Follow-Me Cart

Job description, constraints and safety requirements

Revision 3 — 2026-11-01. Written around the Permobil base and the hardware Mike already owns.
Base: Permobil M300 / M3 / M1 | **Factory R-net electronics:** owned, boxed
Also owned: dual brushed 35 A controllers, several ESP32s, 24 V batteries + charger
STATUS: BRIEFED — nothing bought. Next: bench-test the R-net, photograph the motors.

Follow-Me Cart — Constraints and Job Description

Revision 3 — 2026-11-01. Rev 1 planned a from-scratch salvage build. Rev 2 was written for a Permobil base with its electronics stripped. **Rev 3 is the accurate one:** the base is a Permobil (M300 / M3 / M1) and **Mike kept the entire factory electronics package** — R-net power module, joystick module and cables, boxed — and additionally owns **dual brushed DC motor controllers rated 35 A** and **several ESP32 boards**.

Answers in §2 are Mike's. They are the authority. Changes are his call, not a worker's.

---

1. What this machine is

A **Permobil power-chair chassis converted into a cargo cart that follows its owner** around the property, carrying tools and gear. Not autonomous; it does not navigate to a destination. The owner walks, the cart follows at walking pace a few feet behind, on gravel, dirt, grass and gentle slopes.

The following is the interesting part. **The carrying is not the hard part** — a power-chair chassis is rated for a 300 lb person, so 150 lb of tools is well inside its design load.

2. The facts

Item	Answer	Consequence
Main job	Haul tools and gear around the property while following me	Cargo box + follow controller
Surface	Gravel, dirt, grass, some slope	Already the chassis's design case
Payload	100–150 lb	**Easy** — base is rated for 300 lb of person
Budget, first build	A few hundred dollars — prove it, then scale	Comfortable: most of the machine is already owned
Base	**Permobil M300 / M3 / M1** — motors, wheels, frame	Mid-wheel drive, 24 V, outdoor, ~300 lb rated
Batteries + charger	**Owned**	Saves $150–250
Factory electronics	**Owned, off the chair and boxed** — R-net **power module**, **joystick module**, cables	**Path A is available (§5)**
Other hardware owned	**Dual brushed DC controllers, 35 A** + **several ESP32s**	**Path B is available (§5)**
Motor type	**Not yet established — pending photos**	Decides Path B; Path A is immune to it (§4)

Derived targets — the worker may refine the margins, not the intent:

- **Speed:** hard cap **3 mph / 1.35 m/s**. The base can do ~6 mph; the cap is ours to enforce.
- **Follow distance:** 3–6 ft, adjustable. Close-follow crawl around **0.5 mph**.
- **Slope:** sustained **10%**, short stretches **15%**.
- **Voltage:** stay **24 V nominal**. Do not series the packs to 48 V.

3. The base — what it gives us

- Two matched **geared drive motors with fail-safe electromagnetic brakes**, on a suspended chassis

with pneumatic wheels, already engineered for gravel and slopes, with a **low centre of gravity and mid-wheel drive** — the drive wheels sit under the load rather than behind it.

- **Batteries and charger.**
- **The complete factory control system, boxed** — and this is the important one. The R-net power

module is the unit that originally drove *these exact motors*. It already contains the brake sequencing, current limiting and speed limiting that would otherwise have to be written from scratch, correctly, by us, on a 300 lb machine.

4. The one physical fact still open

Are the drive motors brushed or brushless? Photos are on their way; until they arrive this is unknown, and it matters for **Path B only**:

Heavy conductors at the motor	Motor type	Consequence
2 wires (+ separate thin brake wires)	**Brushed DC**	Path B works with Mike's 35 A brushed controllers
3 wires + ~5 thin sensor wires	**Brushless**	**Mike's brushed controllers cannot drive these** — Path A, or brushless controllers

Path A does not care either way, because the factory power module drove these exact motors.

Record the answer, with wire counts, part numbers and brake voltage, in `docs/base-survey.md`.

5. Electronics — two paths, not equally good

Brakes first, because they are the safety system and they are already on the machine. Each motor's brake is **normally-closed and spring-applied**: it **locks when power is removed** and releases when fed its rated voltage on its dedicated wires. *A power cut stops the cart through the chair's own brakes, with no software in the loop.* Every design below must preserve that property, never bypass it.

Path A — drive the factory R-net system over CAN (preferred)

R-net is a **CAN 2.0B bus at 125 kbps** between the joystick module (JSM) and the power module (PM). With the PM, JSM and cables in hand, the chair can be commanded by our own computer instead of the joystick. This is **not speculative**: there is open-source reverse-engineered R-Net code (`open-rnet`, and Stephen Chavez's `can2RNET`), and a published teleoperable-wheelchair project that did it on a **Permobil M3** — the same family as this base.

- **Hardware:** an ESP32 plus a CAN transceiver (e.g. SN65HVD230) is about **$5–15** and Mike already

has the ESP32s. A Raspberry Pi + PiCAN2 HAT (~$115) is the better-documented alternative.

- **What we get for free:** factory brake sequencing, current limiting, speed limiting, ramping, and

a system that fails stopped if the bus goes quiet. On a 300 lb chassis, that is the safety case.

