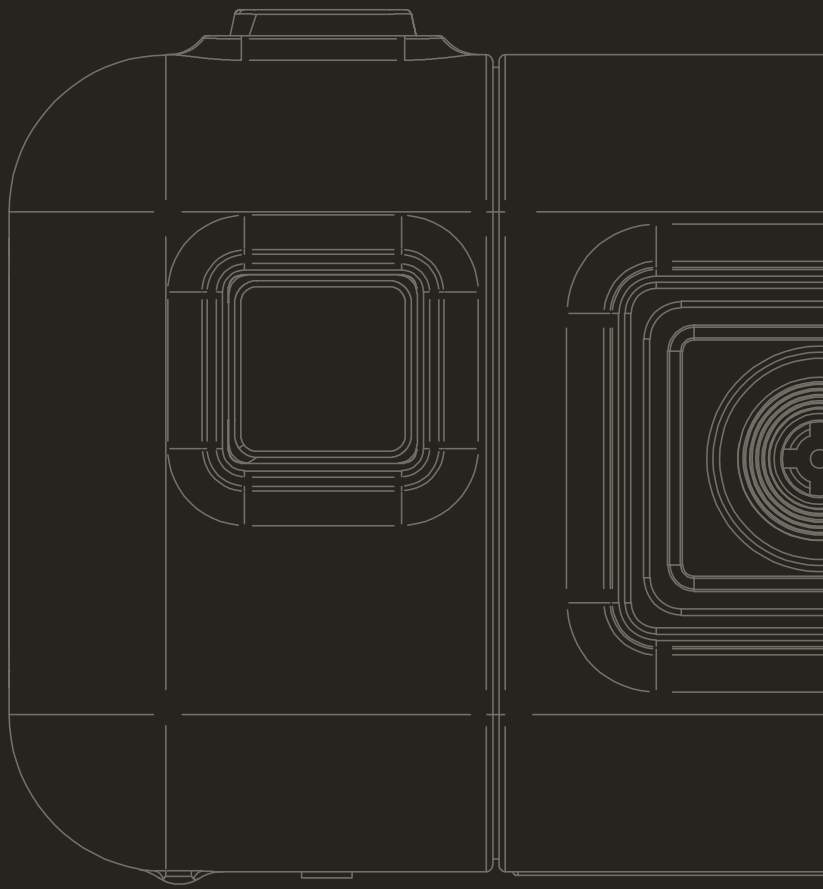
SATURNIX

ASSEMBLY MANUAL



FREE & OPEN-SOURCE

Open hardware + open firmware. Source code, STL & STEP files and full documentation are public.

Firmware: MIT · Hardware: CERN-OHL-S-2.0 · Docs: CC-BY-4.0

DESIGNED BY YUTANI
OPEN-SOURCE PROJECT

SOURCE · STL · STEP · DOCS — PUBLIC
github.com/Yutani140x/saturnix-camera

00 ABOUT THIS DOCUMENT

PROJECT: SATURNIX — open-source camera

MODEL: Dione

LICENSE: MIT (firmware) · CERN-OHL-S-2.0 (hardware) · CC-BY-4.0 (docs & media)

⚠ DISCLAIMER — READ BEFORE YOU BUILD

SATURNIX is a free, open-source DIY project provided "as is", without any warranty of any kind, express or implied. You build, modify, flash and operate this device entirely at your own risk.

The creator and contributors accept **no responsibility or liability** for any injury, death, property damage, data loss, financial loss or any other harm arising from building, using, or attempting to build this project, including from errors or omissions in this document.

You are solely responsible for verifying every step, every wiring connection, every component, and every safety precaution before proceeding. If you are not confident working with lithium batteries, soldering, electronics, or resin 3D printing, do not attempt this build, or seek qualified help.

Nothing in this document is professional, legal, electrical, or safety advice. You are responsible for complying with all laws and regulations in your country. Component brand names are property of their respective owners. SATURNIX is an independent project and is not affiliated with, sponsored, or endorsed by any company or component manufacturer.

BY CONTINUING PAST THIS POINT YOU ACCEPT FULL RESPONSIBILITY FOR YOUR OWN SAFETY AND ACTIONS.

Two high-risk areas — resin printing (§03) and lithium battery handling (§04, §07) — carry dedicated safety notices at the start of those sections. Read them before beginning that work.

CONTENTS

01 INTRODUCTION

SATURNIX is a DIY digital camera inspired by cassette futurism, built around a Raspberry Pi Zero 2 W, a 16-megapixel autofocus sensor, and a small LCD viewfinder. The camera captures photos in DNG+JPEG format and processes them directly on the device using simple JPEG presets inspired by film photography. Image capture, storage, and processing are handled entirely by the Raspberry Pi Zero 2 W. No external apps or computer connection are required.

Project background

I have always been drawn to cassette futurism — the look of old analog machines, industrial computers, worn technical equipment, and retro-future interfaces. I also love cyberpunk worlds like Blade Runner and Ghost in the Shell, and I have always been interested in photography, especially cameras as physical technical devices.

About a year ago, I wanted to build a camera of my own: something simple, tactile, and fully under my control. I wanted a device that I could tune, modify, and shape around my own idea of what a camera should feel like — from the user interface to the physical body design.

With SATURNIX, I wanted to capture the feeling of cassette-futurist hardware. I focused a lot on how the camera feels in the hand and how it feels to use. For example, instead of using small cheap push buttons, I used full-size mechanical keyboard switches to make the controls feel more physical and deliberate.

This project is not trying to replace a commercial camera. It is a personal experiment around:

- designing a real physical camera body;
- creating a custom camera UI;
- building a simple on-device image processing system;
- working with Raspberry Pi camera hardware;
- experimenting with resin 3D printing, post-processing, painting, and weathering;
- making a device that feels personal, repairable, and open.

This was my first time building a device like this, so not everything is perfect. But looking back at where the project started about a year ago, I am happy with how far it has come.

The software was also a major part of the challenge. I am not primarily a software developer, and the code was one of the hardest parts of the project. A lot of the development was AI-assisted, starting with earlier AI models and improving significantly as newer coding models became available. This helped me finally bring the project much closer to the version I had imagined.

SATURNIX started as a personal project. After sharing it online and seeing the level of interest, I decided to make it publicly available as open source.

With the growing interest in old digital compact cameras, I also want this project to show that custom open-source digital cameras are possible. Instead of depending only on large companies, closed firmware, and fixed design choices, we can build devices that we fully understand, modify, repair, and shape for ourselves.

Maybe SATURNIX will inspire someone else to build something even better using this code, these files, or simply the idea behind it.

P.S. I am not a programmer. The firmware was developed with **Claude Code** (Anthropic) and **Codex** (OpenAI): I designed the camera, the hardware and the experience, described what I wanted, tested on the real device and iterated until it worked. If you are an experienced developer and see something to improve — pull requests are welcome. SATURNIX is an independent project, not affiliated with or endorsed by Anthropic/OpenAI. The text was translated with the help of AI, so there may be inaccuracies or errors.

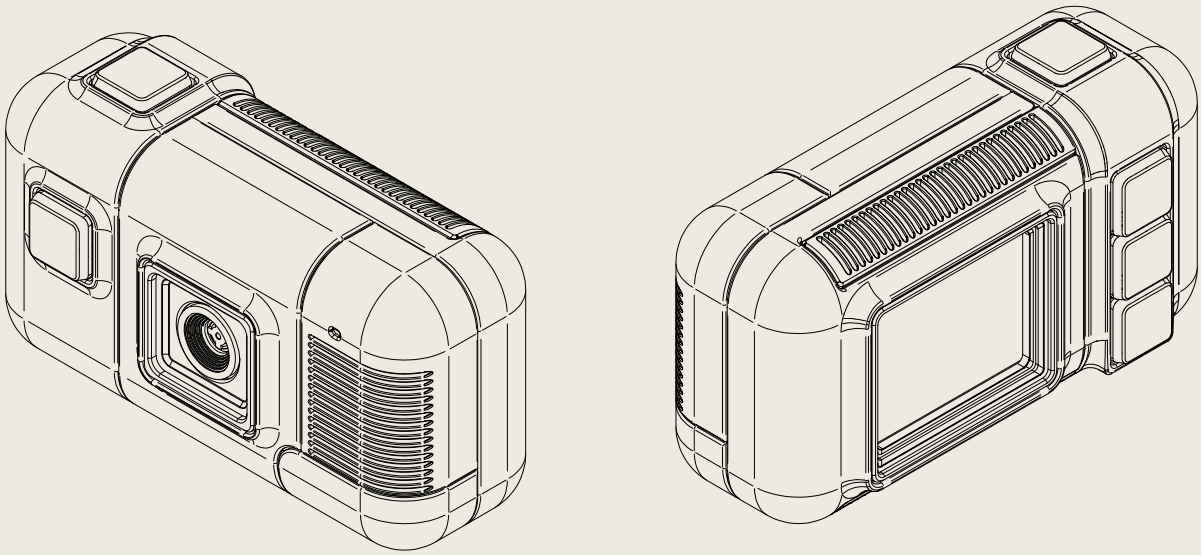
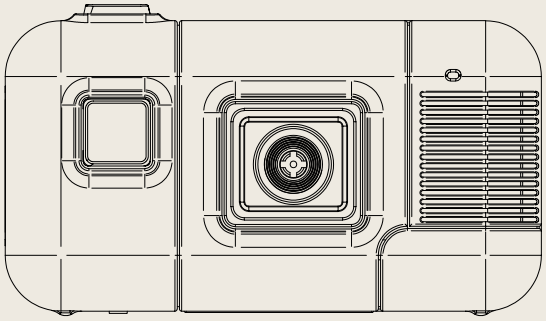
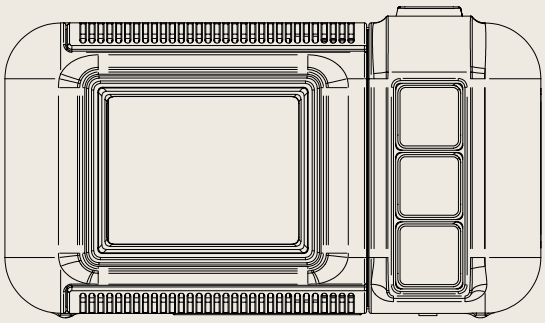


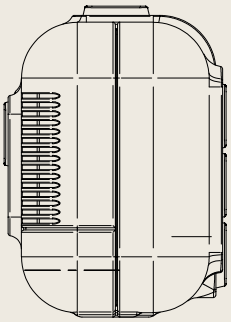
FIG. 01-1 — SATURNIX Dione: front and rear views.



