



User Manual

Openloop Connect Companion App



Haudi Crypto, Inc.

Version 0.9.3
2026-08-08

Contents

1. Introduction
2. Download and Installation
3. Main Screen
4. Registering a Device
5. WalletConnect Integration
6. Signing Request Screen
7. Settings Screen
8. About Screen
9. Credential Manager (Passkeys over BLE)
10. Safari Web Extension
11. Openloop API Server
12. PKCS#11 Dynamic Library
13. Troubleshooting

1. Introduction

1.1 What is Openloop Connect?

Openloop Connect is the companion application for the Openloop hardware wallet. It provides the following five features that let apps and operating systems talk to your Openloop.

#	Feature	Description	Supported platforms
1	Firmware update	Update the Openloop firmware to the latest version over BLE (mobile) or USB (desktop)	Mobile / Desktop
2	WalletConnect relay	Use your Openloop from any WalletConnect-compatible dApp	Mobile / Desktop
3	Openloop API server	A local WebSocket server that lets web apps and native apps use your Openloop	Desktop / Android

#	Feature	Description	Supported platforms
4	Safari Web Extension	Container for the extension that lets Safari use your Openloop	iOS
5	PKCS#11 library	Bundles the PKCS#11 dynamic library used for PIV signing	Desktop

1.2 Feature Support by Platform

Feature	iOS	Android	macOS	Windows	Linux
Firmware update	✓	✓	✓	✓	✓
WalletConnect relay	✓	✓	✓	✓	✓
Openloop API server	—	✓	✓	✓	✓
Safari Web Extension	✓ ※	—	—	—	—
PKCS#11 library	—	—	✓	✓	✓

※ The Safari Web Extension requires **iOS 26.2 or later** (see [Chapter 10](#)).

1.3 Connection Methods

Platform	Connection	Used for
Mobile (iOS / Android)	BLE (Bluetooth Low Energy)	Wallet features, WalletConnect, firmware updates
Desktop (macOS / Windows / Linux)	USB HID	Wallet features, WalletConnect, firmware updates, PIV/PKCS#11, API server

1.4 A Note on Air Gap Mode

Openloop supports Air Gap mode, in which crypto-asset transactions are signed without any communication link. **Openloop Connect is not used in Air Gap mode** — everything happens on the Openloop device itself. See the [Openloop User Manual](#) for details.

2. Download and Installation

2.1 System Requirements

Platform	Required version	Notes
iOS	iOS 15.1 or later	Passkey features require iOS 17 or later; the Safari extension requires iOS 26.2 or later
Android	Android 7.0 or later	Passkey features require Android 14 or later
macOS	macOS 12 (Monterey) or later	Apple Silicon and Intel both supported
Windows	Windows 10 / 11 (64-bit)	—
Linux	Debian / Ubuntu family (64-bit)	DEB package

2.2 Mobile (iOS / Android)

OS	Where to get it
iOS	Search the App Store for “Openloop Connect” and install
Android	Search Google Play for “Openloop Connect” and install

The app is free.

2.3 Desktop (macOS / Windows / Linux)

Go to the [Openloop downloads page](#) and download the file for your OS.

The downloads page lists the following files.

File	OS	Description
DMG (arm64)	macOS (Apple Silicon)	For Macs with M1/M2/M3/M4
DMG (x64)	macOS (Intel)	For Intel-based Macs
Microsoft Store	Windows 10/11 (x64)	MSIX build with automatic updates (recommended)
EXE	Windows (x64)	Windows installer
DEB	Linux (x64)	Package for Debian/Ubuntu

Installing on macOS

1. Download the DMG file
2. Double-click the DMG to open it
3. Drag the “Openloop Connect” icon into the “Applications” folder
4. Launch it from Launchpad or `/Applications`

Installing on Windows

Option 1: Microsoft Store (recommended)

1. Search the Microsoft Store for “Openloop Connect”, or click the “Get it from Microsoft Store” badge on the downloads page
2. Click “Get” to install
3. Launch it from the Start menu

The Microsoft Store build updates itself automatically.

Option 2: Installer (EXE)

1. Download the EXE file
2. Double-click the EXE to run the installer
3. Follow the on-screen instructions
4. Launch it from the Start menu

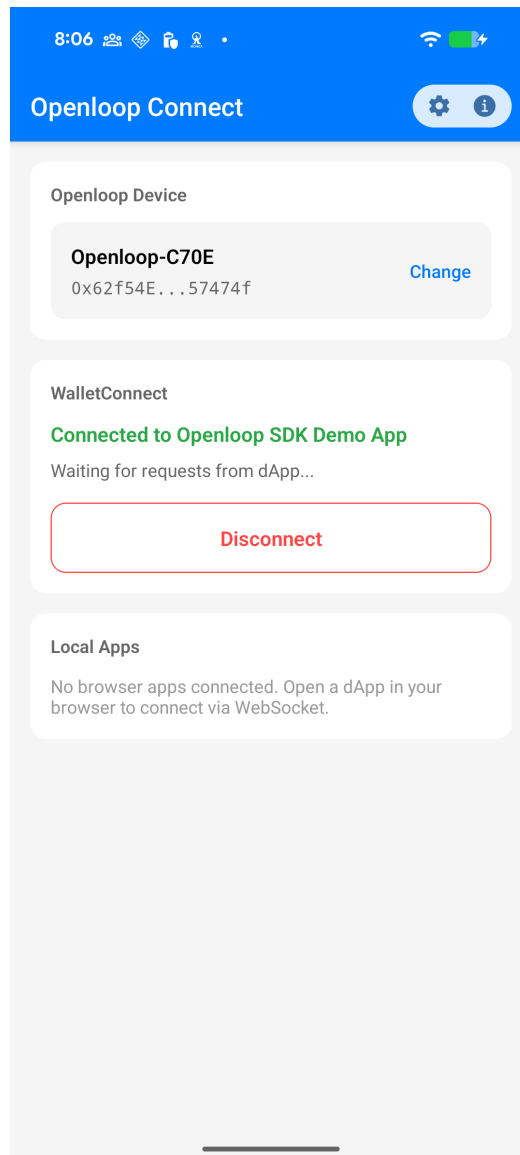
Linux (DEB)

1. Download the DEB file
2. Run in a terminal: `sudo dpkg -i openloop-connect-*.deb`
3. Launch it from the applications menu

 On Linux, connecting the Openloop over USB may require a udev rule. The DEB package includes a script that installs the udev rule for you.

3. Main Screen

This is the screen you see when you launch the app. It consists of a header bar and three sections.



