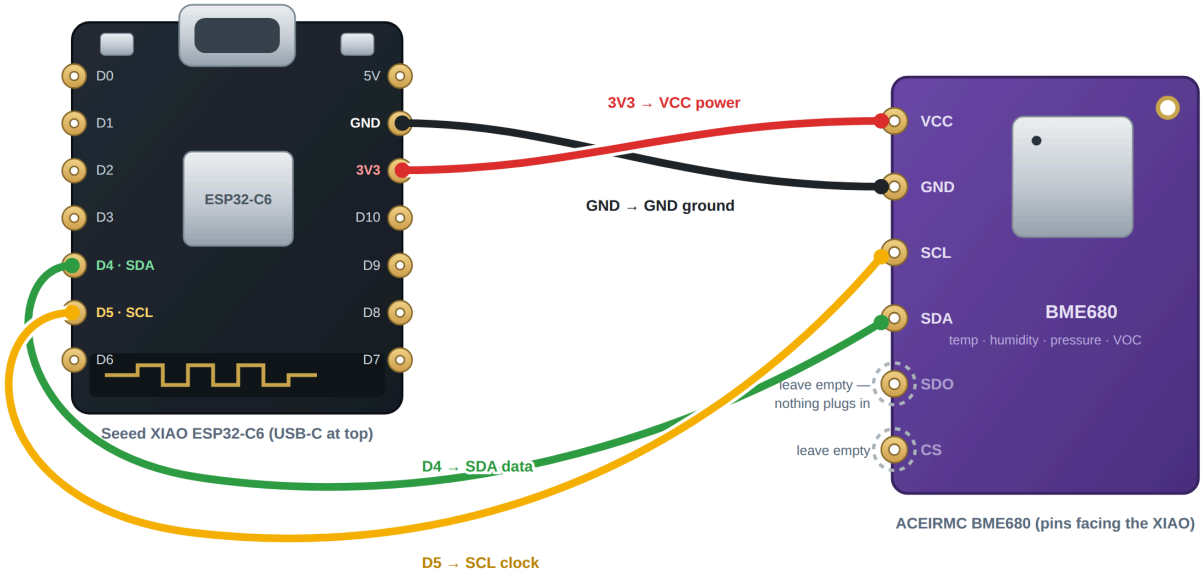
Build your own weather & air-quality station

Seeed Studio **XIAO ESP32-C6** + ACEIRMC **BME680** — four wires, one afternoon, and your sensor is live on the community map.

The whole hookup — four wires, that's it

Power (red), ground (black), clock (yellow), data (green). SDO and CS on the sensor stay empty.

Your jumper wires can be any colors.
Match the printed pin labels — the colors here are just for reading.



Connection checklist (tick as you go)

- ☐ XIAO **3V3** → sensor **VCC**
- ☐ XIAO **GND** → sensor **GND**
- ☐ XIAO **D4 (SDA)** → sensor **SDA**
- ☐ XIAO **D5 (SCL)** → sensor **SCL**

SDO — empty on purpose

CS — empty on purpose

Never use the 5V pin for this sensor's power — 3V3 is the safe choice.

The #1 mistake (everyone does it once)

Swapping SDA and SCL. Nothing breaks — the sensor just shows up as "NOT FOUND". If that happens, swap the green and yellow wires at the sensor end and try again.

Double-check red is on 3V3 (not 5V) before first power-up.

4	2	~1 hr	$0	15 s
wires to connect	numbers to edit	first build, incl. software	you already have the parts	between live readings

What you'll have at the end: a station reporting temperature, humidity, barometric pressure and air quality (VOC) to **sensormap.cloud** every 15 seconds — with live charts, history, phone & email alerts, CSV export, Home Assistant integration, and a free API. It sits on the same map as 8,000+ airports, EPA monitors and community stations.

What you need

PART	NOTES
Seeed Studio XIAO ESP32-C6	The brains + WiFi. Tiny black board with USB-C.
ACEIRMC BME680 board	The sensor. Purple/blue board with a silver square can.
USB-C data cable	Must be a **data** cable — many "charging" cables can't talk to the board. If in doubt, use the cable that came with a phone.
4 jumper wires	Male-to-male for a breadboard build (or see the no-breadboard options on page 5).
Small breadboard	Any half-size board. Optional but recommended for a first build.
Soldering iron + solder	Only for attaching the header pins — about 20 joints, 10 minutes.

