OpenGizmo

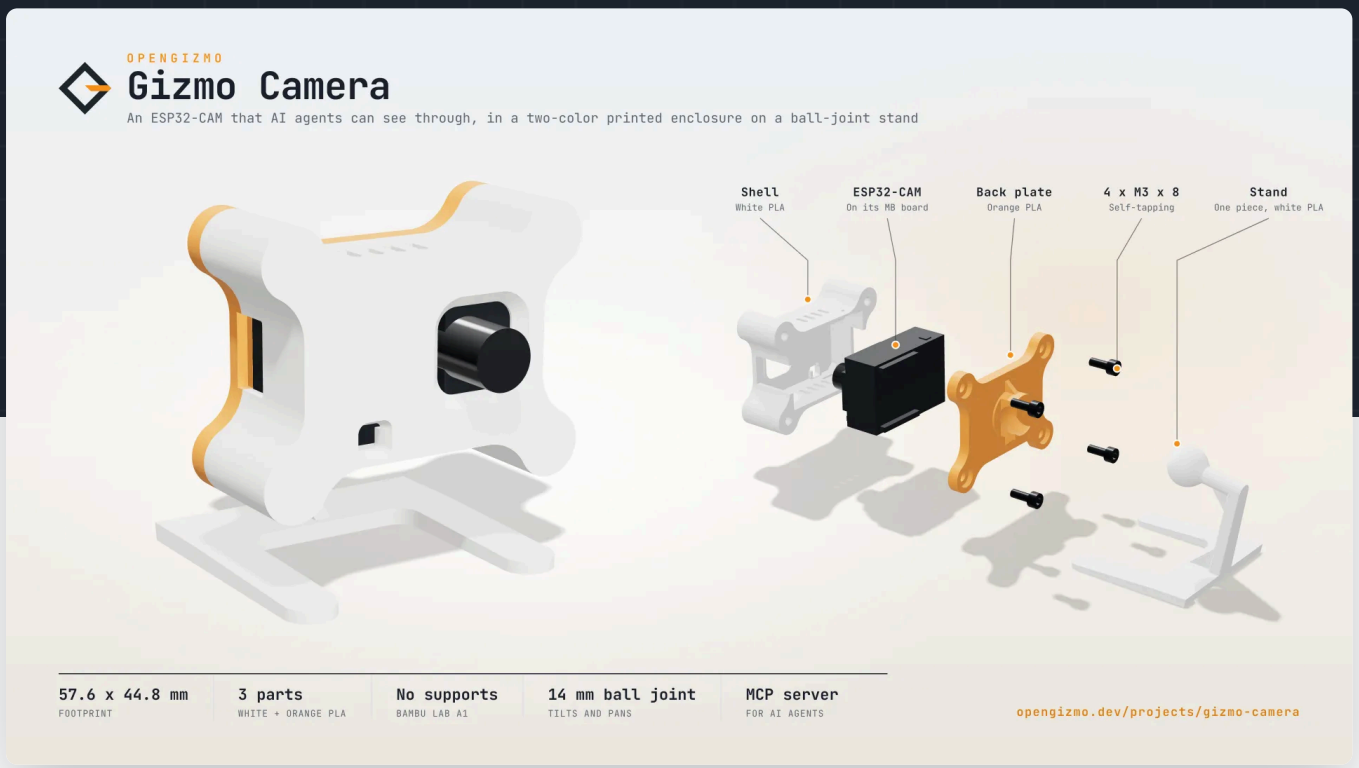
DATASHEET

ESP32-CAM · WI-FI SNAPSHOT CAMERA

Gizmo Camera

A Wi-Fi camera that answers every request with a fresh JPEG, set up from your phone, and an MCP server AI agents can see through.

AI Thinker ESP32-CAM | OV2640 camera | MCP server for AI agents | Wi-Fi setup hotspot | Web app | Open source



BOARD
ESP32-CAM
AI Thinker, on its MB USB board

CAMERA
OV2640
JPEG up to 1600 × 1200

AI AGENTS
MCP
An MCP server on the camera

SETUP
Browser
Hotspot, then gizmo-camera.local

01 Overview

Gizmo Camera gives a cheap ESP32-CAM module a proper job: a camera on your network that answers every request with a fresh photo. Point it at a 3D printer, a plant, a door or a parking spot, and anything that can make an HTTP request can see what it sees.

