Anchorline SOLO bridge: user manual

Firmware 1.0.0. Last updated: 9 November 2026

The Anchorline SOLO bridge is a small board that sits between a Divesoft SOLO gas analyzer and the Anchorline app. It reads the SOLO over USB and passes the readings to your phone or Mac over Bluetooth. With it, you can analyse a cylinder with the phone in your pocket: press OK on the SOLO, and the app saves the analysis and, if you want, prints the cylinder label.

This manual is for divers and fill-station people. You don't need to be a developer to set it up.

What the bridge does

- **Live readings over Bluetooth.** The app's Analyse tab shows the SOLO's O_2, He, temperature and pressure through the bridge. This works on iPhone and iPad too, which can't talk to the SOLO over USB.
- **Log on OK.** With "Log on OK" running in the app, pressing OK on the SOLO (which freezes its display) saves the analysis in the app's analysis log. With a label printer set up, it also prints the label.
- **Nothing gets lost.** The bridge keeps each OK press in its memory until the app confirms it saved it. If the phone is out of range, the app is closed, or the bridge is switched off, the presses wait on the bridge and the app collects them the next time it connects.
- **A status screen.** The bridge's 240×240 screen shows what is going on, in big text you can read at arm's length: settling, stable, saved, printed, or what went wrong.

The bridge never judges the gas itself. Mix names, stability and mismatch checks are all done by the app.

What you need

Part	Notes
Divesoft SOLO gas analyzer	In the SOLO's **Menu** → **Preferences**, set **"Send data to USB"** above zero and turn **printer mode off**. Without this the SOLO sends nothing over USB.
Espressif ESP32-S3-USB-OTG board	Version V1.1. Copies sold on AliExpress and similar shops work (the developer uses one). The plugs on a copy can differ slightly from Espressif's original; check the labels printed on your board.
USB power bank	Powers the board through its **DEV** plug or its **DEBUG** port. See "Power" below.
USB cable for the SOLO	USB-A male to USB-C male, a data cable (not a charge-only cable). It goes from the board's USB-A host socket to the SOLO.
The Anchorline app	On iPhone, iPad, Android or a Mac, in a version that has **Settings** → **Gas** → **Analyzer bridge**.
Phomemo M110 label printer (optional)	With **50×30 mm gap labels**. Only needed if you want labels printed.
A computer with Chrome or Edge (once)	To install the firmware on the board. Windows, macOS or Linux.

The board's plugs and buttons

Name on the board	What it is	What you use it for
USB host (USB-A socket)	The board's USB host port	The SOLO plugs in here
DEV (USB-A plug)	The chip's own USB	Power from a power bank or charger; installing firmware

Name on the board	What it is	What you use it for
DEBUG (USB-C on the developer's V1.1 copy; it may differ on other copies)	A separate USB-to-serial chip (WCH CH342)	Power; installing firmware with esptool as a fallback; the developer's log
OK	Also the BOOT button	Holding it while pressing RESET puts the board in download mode
MENU		Shows the info screen for 5 seconds
RESET		Restarts the board
UP+ / DW–		Not used by the firmware

Power

- **The board** runs from a USB power bank or charger on its DEV plug or DEBUG port. The DEV plug is a USB-A plug, so it goes straight into the USB-A socket of a power bank. If your power bank only has USB-C sockets, use the DEBUG port with a USB-C cable, or a small USB-A-socket-to-USB-C-plug adapter on the DEV plug.
- **The SOLO runs on its own battery.** By default the bridge does **not** power or charge the SOLO: the 5 V on the host socket is off. The SOLO works fine this way.
- **"Power the analyzer from the bridge"** (in the app, Settings → Gas → Analyzer bridge) turns the host socket's 5 V on, current-limited to 500 mA. It is **off by default**. When it is on, the bridge also charges the SOLO. Leave it off unless you know you want it. When the board runs from its DEBUG port, the host socket never gets 5 V, whatever the switch says.
- **Optional battery.** The board has solder pads for a single Li-ion cell (for example an 18650) and an on-board charger that charges from the DEV plug (a red LED shows while charging). The board has no documented low-voltage cut-off, so use **protected cells only**, and switch the board's battery switch **off while charging**. A power bank is the simpler choice.