- **What is uncertain:** whether the boxed PM powers up and enables drive without a factory

programming session, and how much the protocol needs coaxing. **That is testable in an afternoon** (§6, Phase 1a) and should be tested before anything else is built.

Path B — Mike's 35 A brushed controllers + an ESP32 (fallback, and only if the motors are brushed)

Straightforward, well-trodden, and everything is already owned. **But every safety behaviour becomes ours to implement:** brake release and sequencing, current limiting, speed cap, ramp rates, and the failsafe stop. That is real work on a machine that can break a leg.

- **35 A is the concern.** A wheelchair drive motor can draw **120 A peak** per channel; the factory

controller folds back to 100 A. For cart duty at walking pace, 35 A is plausible — but a 150 lb load on a 10% gravel slope can pull far more than that from a standstill. **Measure the real current before committing**; if it trips, the options are gearing, a current ramp, or bigger controllers.

- If a stronger controller is wanted, the **Sabertooth 2x32** (6–33.6 V, 32 A continuous / 64 A peak

per channel) is the proven unit for wheelchair drivetrains and has **dedicated P1/P2 outputs that drive electromagnetic brakes**, with e-stop logic configured in DEScribe. Brake **+** goes to battery **+**; M1 brake – to **P1**, M2 brake – to **P2**; **blocking diodes are required** if the brakes share the motor feed — missing diodes are the classic half-working failure. The cheaper 2x25 cannot control brakes separately and is not adequate.

Decision rule: bench-test Path A first (Phase 1a). If the factory electronics drive the motors, **use Path A** — not because it is cheaper, but because its safety behaviour is already correct. Fall back to Path B only if Path A cannot be made to work.

6. Phases

Phase 1a — bench-test the boxed factory electronics (an afternoon, ~$15)

Power the PM and JSM on the bench, **drive wheels off the ground**, and find out whether:

1. The system powers up and leaves its factory fault states.
2. Our computer can see R-net traffic on CAN at **125 kbps** (`candump`).
3. A CAN command can move a motor.

Write the result, including every frame ID observed, into `docs/rnet-bench-test.md`. **This decides the whole architecture and costs almost nothing** — do it before specifying anything else.

Phase 1b — make the base drive safely under our control

Done when, wheels off the ground first and then on gravel:

1. Each motor drives forward and reverse, and the cart steers, from our controller.
2. The brakes **lock when power is cut** — verified by *pulling the power*, not by reading the code.
3. The **hand-held e-stop stops it from full speed in under 1 second**, tested **with the software

killed**, not by pressing a software button.

1. It is **capped at 3 mph** and cannot be made to exceed it from any input.
2. It carries **150 lb** of tools around a 100 ft gravel loop without stalling.
3. It runs **30 minutes** of stop-start driving on one charge.

Phase 2 — make it follow

Add camera and tracking on top of a chassis already known to be safe. With ESP32s in hand the tracker can be its own board commanding the drive electronics — the same ESP32 can do the tracking and the commanding, which removes the need for a separate computer.

7. Following — the second decision, after the drive

The objective says **camera-based**; that stands. Options, cheapest first:

- **ESP32-CAM / ESP32 + camera** — owned already; weakest compute, but may be plenty for tracking a

marker.

- **An old Android phone** — camera, battery and Wi-Fi built in, runs person detection offline, and

can command the drive over Bluetooth. Usually the cheapest *capable* option.

- **Raspberry Pi + camera** — the conventional choice, and the natural companion to Path A.

A **retro-reflective marker or vest** the owner wears is a legitimate simplification and must be compared, not dismissed: far cheaper and more reliable than person detection, at the cost of wearing something. **Whichever wins must work in bright sun and low light** — outdoor Colorado, year round.

8. Safety — mandatory

The brakes are the safety system. These are requirements, not options:

- **A hand-held e-stop that cuts motor power independently of software.** On this base a battery

disconnect is a good design, because the brakes clamp when power drops. It must work with the computer crashed or unplugged.

- **Failsafe stop on lost sight or signal.** Losing the target means *stop*, never *continue forward

looking for it*. The code's default, not a setting.

- **A hard speed cap** no input or bug can exceed.
- **No autonomous operation.** It follows the owner, or it stops.
- **Free-wheel levers checked in "drive" before any powered test.** A chair in free-wheel has **no

brakes at all** and will roll on a slope.

- **Nothing above 48 V DC.** No mains wiring on the machine.
- Battery secured to the chassis, nothing loose on top, and the cargo box placed so the load cannot

shift the centre of gravity past the casters.

- **Name every pinch point and tipping risk in the build doc.** A chair drivetrain will not stop for

a hand.

9. Out of scope

Snow, mud, steep ground, above 15% slope. Riding on it — the seat and its actuators go; this is a cart. Autonomy, route planning, return-to-base, app control, following anyone but the owner. Weatherproofing beyond "lives in the shop and tolerates a drizzle."

10. What the worker does not decide

Brake wiring, e-stop design, current limits and the speed cap go to Mike as written decisions, with the numbers and the alternatives, clearly flagged. These are the choices that can hurt someone. Present them, and continue with what does not depend on them. Never settle one inside a parts list.