AI agents get it as an MCP server: they connect to it and take a snapshot to look through it. A web app embedded in the firmware sets up the Wi-Fi and takes photos from your phone, with no credentials in the code.

- JPEG snapshots on request, from 160 × 120 up to 1600 × 1200
- Adjustable JPEG quality, and the on-board flash LED for dark scenes
- A small HTTP API for scripts and other devices
- An MCP server and a guide for AI agents
- Wi-Fi setup from a hotspot with a captive portal, no credentials in the firmware
- A printed enclosure in two colors, 57.6 × 44.8 mm, on a ball-joint stand that tilts and pans
- Prebuilt firmware, flashed from the browser on opengizmo.dev

02 Specifications

CONTROLLER

Board	**AI Thinker ESP32-CAM (esp32cam)**
Memory	**4 MB PSRAM, for the frame buffer**

CAMERA

Sensor	**OV2640, 2 MP**
Resolutions	**Up to UXGA, 1600 × 1200**
JPEG quality	**Adjustable**
Flash	**White LED next to the camera**

INTERFACES

Snapshots	**JPEG over HTTP, on request**
AI agents	**MCP server, and a guide for agents**

CONNECTIVITY AND SETUP

Wi-Fi	**2.4 GHz**
Setup hotspot	**Gizmo-Camera**
Web app	**http://gizmo-camera.local**

03 Pinout and wiring

Everything is wired on the ESP32-CAM: the camera plugs into its ribbon connector, and the board sits on its ESP32-CAM-MB, which powers it and flashes it over USB. GPIO0 is both the camera clock and the BOOT pin, so the firmware never reads it as a button.

Pins checked against the firmware v1.1.1 (commit 1382aa1) when this datasheet was generated.

SIGNAL	BOARD PIN	MODULE PIN	NOTES
XCLK	GPIO0	XCLK	Camera clock, also the BOOT strapping pin
SIOD	GPIO26	SDA	Camera control (SCCB) data
SIOC	GPIO27	SCL	Camera control (SCCB) clock
D0	GPIO5	Y2	Camera data
D1	GPIO18	Y3	Camera data
D2	GPIO19	Y4	Camera data
D3	GPIO21	Y5	Camera data
D4	GPIO36	Y6	Camera data
D5	GPIO39	Y7	Camera data
D6	GPIO34	Y8	Camera data
D7	GPIO35	Y9	Camera data
VSYNC	GPIO25	VSYNC	Frame sync
HREF	GPIO23	HREF	Line sync
PCLK	GPIO22	PCLK	Pixel clock
PWDN	GPIO32	PWDN	Camera power down, which also resets it
Flash	GPIO4	Flash LED	The white LED next to the camera

04 Parts list

QTY	PART	NOTES
1	**AI Thinker ESP32-CAM, with its OV2640 camera**	Every AI Thinker module has the 4 MB of PSRAM the firmware needs
1	**ESP32-CAM-MB USB board**	The ESP32-CAM has no USB port of its own

1	**Printed shell, back plate and stand**	The 3MF plates are in the gizmo-docs repository: white and orange PLA
4	**M3 × 8 screws**	They cut their own thread in the shell
1	**Micro-USB data cable and 5 V USB power supply**	A good one: the flash and the Wi-Fi draw current peaks

05 Setup

01 Plug in the board

Sit the ESP32-CAM on its MB board, and connect it to your computer with a data USB cable. Use Chrome or Edge.

02 Flash the firmware

Open the project page on opengizmo.dev and press Flash firmware. If it can't connect, hold IO0, press and release RST, then release IO0 and try again.

03 Start the camera

When the flashing is done, press RST: the camera opens the Gizmo-Camera Wi-Fi.

04 Pick your Wi-Fi

Join the hotspot from your phone, choose your network on the page that opens and type its password.

05 Take a photo

Open http://gizmo-camera.local, or its IP address, choose the resolution, the quality and the flash, and take a snapshot.