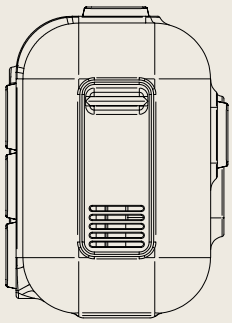
FRONT



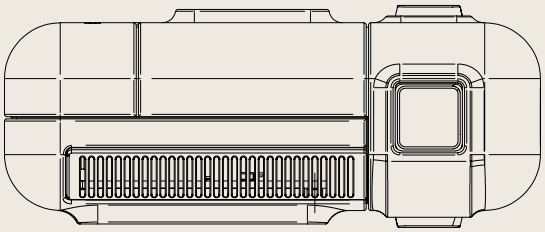
REAR



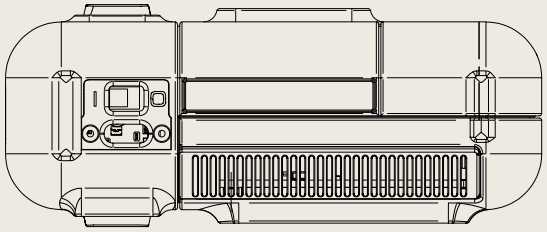
LEFT



RIGHT



TOP



BOTTOM

FIG. 01-2 — Orthographic projections of the assembled camera.

02 BILL OF MATERIALS (BOM)

REPOSITORY FILES — github.com/Yutani140x/saturnix-camera

Everything you download for the build lives in one repository. What's where:

Path	What it is
firmware/	Camera source code (MIT). Contains the **python/** and **saturnix-dione/** folders you copy onto the Pi in §05
hardware/saturnix-camera-print-files-plain.zip	Main camera body — no logos or text markings. STL + STEP
hardware/saturnix-camera-print-files-with-logo.zip	Main camera body with SATURNIX logos and markings. STL + STEP
hardware/saturnix-camera-buttons.zip	Keycaps for the five mechanical buttons. STL + STEP
hardware/LICENSE-HARDWARE.md hardware/NOTICE.txt	CERN-OHL-S-2.0 license text and notice; a copy of NOTICE.txt is inside every CAD zip
docs/	Build guide, images and brand assets (CC-BY-4.0; brand assets follow the naming policy)
SATURNIX_Assembly_Manual.pdf	This document

ELECTRONICS & CORE HARDWARE

Component	Qty
Raspberry Pi Zero 2 W	1
Arducam IMX519 16MP Autofocus	1
Waveshare 2" IPS LCD (SPI)	1
Waveshare UPS HAT (battery included)	1
MH-FMD passive buzzer	1
Kailh Low Profile Switch 1350	5
Mini switch SS12D10	1
USB-C Panel Socket (Type-C)	1
microSD card (32 GB+)	1
Heatsink for Raspberry Pi	1
Silicone hook-up wire	as needed

FASTENERS

Spec	Size	Qty
M1.7	10 mm	6
M1.7	4 mm	16
M2	4 mm	1

3D-PRINTED PARTS — from /hardware (STL / STEP)

Part	Notes
Main body / front shell	Sensor & viewfinder side, body
Rear cover	Display side — display mounts to it (integrated bezel)
DIONE main board mount	Internal frame for Pi + boards
Keycaps ×5	REC · NAV+ · NAV− · SET · AF
I/O switch cover	Power switch (I/O) plate

03 3D PRINTING

⚠ RESIN SAFETY — UNCURED RESIN IS TOXIC

SATURNIX parts are designed for resin (MSLA/SLA) printing. Liquid, uncured resin is a skin, eye and respiratory irritant and a known sensitiser — repeated exposure can cause permanent allergic reactions. Treat it as a hazardous chemical.

- Always wear **nitrile gloves** and **eye protection** when handling resin, prints, or IPA. Never touch uncured resin with bare skin.
- Work in a **well-ventilated area**; use a respirator rated for organic vapours if ventilation is limited. Avoid breathing fumes.
- Wash printed parts thoroughly in IPA, then **fully UV-cure every part before assembly and handling**. An under-cured part inside a hand-held camera means continuous skin contact with reactive resin — do not skip or shorten curing.
- Keep resin, IPA and prints away from children, pets, food and skin.
- **Never** pour liquid resin or resin-contaminated IPA down the drain or into household waste. Cure all waste resin solid under UV/sunlight first, then dispose of it and contaminated consumables as hazardous waste per local regulations.
- Follow your specific resin manufacturer's SDS (safety data sheet) — it overrides this general guidance.

Download the shell files from the repository's **/hardware** folder: pick **saturnix-camera-print-files-plain.zip** (clean body) or **saturnix-camera-print-files-with-logo.zip** (with SATURNIX markings), plus **saturnix-camera-buttons.zip** for the keycaps — each archive contains both STL and STEP. Wash, cure, then test-fit all parts before final assembly.

PRINTER & MATERIAL — reference setup

Item	Used for this build
Printer	Anycubic Photon P1 (14K)
Resin	Anycubic Water-Wash ABS-Like Resin 3.0
Release film	NFEP

Why ABS-like resin matters: the camera is a handheld device used in real-world conditions, and several parts take self-tapping screws directly into the print. Standard rigid resin is brittle — it cracks under screw torque and impact. ABS-like resin has the toughness to hold threads without crumbling. Do not substitute standard resin for the structural parts.

PRINT SETTINGS — Anycubic Photon P1, ABS-Like 3.0

Parameter	Value
Layer thickness	0.030 mm
Normal exposure time	2.2 s
Off time	2.0 s
Bottom layers	5
Bottom exposure time	25 s
Bottom off time	30 s
Transition layer count	8
Z lift (normal), Step0 / Step1	4.0 mm @ 2.0 mm/s · 5.0 mm @ 6.0 mm/s
Z retract (normal), Step0 / Step1	3.0 mm/s · 8.0 mm/s
Z lift (bottom), Step0 / Step1	5.0 mm @ 3.0 mm/s · 5.0 mm @ 3.0 mm/s
Z retract (bottom)	3.0 mm/s

These values are the author's verified profile for this printer/resin pair. A different printer or resin will need its own exposure calibration — run a calibration print first, then adjust exposure time and lift speeds accordingly.

BUILD PLATE LAYOUT — orientation & supports

Both shells print tilted (roughly 45°) with medium supports on the inner faces — never on the visible outer surfaces. Keycaps print face-up on light supports so the legend surface stays clean. Small parts (DIONE mount, I/O cover) stand on their own support islands. Everything fits on a single Photon P1 plate in one job.

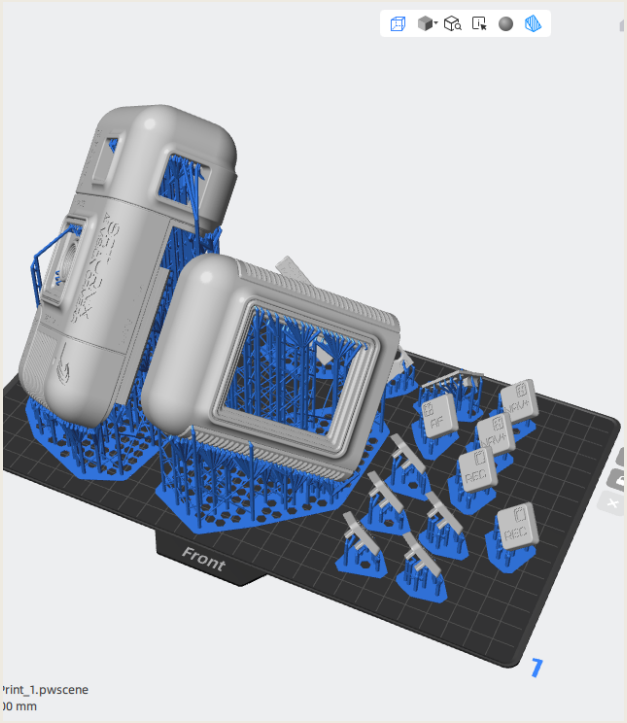


FIG. 03-1 — Reference plate layout, front view: main body and rear cover tilted with supports on inner faces; keycaps face-up on light supports.

