controller.rs — Gamepad Input

Runs a background listener for gamepad input using the `gilrs` library, translating button and axis events into UDP packets sent to the rover. All dispatching is gated on the current mode (drive vs. pickup) and whether the relevant manual mode is active.

Modes

The controller operates in one of two top-level modes, toggled with the **Start** button:

Mode	Active when	Controls
Drive	`pickup_mode = false`	Left stick (forward/turn), triggers (brake)
Pickup	`pickup_mode = true`	Both sticks (X/Y/rotate/flick), D-pad (Z), triggers (gripper)

Within each mode, the **Select** button toggles the relevant manual mode (`drive_manual_mode` or `arm_manual_mode`). Commands are silently suppressed when the associated manual mode is inactive.

Threads

Three concurrent threads are spawned on startup:

`start_controller_listener()` — Entry point. Spawns all threads and owns the shared `CommandState`.

Event thread — Polls `gilrs` at ~125 Hz (8 ms sleep) and routes each `Event` to the appropriate handler. Holds the `shared` mutex only for the duration of each state update.

Heartbeat thread — Wakes every 2 seconds and re-sends the current state (drive axes + brake, or arm + brake). Ensures the rover never silently drifts from its commanded state if packets are dropped.

Ramp threads — Spawned on demand when a ramped button (D-pad up/down, left/right trigger in pickup mode) is pressed. Ticks at ~60 Hz and increments the axis value from `0.0` toward `±1.0` over `RAMP_DURATION_SECS` (1.0 s). The thread exits when the direction is set back to `0.0` on button release.

Button mapping

Button	Drive mode	Pickup mode

Start	Toggle pickup mode	Toggle pickup mode
Select	Toggle `drive_manual_mode`	Toggle `arm_manual_mode`
Left trigger (LB)	Toggle latching brake	Ramp gripper speed → close (−1.0)
Right trigger (RB)	—	Ramp gripper speed → open (+1.0)
Left trigger 2 (LT)	Toggle latching brake	—
Right trigger 2 (RT)	Momentary brake (hold)	—
D-pad up	—	Ramp Z → up (+1.0)
D-pad down	—	Ramp Z → down (−1.0)

Axis mapping

Axis	Drive mode	Pickup mode
Left stick Y	Forward / backward	Flick
Left stick X	Turn	Rotate
Right stick X	—	End effector X (left/right)
Right stick Y	—	End effector Y (forward/backward)

All analog axes pass through a deadzone (`±0.05`) and are only dispatched when the change from the last-sent value exceeds `AXIS_CHANGE_THRESHOLD` (0.05). Z and gripper speed are not axis-driven — they are ramped from button presses.

Packets sent

Packet	When
`BasestationManualDrive`	Drive mode, on axis change or heartbeat
`BasestationManualBrake`	Drive mode, on brake toggle/press/release or heartbeat; also sent continuously (engaged) during pickup heartbeat
`BasestationManualArmMovement`	Pickup mode, on any arm state change or heartbeat

All values are scaled from `[−1.0, 1.0]` to the full `sint32` range before transmission.

Constants

Constant	Value	Purpose
`AXIS_CHANGE_THRESHOLD`	`0.05`	Deadzone boundary and minimum delta before a packet is sent
`HEARTBEAT_INTERVAL`	`2 s`	How often state is re-sent without an input event

`RAMP_DURATION_SECS`	`1.0 s`	Time for a ramped axis to travel from `0.0` to `±1.0`
`RAMP_TICK_MS`	`16 ms`	Ramp thread tick interval (~60 Hz)
`MOMENTARY_BRAKE_DURATION`	`500 ms`	Defined but unused — auto-release was commented out

---

Revision #4
Created 2026-04-14 09:50:03 UTC by Candela Cimadevilla Gonzalez
Updated 2026-05-05 10:26:33 UTC by Candela Cimadevilla Gonzalez