB/Gerber data, simulation configurations, October 12, 2026 arm-test records, and reliability_v1 workspace state dated October 17, 2026. Internal access required; evidence classes are indexed in Appendix G.

[22] NVIDIA, "SOMA-Retargeter," software repository. github.com/NVIDIA/SOMA-Retargeter.

[23] Amazon FAR, "holosoma," humanoid learning and retargeting software repository. github.com/amazon-far/holosoma.

[24] Newton contributors, "Newton," physics-simulation software repository. github.com/newton-physics/newton.

[25] NVIDIA, "Warp," GPU simulation and computation framework. github.com/NVIDIA/warp.

[26] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, M. J. Black, "AMASS: Archive of Motion Capture as Surface Shapes," arXiv:1904.03278, 2019. doi: 10.48550/arXiv.1904.03278. arxiv.org/abs/1904.03278 | amass.is.tue.mpg.de/.

[27] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, C. Pal, "Robust Motion In-betweening," arXiv:2102.04942, 2021. doi: 10.48550/arXiv.2102.04942. arxiv.org/abs/2102.04942 | doi.org/10.1145/3386569.3392480 | github.com/ubisoft/ubisoft-laforge-animation-dataset.

[28] I. Mason, S. Starke, and T. Komura, "Real-time style modelling of human locomotion via feature-wise transformations and local motion phases," ACM Transactions on Graphics, vol. 41, no. 3, 2022. doi: 10.1145/3522618. Associated resource: 100STYLE. doi.org/10.1145/3522618 | www.ianxmason.com/100style/.

[29] J. Saito, J. Li, M. de Ruyter, et al., "SOMA: Unifying Parametric Human Body Models," arXiv:2603.16858, 2026. doi: 10.48550/arXiv.2603.16858. arxiv.org/abs/2603.16858 | huggingface.co/datasets/bones-studio/seed.

[30] J. Lin, A. Zeng, S. Lu, Y. Cai, R. Zhang, H. Wang, L. Zhang, "Motion-X: A Large-scale 3D Expressive Whole-body Human Motion Dataset," arXiv:2307.00818, 2023. doi: 10.48550/arXiv.2307.00818. arxiv.org/abs/2307.00818 | github.com/IDEA-Research/Motion-X.

[31] K. Lee, S. Kim, Y. Lee, et al., "PHUMA: Physically Reliable Humanoid Locomotion Dataset," arXiv:2510.26236, 2025. doi: 10.48550/arXiv.2510.26236. arxiv.org/abs/2510.26236 | github.com/DAVIAN-Robotics/PHUMA.

[32] Z. Shen, H. Pi, Y. Xia, et al., "World-Grounded Human Motion Recovery via Gravity-View Coordinates," arXiv:2409.06662, 2024. doi: 10.48550/arXiv.2409.06662. arxiv.org/abs/2409.06662 | doi.org/10.1145/3680528.3687565 | github.com/zju3dv/GVHMR.

[33] J. Li, J. Wu, C. K. Liu, "Object Motion Guided Human Motion Synthesis," arXiv:2309.16237, 2023. doi: 10.48550/arXiv.2309.16237. arxiv.org/abs/2309.16237 | huggingface.co/datasets/YaojieShen/hhtools_omomo.

[34] R. Yu, X. Lin, J. Ma, et al., "OmniContact: Chaining Meta-Skills via Contact Flow for Generalizable Humanoid Loco-Manipulation," arXiv:2606.26201, 2026. doi: 10.48550/arXiv.2606.26201. arxiv.org/abs/2606.26201 | huggingface.co/datasets/lightcone02/OmniContact-Dataset.

[35] L. Yang, X. Huang, Z. Wu, et al., "OmniRetarget: Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction," arXiv:2509.26633, 2025. doi: 10.48550/arXiv.2509.26633. arxiv.org/abs/2509.26633.