3.1 Header Bar

The top of the screen (a blue bar on mobile) shows:

- The “**Openloop Connect**” title
- ⚙️ (**gear**) icon: opens the [Settings screen](#)
- ⓘ (**info**) icon: opens the [About screen](#)

3.2 Openloop Device Section

This section shows information about the connected Openloop device.

When a device is registered:

- **Device name:** for example “Openloop-C70E” — the last four characters are a device-specific identifier (the low two bytes of the BLE MAC address)

- **Address:** the Ethereum address. Mobile shows it abbreviated (for example `0x62f54E...57474f`); desktop shows it in full
- **“Change” link** (mobile only): opens the device selection modal so you can switch devices
- **“Disconnect” button:** a red text button shown only while BLE is connected. Tapping it drops the BLE connection to the Openloop

When no device is registered (mobile):

- An **“+ Add Wallet” button** (with the subtitle “Tap to scan for devices”) appears. Tapping it starts [device registration](#)

On desktop:

- **After disconnecting:** if you disconnect manually, the app shows “Disconnected” along with a **“Reconnect” button**, so you can re-establish the connection without unplugging the USB cable
- **When you connect a different device:** if you plug in an Openloop other than the previous one, a “Different device” alert appears and all WalletConnect sessions are disconnected (this prevents the app from answering a dApp with a different wallet’s address)
- **Footer:** the app version is shown at the bottom of the window (for example `Openloop Connect v0.9.3`)

Device Selection Modal (mobile only)

This modal appears when you tap the “Change” link.

- A list of registered devices (device name, address, and a radio button showing which is selected)
- A **“Delete”** button next to each device (tapping it opens a confirmation dialog)
- **“+ Add new device”** — starts [device registration](#)
- **“Close”** button

3.3 WalletConnect Section

This section shows the WalletConnect connection status.

When not connected:

- A text field for the WalletConnect URI
- **“Connect” button:** tap after entering a URI to connect to the dApp
- **QR code icon** (mobile only): shown at the right edge of the “WalletConnect” title. Tapping it opens the camera so you can scan a dApp’s QR code directly

When connected:

- **List of connections:** connected dApps are listed with a green dot. **On desktop you can connect to several dApps at once**, and each dApp name has its own **“Disconnect” button** (disconnecting one leaves the others connected). Mobile connects to a single dApp.
- **“Waiting for requests from the dApp…”:** the idle message
- **“Disconnect” button:** a red text button that ends every WalletConnect connection

 You normally do not need to enter a WalletConnect URI by hand. Selecting “Openloop Connect” in a dApp connects automatically via a deep link. See Chapter 5 for details.

 If the dApp disconnects while a signing confirmation is on screen, the confirmation closes automatically (a disconnected dApp can never receive a signature).

3.4 Local Apps Section

This section lists the applications currently connected to the local WebSocket server (`ws://127.0.0.1:21320`).

When nothing is connected:

- Shows “No browser app is connected. Open a dApp in your browser and connect over WebSocket.”

When connected:

- Each connected app’s origin (URL) is listed with a green dot
- Each app has a **“Disconnect” button** that disconnects that app individually

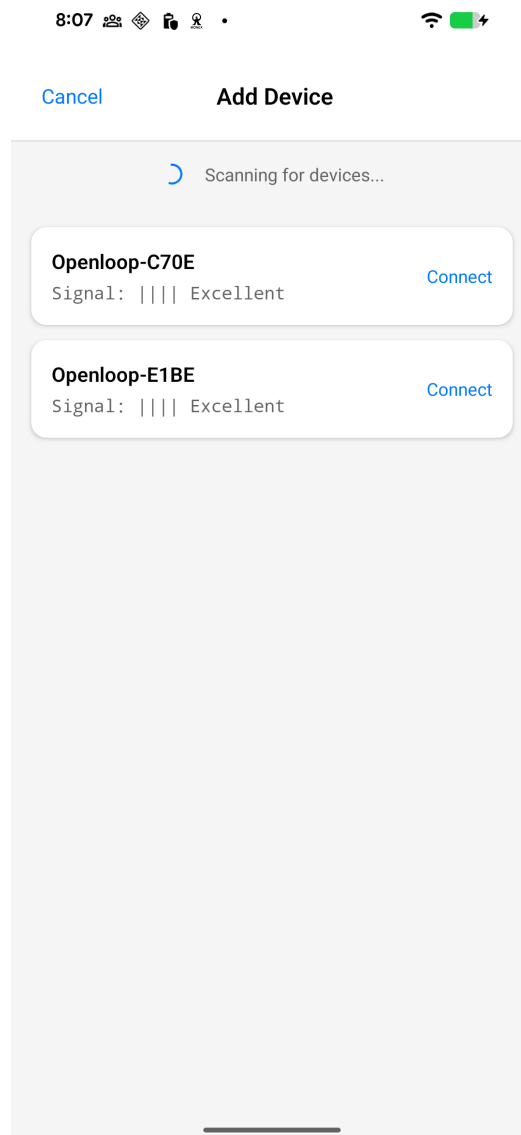
 On iOS the local WebSocket server does not work, because of OS restrictions. This section will always show as not connected. On iOS, use the Safari Web Extension instead.

4. Registering a Device

4.1 Mobile (BLE)

You can start device registration from the “+ Add Wallet” button on the main screen, “+ Add new device” in the device selection modal, or “+ Add new device” in the [Settings screen](#).

Device Scan Screen



A BLE scan starts automatically and lists nearby Openloop devices.

- **Device name:** for example “Openloop-C70E”
- **Signal strength:** shown in four levels — Excellent / Good / Fair / Weak — along with a bar (||||)
- **“Connect” link:** tap to connect to the device