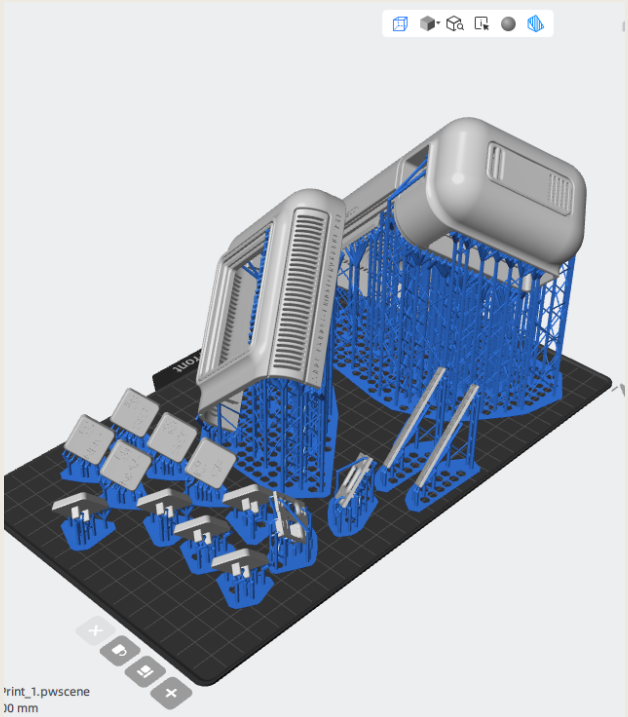


FIG. 03-2 — Same plate from the rear: support structures land only on interior surfaces, keeping the outer shell texture untouched.

- **Support contact points:** inner faces and hidden edges only. A support scar on the outer shell is permanent.
- **After printing:** wash (water — the resin is water-washable), remove supports while the part is still slightly soft, then fully UV-cure per §03 safety notes.
- **Test-fit before assembly:** keycaps into the rear cover, boards into the DIONE mount. Light sanding of support nubs is normal.

Surface finishing & paint — reference recipe

This is the finish used on the reference build. All painting is done with an airbrush after full UV-cure, in a ventilated area with a respirator.

Step	Material / detail
1 · Sanding	Wet-sand progressively: 400 → 600 → 800 grit
2 · Prep	Clean and degrease the surface thoroughly
3 · Primer	Vallejo Grey Primer 74.601
4 · Paint	Tamiya XF series (flat acrylics)
5 · Clear coat	Tamiya satin (semi-gloss) lacquer
Tool	Iwata CR airbrush

Skipping the degrease step is the most common cause of primer peeling on resin prints — residue from washing prevents adhesion. Colours are your choice; the reference build uses the weathered off-white / industrial look shown on the cover photos.

04 ELECTRONICS & GPIO WIRING

⚠ LITHIUM BATTERY SAFETY

The LiPo cell used in SATURNIX stores significant energy and can catch fire or explode if mishandled. Read this before touching the battery (applies here and in §07).

- **Never** puncture, cut, crush, bend, drop, or disassemble the cell. A damaged cell can ignite without warning.
- Double-check **polarity** before connecting. Reversed polarity can destroy the UPS HAT and ignite the cell.
- **Never** short the battery terminals or let them touch metal (tools, screws, the case).
- Charge only via the UPS HAT, on a hard non-flammable surface, away from flammable materials, and **never leave charging unattended**.
- Stop using immediately if the cell is swollen, hot, deformed, punctured, smells, or leaks. Place it in a fire-safe container away from anything flammable.
- In case of fire, do not inhale fumes; move the device away from flammables if safely possible and use an appropriate extinguisher. Lithium fires are intense — prioritise your safety, not the device.
- Dispose of used or damaged cells at a proper battery-recycling point. **Never** put them in normal household waste. Keep batteries away from children and pets.

Power flows from the USB-C panel socket into the Waveshare UPS HAT (C), which charges the LiPo cell and feeds the Raspberry Pi. The HAT does **not** stack on the 40-pin header in this build: its factory GPIO header and onboard switch are desoldered (it would not fit the shell otherwise), and the HAT is hard-wired to the Pi's pins per the pinout below — see the preparation notes in §06. The Pi drives the LCD over SPI, polls the five buttons over GPIO and beeps the buzzer over a PWM pin. The camera sensor connects to the Pi separately via the CSI ribbon connector (no soldering).

MASTER GPIO PINOUT — Raspberry Pi Zero 2 W (BCM)

Signal	BCM	Phys. pin	Goes to
5V	—	2, 4	UPS HAT 5V (wired)
3.3V	—	1, 17	UPS HAT 3.3V · LCD VCC · Buzzer VCC
GND	—	6, 9, 14, 20, 25, 30, 34, 39	Common ground (UPS, LCD, buttons, buzzer)
UPS SDA	GPIO 2	3	UPS HAT — I²C fuel gauge
UPS SCL	GPIO 3	5	UPS HAT — I²C fuel gauge
LCD BL	GPIO 18	12	Waveshare LCD — backlight
LCD RST	GPIO 27	13	Waveshare LCD — reset
LCD DC	GPIO 25	22	Waveshare LCD — data/command
LCD DIN / MOSI	GPIO 10	19	Waveshare LCD — SPI data in
LCD CLK / SCLK	GPIO 11	23	Waveshare LCD — SPI clock
LCD CS / CE0	GPIO 8	24	Waveshare LCD — SPI chip select
BTN REC	GPIO 23	16	Kailh switch — REC (one side to GND)
BTN AF	GPIO 24	18	Kailh switch — AF (one side to GND)
BTN NAV+	GPIO 20	38	Kailh switch — NAV+ (one side to GND)
BTN NAV−	GPIO 16	36	Kailh switch — NAV− (one side to GND)
BTN SET	GPIO 21	40	Kailh switch — SET (one side to GND)
BUZZER I/O	GPIO 26	37	MH-FMD passive buzzer

Pin numbers come from the project's config.json (gpio_pins). Double-check against your wired build before powering on — reversed or misplaced wires can damage the Pi.

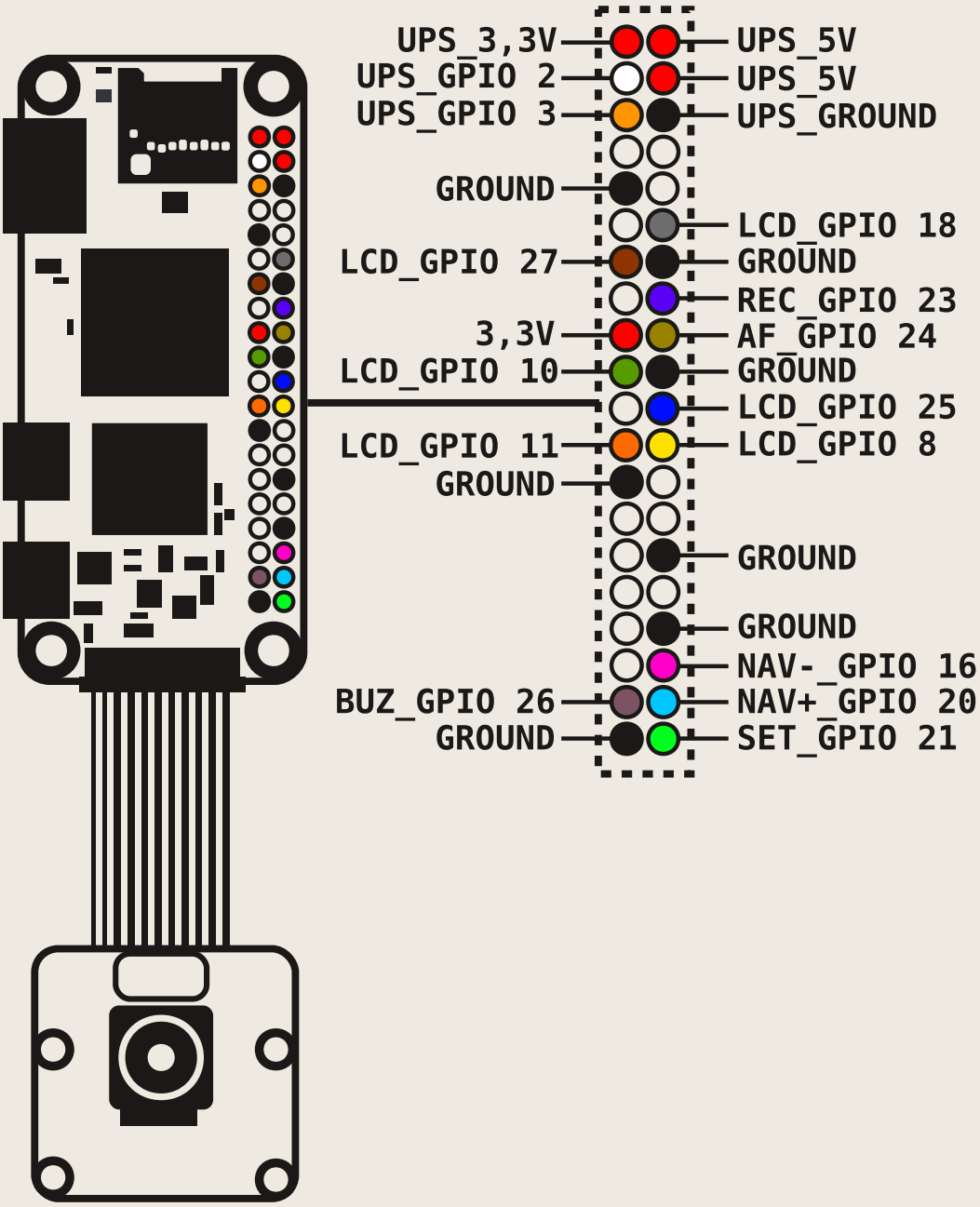


FIG. 04-1 — Raspberry Pi Zero 2 W: master GPIO assignment for SATURNIX Dione. Camera sensor connects via the CSI ribbon (left side, no soldering).

