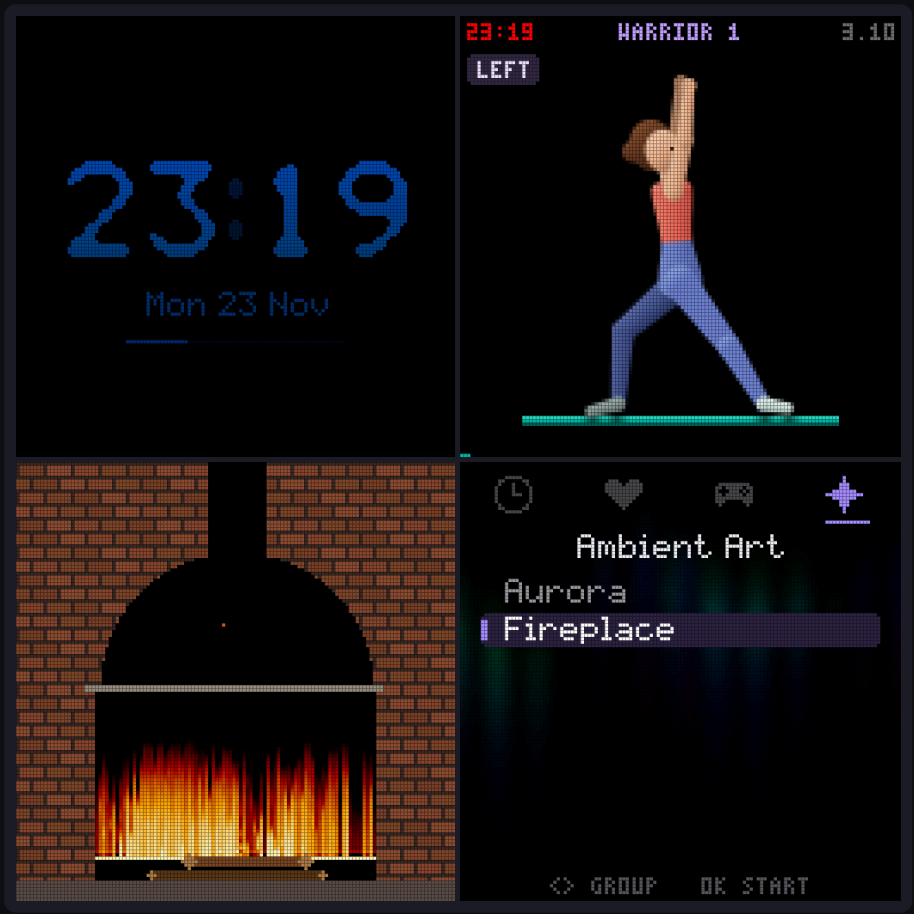
Build **Guide**

Parts, sourcing, wiring, assembly and bring-up for the 128 × 128 LED wall display — Teensy 4.1, SmartLED Shield V5 and four 64 × 64 HUB75 panels.



Prices checked October 2026 (EUR, indicative).
Items marked *est.* are estimates.

Revision 2
supersedes the October 23 hardware manual

STEP 1

Choose the panels

Four 64 × 64 HUB75 panels in a 2 × 2 square make the 128 × 128 display. The pixel pitch sets the size; the firmware is the same for all of them.

PITCH	ONE PANEL	DISPLAY	AVAILABILITY
P2.5	160 mm	320 mm	Waveshare, common. Small for a wall.
P3	192 mm	**384 mm**	Waveshare 64×64 P3 (RobotShop EU ≈ €33), Adafruit #4732. Easiest to source.
P4	256 mm	**512 mm**	HackSpark (FR) ≈ €35, AliExpress. The planned size — **verify the scan rate**.
P5	320 mm	640 mm	64×64 P5 panels are rare (most P5 panels are 64×32). Not recommended.

Recommendation: P4 when the seller confirms the checklist below; otherwise P3.

Compatibility checklist — ask before paying

- **64 × 64, 1/32 scan, HUB75E** (address lines A–E; E on HUB75 pin 8). The SmartLED Shield V5 drives E natively.
- **Driver chips:** ICN2028, FM6124, DP5220 and similar are fine. **FM6126A** works with one firmware option (page 7). **ICN2038S, ICN2053, GS6238S** and "AB" panels (only A/B address lines) do **not** work with SmartMatrix.
- Ask for a photo of the back showing the chips, and order **all four in one batch** so colour and wiring match.
- HackSpark's P4 listing says "1:16 scan" — ask. A true 1/16-scan 64×64 needs a different panel type and can't use C-shape stacking.