First-time setup

1. Install the firmware on the board

A new board doesn't have the Anchorline firmware yet. Install it once from a computer; see "Installing and updating the firmware" below. The quickest way is the **Install firmware** button on the Anchorline website's download page (https://anchorline.eu/download#solo-bridge). After installing, press RESET on the board: the screen shows `WAITING FOR PHONE`.

2. Set up the SOLO

On the SOLO, open **Menu** → **Preferences**, set **"Send data to USB"** above zero, and make sure **printer mode is off**. Plug the SOLO into the board's USB host socket with the USB-A-to-USB-C cable, and switch the SOLO on.

3. Pair the bridge in the app

1. Power the board (power bank on DEV or DEBUG). Keep it near the phone.
2. In the app, open **Settings** → **Gas** → **Analyzer bridge** and tap **Find bridge**.
3. Pick **Anchorline Bridge XXXX** from the list. The four characters are the end of the board's ID; the info screen (press MENU) shows the full ID.
4. The section now says the bridge is set up on this device. **Forget bridge** removes it again.

Each device (phone, tablet, Mac) is paired on its own. The bridge talks to one device at a time.

4. Settings that matter

Where	Setting	What it does
Settings → Gas → Analyzer bridge	**Power the analyzer from the bridge**	Off (the default): the SOLO runs on its own battery. On: the bridge supplies 5 V, which also charges the SOLO.
Settings → General → You	**Your initials**	Fills "Analysed by" and goes on cylinder labels. Printing on OK needs a name in "Analysed by".
Settings → Gas → Label printer	**Find printer**, **Print test label**	Pairs the Phomemo M110. Only needed for printing.
Settings → Gas → Analyzer	**MOD rows at ppO_2**, **Fill pressure starts at**	As for any analysis.

Using the bridge

Live readings

1. Switch on the board and the SOLO, with the SOLO plugged into the host socket.
2. In the app, open **Gas** → **Analyse**, choose the source **Bridge**, and tap **Connect**.
3. The connection strip says **Bridge · SOLO connected** (or **Bridge · no SOLO**), with the bridge's battery voltage or "USB powered", and "5 V on" when the bridge powers the SOLO.
4. Flow gas over the SOLO's sensor as usual. The app shows the reading and whether it has settled.

Log on OK

1. Connect as above.
2. In the **Save analysis** section, fill in what applies: the desired mix, the gas role, "Analysed by", the cylinder, fill pressure and notes. These are the setup for the next OK press.
3. Tap **Log on OK** at the right of the section heading. The heading changes to **● Logging on OK**, and the bar at the bottom shows the status with **Stop**. Log on OK is remembered on the device and starts again on the next connect, until you tap **Stop**.
4. With a printer set up, the row **Print a label on each OK** decides whether each OK press prints.
5. You can now lock the phone and put it in your pocket. Analyse the cylinder and press **OK** on the SOLO when you are happy with the reading. The SOLO freezes its display, the bridge shows `HELD – SAVING...`, and a moment later the result, for example `PRINTED EAN32`.
6. Press OK again on the SOLO to resume measuring for the next cylinder.

What carries over between cylinders: "Analysed by", the desired mix you picked, the gas role, "For" and "Print a label on each OK" stay until you change them. The cylinder, fill pressure and notes are for one cylinder only and are cleared after a complete save.

When a label is printed. A press prints only when all of these hold: printing is on and a printer is set up; "Analysed by" is filled in; the app was connected when you pressed OK; the SOLO had a usable reading; and, if you set a desired mix, the reading matches it closely enough. Otherwise the analysis is still saved, marked **Needs sign-off** in the log, and you sign it off and print it from there.

Room air is never logged by OK. A press while the SOLO reads 20.5–21.5 % O_2 and under 1 % He saves nothing; the bridge shows `AIR – NOT LOGGED`. Save air cylinders by hand with the **Save** button on the Analyse tab. This assumes a SOLO calibrated on air: a SOLO that reads air as, say, 20.2 % will save it as "Hypoxic 20". Calibrate before relying on it.