No soldering iron?
You can twist stripped wire ends through the holes for a test — it works, just not reliably. For anything permanent, solder (or ask a friend with an iron; it's a five-minute favor).

What this station reports

READING	FROM
Temperature (°C/°F)	BME680
Humidity (%)	BME680
Barometric pressure	BME680
Air quality — VOC index	BME680 gas sensor
WiFi signal, chip temp, uptime, free memory	the XIAO itself

SensorMap also computes dew point and "feels like" for you automatically.

How the pieces talk

BME680 measures the air → whispers numbers to the **XIAO** over two wires (that's "I²C") → XIAO posts them over your **WiFi** to **sensormap.cloud** → they appear on the live map, in your charts, and in any alerts you've set.

The build at a glance

1. **Solder the header pins** onto both boards (one-time, ~10 min).
2. **Connect four wires** — power, ground, clock, data.
3. **Add the device on sensormap.cloud** and download your personal, pre-configured firmware zip.
4. **Set up Arduino IDE** (free) with the ESP32 boards package and three libraries.
5. **Change two numbers** in one file so the firmware uses the XIAO's pin numbering.
6. **Hit Upload,** watch the serial monitor say `HTTP 200` — you're on the map.

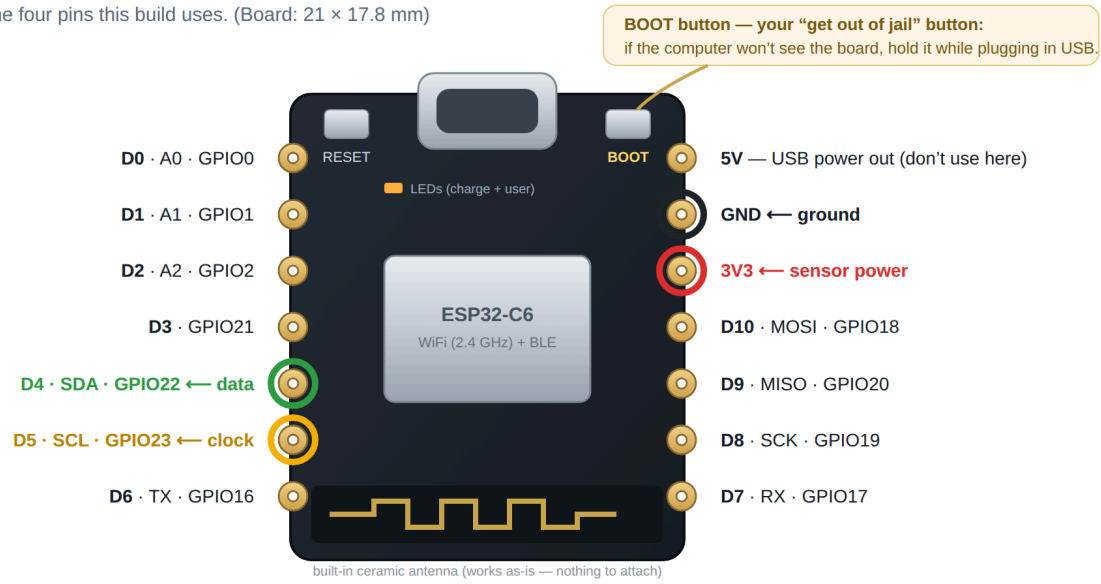
Nothing here can really go wrong.
The boards are polite about mistakes at 3.3 V — the classic slip-ups (swapped wires, wrong pin numbers) just make the sensor invisible until you fix them. The troubleshooting page at the back covers every "NOT FOUND" we know about.

Meet board #1 — the XIAO ESP32-C6

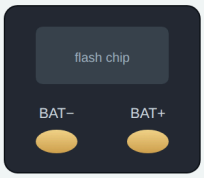
This is the computer. It has WiFi, a USB-C port for power and programming, and 14 gold pads. You will use exactly four of them.