Per-module wiring

Wire colours are consistent across all diagrams — match them to the master diagram above when soldering.

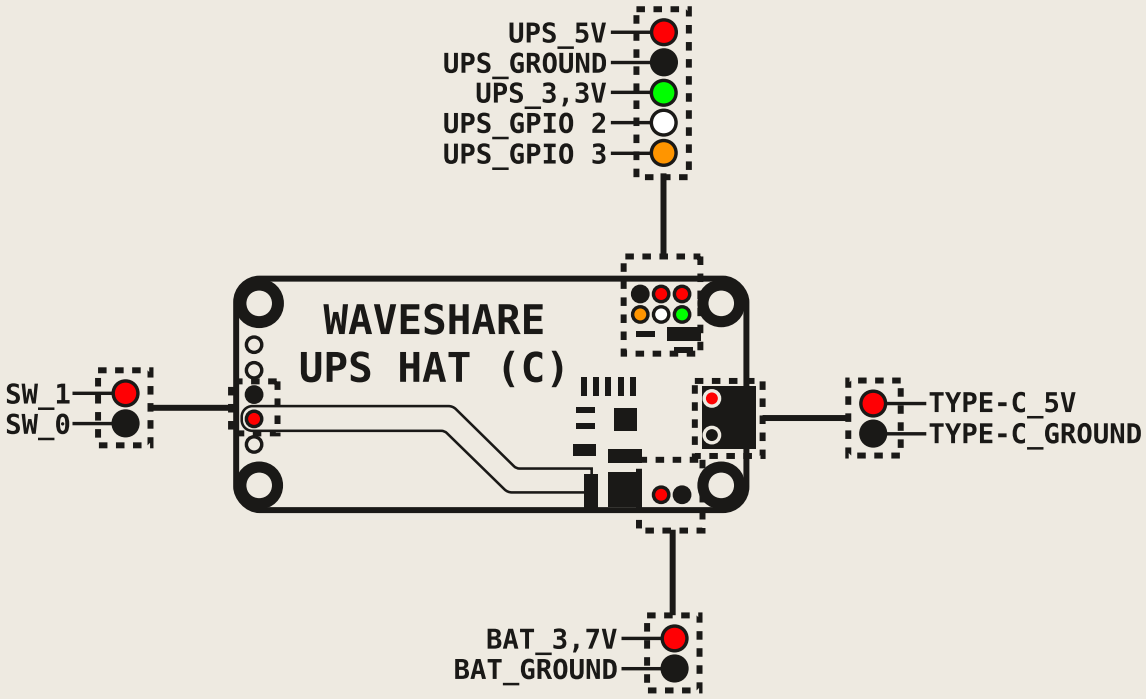


FIG. 04-2 — Waveshare UPS HAT (C). The factory GPIO header and onboard switch are desoldered; the HAT is hard-wired to the Pi (5V, 3.3V, GND, SDA, SCL — see the master pinout). External connections: power switch (SW_0 / SW_1), USB-C input (TYPE-C 5V / GND), and the LiPo cell (BAT 3.7V / GND).

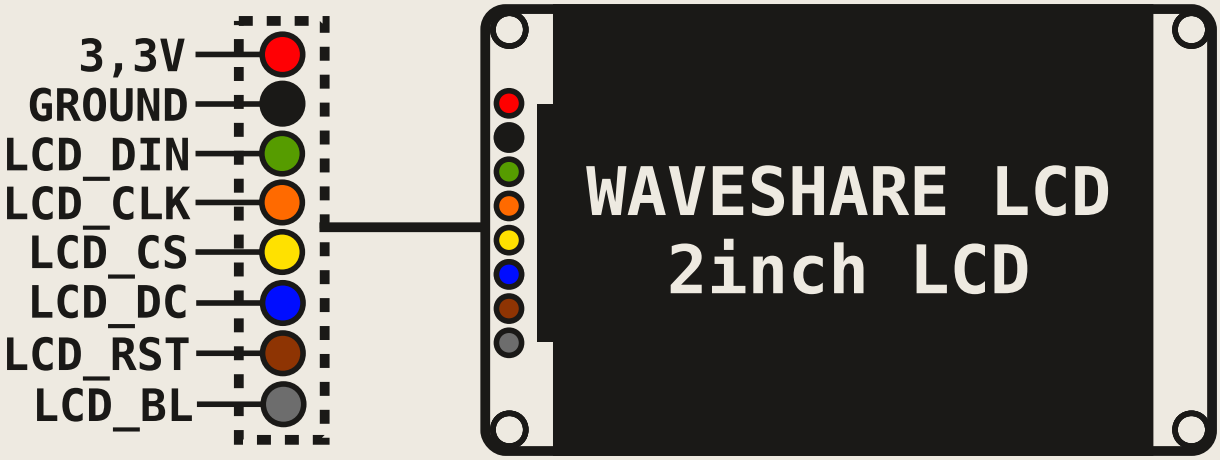


FIG. 04-3 — Waveshare 2" IPS LCD. Eight wires total: 3.3V, GND, plus six GPIO lines (DIN, CLK, CS, DC, RST, BL).

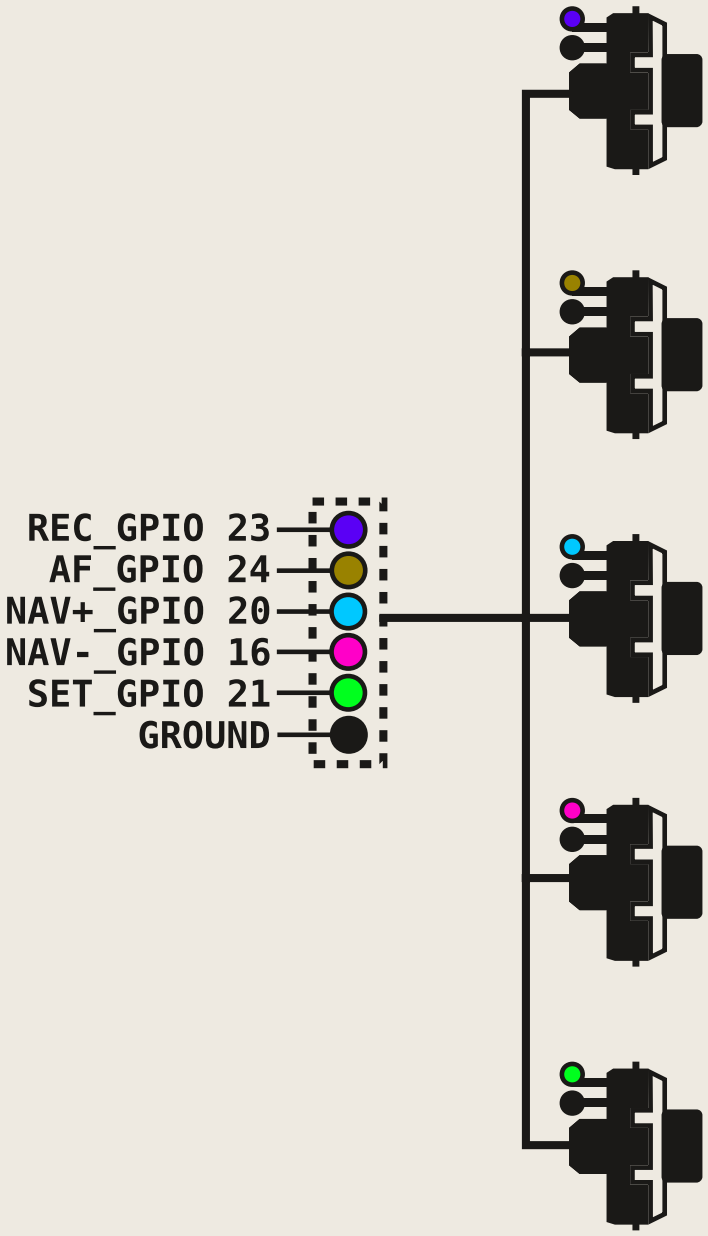


FIG. 04-4 — Kailh Low Profile 1350 switches (×5). Each switch has two electrically identical contacts: one to its assigned GPIO, the other to a shared ground rail. Polarity does not matter on the switch itself.

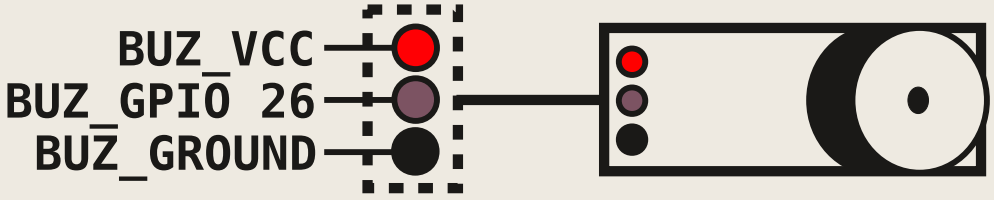


FIG. 04-5 — MH-FMD passive buzzer. Three pins: VCC → 3.3V, I/O → GPIO 26, GND → ground.

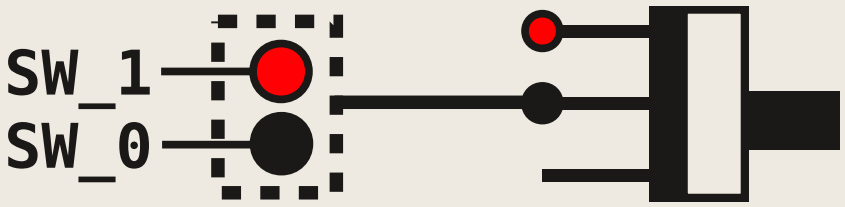


FIG. 04-6 — SS12D10 mini slide switch (power on/off). Two contacts wired to SW_0 and SW_1 on the UPS HAT. The third pin is unused.