06 Connect an AI agent

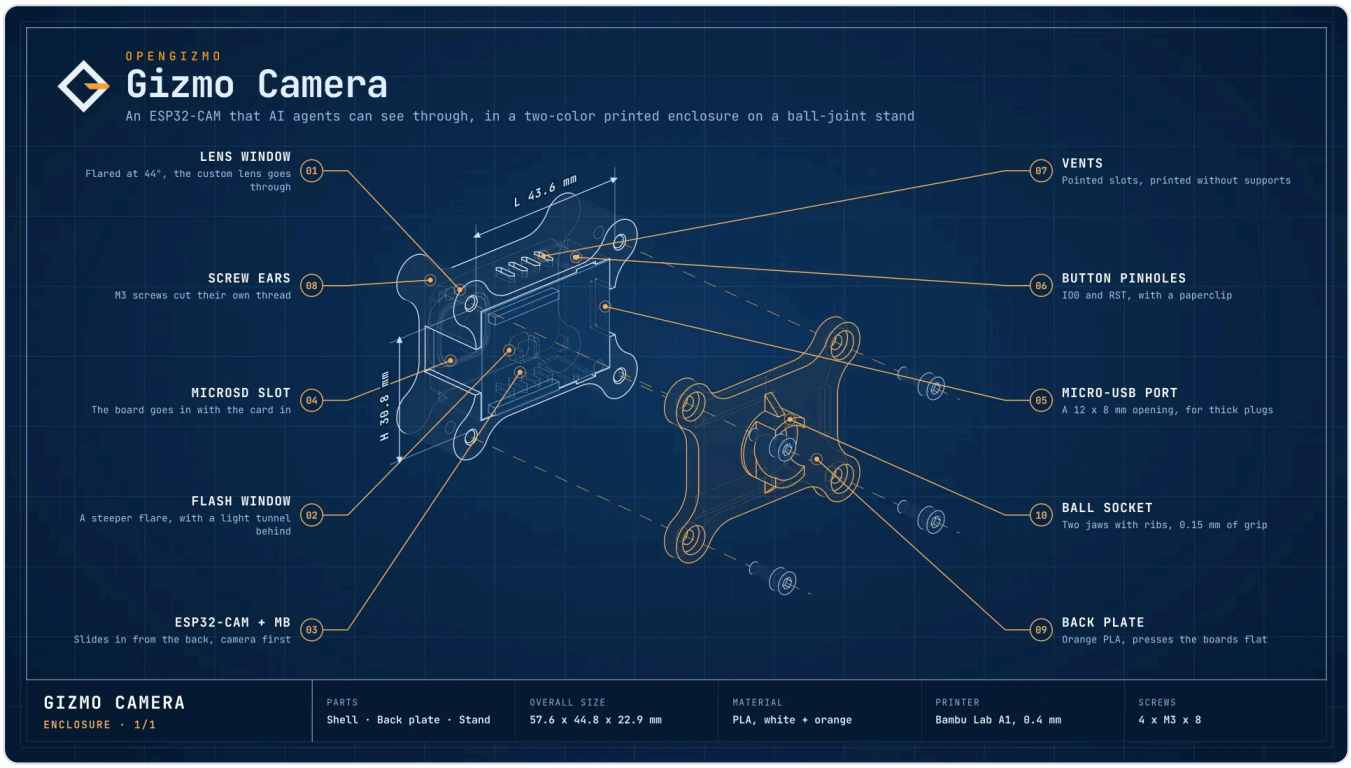
Add the camera as an MCP server in your AI agent, with its address: the README has the details.

06 Troubleshooting

The flasher can't connect to the board	Check that the ESP32-CAM sits fully in its MB board, try another USB cable (some only charge), and enter flashing mode by hand with IO0 and RST. In the enclosure, press them with a paperclip through the pinholes.
The web app can't find the camera sensor	The sensor didn't start. Power the board off, check that the ribbon cable is fully inserted and its latch closed, then power it on again.
The board restarts when the flash fires or during the Wi-Fi setup	A brownout: a weak USB port or a thin cable can't supply the current peaks. Use a powered USB port or a good 5 V supply.
A snapshot request is refused as busy	Another snapshot was being taken or sent at the same moment. Try again once it's done.

An AI agent can't connect to the MCP server	The agent's computer must be on the camera's network: use the IP address if gizmo-camera.local doesn't resolve there. Cloud agents can't reach a camera on your home network.

07 Enclosure



Every detail of the enclosure, from the flared lens window to the ball socket, on a 57.6 x 44.8 mm footprint.

Gizmo Camera is an open-source OpenGizmo project

Project page: https://opengizmo.dev/projects/gizmo-camera

Source code: https://github.com/open-gizmo/gizmo-camera · MIT license