Seeed Studio XIAO ESP32-C6 — pin map (top view, USB at the top)

Colored rings = the four pins this build uses. (Board: 21 × 17.8 mm)



Flip side (back of the board)



- The two oval pads are for an optional 3.7 V lithium battery — the board even charges it over USB. You don’t need this: a plain USB-C phone charger runs the station forever.
- Nothing on the back needs soldering for this build.
- Powering by USB? The whole station draws less than a night-light (~0.5 W).

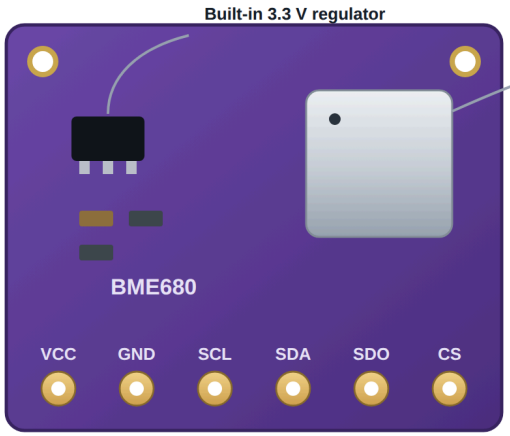
⚠ One habit to build right now: power sensors from 3V3, not 5V.

The 5V pad carries raw USB power. Our BME680 board happens to tolerate it, but many other sensors don't — keeping red wires on 3V3 means you never have to think about it.

Meet board #2 — the BME680

ACEIRMC BME680 sensor board — meet the pins

One chip measures four things: temperature, humidity, barometric pressure, and air quality (VOC gas).



The silver can is the sensor itself.
The pin-hole in its corner is the air vent — never cover it, poke it, or get it wet.

Where each pin goes

- **VCC** → XIAO **3V3** (power)
- **GND** → XIAO **GND** (ground)
- **SCL** → XIAO **D5** (clock)
- **SDA** → XIAO **D4** (data)
- **SDO** → leave empty (this sets address 0x77)
- **CS** → leave empty (selects I²C mode)

Two things worth knowing: ① This board is happy on 3.3 V or 5 V — we use 3V3 because it's always safe.
② Some batches print the pins in a different order. Wire by the **printed labels on your board**, never by position.
Sold as ACEIRMC / GY-BME680 / CJMCU-680 — they're all the same design. Yours may be purple or blue.

STEP 1 · SOLDER THE HEADERS

1. Snap the included pin strips to length if needed: **7 + 7** for the XIAO, **6** for the sensor (a 4-pin strip is fine too — SDO and CS can go pinless).
2. Push the **short** end of the pins up through the **bottom** of each board, long legs pointing **down** — that's the breadboard side. The XIAO's USB-C port faces **up**; the sensor's silver can faces **up**.
3. Park the pins in the breadboard to hold everything square while you solder the top side. One pad at a time: touch iron to pad+pin, feed a little solder, two seconds, done.

Good joint / bad joint

A good joint looks like a tiny shiny volcano hugging the pin. A bad one is a dull ball sitting on top, or a bridge of solder connecting two neighbors. Bridges are the #1 cause of "it worked, then it didn't" — drag the hot iron between the pads to clear one.

Keep heat away from the sensor can.

Solder from the top surface, be done in a couple of seconds per pad, and never touch the silver can with the iron. The BME680 survives soldering fine — it's assembled in an oven, after all — but the little vent hole should stay clean.

STEP 2 · CONNECT FOUR WIRES

The wiring

#	FROM (XIAO)	TO (SENSOR)	WHAT IT IS	SUGGESTED COLOR
1	**3V3**	**VCC**	Power	**red**
2	**GND**	**GND**	Ground	**black**
3	**D4** (SDA · GPIO22)	**SDA**	Data	**green**
4	**D5** (SCL · GPIO23)	**SCL**	Clock	**yellow**
—	**SDO** and **CS** on the sensor		stay **empty** — that's deliberate (see below)	

Why SDO stays empty:
that pin picks the sensor's I²C "street address". Left alone, this board answers at `0x77` — exactly the address your SensorMap firmware calls. Wire SDO to GND and the sensor moves to `0x76` and stops answering. So: empty. (CS empty just means "speak I²C", which is what we want.)