Records waiting while the phone is away. If the phone isn't connected when you press OK (out of range, app closed, Log on OK stopped), the bridge keeps the press in its memory and shows `n waiting for phone`. The app collects waiting presses the next time it connects to the bridge. Because nobody can tell which setup applied at that moment, these are saved **without** the setup, marked **Needs sign-off**, and never printed automatically. The bridge holds up to 50 waiting presses; when it is full it shows `FULL – CONNECT PHONE` and records nothing new until the phone collects them. It never overwrites one.

A quick double press (under about 2 seconds) records nothing. Pressing OK, OK, OK on the same gas makes several entries: the app never drops one on its own, because a missing analysis is worse than a duplicate. Delete the extra entries in the log.

The status screen

The top line shows `SOLO ✓/✗ PHONE ✓/✗`, then the battery voltage (or `USB` when there is no battery), and `5V` when the bridge powers the SOLO. Under it, `3 waiting for phone` when presses are waiting.

The main text shows one thing at a time. When several apply, the one highest in this table wins.

Screen text	Colour	Meaning	What to do
`BATTERY EMPTY`	red	The battery is below 3.3 V	Charge it or use a power bank
`WAITING FOR PHONE`	white	No phone has connected since the bridge was switched on	Connect from the app (Gas → Analyse → Bridge → Connect)
`CONNECTION LOST`	red	The phone was connected and the link dropped	Bring the phone closer, or reconnect in the app. OK presses are kept meanwhile
`PHONE NOT RESPONDING`	red	Connected, but the app has sent nothing for 10 s while the SOLO is streaming	Open the app; check it is still on the Analyse tab or in a Log on OK session
`NO SOLO`	amber	No SOLO on the host socket	Check the cable and that the SOLO is on
`SOLO: TURN ON SEND DATA TO USB`	amber	The SOLO is plugged in but has sent nothing	On the SOLO: Menu → Preferences → "Send data to USB" above zero, printer mode off
`HELD – SAVING...`	amber	An OK press was recorded and the bridge is waiting for the app's answer	Wait a moment
`FULL – CONNECT PHONE`	amber	50 presses are waiting; no new ones are recorded	Connect the app so it can collect them
`BATTERY LOW`	amber	The battery is below 3.5 V	Charge soon
`PRINTED EAN32`	green	Saved and the label printed (the mix name varies)	Stick the label on
`SAVED – EAN32`	green	Saved complete, no label wanted	—
`SAVED – SIGN OFF LATER`	green	Saved, but it needs sign-off in the log (for example a press collected late)	Sign it off in Gas → Log
`SAVED – NO NAME SET`	green	Saved, but "Analysed by" was empty, so no label	Fill in "Analysed by"; sign off in the log
`SAVED – ALREADY SAVED`	green	The app already had this press (sent twice after a dropped link)	—
`CHECK PHONE: MISMATCH`	amber	The reading doesn't match the desired mix; saved, not printed	Look at the phone; check the cylinder
`CHECK PHONE: PRINT FAILED` / `NO PRINTER`	amber	Saved, but the label didn't print	Check the printer; print from the log after sign-off

Screen text	Colour	Meaning	What to do
`AIR – NOT LOGGED`	white	The reading was room air; nothing saved	Save air by hand if you want it logged
`NO READING`	red	The SOLO had no usable reading; nothing saved	Check the sensor and the flow
`SAVE FAILED` / `LOG FAILED`	red	The app couldn't save the press	Look at the phone; the analysis may need to be repeated
`SETTLING... TX21/35`	amber	Live: the reading is still changing	Wait
`STABLE TX21/35`	green	Live: the reading has settled	Press OK when ready
`SENSOR ERROR`	red	Live: the SOLO reports a sensor problem	See the SOLO's manual
`AIR`	white	Live: the reading is room air	—
`READY`	white	Connected and nothing else to report	—

Results (`PRINTED`, `SAVED`, `CHECK PHONE`, …) stay on screen for 10 seconds, then the live state returns.

The LEDs

LED	Meaning
Green, short blink	The app confirmed it received an OK press
Yellow, on	Presses are waiting for the phone, the memory is full, the connection is lost, or the battery is low or empty
Red (on the board's charger)	The battery is charging

The info screen

Press **MENU** to show the info screen for 5 seconds:

- `ID`: the bridge's ID (its factory MAC address).
- `FW`: the firmware version.
- `BATT`: the battery voltage in mV (near 0 without a battery).
- `HOST VBUS`: the voltage on the SOLO's socket in mV. With the 5 V switch off this can still read about 5 V: the SOLO seems to put its own voltage on the line.
- `5V SRC`: where the host socket's 5 V comes from: `off`, `dev` (the DEV plug) or `battery`.
- `EPOCH`: an internal number that changes when the bridge's memory is erased.