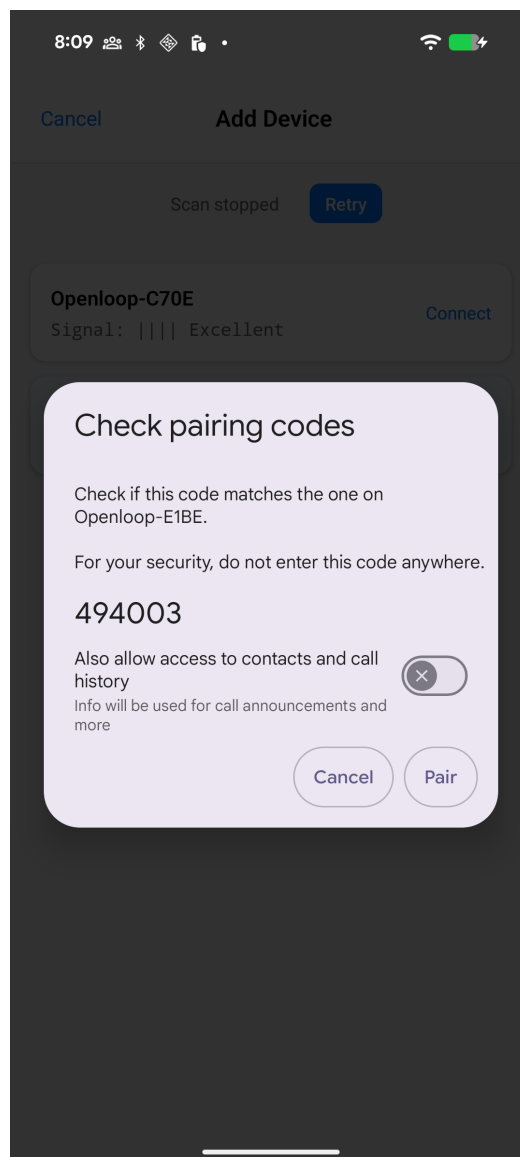
If no device is found, the screen shows “No devices found” and prompts you to check that the device is powered on. The “Retry” button restarts the scan.

The “Cancel” button at the top of the screen returns to the previous screen.

Connecting

Tapping “Connect” on a device shows a connection overlay.

1. “Connecting to device…” — establishing the BLE connection
2. “Getting address…” — reading the Ethereum address from the device
3. When your phone shows the BLE pairing dialog, tap “Pair”



Registration Complete

Once pairing finishes, the registration complete screen appears.



Device Added

Openloop-E1BE

0x09F883Eb...c8ea12bB

Your wallet has been registered successfully.

Done

- A green checkmark icon
- The message “**Device added**”
- The device name and Ethereum address

Tap “Done” to return to the main screen.

 Registering a device in Openloop Connect also performs the phone’s BLE pairing at the same time. There is no separate step.

If Pairing Fails

The most common problem is **stale pairing data left on your phone**. After a factory reset of the Openloop, or after unpairing and reconnecting, remnants of the old pairing can remain on the phone side.

How to fix it:

1. Open **Settings** → **Bluetooth** on your phone
2. Find “Openloop” in the list of paired devices
3. Tap the “i” or settings icon and choose **“Forget This Device”**
4. Register the device again in Openloop Connect

4.2 Desktop (USB)

On desktop, you connect the device with a USB cable. There is no registration or pairing step.

1. Connect the Openloop to your PC with a USB cable
2. Launch Openloop Connect
3. When device information appears in the Openloop device section, you are connected

▲ Charge-only USB cables will not work. Use a cable that supports data transfer.

5. WalletConnect Integration

5.1 Overview

WalletConnect is the global standard protocol for connecting wallets to dApps (decentralized applications) and Web3 platforms. With Openloop Connect running, you can use your Openloop from the many dApps that support WalletConnect.

5.2 Connecting to a dApp

Use any of the following methods to connect a dApp to your Openloop.

Method 1: Pick it from the dApp’s wallet list (recommended)

This is the easiest approach. You connect directly from a dApp on your phone (in a browser or as an app).

1. Launch Openloop Connect and confirm the device is connected
2. In the dApp (for example Uniswap or OpenSea), tap “Connect Wallet”
3. Choose **“Openloop Connect”** from the wallet list
4. Openloop Connect comes to the foreground automatically and shows the connection approval dialog

5. Review the request (dApp name, requested chains, and so on), **choose which account to expose for each currency (5.3)**, then tap **“Approve”**
6. Return to the dApp — you are connected

Once connected, the WalletConnect section on the main screen shows “Connected to: [dApp name]”.

 With this method, the dApp’s deep link (`openloop://` / `wc://`) or Universal Link (`https://www.crypto.haudi.jp/wc...` , WalletConnect Link Mode) handles the WalletConnect URI automatically.

Method 2: In-app QR code scanner

Handy when you want to connect a dApp running on a PC to Openloop Connect on your phone.

1. On the Openloop Connect main screen, tap the **QR code icon** at the right edge of the “WalletConnect” title
2. The camera opens (the first time, you will be asked for camera permission)
3. In the dApp on your PC, choose “Connect Wallet” → “WalletConnect” and display the QR code
4. Point the camera at the QR code; it is read automatically and the connection approval dialog appears
5. Tap **“Approve”**

Method 3: Enter the URI manually

A fallback for when the methods above are not available.

1. In the dApp, choose “Connect Wallet” → “WalletConnect”
2. Use “Copy to Clipboard” (or similar) below the QR code to copy the URI
3. Paste the URI into the text field in the WalletConnect section of the main screen
4. Tap the **“Connect”** button

5.3 Choosing Accounts When Connecting

Before you approve a connection, an **account selection screen** appears. For each dApp you can expose **one account per currency (chain)**.

- Only the currencies the dApp requested (Ethereum / Bitcoin / TRON / Solana, and so on) are shown.
- Each currency gets one **dropdown**. Closed, it shows the selected account name and abbreviated address; tap to open it and five accounts are listed **with their addresses**.
- **“Previous 5 / Next 5”** lets you reach account number 5 and beyond.

- Solana offers both conventions (**Phantom = 4-level** and **Solflare = 3-level**).
- Only the **address of the account you choose** is disclosed to the dApp — the others stay private.

When you are done, tap **“Approve”**. From then on, signing for that dApp uses the account you selected.

How the selection is remembered:

Platform	Scope
Desktop (USB)	Remembered per device , and preserved across disconnect/reconnect and app restarts. Connecting a different Openloop switches to that device’s own memory (account 0 the first time).
Mobile (BLE)	Remembered per device , and kept until you delete that wallet’s registration. Reconnecting to the same dApp preselects your previous account.