Three ways to physically build it

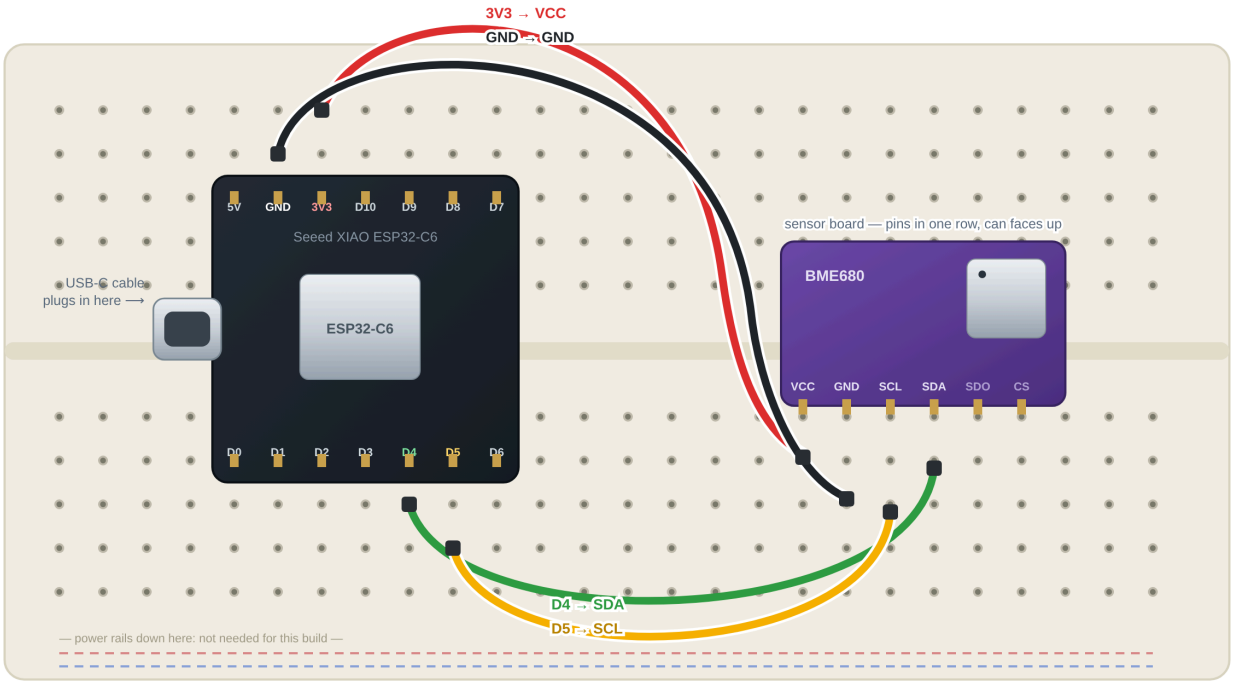
	HOW	BEST FOR
A · Breadboard recommended	Solder headers on both boards, push them into a breadboard, connect with 4 male-male jumpers (next page).	First build — easy to fix, easy to expand later.
B · Direct jumpers	Solder headers, then run 4 **female-female** jumper wires pin-to-pin. No breadboard at all.	Compact installs in a small case.
C · Solder wires	Skip headers; solder 4 short wires (~7 cm) directly pad-to-pad.	The permanent, vibration-proof version. Do this after A has run happily for a week.

Keep the wires shortish.
I²C likes tidy: anything up to ~20 cm is trouble-free. If you want the sensor far from the XIAO (outside a case, say), extend the red/black power wires — not the green/yellow ones — or just mount the whole XIAO at the sensor.

Breadboard layout (option A)

On the breadboard — how it actually looks

XIAO straddles the middle groove; the sensor's six pins each get their own column. Four jumpers connect them.



Each jumper goes into the **same column** as the pin it connects to — that's how a breadboard joins them. Left holes of the sensor row and everything else stay empty. Colors don't matter; columns do.

