15 NOVEMBER 6, 2026 3 PP

Broad Daylight: open-sourcing the tools we use to put AI to work in the physical world

Tim Child · Felipe Arce · Collin Rijock

Broad Daylight is Exowatt's open-source initiative: six tools a small team and its agents use to take code from a phone scan of a site, through simulation, to real hardware. This paper covers what each one does, the numbers we measured, and which parts are public today.

BROAD-DAYLIGHT OPEN-SOURCE PHYSICAL-AI SIMULATION AGENTS INFRASTRUCTURE



Exowatt builds modular, dispatchable solar power for data centers. Our software team is small, so we are building it a software factory: people set the goals, agents do much of the work, and our own tools test and ship it.

Here is what it did in October. On October 8 a DEXI-5 quadcopter flew a 3 m square on its own while a box fan pushed it off course, and a small action model pulled it back half a metre at a time. A coding agent got the request at 20:07 UTC, and the drone was flying it before 23:00.[1] Four days later we compiled a phone scan of our office copy room into scenes for MuJoCo and Isaac Sim.[2] We did both with our own tools.

Broad Daylight, Exowatt's open-source initiative, is how we are sharing the crucial infrastructure behind that factory. It covers six projects: Orb and OrbLang, Copper,

phi, Roadrunner, Feads and Forca. Each one exists to get code working in the physical world: model a real place, test against the model, then run on real equipment.

Why an energy company builds a software factory

Our code runs next to equipment: each site's energy management system runs on a small host at the plant, speaking Modbus and OPC UA to batteries and inverters. In July an agent asked our deploy server for a rollout without naming a version, got an older build, and kept one site's only control process down until we rolled forward within the hour.[3] Agents are now fast enough to do real work on real systems, so we build the tools that keep that work safe.

Simulation needs the same care. On the drone's first real flight, the simulator and the aircraft ran two different control programs, so passing in Isaac Sim had proved little.[1] Physical AI fails in gaps like these, between the model and the place, and we built Orb to narrow them.

What is in it today

Each project has a paper here. One measured number each:

Project	What it does	One number	Code
Orb + OrbLang	Turns a phone scan into a typed program of the site	Scan to Isaac Sim render in about 15 minutes	Private
Copper	A browser agents drive over MCP, plus Copper Cloud	24 of 24 benchmark tasks, 1.9 s median	Public, MIT
phi	Our fork of the pi coding agent	1.24 million requests from ten engineers by mid-October	Private
Roadrunner	Self-hosted CI and deploys on Woodpecker and Dagger	Default deploy path for 199 repos in eight days	Private
Feads	A board where agents claim work and declare files	4,732 overlap answers, 935 across people	Private

Project	What it does	One number	Code
Forca	Our fork of the Orca agent workspace	9,043 upstream commits merged in one sync	Private

Orb and OrbLang

Orb's README calls it "real world as code". OrbLang numbers carry units, so 1 m + 1 kg fails to compile, and one program feeds MuJoCo for fast checks and Isaac Sim for pictures. A test checks that every element lands within 2 mm across six outputs.[2] Collin Rijock designed the language; Felipe Arce built the capture and GPU rendering around it.

Copper and Jev

Copper runs Jev, a small action model from TypeSafe, inside the open tab. Jev picks a browser step in 110 to 130 ms, so a frontier model hands it a whole goal once instead of paying for every click.[4] Jev also flew the drone: given its state as JSON instead of a page, it chose a bounded move every 0.4 seconds and brought a simulated quadcopter back from 12 of 12 gusts.[1] The flight code is not part of Broad Daylight, but the pattern is Copper's: a small, fast model picks bounded actions while a person or a larger model sets the goal.