[36] M. Xu, Y. Shi, K. Yin, X. B. Peng, "PARC: Physics-based Augmentation with Reinforcement Learning for Character Controllers," arXiv:2505.04002, 2025. doi: 10.48550/arXiv.2505.04002. arxiv.org/abs/2505.04002 | doi.org/10.1145/3721238.3730616 | huggingface.co/datasets/YaojieShen/hhtools_parc_ms.

[37] MotionDecode contributors, "MotionDecode," motion-data project and dataset. Version and release date are not specified in the source manuscript. chingmudata.github.io/MotionDecode/ | huggingface.co/datasets/CMRobot/MotionDecode.

[38] RoboParty, "hhtools demonstration assets," project videos and animated examples, 2026. Figure frames are extracted from the original project assets. Demonstration gallery | URDF import | Calibration | Batch export. Accessed 19 Oct. 2026.

[39] RoboParty, "gpt-6-astra-real2sim-workflow," software repository, 2026. Performance claims in Section 5 follow the project report; exact run settings were not included in the manuscript. github.com/Roboparty/gpt-6-astra-real2sim-workflow.

[40] RoboParty, "LabWeft," research-workbench software repository, 2026. github.com/Roboparty/LabWeft.

[41] Sim2Real Benchmark, "Humanoid motion-tracking leaderboard," online benchmark. Rankings are time-dependent; the manuscript does not retain an immutable leaderboard snapshot. egalahad.github.io/sim2real/leaderboard/.

[42] J. M. Soucie et al., "Range of motion measurements: Reference values and a database for comparison studies," Haemophilia, vol. 17, no. 3, pp. 500-507, 2011. doi: 10.1111/j.1365-2516.2010.02399.x. doi.org/10.1111/j.1365-2516.2010.02399.x | archive.cdc.gov/www_cdc_gov/ncbddd/jointrom/index_1715172647.html.

[43] A. Roaas and G. B. J. Andersson, "Normal range of motion of the hip, knee and ankle joints in male subjects, 30-40 years of age," Acta Orthopaedica Scandinavica, vol. 53, no. 2, pp. 205-208, 1982. doi: 10.3109/17453678208992202. doi.org/10.3109/17453678208992202.

Appendix E Humanoid Platform Index

The 49 entries correspond to Figure 1.2. Heights are nominal figure values, not standardized measurements. Sources identify model versions and unresolved entries; asterisks follow the figure.

* YOMI: 120 cm is a provisional author-supplied figure value. Other entries with differing published generations are annotated in the source column. Source links were recovered from the figure-production records; they do not establish uniform measurement conditions.

ID	Platform	Height	Specification and image provenance
E01	ROBOTIS MINI	26.9 cm	Manufacturer brochure hosted by distributor; humanoid kit configuration. Specification: www.consulta.cz | Image: emanual.robotis.com
E02	Zeroth M1	49.4 cm	494 mm specification; product image via a design publication. Specification: www.zeroth0.com | Image: www.yankodesign.com
E03	ROBOTIS OP3	51 cm	510 mm specification; image via distributor. Specification: en.robotis.com | Image: api.robomercato.com
E04	Luvbotics RUMI	65 cm	65 cm shown in the manufacturer dimension illustration. Specification: www.luvbotics.com | Image: luvbotics.com
E05	HighTorque Mini Pi Plus	68 cm	680 mm on the product page; store material also reports 65 cm. Version-specific. Specification: www.hightorque.cn | Image: store.hightorque.cn
E06	Zeroth Bridge	88 cm	88 cm specification; image from a public product report. Specification: zeroth-openbridge.com | Image: n.sinaimg.cn
E07	PrimeBOT Q1	88 cm	88 cm specification; low-resolution official image. Specification: www.primebot.com | Image: t-vcsh-www-tos-01.tos-cn-shanghai.volces.com
E08	Lumos Xiaoming	89 cm	89 cm from a news report; official brand image. Specification: www.dutenews.com | Image: www.lumosbot.tech
E09	Booster K1	95 cm	Approximately 95 cm on the product page. Specification: www.booster.tech | Image: www.booster.tech
E10	Noetix Bumi	98 cm	98 cm specification; distributor image. Specification: www.noetixrobotics.com | Image: sonnyrobotics.com