Installing and updating the firmware

The firmware is one file, `anchorline-bridge-<version>.bin`, written at address 0x0. The download page lists its size and SHA-256 checksum.

Before updating: connect the bridge to the app once, so presses waiting on the bridge are collected (no `waiting for phone` line on the screen). Installing resets the bridge's own settings to the defaults (5 V switch off); the app sends its settings again when it next connects. If the installer offers to **erase** the device, erasing also deletes any presses still waiting.

Download mode

To install firmware the board must be in **download mode**:

1. Hold **OK** (also labelled BOOT).
2. Press and release **RESET**.
3. Release **OK**.

The screen goes dark or stays frozen; that is normal. After installing, press **RESET** to start the new firmware.

Which plug to use

- **DEV plug (recommended).** The chip's own USB. In download mode it shows up on the computer as a USB serial device with no driver needed on Windows 10/11, macOS or Linux. The DEV plug is a USB-A plug: on a computer with only USB-C sockets, use a USB-A-socket-to-USB-C adapter. When the Anchorline firmware runs, the board switches the chip's USB over to the SOLO's socket, so outside download mode the computer won't see the board on the DEV plug. That is expected.
- **DEBUG port (fallback).** A WCH CH342 USB-to-serial chip. It can need a driver (see per system below). It shows up as two serial ports; use the first one, or try the other if the first doesn't connect. esptool can usually put the board in download mode by itself over this port; if it can't, use the button sequence.

Recommended: the website's "Install firmware" button

This uses ESP Web Tools in the browser. It needs **Chrome or Edge on a desktop computer** (Windows, macOS or Linux). Safari, Firefox and phones can't do it.

1. Open the download page (https://anchorline.eu/download#solo-bridge) in Chrome or Edge.
2. Put the board in download mode (above) and connect it with the **DEV** plug.
3. Click **Install firmware**, choose the board's port in the browser's list, and confirm. On a first install, choosing to erase the device is fine.
4. Wait until the installer says it is done. Then press **RESET** on the board.

Fallback: esptool from the command line

esptool is Espressif's flashing tool. It needs Python 3.

1. Download `anchorline-bridge-<version>.bin` from the download page and check its SHA-256 (commands below) against the one on the page.
2. Install esptool: `pip install esptool` (on Windows `py -m pip install esptool`).
3. Put the board in download mode and connect it (DEV plug, or the DEBUG port).
4. Find the port name (per system below).
5. Run, with your port and the file name:

   ```
   python -m esptool --chip esp32s3 --port <PORT> write_flash 0x0 anchorline-bridge-1.0.0.bin
   ```

   Newer esptool versions spell the command `write-flash`; both work.
6. Press **RESET** on the board.

Windows

- **Drivers.** DEV plug: none; Windows 10 and 11 use their built-in USB serial driver. DEBUG port: Windows usually installs the WCH driver by itself through Windows Update. If the port doesn't appear, install WCH's CH343 serial driver (CH343SER) from wch-ic.com.
- **Finding the port.** Open **Device Manager** → **Ports (COM & LPT)**, plug the board in, and look for the new entry: "USB Serial Device (COM5)" on the DEV plug, or a "USB-SERIAL CH342" entry on the DEBUG port. The port is the `COM` name.
- **Checksum.** In PowerShell: `Get-FileHash anchorline-bridge-1.0.0.bin -Algorithm SHA256`.
- **Command.**

```
py -m esptool --chip esp32s3 --port COM5 write_flash 0x0 anchorline-bridge-1.0.0.bin
```

macOS

- **Drivers.** DEV plug: none. DEBUG port: macOS has no built-in driver for the CH342. Install WCH's **CH34xVCPDriver** (Mac App Store, or wch-ic.com), allow it in **System Settings** → **Privacy & Security** (or General → Login Items & Extensions → Driver Extensions on recent macOS), then unplug and replug the board.
- **Finding the port.** In Terminal, run `ls /dev/cu.*` before and after plugging in. The DEV plug shows up as `/dev/cu.usbmodem…`; the DEBUG port with the WCH driver as `/dev/cu.wchusbserial…`.
- **Checksum.** `shasum -a 256 anchorline-bridge-1.0.0.bin`
- **Command.** Use `python3` if `python` is not found:

```
python3 -m esptool --chip esp32s3 --port /dev/cu.usbmodem1101 write_flash 0x0 anchorline-bri
dge-1.0.0.bin
```