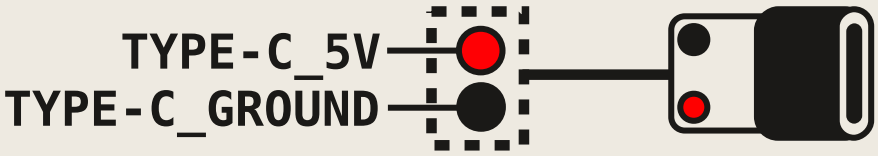


FIG. 04-7 — USB-C panel socket (Type-C). Carries 5V and GND to the UPS HAT's Type-C input pads via a short micro-USB pigtail adapter (avoids desoldering the HAT's connector). Power only — no data lines required.

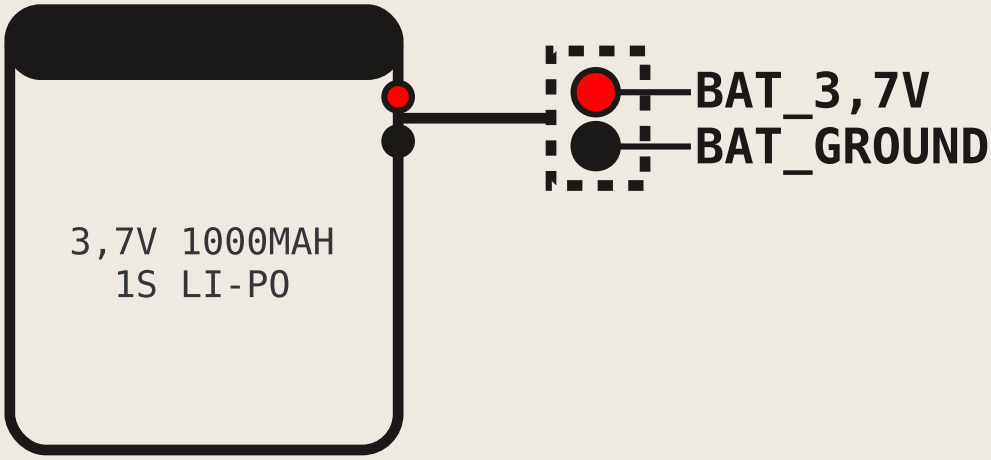


FIG. 04-8 — LiPo cell 3.7V 1000 mAh (1S). Connects to the UPS HAT's BAT pads. **Confirm polarity before soldering** — reversed polarity will destroy the HAT and may ignite the cell. See the Lithium Battery Safety notice above.

05 SOFTWARE INSTALLATION

Install the software **before** final assembly so you can verify the electronics with the case still open. The overall flow: flash the base OS, connect the Pi to a PC/display, enable interfaces, install packages, place the project folders, adjust **config.json**, then run the fast-boot script — after which the camera starts automatically on every power-up.

⚠ IDENTITY VALUES ARE HARD-CODED — DO NOT CHANGE

The SATURNIX firmware is tied to these exact system names. Use them verbatim during flashing, or the software will not run:

- Hostname: **saturnix-dione**
- Username: **saturnix**
- Password: **saturnix**

Step 1 — Flash the base OS. Download **Raspberry Pi Imager** (raspberrypi.com/software) and walk through it exactly as shown:

Imager screen	Choose
Device	Raspberry Pi Zero 2 W
Operating System	Raspberry Pi OS (Legacy, 32-bit) — Debian Bookworm
Storage	Your microSD card
Customisation → Hostname	saturnix-dione
Customisation → User	saturnix / saturnix
Customisation → Wi-Fi	Your home network (needed for Step 2 downloads)
Customisation → Remote access	Enable SSH

Step 2 — Update the system. Boot the Pi, then:

bash

```
sudo apt update
sudo apt full-upgrade -y
sudo reboot
```

Step 3 — Enable interfaces.

```bash
sudo raspi-config
```

- Interface Options → **I2C** → Enable (UPS HAT)
- Interface Options → **SPI** → Enable (LCD)
- Finish → reboot

Step 4 — System packages + Arducam driver + camera overlay. Install packages:

```bash
sudo apt install -y \
  python3-pip python3-picamera2 python3-opencv \
  python3-smbus python3-rpi.gpio python3-pil \
  i2c-tools rpicam-apps git
```

Install the Arducam IMX519 driver:

```bash
cd ~
wget -O install_pivariety_pkgs.sh \
  https://github.com/ArduCAM/Arducam-Pivariety-V4L2-Driver/releases/download/install_script/install_pivariety_pkgs.sh
chmod +x install_pivariety_pkgs.sh
./install_pivariety_pkgs.sh -p libcamera_dev
./install_pivariety_pkgs.sh -p libcamera_apps
```

Edit the boot config:

```bash
sudo nano /boot/firmware/config.txt
```

Add at the end (no spaces around **=**):

```config.txt
camera_auto_detect=0
dtoverlay=imx519
```

Save (Ctrl+O, Enter, Ctrl+X), then reboot.

Step 5 — Place the project folders. Copy two folders from the repository onto the Pi — that's all this step is:

- **~/Desktop/python/** — the LCD driver (LCD_2inch.py, lcdconfig.py)
- **/home/saturnix-dione/** — the camera code and scripts (liveview.py, main.py, capture.py, buzzer.py, ups.py, logger.py, config.json, splash.jpg, setup_fast_boot.sh, setup_wifi_hotspot.sh, and a Pictures/ folder for photos). The .sh scripts are run from this folder via the console.

Folder names and locations are hard-coded, same as the system identity — keep the exact paths above.

Step 6 — Set up the WiFi hotspot.

```bash
cd /home/saturnix-dione
chmod +x setup_wifi_hotspot.sh
sudo ./setup_wifi_hotspot.sh
```

Creates the hotspot **SaturnixCam** (password **saturnix24**, camera address **192.168.4.1**) used for photo transfer.

Step 7 — Adjust config.json. The configuration file lives in the **saturnix-dione** folder. Open it in any text editor and review the settings before enabling autostart — the full parameter reference is in §07. At minimum, check the battery behaviour below.

⚠ BATTERY AUTO-SHUTDOWN — READ BEFORE FIRST RUN

By default the camera **shuts down at 5% battery** to protect the SD card from corruption. If the UPS HAT battery gauge is not detected on first boot, the reading is treated as empty — and the camera will simply **power itself off** shortly after starting.

If your camera keeps switching off right after boot: check the UPS HAT wiring to the Pi's GPIO pins (SDA / SCL / power) and that I²C is enabled (Step 3), or temporarily change **battery_shutdown_pct** in config.json.

Building without a battery? Set **"battery_auto_shutdown": false** in config.json — this disables both the auto-shutdown and the low-battery warning beeps.

Step 8 — Fast boot + autostart.

```bash
cd /home/saturnix-dione
chmod +x setup_fast_boot.sh
sudo ./setup_fast_boot.sh
sudo reboot
```

After the reboot the camera starts automatically: splash screen → live preview. Installation complete — power on and verify the display, buttons, autofocus, buzzer and hotspot while the case is still open, then continue to final assembly (§06).

Returning to a normal desktop later (undo autostart):

```bash
sudo systemctl set-default graphical.target
# remove the SATURNIX_AUTOSTART block from ~/.bashrc
sudo reboot
```

06 FINAL ASSEMBLY

Reminder: this step involves handling and enclosing the LiPo cell. Re-read the Lithium Battery Safety notice in §04 before proceeding.

- Mount the boards onto the DIONE frame, route cables, fit the display into the rear cover, seat the keycaps.
- Place the battery, close the shell, drive the screws (M1.7×4 for internals, M1.7×10 for the shell, M2×4 where marked).

Two preparation notes for the UPS HAT. First, the board ships with a GPIO header and an onboard power switch pre-fitted — both are **desoldered** and replaced with wires soldered directly to the pads, otherwise the HAT does not fit inside the shell. Second, charging power enters through a micro-USB connector with pre-soldered leads plugged into the HAT's **CHG** port and wired to the USB-C panel socket — plugging in a pigtail is far safer for beginners than desoldering the port. Treat the HAT and the LiPo cell with extreme care throughout (§04).

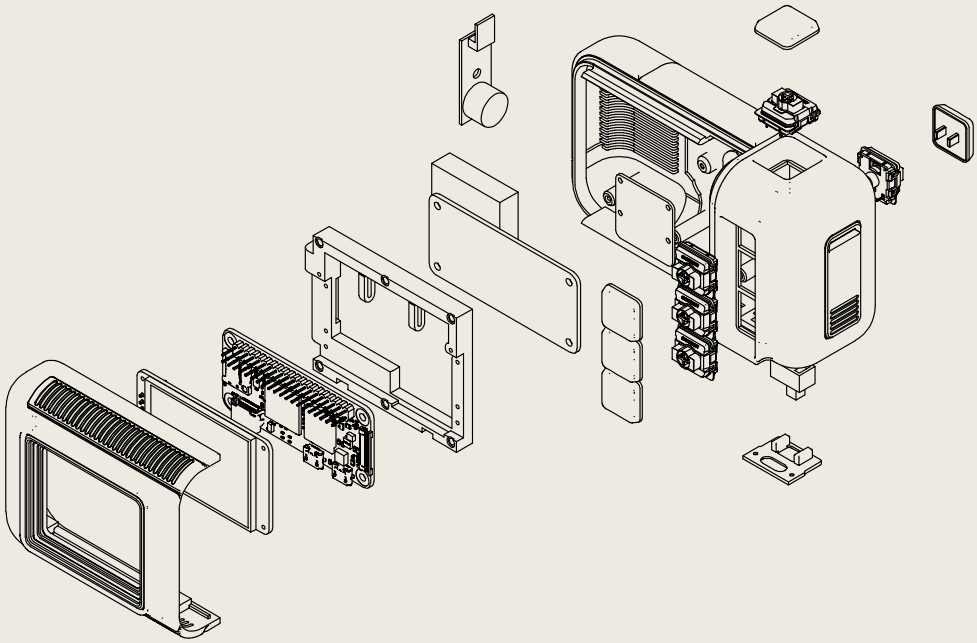


FIG. 06-1 — Exploded view: how the printed parts, boards and battery stack inside the shell.