ID	Platform	Height	Specification and image provenance
E11	PrimeBOT T1	100 cm	100 cm specification; standing frame from manufacturer sequence. Specification: www.primebot.com | Image: www.primebot.com
E12	Fauna Robotics Sprout	107 cm	Approximately 107 cm in the manufacturer paper; distributor image. Specification: arxiv.org | Image: valumech.com
E13	Noetix N2	118 cm	118 cm retained; some material rounds to 120 cm. Distributor image. Specification: www.noetixrobotics.com | Image: sonnyrobotics.com
E14	Luvbotics YOMI	120 cm*	120 cm is an author-supplied provisional layout value, NOT a verified manufacturer height. Specification: luvbotics.com | Image: cdn.luvbotics.com
E15	JAKA pi	122 cm	1220 mm specification. Specification: www.jaka.com | Image: www.jaka.com
E16	Unitree R1	123 cm	1230 mm on the retained product page; configurations vary. Specification: www.unitree.com | Image: gmo-humanoid.shop
E17	Galbot ET1	125 cm	1250 mm specification. Specification: www.galbot.com | Image: oss-cn-beijing.galbot.com
E18	AgiBot Lingxi X2	131 cm	1310 mm specification. Specification: www.agibot.com.cn | Image: agibot.com
E19	Unitree G1	132 cm	1320 mm on the retained current page; earlier material differs. Specification: www.unitree.com | Image: unitree.ru
E20	Tiangong Omni	135 cm	135 cm specification. Specification: www.x-humanoid.com | Image: www.x-humanoid.com
E21	RoboParty RP1	135 cm	Author-supplied platform height and render [21].
E22	HONOR Yuanqi A1	136.9 cm	136.9 cm reported by news media citing the manufacturer; official brand image. Specification: www.bjnews.com.cn | Image: www-file.honor.com
E23	Galaxea Kengo	139.6 cm	1396 mm specification; product-launch image. Specification: galaxea-dynamics.com | Image: file.aibangbots.com
E24	Booster T2	140 cm	Approximately 140 cm on the product page. Specification: www.booster.tech | Image: www.booster.tech
E25	Noetix E1	140 cm	Current retained documentation: 1400 mm; earlier material: 1360 mm. Match image and version. Specification: web.noetixrobotics.com | Image: sonnyrobotics.com
E26	ENGINEAI PM01	140 cm	Current retained documentation: approximately 1400 mm; earlier: 1388 mm. Specification: engineai.com.cn | Image: cn-engineai-1304599088.cos.ap-guangzhou.myqcloud.com
E27	Boston Dynamics Atlas HD	150 cm	Legacy hydraulic Atlas HD, retired in 2024. Height and Boston Dynamics-credited photo: IEEE ROBOTS guide. Distinct from the 190 cm 2026 production Atlas. Specification: robotsguide.com | Image: cdn.sanity.io
E28	Lumos Robotics LUS2	160 cm	Current LUS2 official product-page render, bearing LUS II 2+ markings; official page height 160 cm. Specification: www.lumosbot.tech | Image: www.lumosbot.tech
E29	LimX Dynamics Luna	160 cm	160 cm from the official specification. Original full-body product render from the official Luna page; background removal only. Specification: www.limxdynamics.com | Image: limx-video-oss.limxdynamics.com
E30	Fourier GR-1	165 cm	Official GR-1 dimensions: 1650 x 524 x 325 mm. Distributor product render. Specification: support-old.fftai.com | Image: sonnyrobotics.com
E31	Fourier GR-3	165 cm	Model-specific source retained in the figure-production manifest. Specification: support.fftai.com | Image: sonnyrobotics.com
E32	LimX Dynamics Oli	165 cm	Model-specific source retained in the figure-production manifest. Specification: www.limxdynamics.com | Image: aifitlab.com