! **Buy the SmartLED Shield V5 first.** It is discontinued at Crowd Supply, SparkFun and Adafruit; Smart Prototyping still lists it (ships from China). The design is open source: the Gerbers and BOM in the SmartMatrix repository can be fabricated and assembled (e.g. JLCPCB) as a fallback.

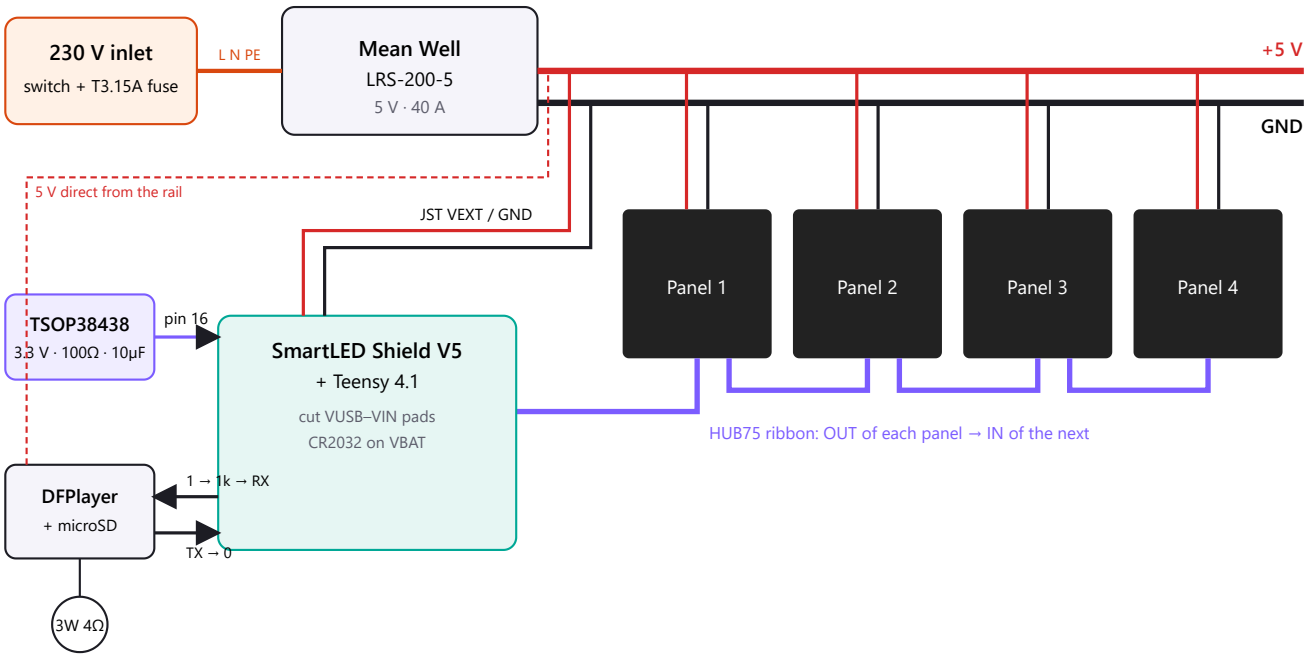
STEP 2

Shopping list

ITEM	QTY	RECOMMENDED PART / WHERE	≈ EUR
SmartLED Shield for Teensy 4 (V5)	1	smart-prototyping.com (or fabricate from the SmartMatrix repo)	20–25 + VAT
Teensy 4.1, with headers	1	SparkFun DEV-16771 via BerryBase (DE), The Pi Hut, Electrokit, PL shops	32–49
64×64 HUB75E panels, 1/32	4	HackSpark P4 · Waveshare P3 · AliExpress (*est.* €18–30)	120–140
5 V power supply	1	**Mean Well LRS-200-5** (5 V 40 A, fanless) · LRS-150F-5 (22 A) with a brightness cap	28–40 *est.*
IEC inlet, switch + fuse	1	C14 inlet with switch and fuse holder, T3.15 A slow-blow, 3-core lead	8–12 *est.*
DC wiring	1 lot	1.5 mm² red/black, ferrules, lever connectors, 2 × 1000–2200 µF 10 V caps	15–25 *est.*
DFPlayer Mini	1	genuine DFRobot DFR0299 — RobotShop EU, Arduino Store	7
microSD card	1	8–32 GB, FAT32	5
Speaker	1	40 mm, 3 W, 4 Ω	3–5
IR receiver kit	1	Vishay TSOP38438 (or TSOP38238), 100 Ω, 10 µF	2
Remote	1	Apple Remote (A1294 aluminium / A1156), or 17-key NEC kit remote	2–12
RTC battery	1	CR2032 + holder	3
Small parts	1 lot	1 kΩ resistor, hook-up wire, Dupont / JST leads	3
Frame	1	2020 extrusion + corner brackets + M5 T-nuts, or plywood shadow box	30–45
Panel mounts	~16	12 mm M3 magnetic mounts + galvanised steel strip	12–20
Diffuser	1	3 mm smoked / "black LED" cast acrylic, cut to size (metacrilato a medida)	20–45 *est.*
		Total	**≈ 330–450**
OPTIONAL — RESERVED FOR FUTURE FIRMWARE FEATURES			
Light sensor TSL2591	1	Adafruit #1980 — auto-brightness (night dimming is schedule-based today)	7–8
ESP32-C3 board	1	Weather feed over UART (the Weather scene waits for it)	3–12
PSRAM APS6404L	1	Not needed at 128 × 128	2.50