How the factory runs

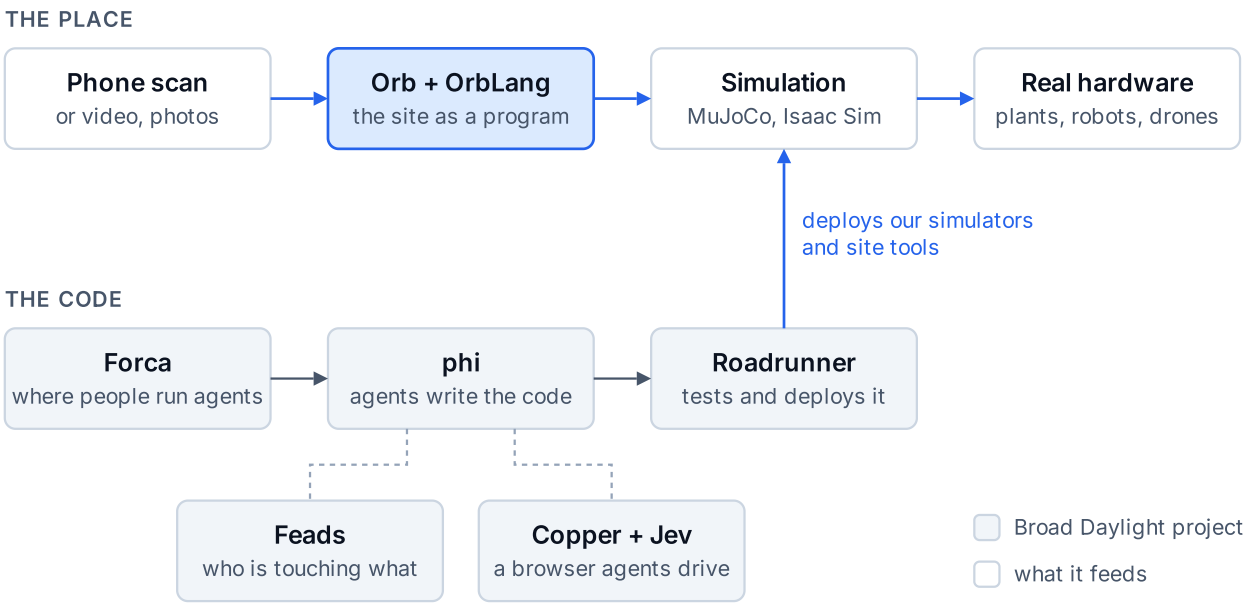


Figure 1. Where the six projects sit between code and the physical world. The blue path is the one that matters for physical AI: a place becomes a program, the program becomes a simulated site, and code passes there before it runs on hardware.

A person opens Forca and starts several phi sessions side by side, each reaching its models through the gateway Tim Child built. Before an agent edits a file it declares the path in Feads; agents have stayed off a claimed path 826 times.[5] When a task lives in a web page, the agent drives Copper, signed in as the person. Roadrunner tests and deploys every push on our own hardware, including our energy-site console, a PLC simulator and a drone simulator.[6]

Why we are sharing it

We started Broad Daylight because these tools are not what Exowatt sells, and because most of them began as someone else's open-source code. phi is a fork of pi, which Mario Zechner started.[7] Forca is a fork of Stably AI's Orca; 237 of its 16,174 commits are ours.[8] Feads began as beads, and Copper's step loop is a port of browser-use's. Changes we land upstream make the next merge smaller; Forca waited long enough between syncs for 221 conflicts to become 663. People running these tools on robots and sites we do not have will find failures we cannot.

What is open today

Today only Copper is public: the browser, its Linux and Windows builds and Copper Cloud, under MIT in the copper-browser organization on GitHub. `brew install --cask copper-browser/copper/copper` puts it on a Mac, and its issues and pull requests are open. The other five are private and will open one at a time; we have no dates yet. Our plant energy management system is not part of Broad Daylight. We built all six for one team's machines, so expect rough edges and tell us where it breaks.

What comes next

The open problems are physical. We designed Orb to compare a fresh scan with the program and mark each element ok, drifted, missing, unexplained or unverified; we have not built that yet. Our Jev drone loop drifted 0.55 to 1.07 m per lap in a GPS-off simulator, because it corrects against the drone's own position estimate; the fix is an outside reference such as AprilTags.[1] We will write each up as the code opens.

Exowatt, Putting a text-only model in a quadcopter's control loop.

Exowatt, From a phone scan to a film, through OrbLang.

Exowatt, Safe rollouts and a dial-out WireGuard tunnel for on-site energy controllers.

Exowatt, Copper Cloud and the tenth-of-a-second browser step.

Exowatt, Feads, a work board for agents that people read.

Exowatt, Moving 199 repos off GitHub Actions in eight days.

Exowatt, Running a company on a forked coding agent.

Exowatt, Shipping our fork of Orca, and merging 9,043 upstream commits back into it.