ID	Platform	Height	Specification and image provenance
E33	1X NEO	168 cm	Official height 5 ft 6 in = 167.64 cm; rounded to 168 cm. Specification: www.1x.tech | Image: newatlas.com
E34	AgiBot A2 Ultra	169 cm	Model-specific source retained in the figure-production manifest. Specification: agibot.com | Image: agibot.com
E35	AgiBot A3	173 cm	Model-specific source retained in the figure-production manifest. Specification: agibot.com | Image: agibot.com
E36	Apptronik Apollo	173 cm	Original Apollo, not Apollo 2. Height 5 ft 8 in = 172.72 cm, rounded. Specification: apptronik.com | Image: assets.website-files.com
E37	Figure Figure 03	173 cm	Official current Figure page: 5 ft 8 in = 172.72 cm; rounded to 173 cm. Photo is the Figure 03 launch image. Specification: www.figure.ai | Image: ixbt.online
E38	Tesla Optimus Gen 2	~173 cm	Approximate published height; IEEE ROBOTS guide, manufacturer-credited photograph. Specification: robotsguide.com | Image: cdn.sanity.io
E39	ENGINE AI T800	173 cm	Model-specific source retained in the figure-production manifest. Specification: www.engineai.com.cn | Image: cn-engineai-1304599088.cos.ap-guangzhou.myqcloud.com
E40	Fourier GR-2	175 cm	Official GR-2 height 1.75 m; product render from the RBTX marketplace. Specification: support-old.fftai.com | Image: images.cdn.europe-west1.gcp.commercetools.com
E41	Agility Robotics Digit (V4)	175 cm	2023 commercial Digit with head and hands, not the 2026 V5. Deployment partner GXO reports 5 ft 9 in (175.26 cm), rounded to 175 cm. Distributor product cutout. Specification: gxo.com | Image: api.robgen.com
E42	Kepler K2	175 cm	Bipedal K2; current official specification is 175 cm. Specification: www.gotokepler.com | Image: images.cdn.europe-west1.gcp.commercetools.com
E43	PAL Robotics TALOS	175 cm	Model-specific source retained in the figure-production manifest. Specification: pal-robotics.com | Image: pal-robotics.com
E44	UBTECH Walker S2	176 cm	Model-specific source retained in the figure-production manifest. Specification: docs.ubtrobot.com | Image: images.cdn.europe-west1.gcp.commercetools.com
E45	Xiaomi CyberOne	177 cm	2022 CyberOne, 177 cm in Xiaomi launch release. Original Xiaomi award-entry render from iF Design, project 615888. Specification: new.c.mi.com | Image: ifdalivestorage.blob.core.windows.net
E46	NEURA Robotics 4NE1	180 cm	Model-specific source retained in the figure-production manifest. Specification: neura-robotics.com | Image: neura-robotics.com
E47	Unitree H1	180 cm	Original H1; the official height is approximately 180 cm. Not H1-2 (178 cm). Specification: www.unitree.com | Image: robostore.com
E48	Unitree H2	182 cm	Official technical table: 1820 mm. Human-face H2, not H2 PLUS. Specification: www.unitree.com | Image: www.unitree.com
E49	Boston Dynamics Atlas (2026)	190 cm	2026 production design and specification; not the 2024 electric prototype. Specification: bostondynamics.com | Image: bostondynamics.com

Additional software-supported platforms are listed below. Comparable heights were not supplied.

Organization	Model	Height	Location / source
RoboParty	RPO	Unspecified	hhtools integration list [9]
RoboParty	ROBOTO_ORIGIN	Unspecified	hhtools integration list [9]
Asimov	Asimov 1	Unspecified	hhtools integration list [9]
Berkeley	Humanoid Lite	Unspecified	hhtools integration list [9]

Appendix F Figure and Table Index

Chapter-numbered figures and tables with source attribution. Internal engineering records are distinguished from external benchmarks and product/anatomy references.