Linux

- **Drivers.** DEV plug: none (the kernel's `cdc_acm` driver). DEBUG port: recent kernels normally recognise the CH342 by themselves; if no port appears, WCH publishes a Linux driver on wch-ic.com.
- **Permission.** Your user needs access to serial ports: add yourself to the `dialout` group (Debian, Ubuntu, Fedora) or `uucp` (Arch), with `sudo usermod -aG dialout $USER`, then log out and back in. This also applies to the browser route. If you run Chromium as a Snap, also run `sudo snap connect chromium:raw-usb`.
- **Finding the port.** Run `ls /dev/ttyACM* /dev/ttyUSB*` before and after plugging in, or watch `sudo dmesg -w`. The DEV plug is usually `/dev/ttyACM0`; the DEBUG port `/dev/ttyACM*` or `/dev/ttyUSB*`. If the port vanishes or flashing stops at random, ModemManager may be probing it: stop it for the duration (`sudo systemctl stop ModemManager`).
- **Checksum.** `sha256sum anchorline-bridge-1.0.0.bin`
- **Command.**

```
python3 -m esptool --chip esp32s3 --port /dev/ttyACM0 write_flash 0x0 anchorline-bridge-1.0.
0.bin
```

Troubleshooting

Problem	Try this
The screen says `SOLO: TURN ON SEND DATA TO USB`	On the SOLO: Menu → Preferences → "Send data to USB" above zero, printer mode off
`NO SOLO` with the SOLO plugged in	Use a data cable, not a charge-only cable; plug it into the USB-A host socket; switch the SOLO on
"Find bridge" finds nothing	Switch the bridge on and keep it close. Check Bluetooth is allowed for Anchorline in the phone's settings. The bridge talks to one device at a time: disconnect it from any other phone or Mac
The bridge source is missing in the app	The bridge works on iPhone, iPad, Android and macOS, in a version of Anchorline that has Settings → Gas → Analyzer bridge
OK presses save but never print	Check "Analysed by" is filled in, "Print a label on each OK" is on, and the printer is on and paired. Presses collected late never print automatically: sign them off in the log
Room air saves as "Hypoxic 20"	The SOLO reads air low. Calibrate it on air
`CONNECTION LOST` all the time	Bring the phone closer. Metal cylinders and bodies block Bluetooth. Presses are kept on the bridge meanwhile
The browser installer doesn't list a port	Put the board in download mode again (hold OK, press RESET, release OK) and use the DEV plug. Use Chrome or Edge on a desktop. On Linux, check the `dialout` group
esptool says "Failed to connect"	Download mode again; try the other plug or the other DEBUG port; on macOS with the DEBUG port, check the WCH driver is allowed
After installing, the board seems dead	Press RESET. The screen should show `WAITING FOR PHONE`
The info screen shows `HOST VBUS` about 5000 mV with the switch off	Normal: the SOLO seems to supply that voltage itself

Safety

- **The bridge and the app are an aid, not a replacement for your analysis procedure.** Read the SOLO's own display, verify every mix with your normal procedures, and check every label against the analyzer before you breathe from a cylinder. The person who dives the gas is responsible for knowing what is in it.
- **The app's match checks are for orientation only.** A label printed from the pocket still needs a look.
- **Calibrate the SOLO** as Divesoft describes. The room-air rule and the mix names depend on it.
- **Batteries:** use protected Li-ion cells only, as the board has no documented low-voltage cut-off. Switch the battery switch off while charging. Don't leave a charging board unattended. A power bank avoids all of this.
- **The 5 V switch:** keep "Power the analyzer from the bridge" off unless you need it. When on, the bridge charges the SOLO from its battery or power source.
- **Keep the board dry.** It is a bare circuit board; keep it away from water and cylinder valves being opened. A small case helps.