Assembly sequence — reference photos

The photos below document the reference build in order. Cross-check every connection against the wiring in §04 as you go — nothing here replaces the master pinout.

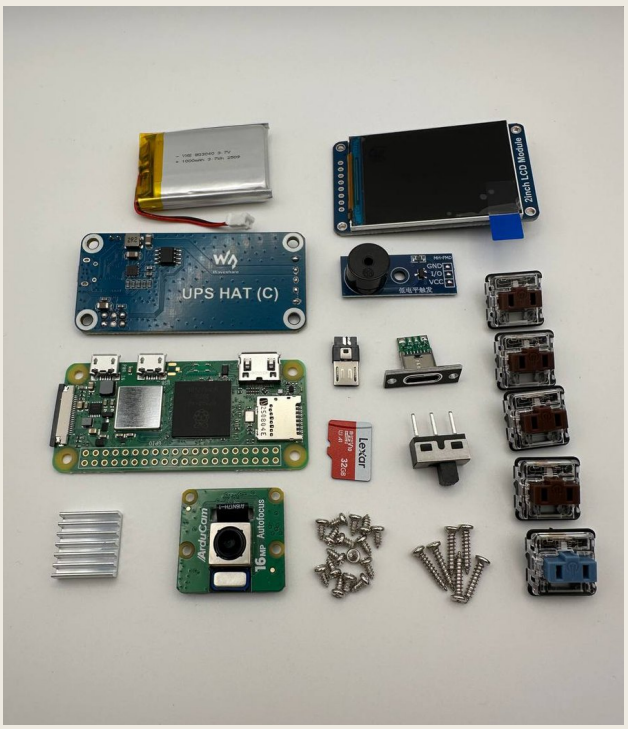


FIG. 06-2 — Parts check: the full BOM (§02) laid out before assembly — boards, sensor, LCD, cell, switches, fasteners.

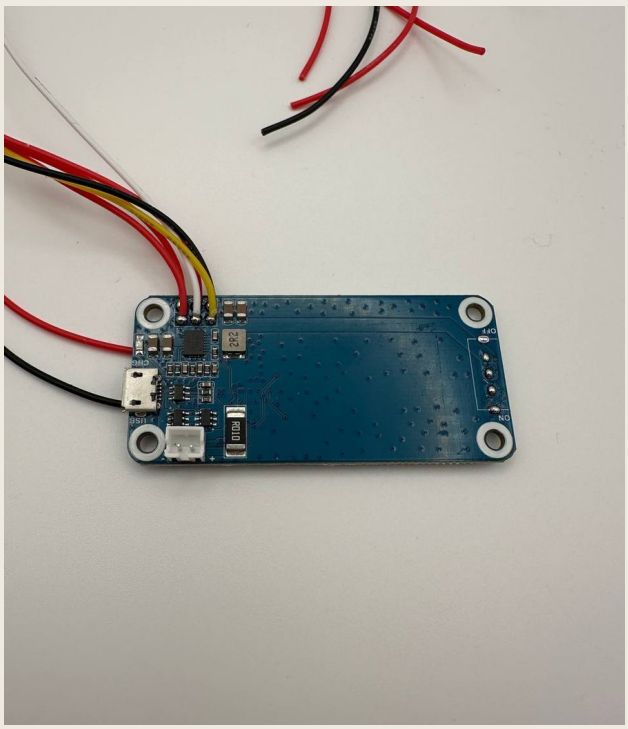


FIG. 06-3 — UPS HAT (C) prep: the factory GPIO header and onboard switch are desoldered; wires go straight to the pads (FIG. 04-2) — the HAT does not fit the shell otherwise.

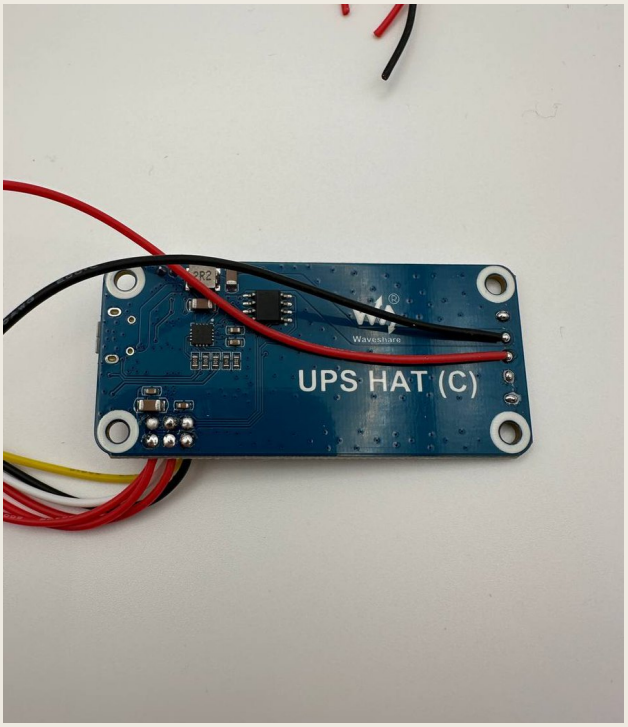


FIG. 06-4 — Power-switch leads (SW_0 / SW_1) soldered in place of the removed onboard switch. Keep all leads short so they route cleanly.

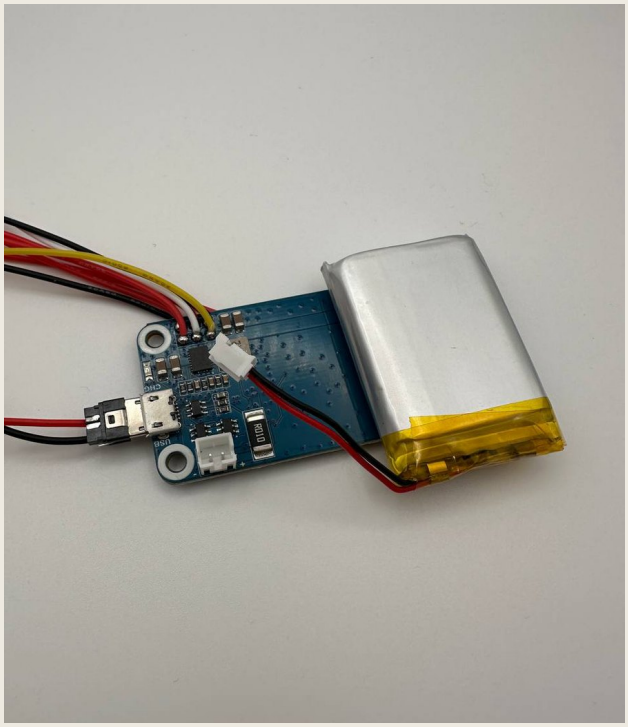


FIG. 06-5 — Charging input: a micro-USB connector with pre-soldered leads plugs into the HAT's CHG port and is wired to the USB-C panel socket (FIG. 04-7) — no risky port desoldering.

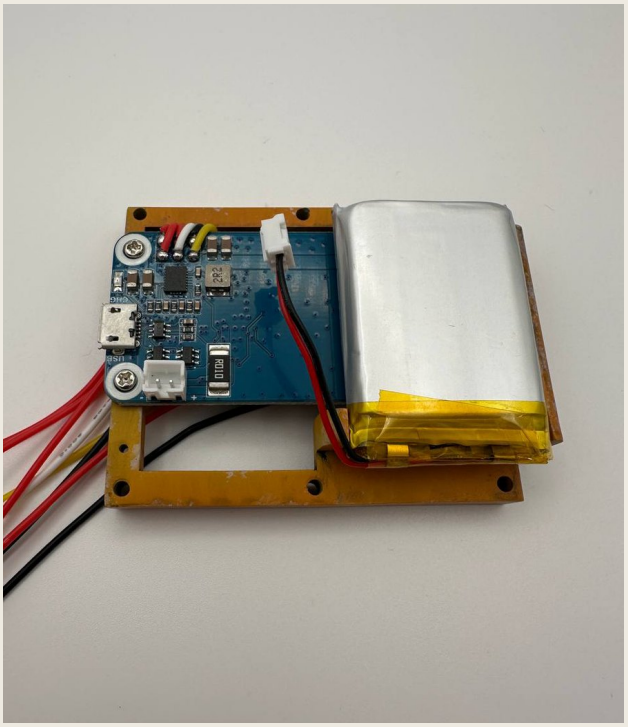


FIG. 06-6 — HAT mounted on the DIONE frame (M1.7×4); LiPo cell plugged into the BAT socket and secured with tape. Confirm polarity first.