STEP 3

Wiring



Teensy pins used by the shield

2 (OE), 3 (LAT), 6, 7 (CLK), 9–13 (panel data), and 4, 5, 22 reserved for its APA102 output. Pin 13 is panel data, so the Teensy's LED flickers — normal.

> ! The previous hardware manual put the IR receiver on **pin 2**. That is the panels' OE line — use **pin 16**.

Coriolis connections

IR receiver OUT	pin 16 (TSOP on 3.3 V)
DFPlayer TX	pin 0 (RX1)
DFPlayer RX	pin 1 (TX1) via 1 kΩ
RTC battery	VBAT / GND
Future: light sensor	18 SDA / 19 SCL
Future: ESP32	14 TX3 / 15 RX3

Power

- Four 64 × 64 panels draw **4–5 A each at full white** (16–20 A in total). The LRS-200-5 (40 A) has plenty of headroom. With the 22 A LRS-150F-5, cap brightness at compile time: `-DCORIOLIS_MAX_BRIGHTNESS=180`, or limit only the brightest frames with `-DCORIOLIS_MAX_MILLIAMPS=16000`. Showing the clock, Coriolis uses a few amps.
- Run each panel's power pigtail straight to the PSU terminals, or one 1.5 mm² pair per two panels (2.5 mm² for a single trunk). Ferrule every end. Fit a 1000–2200 µF capacitor across each pair at the panels.
- The shield feeds the Teensy from its JST input through a 0.5 A diode. **Cut the VUSB–VIN pad bridge** on the underside of the Teensy so USB and the 5 V rail never fight. Afterwards the Teensy only runs with the PSU on — flash it with the PSU on.
- Power the DFPlayer (and later the ESP32) from the 5 V rail, not through the Teensy.
- The TSOP must run from **3.3 V**: Teensy 4.1 pins are not 5 V tolerant.

! **Mains safety.** Earth the PSU's ⏚ terminal and any metal frame. Keep the mains terminals covered. Use the fused, switched IEC inlet and strain-relieve the mains lead. If you are not comfortable wiring mains, have an electrician do this part.

STEP 4

Assembly

1. **Teensy.** Solder headers if needed. Cut the VUSB–VIN pads. Solder the CR2032 holder to VBAT and GND.
2. **Shield.** Plug the Teensy into the shield (the shield covers the 4.0 footprint; snap off its perforated tab to reach pins 24–41 if ever needed). Fit the JST power pigtail.
3. **IR receiver.** TSOP pins, lens facing you: OUT · GND · Vs. Vs → 100 Ω → 3.3 V, 10 µF from Vs to GND at the sensor, OUT → pin 16. Mount it behind a small hole at the frame edge.
4. **DFPlayer.** VCC → 5 V rail, GND → common ground, TX → pin 0, RX ← 1 kΩ ← pin 1, SPK1/SPK2 → speaker (neither side to ground).
5. **Bench-test with one panel** before building the frame (Step 6).
6. **Frame.** For P4: 2020 rails sized so the 512 mm panel square sits inside with ~2 mm play (≈ 550 mm outside), plus 2–3 cross-members behind the panel seams carrying steel strip for the magnetic mounts. Butt the panels seam-to-seam before tightening. A wooden box: 70–90 mm deep sides, MDF or ACM back with steel strips, French cleat (≈ 5–6 kg total).
7. **Chain the panels** (Step 5). Mount the electronics plate on a cross-member and the PSU low in the frame with space around it. Cut ventilation slots top and bottom.
8. **Diffuser.** Smoked acrylic directly on the panels gives crisp, high-contrast pixels; a grid louver (walls ≤ 1 mm at P4) under a tinted sheet gives square pixels.