Before you plug in USB — the 60-second check

- **Red** goes 3V3 → VCC (not 5V, not GND).
- **Black** goes GND → GND.
- **Green** D4 → SDA, **yellow** D5 → SCL (the crossover mistake is the classic).
- SDO and CS have **nothing** in their columns.
- Each jumper shares a **column** with its pin — same vertical strip of 5 holes.
- No stray wire ends touching each other.
- Solder side: no bridges between neighboring pins.
- Sensor can faces up, nothing covering its vent.

Then plug it into USB once, just to see it wake up.

A tiny LED on the XIAO lights. Nothing else visible happens yet — the brains are empty until Step 6. Unplug again while we get the firmware ready.

STEP 3 · CREATE THE DEVICE ON SENSORMAP

Get your personal firmware zip

SensorMap builds the firmware *around your device* — WiFi name, device key, sensor list all baked in — so there are no config files to hand-edit (well, two numbers, but that's Step 5).

1. Go to **www.sensormap.cloud** and sign in (or create your free account).
2. Open **Devices → + Add device**. Name it (e.g. "Backyard station"), drop the map pin where it will live — it's fine to place the pin down the street for privacy — and choose public (on the community map) or private (only you).
3. Open your new device and hit **Manage**, then find the **Ready-to-flash firmware** panel.
4. Fill it in exactly like this:

What are you connecting?	**ESP32 (Arduino)**
Your board	**ESP32-C6**
Role	**Standalone (WiFi)**
WiFi network	**your home WiFi — 2.4 GHz**
WiFi password	••••••••
Sensors wired to this device	✓ **BME680** · ✗ **un-tick BME280**

5. Click **Save settings**, then **Download ready-to-flash firmware**. You get a zip named after your device — unzip it somewhere you can find (it contains `sensormap.ino`, a `src/` folder, and a START-HERE.txt).

Two details that matter here:

① The sensor checkbox list starts with **BME280** ticked (it's the most common sensor). Your board is the **BME680** — tick that one and **un-tick BME280**, or the firmware will look for the wrong chip. ② The C6 talks **2.4 GHz WiFi only**. If your router has one combined name for 2.4+5 GHz, that's fine — the C6 will find the 2.4 side by itself.

Your WiFi password, plainly:

it's stored in your SensorMap account (encrypted at rest, visible only to you) so the zip can be pre-configured. It travels in that zip and lives inside your own board — it is never shown on the site or shared anywhere.

STEP 4 · SET UP ARDUINO IDE

One-time computer setup (~15 min, mostly downloads)

1. **Install Arduino IDE** (free) from **arduino.cc/en/software** — Windows, Mac or Linux.
2. **Teach it about ESP32 boards:** File → Preferences → "Additional boards manager URLs", paste:

   ```
   https://espressif.github.io/arduino-esp32/package_esp32_index.json
   ```

3. **Install the boards package:** Tools → Board → Boards Manager → search **esp32** → install "**esp32 by Espressif Systems**". Version 3.0 or newer — the C6 is only in 3.x. This is the big download (a few hundred MB); good moment for coffee.
4. **Install three libraries:** Tools → Manage Libraries, search and install each, exact names:

`Adafruit BME680 Library`	the sensor driver (say **yes** when it offers to install dependencies)
`Adafruit Unified Sensor`	helper the driver needs
`ArduinoJson`	by Benoît Blanchon — builds the messages sent to SensorMap

5. **Open the sketch:** File → Open → the unzipped folder → `sensormap.ino`.
6. **Pick the board:** Tools → Board → esp32 → **XIAO_ESP32C6** (type "xiao" in the board search box to find it fast).

> ⚠ **The zip's own instructions will say board "ESP32C6 Dev Module" — use XIAO_ESP32C6 instead.**
> Both can flash your board, but the XIAO entry also configures the board's built-in antenna switch correctly, which means full WiFi range. (The generic entry can leave the radio whispering into the wrong antenna.)