5.4 How Signing Works

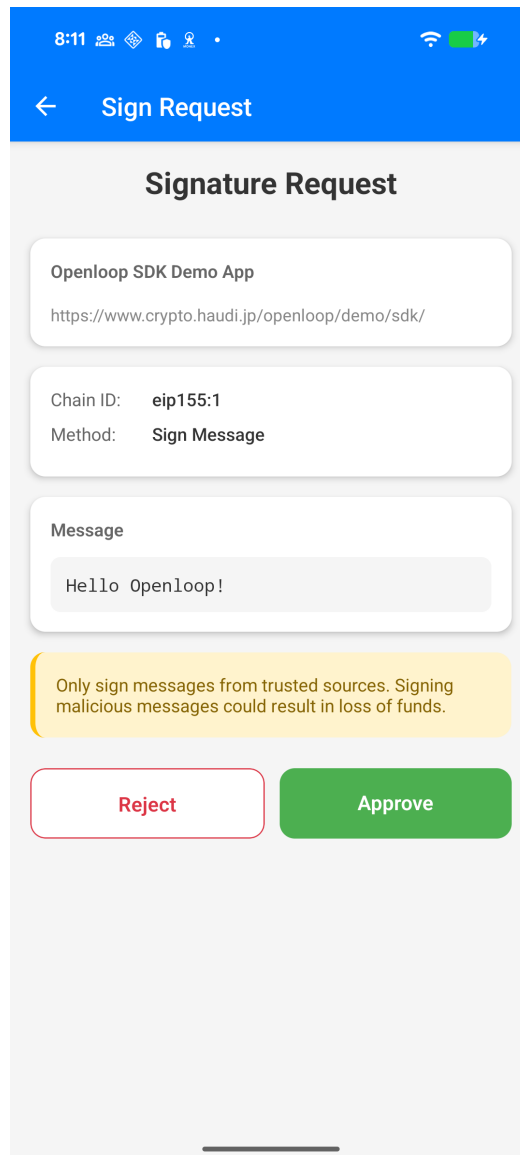
When you perform a transaction in a dApp, the signing request is delivered to Openloop Connect automatically and the [signing request screen](#) appears. Signing uses the account you selected when connecting.

5.5 Disconnecting

Tap the **“Disconnect”** button in the WalletConnect section of the main screen to end the connection to the dApp.

6. Signing Request Screen

When a signing request arrives from a dApp, this screen appears as a modal.



6.1 Screen Layout

dApp information

- **dApp name:** the name of the requesting dApp (bold)
- **URL:** the dApp's URL (small grey text)

Request details

- **Chain ID:** the target blockchain network (for example `bip122:0000000000019d6689c085ae165831e93`)
- **Method:** the type of signature, shown as a readable name (for example “Bitcoin Message Signing”, “Send Transaction”, “Sign Typed Data (EIP-712)”)

Message

The message or transaction data to be signed, shown as scrollable monospace text. Hex-encoded data is decoded to UTF-8 text.

Warning message

On a yellow background: “Only sign messages from sources you trust. Signing a malicious message can result in loss of funds.”

Action buttons

- **“Reject” button** (left, white with a red border): reject the request
- **“Approve” button** (right, green): approve and ask the Openloop device to sign

6.2 After Approving

When you tap “Approve”:

1. **“Please confirm on your device”** — the signing details appear on the Openloop’s screen
2. Touch the approve button on the device
3. The signature is produced inside the device’s secure element
4. Signing complete — a green checkmark with **“Request approved”** and “The transaction was signed on your Openloop device.” appears (it closes automatically after 3 seconds)

6.3 Rejection and Expiry

- **Rejecting in the app:** tapping “Reject” notifies the dApp and closes the screen immediately
- **Rejecting on the device:** if you reject on the Openloop’s screen, a red X appears with **“Signing rejected”** and “Signing was rejected on the device.” (it closes automatically after 3 seconds)
- **Expiry:** when a request expires, **“Request expired”** and “The dApp request timed out. Please try again from the dApp.” appear and the screen closes automatically

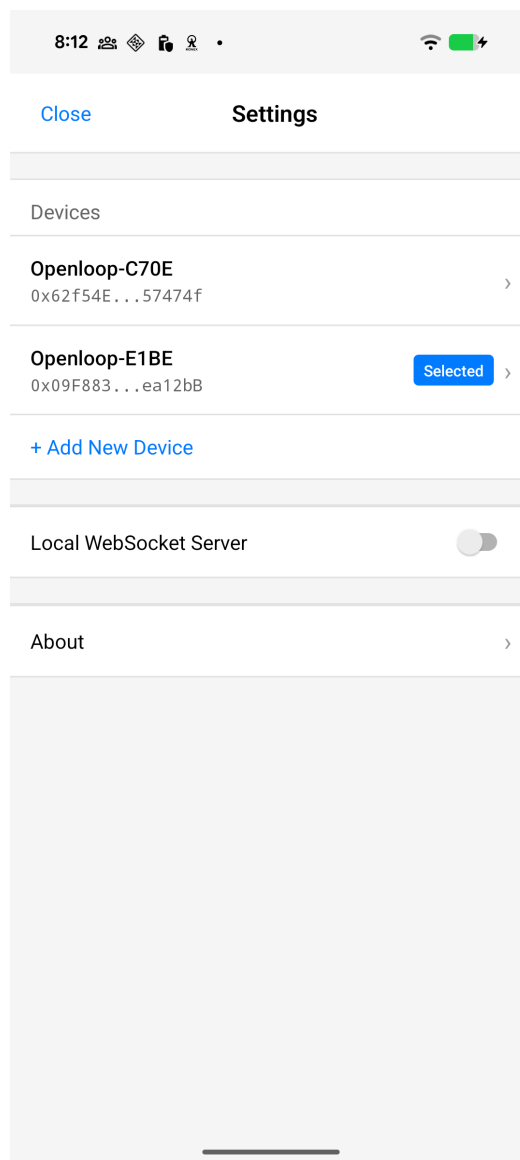
| 7. Settings Screen

Tap or click the  (**gear**) icon in the header bar to open it.

The layout differs by platform.

Platform	Sections
Mobile (iOS / Android)	Devices / Local WebSocket Server (Android only) / About
Desktop (macOS / Windows / Linux)	Firmware / Local WebSocket Server / PKCS#11 Cryptographic Module

On mobile, the “Close” button at the top returns you to the main screen; on desktop, use the “x” button at the top right (both are disabled during a firmware update).