STEP 5

Panel layout

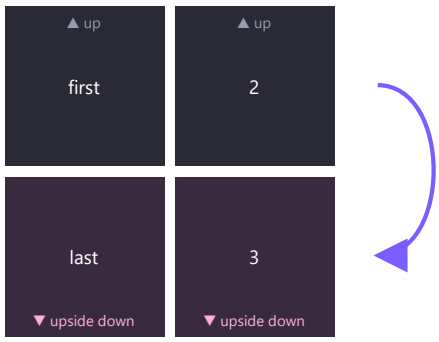
The firmware drives one chain of four panels as 128 × 128. How they're chained is set by `CORIOLIS_PANEL_OPTIONS` at the top of `firmware/coriolis/coriolis.ino`.

Default (no options) — all upright



top row first, long cable back across to the bottom row

C_SHAPE_STACKING — snaking chain



bottom row mounted upside-down, shortest cables

Data order within a row depends on the panels' IN/OUT direction — so **don't guess: use the test pattern**. Over USB serial (115 200 baud) type `t`: four numbered, coloured quadrants appear with an arrow in the top-left corner. When 1-2 / 3-4 read in order and the arrow points at the top-left, the layout is right. If the whole picture is merely rotated, use **Settings** → **Rotate** instead. `SM_HUB75_OPTIONS_BOTTOM_TO_TOP_STACKING` starts the chain from the bottom row; combine options with `|`.

STEP 6

Firmware and bring-up

Toolchain (Windows)

```
winget install ArduinoSA.CLI
powershell -File firmware/setup.ps1     # Teensy core, SmartMatrix 4 + patch, shared code
arduino-cli compile --fqbn teensy:avr:teensy41 firmware/coriolis
arduino-cli upload  --fqbn teensy:avr:teensy41 firmware/coriolis
```

If your Windows user name has non-ASCII letters (e.g. “Nykänen”), GCC can't find its own headers: map the folders to a drive letter first (`subst X: C:\path`) and build from there. Options such as `CORIOLIS_PANEL_OPTIONS` and `CORIOLIS_MAX_BRIGHTNESS` are defines at the top of the sketch.

Bring-up, one stage at a time

1. **Teensy alone over USB** (before cutting VUSB–VIN, or with the PSU on). Flash, open a serial monitor at 115 200 baud, type `i`: it prints the time and the active scene.
2. **One panel** as panel 1, powered. Type `t`: red quadrant 1 appears. Blank or noise → wrong panel type, or FM6126A chips: set `SM_HUB75_OPTIONS_FM6126A_RESET_AT_START`.
3. **All four panels.** Test pattern again; fix the chain option (Step 5). `t` again leaves it.
4. **Remote.** Each button prints `[IR] 0x........ -> n`. `n` = 0 means the code isn't mapped yet — add it in `RemoteMap::decode()`.
5. **Sound.** Generate the card (`python tools/generate_voice_pack.py`), copy `01`, `02` and `ADVERT` to a FAT32 microSD, insert it in the DFPlayer. Start a Yoga session: countdown ticks, then the pose name.
6. **Time.** The Teensy Loader sets the RTC to your PC's local time on upload; the first boot of each new build converts it to UTC (the RTC keeps UTC and the display applies the zone and EU summer time). Check it; fix it in **Settings** → **Clock**, or send `T<unix seconds UTC>` over serial — in PowerShell: `"T$([DateTimeOffset]::UtcNow.ToUnixTimeSeconds())"`.

Serial bench commands

u d l r	arrows	**k**	OK	**b**	back
m	menu	**s**	settings	**c**	clock overlay
n	next scene	**h**	home	**t**	test pattern
i	info	**T<unix>**	set the clock (UTC seconds)		

REFERENCE

Troubleshooting

SYMPTOM	LIKELY CAUSE / FIX
Panels dark, Teensy LED flickers	No panel power; ribbon on a panel's OUT side; FM6126A panels (Step 6).
Random pixels / garbage rows	1/16 vs 1/32 scan mismatch; incompatible driver chips; long or poor ribbons.
Quadrants in the wrong place	Chain option (Step 5). Whole picture rotated → Settings → Rotate.
Red and blue swapped	Some panels are RBG/BGR: swap the channels in `blit()`.
Clock 1–2 h off after flashing	Check Zone / Summer in Settings → Clock, then the hour.
Clock lost after power cut	CR2032 missing or flat on VBAT.
Remote ignored	Receiver on 3.3 V and pin 16? Watch for `[IR]` lines; map unknown codes.
No sound	FAT32 card with `01`, `02`, `ADVERT`; volume in Settings; 1 kΩ on DFPlayer RX.