Asteria

Command Station

A local daemon that lets any AI agent drive a real robot — and the filesystem behind it.

Noah Hicks · Cognitive Robotics · 2026

The daemon is the product.

Asteria is a long-running local service that exposes a safe, structured interface to the robot and the filesystem around it. Agents talk to the daemon — the daemon does the driving.

AA

Agent-agnostic

Any local or cloud agent — Codex, Claude, a custom script — drives through the same daemon API. No per-agent glue code.

Local · Cloud · Custom

RF

Robot + filesystem, together

One session can move the robot, compile a new FSM, edit a file on disk, and capture vision — without context-switching tools.

AIM runtime · FSM · Vision

LS

Lease-gated & safe

Every command is claimed, leased, and stoppable. The daemon owns authority; agents request it. Emergency stop is always one call away.

Claim · Lease · E-Stop

One daemon, many clients.

AGENTS

Codex
Prompt forwarding

Claude
Pair programming

Local scripts
Automations

Asteria DS
Handheld pilot

ASTERIA DAEMON

`aim_fsm_headless`

- Claim / lease authority
- Compile & run FSM programs
- Direct motion (move, strafe, turn)
- Vision capture + preview
- Filesystem read / write / edit
- Prompt queue + session log

WORLD

AIM Robot
Motors · Kicker · Camera

Filesystem
FSM sources · artifacts

AprilTags
World-map anchors

Operator
Claim · Stop · Review

Agents never touch the robot directly. The daemon enforces lease, logs every command, and can E-Stop on demand.

An agent with a body.

The same daemon supports very different users — from a developer pair-programming behaviors to a kid teaching a robot to dance.

FOR DEVELOPERS

Robotics pair programmer

- Describe a behavior — agent drafts an FSM, compiles it, runs it live.
- Iterate in the loop: watch the robot, edit the code, re-run in seconds.
- Vision + AprilTag grounding makes plans check themselves against reality.

Codex · Claude · Local scripts

FOR LEARNERS

Advanced toy & learning system

- "Find the soccer ball and kick it at AprilTag 0" becomes a working program.
- Kids get a real robot that responds to plain language — no IDE required.
- Every session is logged, so lessons can replay, remix, and build on prior runs.

Say it · See it · Save it

Real FSMs, written by the agent.

zigzag_20s.fsm — asteria/artifacts/fsm/ Compiled

```
from aim_fsm import *

class Zigzag20s(StateMachineProgram):
    # Approximate 20-second zigzag. Actual runtime depends on drive
    # floor traction, and the current AIM runtime timing.
    $setup{
    start: Say("Starting zigzag run")
        =C=> Forward(160)
        =C=> Turn(28)
        =C=> Forward(160)
        =C=> Turn(-56)
        =C=> Forward(160)
        =C=> Turn(56)
        =C=> Forward(160)
        =C=> Turn(-56)
    }
```

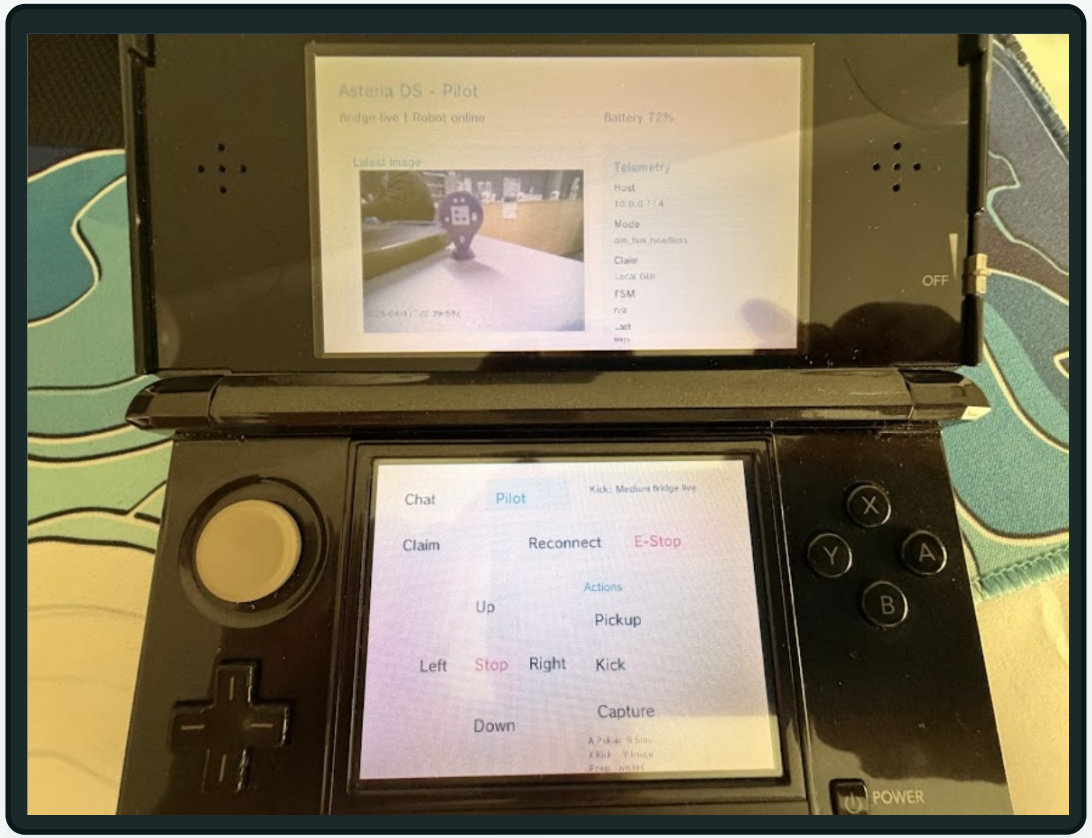
FROM THE FSM BROWSER

A sample of agent-authored state machines:

soccer_tag0_shot.fsm	Find ball · aim at AprilTag-0 · kick
zigzag_20s.fsm	~20s zigzag sweep across the floor
approach_operator_scan.fsm	Approach operator, scan for tags
area_context_scan.fsm	Rotate-scan the room, log findings
follow_me_tag.fsm	Keep a tracked tag centered in view
asteria_star_dance.fsm	Choreographed star pattern
agent_safe_observer.fsm	Idle, watch, narrate — no motion

The daemon in your pocket.

The 3DS acts as a physical command station — a dedicated pilot + chat client that makes the whole system feel like a toy, not a terminal.



1 Dedicated device

A purpose-built controller beats juggling tabs — claim, pilot, chat, stop, all in reach.

2 Physical affordance

Hardware buttons turn abstract commands into motor muscle memory. It feels like a robot toy, not a CLI.

3 Safe by default

Stop and E-Stop are always a thumb away. Authority stays with the human holding the device.

Asteria

Build the daemon once.

Every agent — and every kid — gets a real robot to talk to.

FOR ENGINEERS

A pair programmer that closes the sim-to-real loop.

FOR LEARNERS

A toy that rewards curiosity with real motion.

FOR THE FIELD

A safe, observable bridge between AI and hardware.