7.1 Device List (mobile only)

The “**Devices**” section lists your registered Openloop devices. Each row shows:

- **Device name:** for example “Openloop-E1BE”

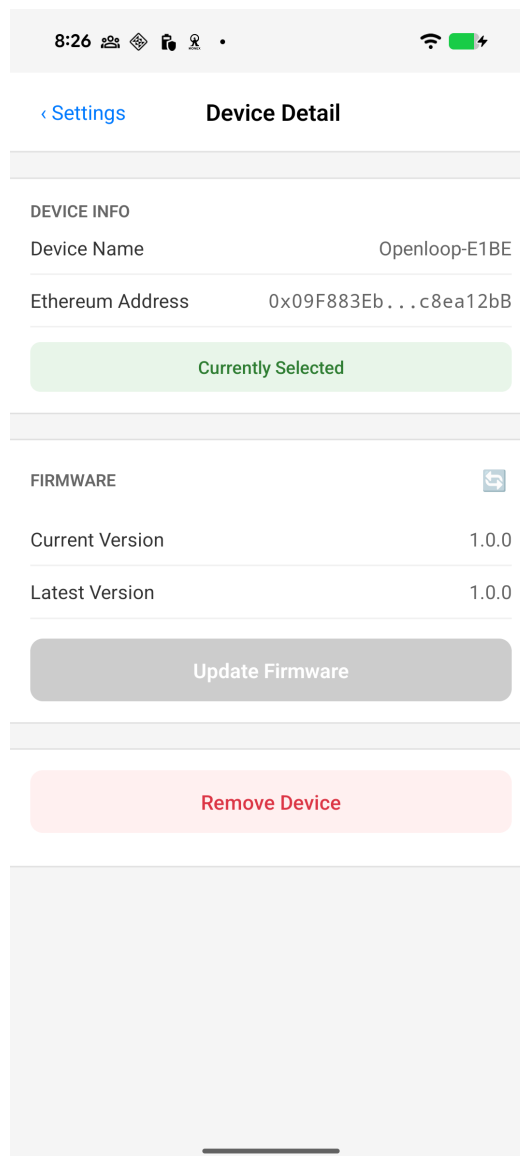
- **Address:** the Ethereum address (abbreviated)
- **“Selected” badge** (mobile only): a blue badge on the device currently in use on the main screen
- **“>” arrow:** opens the [device detail screen](#)

Tapping **+ Add new device** (mobile only) starts [device registration](#).

Tapping **About** opens the [About screen](#).

7.2 Device Detail Screen

Opens when you tap a device in the settings screen.



Device information section

- **Device name:** the device's name (for example "Openloop-E1BE")
- **Ethereum address:** first 10 and last 8 characters

- **“Currently selected” badge** (mobile only): shown in green only when this device is the one selected on the main screen

Firmware section

- **Check for updates button** (firmware update screen

Recovery firmware

The recovery firmware restores the device if the main firmware becomes corrupted. A **“Recovery Firmware”** section appears below the main firmware only when an update is needed. It shows the same items as the main firmware (current version, latest version, update button).

Actions section

- **“Switch to this device” button** (mobile only): shown only when this device is not the one selected on the main screen. Tapping it starts a BLE connection and switches the main screen to this device
- **“Delete device” button**: a red button. Tapping it opens a confirmation dialog; confirming removes this device’s registration from Openloop Connect

▲ Deleting a device does not remove the Bluetooth pairing from your phone. If you hit pairing problems when re-registering, see [“If Pairing Fails”](#) in section 4.1.

7.3 Firmware (desktop)

On desktop, a **“Firmware”** section appears at the top of the settings screen. There is no device list or device detail screen as on mobile — you update the firmware of the USB-connected device directly from this section.

- **Check for updates button** (

19

While updating:

A progress bar and the text below it change as the update proceeds.

1. “Downloading firmware…”
2. “Preparing…”
3. “Waiting for device…” — approve the update on the Openloop’s screen
4. “Sending to device… N%”
5. “Verifying firmware…”

The **“Cancel”** button aborts the update, though this is not recommended. On success, **“Update complete!”** appears. On failure, the error details and a **“Retry”** button are shown.

Recovery firmware

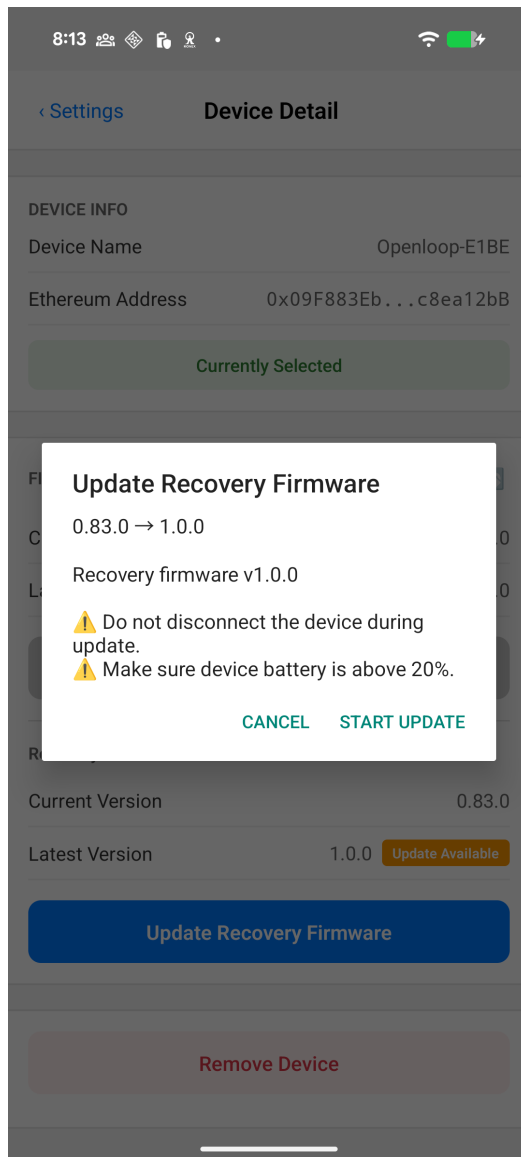
A **“Recovery”** section appears below the main firmware only when the recovery firmware has an update available (the main firmware’s heading then becomes **“Main”**). It shows the current version, the latest version, and an **“Update Recovery Firmware”** button; the progress display is the same as for the main firmware. On success, **“Recovery firmware updated!”** appears.

Device log

A **“Device Log”** appears at the end of the section only when log output has been received from the device. It is normally not shown.

7.4 Firmware Update Screen

On mobile, this opens from the “Update Firmware” button on the device detail screen. The screen changes as the update proceeds. The “Close” button at the top returns you to the previous screen (it is disabled during the update).



Confirming the update (device detail screen)

When you tap “Update Firmware” on the device detail screen, a confirmation dialog appears first.

- **Current version** → **latest version**
- **Release notes** (when available)
- **Cautions:** “Do not power off the device during the update.” and “Make sure the device battery is at 20% or above.”
- **“Cancel” / “Start Update”** buttons

Tapping **“Start Update”** moves to the firmware update screen and the download begins automatically.

Checking version

A spinner and “Checking version…” appear briefly.

Updating

The update runs in three steps:

1. Downloading the firmware
2. Transferring to the device
3. Verification

A progress bar and percentage are shown as each step proceeds. Completed steps turn green. The red warning “Do not power off the device during the update.” is shown throughout. The “Cancel” button aborts the update, though this is not recommended.