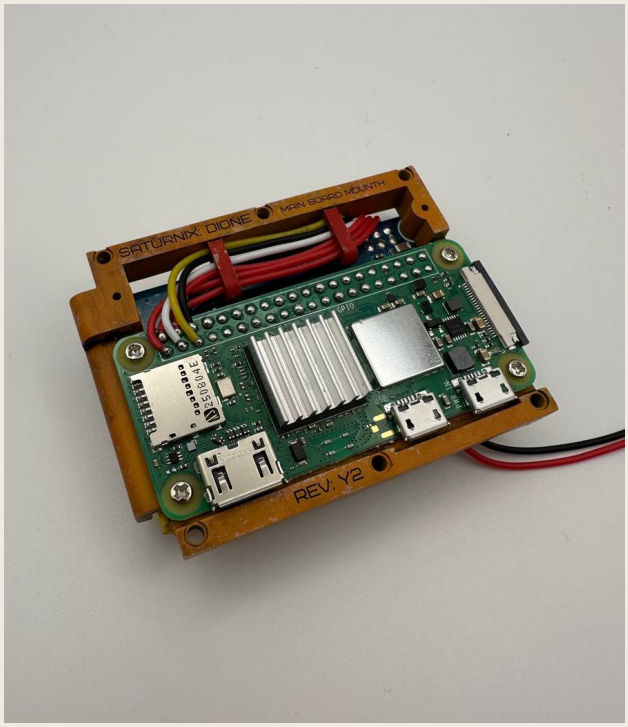


FIG. 06-7 — Raspberry Pi Zero 2 W (heatsink fitted) mounted on the DIONE frame above the HAT and screwed down; microSD inserted.

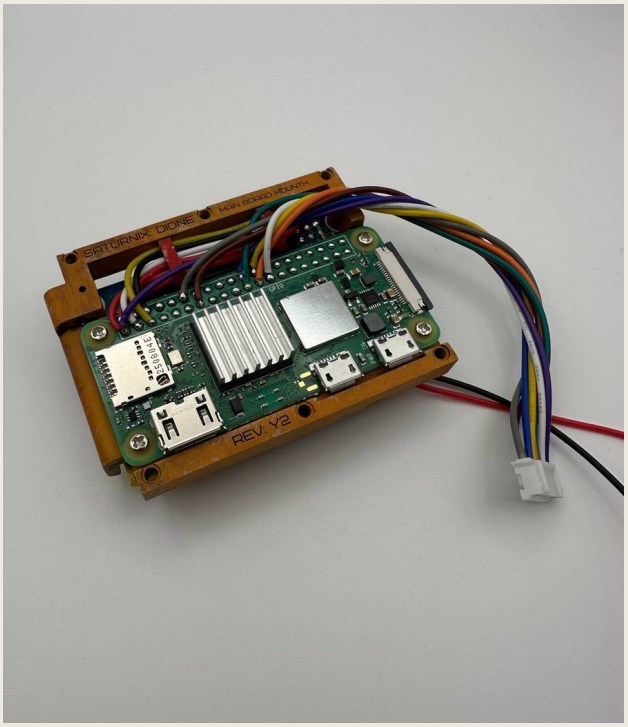


FIG. 06-8 — GPIO harness soldered per the master pinout (§04); LCD lines terminated with a connector, button and buzzer leads routed aft.

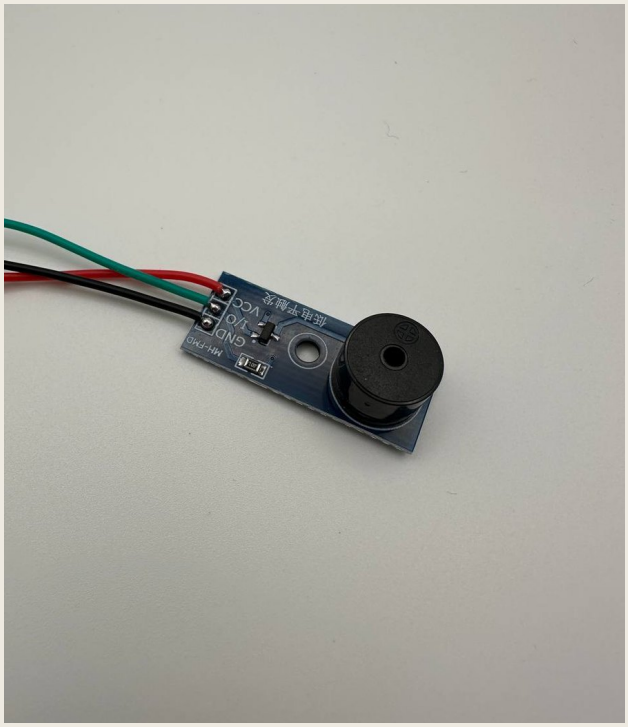


FIG. 06-9 — MH-FMD buzzer wired: VCC, I/O and GND (FIG. 04-5).

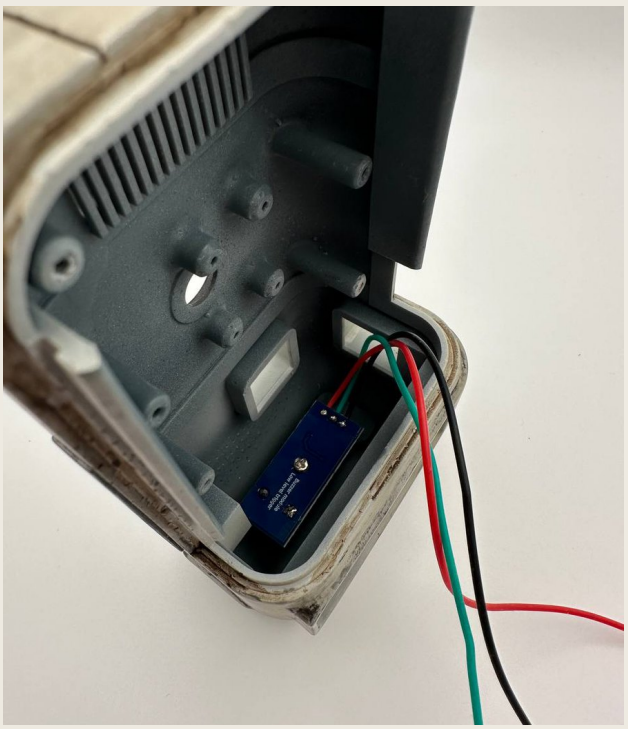


FIG. 06-10 — Buzzer module seated in its pocket in the main body; leads exit through the internal cutout.

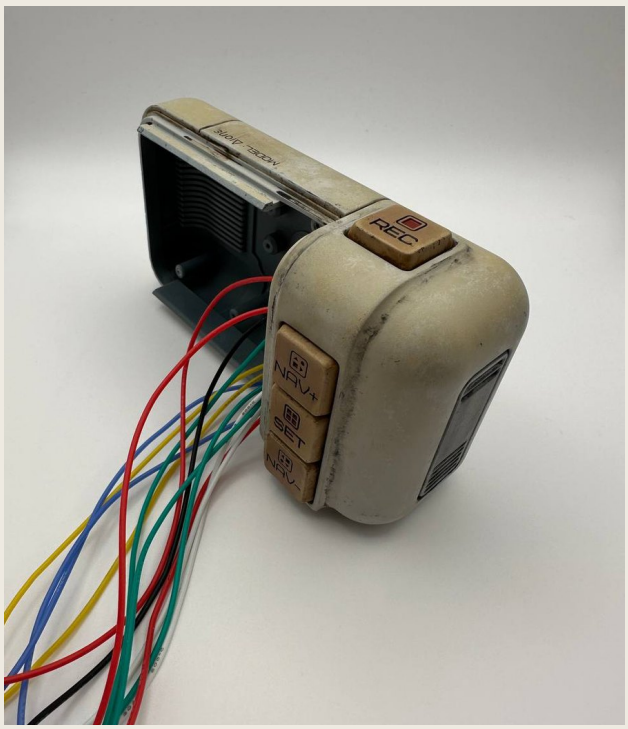


FIG. 06-11 — Switches and keycaps seated: REC on the top edge, NAV+ / SET / NAV– on the side; button wires exit for soldering.

FIG. 06-12 — IMX519 camera module screwed to the bosses inside the front shell, CSI ribbon attached before mounting.

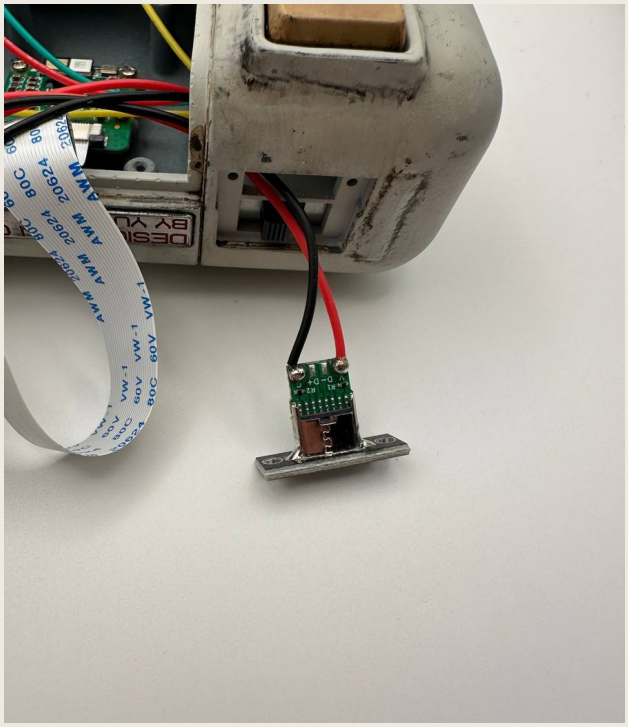


FIG. 06-13 — USB-C panel socket wired (5V / GND only); these leads meet the micro-USB pigtail from FIG. 06-5. Fit into the port cutout next to the power switch.