Identifier	Title and provenance
Figure 1.1	Three views of RP1 in the white-and-blue color scheme. Source: RoboParty [21].
Figure 1.2	Humanoids by nominal height, 26.9-190 cm; RP1 is blue. Versions and sources: Appendix E. Heights do not share a measurement protocol.
Figure 1.3	Human poses, SMPL-X meshes, and RP1 morphology. Sources: SMPL-X [6] and RoboParty [21].
Figure 1.4	RP1 actuator-group correspondence with SMPL-X human joints. Sources: SMPL-X [6] and RoboParty [21]. Anatomical illustration source: Anatomia, https://anatomia.com.mx/anatomia/lineaOsteo/MicroSitio/Esqueleto/index.html.
Figure 1.5	ROMOMO joint module series. Source: RoboParty [21].
Figure 1.6	RMM6562-2E operating points for five representative motion conditions. Source: RoboParty [21].
Figure 1.7	RMM8462-2E operating points for combined lateral raises and elbow lifts with a 5.5 kg end load. Source: RoboParty [21].
Figure 1.8	RMM10062-2E operating points for five representative motion conditions. Source: RoboParty [21].
Figure 1.9	Clamp interface and split-clamp removal. Source: RoboParty [21].
Figure 1.10	RP1.3.4 joint-axis groups, shell layout, and motion-clearance regions. P: pitch; R: roll; Y: yaw. Axis locations and clearance regions are schematic. Source: RoboParty [21].
Figure 1.11	Human and RP1 elbow flexion. Human reference: [42]; RP1 limits and photographs: [21].
Figure 1.12	Left-arm geometry and projected joint-origin spacing. S: shoulder pitch; E: elbow; Wp: wrist pitch. Dimensions follow the supplied CAD drawing and illustrated pose. Source: RoboParty [21].
Figure 1.13	Waist architectures: EngineAI T800, Unitree R1, Unitree G1 EDU, and RP1. P: pitch; R: roll; Y: yaw. Model variants are as labeled; images are not to scale. Source: supplied comparison artwork; manufacturer sources in Appendix E; RP1 [7, 21].
Figure 1.14	Hip reference architectures and RP1 joint layout. Axis sketches indicate neutral-pose directions, not coincident origins. Source: supplied comparison artwork and labeled robot sources; anatomy credited to Omnia Medical; RP1 [7, 21].
Figure 1.15	Human and RP1 hip adduction. Human reference: [43]; RP1 limit and photographs: [21].
Figure 1.16	Human and RP1 knee flexion. Human reference: [42]; RP1 limits and photographs: [21].
Figure 1.17	Left-leg geometry and sagittal-plane axis spacing at the URDF zero pose. H: hip pitch; K: knee; Ap: functional ankle-pitch reference. Source: RoboParty [21].
Figure 1.18	Human and RP1 ankle dorsiflexion. Human reference: [42]; RP1 limits and photographs: [21].
Figure 1.19	Human and RP1 ankle plantarflexion. Human reference: [42]; RP1 limits and photographs: [21].
Figure 1.20	Foot and ankle free-body diagrams. (a) Side view of the foot and ankle-pitch assembly. (b) Front view of the isolated foot. Weight terms follow the corresponding free-body boundary; moment arms depend on the operating condition. Source: RoboParty [21].
Figure 1.21	DFM, DFA, and DFS framework. Source: RoboParty [21].
Figure 1.22	Material tiers and joint module families. Source: RoboParty [21].
Figure 1.23	RP1 body components. Source: RoboParty [21].
Figure 1.24	RP1 core structural components. Source: RoboParty [21].
Figure 1.25	Details of core components. Source: RoboParty [21].
Figure 1.26	RP1 wrist interface and three interchangeable end effectors. Source: RoboParty [21].
Figure 1.27	Removal of the RP1 head perception module. Source: RoboParty [21].