STEP 5 · CHANGE TWO NUMBERS

Point the firmware at the XIAO's I²C pins

The wizard's "ESP32-C6" preset is written for generic C6 dev boards, whose I²C lives on GPIO 6/7. On the XIAO, the I²C pads are **GPIO 22 (D4)** and **GPIO 23 (D5)** — where you wired the sensor. In the Arduino IDE, click the **config.h** tab (it's the file `src/config.h`), find the pins section, and make it read:

```
/* 5. Pins — before */
#define I2C_SDA_PIN   6
#define I2C_SCL_PIN   7
```

```
/* 5. Pins — after  */
#define I2C_SDA_PIN   22
#define I2C_SCL_PIN   23
```

⚠ **The zip contains TWO copies of config.h — edit the right one.**
The Arduino IDE only compiles the one that shows up as the **config.h tab** inside the IDE (that's `src/config.h`). The copy in the `include/` folder is for PlatformIO only — editing it does nothing in Arduino, and the sensor stays "NOT FOUND". (Field-tested: this exact mix-up got our very first builder.)

That's the whole edit. Everything else in the zip is already yours: WiFi, device key, the BME680 switch. A "Seeed XIAO ESP32-C6" preset is coming to the wizard so future downloads won't even need this.

STEP 6 · FLASH IT

1. Plug the XIAO into your computer with the USB-C **data** cable.
2. Tools → Port → pick the new entry that appeared (Windows: a COM number · Mac: usbmodem…).
3. Click **Upload** (the arrow). First build takes a couple of minutes. "Done uploading" = success.
4. Tools → **Serial Monitor**, set **115200 baud**. Within seconds you should see:

```
[init] BME680   ok
[wifi] connected  ip=192.168.1.x  rssi=-52 dBm
[post] HTTP 200
[post] HTTP 200   ← one of these every 15 seconds — you are LIVE
```

Open your device page on **sensormap.cloud** — the readings are already there, charting away. Temperature, humidity, pressure and a **voc** air-quality index (0–500-ish; lower is cleaner air — hold a marker pen near the sensor and watch it spike).

Port never shows up?
Unplug → hold the tiny **BOOT** button (marked B, right of the USB port) → plug back in while holding → release. The board is now in "please program me" mode and the port appears. Still nothing: it's almost always the cable — swap for a known data cable.

STEP 7 · PUT IT SOMEWHERE HONEST

Placement is 90% of data quality

Do

- **Shade, always.** North side of the house, under an eave, a ventilated porch — direct sun turns any thermometer into fiction.
- **Let air flow.** If it lives in a box, drill generous vent holes low and high so air moves past the sensor.
- **Keep rain off.** The BME680 measures humidity happily but must not get *wet*. Outdoors = under cover, can facing down or sideways.
- **Give the sensor distance from the XIAO.** An inch or two of separation (or sensor upwind) keeps the C6's own warmth out of your readings.
- **Power from any USB-C phone charger.** It draws less than half a watt.

Don't

- On a sunny windowsill, above a radiator, near a dryer vent, or on top of the router.
- Sealed in an airtight jar "to protect it" — it will faithfully measure the inside of a jar.
- Directly against a wall that gets sun — walls radiate heat for hours.

Expect +1 °C or so at first.
The BME680 heats a tiny plate inside to smell VOCs, so in still air it reads slightly warm. Good airflow shrinks the effect; SensorMap's charts will show you a rock-steady offset you can mentally subtract — or compare against the nearest airport station on the map.

First 24–48 h of VOC:
the gas sensor "burns in" and drifts while it finds its baseline. Temperature/humidity/pressure are trustworthy immediately; judge air-quality trends after a day or two.

Making it permanent

Any small vented enclosure works — a food container with holes, a 3D-printed case, or the classic: a stack of plant-pot saucers as a DIY radiation shield. Route the USB cable in from below so water can't follow it. When you're happy, consider soldering the four wires (option C) so a bump can't unseat a jumper.