Update complete

A green checkmark and the message “Update complete!” appear. For the main firmware it says “The device will restart automatically.”; for the recovery firmware it says “The device will not restart.” Tap “Done” to return to the previous screen.

Update failed

A red X icon, the message “Update failed”, and the error details appear. If the update can be retried, a “Retry” button is shown. Tap “Close” to return to the previous screen.

▲ While a firmware update is in progress:

- Do not power off the device
- Do not close the app
- Keep the device close by

7.5 Local WebSocket Server

The “**Local WebSocket Server**” section of the settings screen controls whether local apps (dApps in a web browser, native apps, and so on) may connect to your wallet.

Platform	Shown	Notes
Desktop	✓	Local WS + allowed sites
Android	✓	Local WS + allowed sites
iOS	–	

Platform	Shown	Notes
		Hidden, because the local WebSocket server cannot work under OS restrictions

The switch (ON / OFF)

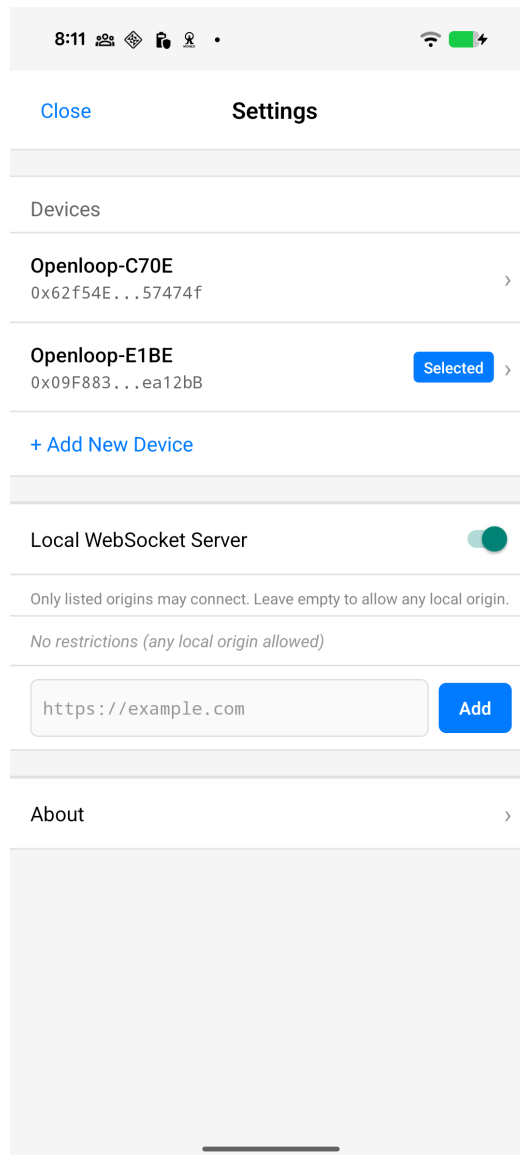
- **ON**: allows connections from local apps
- **OFF**: new connections are refused with HTTP 403 (**OFF is the initial state**)

▲ The switch is OFF right after you install the app. Turn it ON before connecting from a local app (a dApp in your browser, a mini wallet, and so on).

i Active sessions are protected: turning the switch OFF does not drop WebSocket sessions that are already established — only new connections are refused. This keeps something like Acrobat from breaking in the middle of signing a PDF.

Allowed sites (origin allowlist)

Shown when the switch is ON.



- **When the list is empty:** connections from any local origin are allowed
- **When one or more entries exist:** only the registered origins may connect; everything else is refused with HTTP 403

Operations:

- **Add:** type a URL into the input field (for example `https://example.com`) and press “Add”
- **Remove:** press the `x` button on the row

i **Origin normalization:** when you enter a URL, only the `scheme://host[:port]` part is stored. Entering `https://example.com/foo` stores `https://example.com`. This follows the web security specifications (WebSocket / WebAuthn), in which an origin never includes a path. If you need to distinguish individual pages on the same domain, separate them into different subdomains.

▲ **An origin means the whole site:** if the same domain also hosts untrusted content (for example pages that users can upload), allowing the origin allows all of it. Take care with sites on shared hosting.

7.6 PKCS#11 Cryptographic Module (desktop only)

The “**PKCS#11 Cryptographic Module**” section of the desktop settings screen controls device access through the PKCS#11 library (Firefox TLS client authentication, Adobe Acrobat PDF signing, SSH authentication, and so on).

The switch (ON / OFF)

- **ON:** allows connections from the PKCS#11 library
- **OFF:** new PKCS#11 sessions are refused and the library reports “no token” (0 slots) (**OFF is the initial state**)

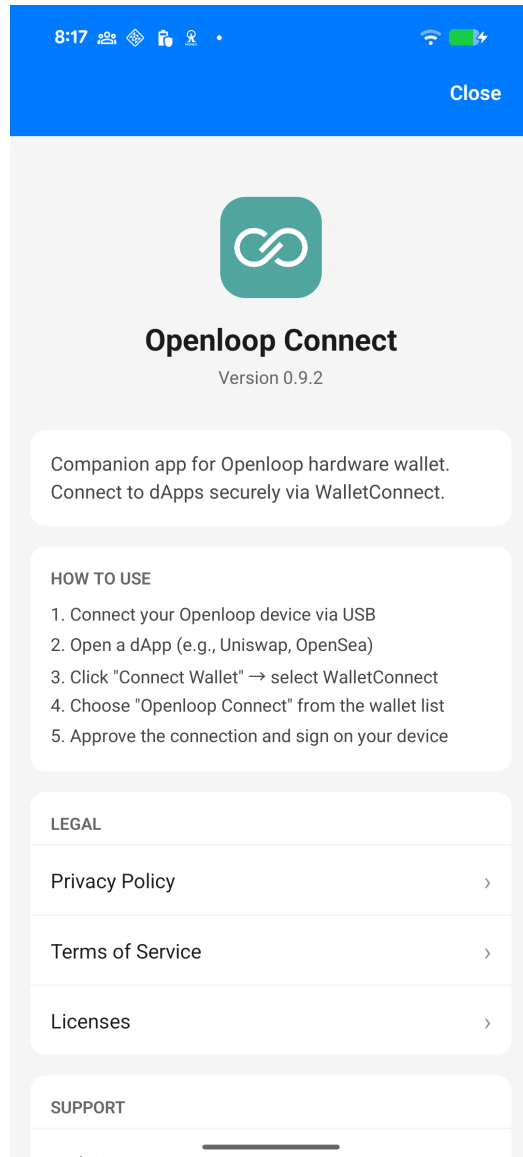
▲ **The switch is OFF right after you install the app.** Turn it ON before using PIV from Firefox, Adobe Acrobat, or SSH. If it stays OFF, the device is treated as having no token and your certificates and keys will not appear in the list.

i **Active sessions are protected:** turning the switch OFF leaves already-open PKCS#11 sessions running. For example, turning it OFF while Adobe Acrobat is signing a PDF does not interrupt that operation — only new `C_Initialize` calls are refused.