Identifier	Title and provenance
Figure 1.28	Perception, data-collection, and motion-control head configurations. Source: RoboParty [21].
Figure 1.29	Selected quick-release interfaces in RP1 leg modules. Source: RoboParty [21].
Figure 1.30	Head-torso link. Source: RoboParty [21].
Figure 1.31	Link hierarchy. Source: RoboParty [21].
Figure 1.32	Principal joint axes and rotation directions. Red arrows indicate axis directions; green arrows indicate the corresponding rotations. The illustration supports checking joint conventions against the mechanical assembly. Source: RoboParty [21].
Figure 1.33	Chest-shell geometry before and after CAD simplification. (a) Original model, with detailed regions highlighted. (b) Simplified model for export. Source: RoboParty [21].
Figure 1.34	Example of surface damage after aggressive mesh decimation. The highlighted head-torso region illustrates why visual inspection and contact-boundary checks are required. Source: RoboParty [21].
Figure 2.1	RP1 electrical topology: power, communication, and stop paths. Source: RoboParty [21].
Figure 2.2	Top and bottom Gerber renderings of MAIN and BACKHUB, not to a common scale. Source: RoboParty [21].
Figure 2.3	Top-layer Gerber renderings of HOTSWAP, HANDPlusPro, and MCAP. Source: RoboParty [21].
Figure 2.4	Front-view line drawing of RP1 external interfaces. Source: RoboParty [21].
Figure 2.5	RP1 main-battery removal and replacement. Source: RoboParty [21].
Figure 2.6	RP1 stop inputs and example software interface. Source: RoboParty [21].
Figure 3.1	Embedded execution architecture. (a) Control, driver, operating-system, and bridge layers. (b) Joint-command and actuator-feedback paths. Source: RoboParty [21].
Figure 3.2	Theoretical CAN/USB traffic and payload timing. The 11.84% value excludes framing and other delays; unused area is not validated margin. Source: RoboParty [21].
Figure 3.3	Hardware interfaces and parallel-joint transforms; robot BSP configuration; GitHub Actions and APT delivery. Source: RoboParty [21].
Figure 3.4	Session archives, checksums, and offline analysis; communication, recovery, and functional checks. Dashed paths denote reanalysis. Coverage does not imply acceptance. Source: RoboParty [21].
Figure 4.1	hhtools retargeting: carrying, dance, steps, and obstacles. Source: hhtools [9, 38].
Figure 4.2	RP1 reference-motion tracking. (a) Sixteen parallel environments. (b) Single-environment policy replay. Source: RoboParty [21].
Figure 4.3	Published humanoid benchmarks [10]. (a) Training throughput by physics preset. (b) Torch and Warp environment-step times. Source: Isaac Lab public benchmark [10].
Figure 4.4	Torque-speed envelope for RP1 waist, hip, and knee actuators. Source: RoboParty [21].
Figure 4.5	Three-frame observation history in (a) obs-major and (b) frame-major order. Source: RoboParty [21].
Figure 4.6	Analytical left-ankle geometry. (a) Zero pose. (b) Combined pitch and roll. Source: RoboParty [21].
Figure 4.7	Whole-body posture tracking and manipulation demonstrations. Source: RoboParty [21].
Figure 4.8	Loaded upper-limb manipulation. Source: RoboParty [21].
Figure 4.9	WBI development workflow. Source: RoboParty [21].
Figure 4.10	Synchronized video and joint-state replay. Source: RoboParty [21].
Figure 6.1	Modular test benches and quick-release fixtures. Design renderings. Source: RoboParty [21].
Figure 6.2	Physical endurance motions: upper-body arm swing and loaded lower-body stepping. Source: RoboParty [21].
Figure 6.3	Joint motion, torque, and 12-axis temperature during squatting with a 50 kg external load. Source: RoboParty [21].
Figure 6.4	Simulated design of continuous single-arm loaded motions at 2.5, 3.5, and 5.5 kg. Source: RoboParty [21].