Troubleshooting — the complete "it doesn't work" table

SYMPTOM	FIX
`no port under Tools → Port`	It's the cable 9 times out of 10 — use a known **data** cable. Otherwise: unplug, hold **BOOT**, plug in while holding, release, look again.
`fatal error: Adafruit_BME680.h: No such file or directory`	A library from Step 4 isn't installed (this exact error names the missing one). Tools → Manage Libraries → install it — exact names matter.
`[init] BME680 NOT FOUND`	In order of likelihood: ① the two-number edit (Step 5) missed **or made in the wrong file** — the Arduino IDE uses the **config.h tab** (`src/config.h`), never the `include/` copy; SDA must be **22**, SCL **23**; ② green/yellow swapped — swap them at the sensor end; ③ a cold solder joint — reheat the six sensor pins; ④ something is touching **SDO** — it must be empty (or tied to 3V3, never GND). Truly stuck? Rare boards are hard-set to address 0x76: in `src/main.cpp` find `g_bme680.begin()` and change it to `g_bme680.begin(0x76)`.
`[wifi] ...retrying / never connects`	The C6 is **2.4 GHz only** — it cannot see 5 GHz-only networks. Check the password character by character (it's in the Manage panel on the site — fix it there and re-download). Corporate/hotel WiFi with login pages won't work; a phone hotspot is a great test.
`[post] HTTP 401`	The device key in your zip no longer matches (usually after the key was regenerated on the site). Re-download the zip from Manage → flash again.
`Upload fails mid-way / times out`	Hold **BOOT** as the IDE prints "Connecting…", release when uploading starts. USB hubs can also cause this — plug straight into the computer.
`temperature reads a bit high`	Normal (self-heating + snug enclosures) — see page 10. More airflow, more distance from the XIAO, out of the sun.
`voc jumps around at first`	Normal for 24–48 h while the gas sensor burns in. It's a relative index (lower = cleaner), not a lab instrument.
`was working, stopped after a move`	A jumper walked out of its column. Re-seat all four; check the 60-second list on page 6. (This is why option C exists.)

Still stuck?
Your device page → Manage shows the last time SensorMap heard from the board, and the serial monitor tells you which side is unhappy (sensor, WiFi, or posting). Mail **hello@sensormap.cloud** with the serial output and we'll figure it out together.

It's alive — now the fun part

ON SENSORMAP.CLOUD	WHAT IT DOES FOR THIS STATION
Alerts	Open your device → Alerts. "Warn me when temperature drops below 32 °F", delivered in-app, by email, by push to your phone, or to any webhook. Add an **offline watch** too — SensorMap emails you if the station goes quiet (power cut, WiFi down).
Phone notifications	Profile → enable notifications (iPhone: Share → "Add to Home Screen" first, then enable from the installed app). No app store involved.
Home Assistant	Manage → Home Assistant gives you copy-paste YAML — your BME680 readings become HA entities, and you can automate on any public station too (the airport's wind, the city's AQI…).
History & CSV	Every chart exports. Your data is never locked in.
Developer API	Profile → API keys → free key, 10,000 reads/day. Docs at **sensormap.cloud/developers**.
Public or private	Flip anytime in Manage. Public stations join the community map (pin can sit down the street from your real address).

Ideas for station #2

The firmware wizard on the site already knows these — same four-wire I²C hookup, same flow you just learned: **SCD40** (true CO_2 — brilliant indoors), **BH1750** (light), **LTR390** (UV index), a **PIR** motion node, or a battery XIAO on the BAT pads. Mesh mode lets far-away nodes relay through one gateway — no WiFi at the node.

www.sensormap.cloud — free, no ads, community-run.
Questions, bugs, or a photo of your finished build (please!): **hello@sensormap.cloud**

BUILD SPEC — FOR REFERENCE	
Microcontroller	Seeed Studio XIAO ESP32-C6 · WiFi 2.4 GHz + BLE · USB-C · 21×17.8 mm
Sensor	Bosch BME680 on ACEIRMC breakout · I²C address 0x77 · powered at 3.3 V
Wiring	3V3→VCC · GND→GND · D4/GPIO22→SDA · D5/GPIO23→SCL · SDO+CS unconnected
Firmware	SensorMap wizard zip (board preset ESP32-C6, sensor BME680) + I²C pins set to 22/23
Arduino setup	esp32 core ≥3.0 · board XIAO_ESP32C6 · Adafruit BME680 + Unified Sensor + ArduinoJson
Reports	temperature_c · humidity_pct · pressure_hpa · voc_index (+ computed dew point)
Interval / power	every 15 s · <0.5 W from any USB-C charger