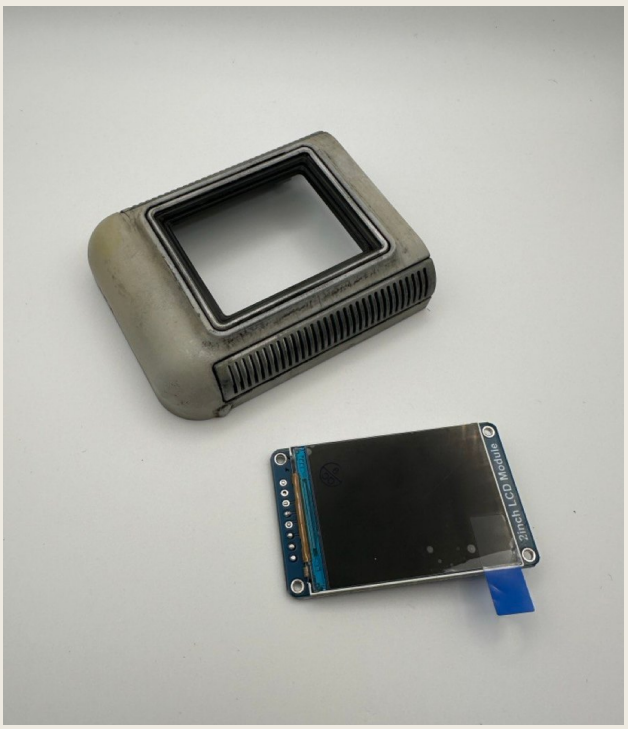


FIG. 06-14 — Rear cover and the 2" LCD module: the display mounts into the cover's integrated bezel.

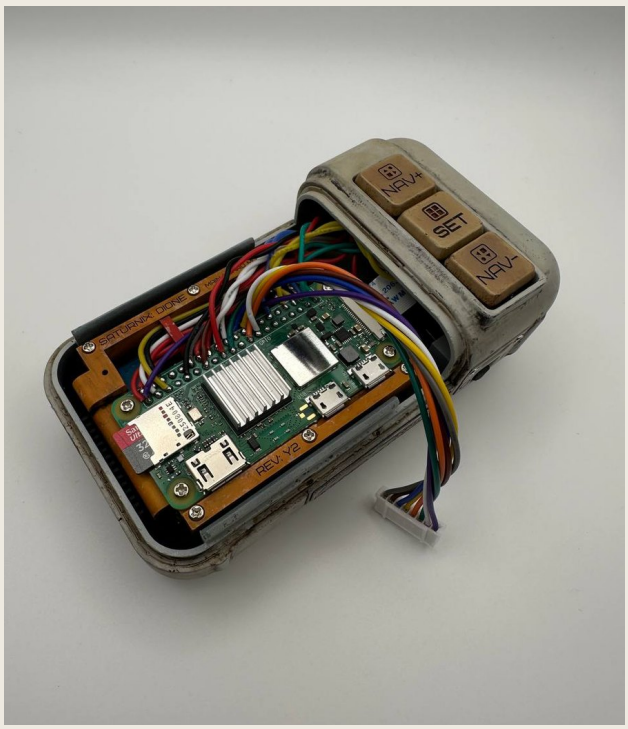


FIG. 06-15 — DIONE frame with the Pi stack seated in the body; LCD harness ready. Check no wires are pinched.

FIG. 06-16 — Rear cover with the display fitted, connected to the body via the LCD harness — last functional check before closing.

FIG. 06-17 — The finished SATURNIX Dione.

Close the shell and drive the screws — **M1.7×4** for internals, **M1.7×10** for the shell, **M2×4** where marked. Power on and re-run the §05 checklist (display, buttons, autofocus, buzzer, hotspot) before the first outing.

07 CONFIGURATION REFERENCE

All camera behaviour is controlled by a single **config.json** file in the root folder. Edit it with any plain-text editor. The camera reads it at startup, so reboot the camera to apply changes. Keep a backup of a working file before editing.

CAMERA IDENTITY

Key	Description
camera_name	Display name shown in UI
firmware_version	Firmware tag (shown in diagnostics)
sensor	Sensor model label
resolution	Full capture resolution [width, height]

GPIO PINS

Key	Description
gpio_pins.left / right / select / capture / focus / buzzer	BCM pin numbers for each button + buzzer. Range 0–27, must be unique. Falls back to defaults if invalid.

CAPTURE OPTIONS — values available in menus

Key	Description
iso_options	Selectable ISO values
shutter_options	Selectable shutter speeds
ev_options	EV compensation steps
jpeg_quality_options	JPEG quality levels
wb_options	White balance presets
denoise_options	Denoise modes
focus_options	AF-C / AF-S / MF
mode_options	Capture modes (JPG+DNG / JPG / Dynamic)
grid_options	Composition grid overlays
film_options	Film simulation presets
burst_options	Burst shot counts (Dynamic only)
burst_interval_ms	Delay between burst frames (ms)
hdr_options	HDR off / 3-FILES
hdr_ev_steps	EV offsets for HDR brackets, e.g. [-2, 0, 2]
timer_options	Self-timer durations (seconds)

WATERMARK & METADATA

Key	Description
watermark_options	On / Off choices
watermark_color	Watermark RGB [r, g, b]
watermark_text	Text printed on JPGs
exif_make	EXIF Make tag
exif_model	EXIF Model tag
exif_software	EXIF Software tag

POWER & STORAGE

Key	Description
battery_auto_shutdown	true / false — master switch for battery protection. Set false for builds without a battery: disables auto-shutdown and warning beeps
battery_critical_pct	% at which warning beep triggers
battery_shutdown_pct	% at which camera auto-shuts down (default 5% — protects the SD card; see the battery notice in §05)
storage_warning_mb	Free MB threshold for storage warning

WIFI HOTSPOT

Key	Description
wifi_ssid	Hotspot network name
wifi_password	Hotspot password
wifi_address	Camera IP on hotspot
wifi_interface	Network interface (usually wlan0)

COLORS — RGB arrays [r, g, b], 0–255

Group	Affects
colors.text	UI text (primary, dim, label)
colors.cursor	Selection cursor fill / text / glow
colors.background	Main + panel backgrounds
colors.accent	Accent, warning, success colors
colors.reticle	Focus reticle + AF states
colors.ui_elements	Grid, meters, battery, noise, dividers, etc.

UI BEHAVIOR

Key	Description
screen_rotation	Display rotation (0 / 90 / 180 / 270)
auto_hide_seconds	Seconds before UI auto-hides
ui_cal_marks	Show calibration ticks (true / false)
ui_static_noise	Show static noise overlay
ui_transitions	Enable screen transitions
ui_technospam	Show sci-fi flavor labels
volume	Buzzer volume: OFF / LOW / MED / HIGH

SAFE TO EDIT FREELY — all *_options, colors, watermark text, WiFi credentials, UI toggles, battery thresholds.
EDIT WITH CARE — gpio_pins (must match your wiring), resolution (must match sensor), screen_rotation (must match physical LCD mount).

08 CREDITS & LICENSE

Firmware developed with **Claude Code** (Anthropic) and **Codex** (OpenAI). SATURNIX is an independent project and is not affiliated with, sponsored, or endorsed by Anthropic/ OpenAI.

SATURNIX is fully open source under a multi-license model. Each component of the project carries its own license — the full texts live in the repository:

Component	License	Commercial use
Firmware (Python)	MIT — **LICENSE** in the repo root	allowed, no restrictions
Hardware (CAD / STL / STEP)	CERN-OHL-S-2.0 — **hardware/LICENSE-HARDWARE.md** + **NOTICE.txt** (a copy of the notice ships inside every CAD zip)	allowed under the license terms
Documentation & media	CC-BY-4.0 — **LICENSE-DOCS.md** (SATURNIX logo and brand assets excluded)	allowed with attribution
SATURNIX name & logo	Naming policy — **BRANDING** in the repo root	only as "SATURNIX-compatible" / "based on SATURNIX"

Hardware reciprocity. CERN-OHL-S-2.0 is strongly reciprocal: if you distribute modified hardware, you must publish your modified design sources under the same license and state the Source Location.

Naming policy. The design is fully open — only the name is protected, the same model as Arduino or Prusa. Forks, mods, compatible parts, kits and builds (including commercial ones) are free to use the wordings **"based on SATURNIX"**, **"SATURNIX-compatible"** or **"SATURNIX-derived"**. The plain name SATURNIX and the bare logo are reserved for official releases by the author; modified versions must clearly state that they are modified and must not present themselves as the original release.

Attribution (a kind request, not a legal demand): *"Based on the SATURNIX Camera open-source hardware project by Yutani140x."*

Contributions welcome — see the repository for how to get involved.

Supporters — thank you

SATURNIX exists in the open thanks to the people who chipped in. In no particular order:

Supporter	Via
Paul Black	Buy Me a Coffee — supporter
michael	Buy Me a Coffee — supporter
LazyDmedia	Patreon — patron

And a wave to the Patreon followers cheering from the sidelines: **Fabrizio Guccione**, **JoshMahan**, **Richard Jewkes**, **RC**, **Nathan Boom**, **IxeLWorx**.

github.com/Yutani140x/saturnix-camera