▲ **When a restart is needed:** Firefox and Adobe Acrobat cache the state of PKCS#11 modules, so after flipping the switch you may need to **restart Firefox or Acrobat** before the device list is re-read.

8. About Screen

Opens from the ⓘ (**info**) icon in the header bar, or from “About” in the [Settings screen](#). On mobile, the “Close” button at the top returns you to the previous screen; on desktop, use the “x” button at the top right.



8.1 App Information

- **App icon:** the Openloop Connect logo
- **App name:** “Openloop Connect”
- **Version:** the app version number (mobile shows “Version 0.9.3”; desktop shows it in its own localized form)

8.2 App Description

An overview of Openloop Connect.

8.3 How to Use

The basic workflow for Openloop Connect, in five steps:

1. Connect the Openloop device over USB
2. Open a dApp (for example Uniswap or OpenSea)
3. Choose “Connect Wallet” → WalletConnect
4. Choose “Openloop Connect” from the wallet list
5. Approve the connection and sign on the device

8.4 Legal Documents

Tapping each row opens it in your browser. It is shown in Japanese or English according to your device’s language setting.

- **Privacy Policy**
- **Terms of Service**
- **Licenses:** the list of open-source libraries in use

8.5 Support

- **Website:** opens <https://www.crypto.haudi.jp/openloop/>
- **Contact support:** opens <https://www.crypto.haudi.jp/openloop/contact>

8.6 Footer

“© 2026 Haudi Crypto, Inc. / All rights reserved.”

| 9. Credential Manager (Passkeys over BLE)

9.1 Overview

The mobile version of Openloop Connect includes a **passkey provider** that integrates with your operating system’s Credential Manager. This lets you complete passkey authentication for websites in your iPhone or Android browser **by communicating with the Openloop device over BLE**.

```

[Browser (iOS Safari / Android Chrome, etc.)]
  ↓ navigator.credentials API
[OS Credential Manager]
  ↓ provider selection (by the user)
[Openloop Connect's credential provider]
  ↓ BLE
[Openloop device (passkey signing)]

```

OS	Implementation	Required OS version
iOS	ASCredentialProviderViewController (Credential Provider Extension)	iOS 17 or later
Android	CredentialProviderService (Credential Manager API)	Android 14 or later

9.2 USB vs. BLE (important)

Openloop passkey authentication has **two paths**. The Connect Credential Manager feature handles the **BLE path only**.

Path	Communication	Connect involved?	Where it is used
USB CTAPHID	The OS's built-in security key support	No	Desktop (Windows / macOS / Linux)
BLE (via Connect)	Implemented by Connect itself	Yes	Mobile (iOS / Android)

▲ About the name shown in the iOS picker: the iOS WebAuthn picker shows “Openloop Connect”. This is by design in iOS — an extension’s display name is fixed to the containing app’s name (`CFBundleDisplayName`). The actual communication is over BLE, not USB. Please do not let the name mislead you.

9.3 Enabling It (iOS)

1. Install the Openloop Connect mobile app
2. Pair and select your Openloop device over BLE
3. iOS Settings app > General > AutoFill & Passwords
4. Turn “Openloop Connect” ON (as a passkey provider)
5. It can coexist with other providers such as iCloud Keychain

9.4 Enabling It (Android)

1. Install the Openloop Connect mobile app
2. Pair and select your Openloop device over BLE
3. **Android Settings app > Passwords & accounts > Additional services**
4. Turn **“Openloop via Bluetooth” ON**

 On Android the list shows **“Openloop via Bluetooth”** rather than the app name **“Openloop Connect”**. The name indicates that this is a passkey provider that works over BLE.

9.5 Registering a Passkey

1. Open the target website in your browser and choose the create-passkey option (for example, Google’s “Create a passkey”)
2. The OS Credential Manager picker appears
3. Choose **“Openloop Connect”** on iOS, or **“Openloop via Bluetooth”** on Android
4. Connect communicates with the Openloop over BLE and the registration request appears on the device screen
5. Press the **approve** button on the Openloop device
6. Registration is complete

9.6 Authenticating with a Passkey

1. Go to a website’s sign-in page in your browser and choose passkey authentication
2. In the OS picker, choose **“Openloop Connect”** (iOS) or **“Openloop via Bluetooth”** (Android)
3. Connect communicates with the Openloop over BLE and the authentication request appears on the device screen
4. Press the **approve** button on the Openloop device
5. You are signed in

9.7 USB and BLE Registration Compatibility (generally fine; Google is the exception)

With most relying parties, a single registration works over both USB and BLE. Verified with Monex, Nikko, JAL, Microsoft, SBI VC Trade, and others.

However, **some relying parties (Google being the prime example)** manage transport hints strictly per credential, so **a passkey registered over USB CTAPHID is not visible to Connect’s BLE flow** (and vice versa). Only for such relying parties do you need to **register**

the passkey twice with the same Openloop device — once over USB and once over BLE. Openloop issues a distinct credential ID per transport even on the same hardware, so the server records two independent credentials.

Relying party	USB registration usable over BLE	BLE registration usable over USB	Dual use
Monex / JAL / Microsoft / SBI VC and most others	✓	✓	One registration covers both
Google	✗	△	Register over both USB and BLE

See Chapter 2 of the Security Key Guide for details.

9.8 Troubleshooting

Symptom	What to check
Openloop does not appear in the picker	Is the provider turned ON in your iOS/Android settings (“Openloop Connect” on iOS, “Openloop via Bluetooth” on Android)? Right after installing Connect, the OS may need to be restarted
Nothing happens after choosing it	Is the Openloop within Bluetooth range, is it paired, and is Bluetooth turned on?
“No matching credential” error with Google	That Google account only has a passkey registered over USB. Register again for BLE (see §9.7 above)
It says “Openloop Connect”, so I assumed it was USB	This is the iOS display-name limitation. The communication is over BLE (see §9.2 above)

10. Safari Web Extension

10.1 Overview

The iOS version of Openloop Connect includes a Safari Web Extension. This extension lets Safari communicate directly with the Openloop device over BLE.