Identifier	Title and provenance
Figure 6.5	Physical single-arm motions at 2.5, 3.5, and 5.5 kg. Source: RoboParty [21].
Figure 6.6	Key joint responses during continuous single-arm loading. Source: RoboParty [21].
Figure 6.7	Test software and evidence-data architecture. Source: RoboParty [21].
Figure C.1	RP Hand integration with RP1. Source: RP Hand materials [20].
Figure C.2	RP Hand dimensions and connector pin assignments. Source: RP Hand materials [20].
Table 1.1	RP1 body configurations.
Table 1.2	Transmission characteristics. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.
Table 1.3	Actuator integration comparison. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.
Table 1.4	Reducer comparison. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.
Table 1.5	Planetary arrangements. Source: original RP1 design comparison [21]; qualitative and configuration-dependent.
Table 1.6	RP1 joint limits and human reference ranges. Robot zero positions and signs differ from anatomical conventions; the human references are identified locally below the table.
Table 1.7	Hip serial order.
Table 1.8	Material selection.
Table 2.1	Board roles. Source: RoboParty engineering records [21].
Table 2.2	Target impedances. Source: PCB design references [18] and engineering targets [21].
Table 2.3	Interfaces and nominal rates. Source: protocol specifications [16, 17] and interface configuration [21].
Table 3.1	Hardware abstractions. Source: RoboParty engineering records [21].
Table 3.2	Tests and checks. Source: RoboParty engineering records [21].
Table 4.1	Retargeting workflow comparison. Source: retargeting methods and software [2, 9, 22, 23].
Table 4.2	Dataset inputs. Source: dataset and project records [9, 26-37].
Table 4.3	Policy configurations. Source: robot model [7] and policy configurations [21].
Table 4.4	Recorded data. Source: RoboParty engineering records [21].
Table 6.1	Test conditions and representative physical records. Source: RoboParty [21].
Table C.1	RP Hand specifications. Source: RP Hand product specifications [20].
Table C.2	RP Hand software and models. Source: RP Hand SDK and model records [20].

Appendix G Engineering Evidence Index

ID	Engineering record	Contents	Report location	Provenance
G1	CAD / STEP	Whole robot, legs, left arm, and joint parts	Sections 1.4-1.7	RoboParty internal engineering files; versions as captioned
G2	BOM and drawings	Parts, materials, specifications, and configurations	Section 1.6	RoboParty internal records [3, 21]
G3	RP1_description	URDF, MJCF, meshes, and configurations	Sections 1.7 and 4	Public repository [7]; commit not pinned in source draft
G4	hhtools	Retargeting, calibration, adapters, and quality analysis	Section 4.1	Public repository [9]; implementation described at commit 6e6588c. Experimental run versions require separate records.

ID	Engineering record	Contents	Report location	Provenance
G5	Isaac Lab / Newton / PhysX	Training, dynamics, and replay	Section 4.2	Public tools [10, 24, 25] and project configurations
G6	BeyondMimic / MimicLite / UFO	Training, inference, and deployment	Sections 4.2-4.3 and 5	Research and repositories [11-13]
G7	Schematics / PCB / Gerber	Power, CAN-FD, USB, boards, and interfaces	Section 2	RoboParty internal fabrication records [21]
G8	Test videos and logs	Motion, temperature, torque, and fault records	Section 6	Internal physical and simulation records [21]; loads and dates stated in text
G9	reliability_v1 source	Execution, archive, review, and queries	Section 6.4	Workspace state dated October 17, 2026 [21]
G10	Humanoid comparison	49 models and nominal heights	Section 1.2; Appendix E	Retained specification and image-source manifests
G11	RP Hand materials	Structure, interfaces, models, and SDK	Appendix C	RoboParty product and engineering materials [20]