Openloop Connect acts as the **container app** for the Safari Web Extension. Once the extension is enabled, you can use your Openloop from Safari **even when the Connect app is not running**.

 On iOS every browser is WebKit-based, which restricts local WebSocket connections. The Safari Web Extension is the alternative that works around this limitation.

 The Safari Web Extension requires **iOS 26.2 or later**. On earlier versions of iOS you can still install Openloop Connect itself, but Openloop Connect will not appear in Safari's "Extensions" list.

10.2 Enabling It (two permission steps are required)

Because of iOS privacy protections, a Safari Web Extension needs **two permission steps**.

Step 1 alone is not enough, so be sure to complete both.

Step 1: Enable the extension

1. Open the **Settings** app on your iPhone
2. Tap **Safari** → **Extensions**
3. Find and tap "**Openloop Connect Safari Extension**" (shown in English)
4. Turn on "**Allow Extension**"
5. Set "**All Websites**" to "**Allow**"

Step 2: Per-site access permission (important)

Even after step 1, the extension may still be inaccessible from a website you open in Safari. This is by design in iOS: **each site needs explicit permission**.

1. Open the dApp (for example Uniswap or OpenSea) in Safari
2. Tap the "**Aa**" (**page settings**) icon at the left of the address bar
3. Tap "**Manage Extensions**" (website settings)
4. Tap "**Openloop Connect Safari Extension**"
5. Choose "**Always Allow**"
6. Reload the page

 This setting is **required per site**. You only do it once per site, but you may need to repeat it when you start using a different dApp.

Troubleshooting

If “Openloop Connect Safari Extension” does not appear in the “Aa” menu, or the extension does not work even after choosing “Always Allow”:

- Re-check the step 1 settings
- Quit Safari completely and restart it
- Restart your iPhone
- If you reinstalled the app, per-site permissions are reset, so repeat step 2

10.3 Using It

With the Safari Web Extension enabled, visiting a web app that supports the Openloop API lets you communicate with the Openloop device through the extension.

If a button cannot be pressed on your first visit, or the Openloop device is not recognized, complete step 2 “Per-site access permission” in section 10.2.

11. Openloop API Server

11.1 Overview

Openloop Connect includes a local WebSocket server that lets web apps and native apps access the Openloop device. The server starts automatically when Openloop Connect launches.

You can enable or disable the server, and manage which origins may connect, from [the “Local WebSocket Server” section of the settings screen](#).

11.2 Endpoint

```
ws://127.0.0.1:21320
```

APDU commands are exchanged using the JSON-RPC 2.0 protocol.

11.3 Supported Platforms

Platform	Supported	Notes
macOS	✓	

Platform	Supported	Notes
Windows	✓	
Linux	✓	
Android	✓	A foreground service keeps the BLE connection alive
iOS	✗	Not available under OS restrictions (use the Safari Web Extension instead)

11.4 How It Appears on the Main Screen

When an application connects to the local WebSocket server, its origin (URL) appears with a green dot in the “Local Apps” section of the main screen. The “Disconnect” button next to each app disconnects it individually.

11.5 API Reference

For the full API reference, see the [Openloop Connect Specification](#).

12. PKCS#11 Dynamic Library

12.1 Overview

The desktop version of Openloop Connect bundles a PKCS#11 dynamic library for using the PIV (Personal Identity Verification) feature. It can be used for SSH authentication, TLS client authentication, PDF signing, code signing, and more.

12.2 Library Paths

The PKCS#11 library is bundled in the `pkcs11/` folder under the application’s installation path:

OS	Relative to the app
macOS	<code><app>/Contents/Resources/pkcs11/libopenloop-pkcs11.dylib</code>
Windows	<code><app>/resources/pkcs11/libopenloop-pkcs11.dll</code>
Linux	<code><app>/resources/pkcs11/libopenloop-pkcs11.so</code>

The installation path `<app>` depends on how you installed the app:

Installation method	Location
macOS (DMG)	<code>/Applications/Openloop Connect.app</code>
Windows (official installer)	<code>C:\Program Files\Openloop Connect</code>
Windows (Microsoft Store)	In PowerShell: <code>(Get-AppxPackage 7CA75049.OpenloopConnect).InstallLocation</code>
Linux (deb)	<code>/opt/Openloop Connect</code>

For example, the full path for the macOS DMG build is `/Applications/Openloop Connect.app/Contents/Resources/pkcs11/libopenloop-pkcs11.dylib`.

12.3 Prerequisites

The PKCS#11 library talks to the device through Openloop Connect. Before using it, check that:

1. **Openloop Connect** is running
2. The Openloop device is connected over **USB**
3. **Settings > USB Settings > USB HID** is **ON** on the device
4. **Passkey & PIV > PIV** is **ON** on the device
5. **Settings > PKCS#11 Cryptographic Module** is **ON** in Openloop Connect

12.4 Usage

For detailed usage of the PKCS#11 library, see the [Security Key Guide](#). It covers SSH authentication, Firefox TLS client authentication, key management with pkcs11-tool, setting the PIV PIN, and more.

13. Troubleshooting

The device is not found over BLE

- Check that Bluetooth is enabled
- Check that the device is powered on
- Check that no other app is connected to the device (BLE is a one-to-one connection)
- **Delete the old pairing data** (see “If Pairing Fails” in [section 4.1](#))

The BLE connection is unstable

- Move the device and your phone closer together
- Check for interference from other BLE devices
- Quit the app completely and restart it

WalletConnect will not connect

- Check your internet connection
- Restart Openloop Connect
- Retry the QR code or the connection from the dApp
- Use the “Disconnect” button on the main screen to end the existing session, then reconnect

Signing requests do not arrive

- Check that Openloop Connect is running in the foreground
- Check the WalletConnect connection status (does the main screen show “Connected to: [dApp name]”?)
- Resend the transaction from the dApp

The firmware update fails

- Check the device’s battery level
- Check that the BLE connection is stable (move the device closer)
- Restart the app and try again
- If the device entered recovery mode, run the firmware update again

The PKCS#11 library does not work

- Check that Openloop Connect is running
- Check that the device is connected over USB
- Check that the library path is correct (see [section 12.2](#))
- Check that PIV is turned ON on the device
- For details, see the [Security Key Guide](#)

The Safari Web Extension does not work

- Check that “Openloop Connect Safari Extension” is enabled under Settings → Safari → Extensions
- Quit Safari completely and restart it
- Restart your iPhone

Cannot connect to the local WebSocket server

- Check that Openloop Connect is running
 - Check that the address is `ws://127.0.0.1:21320`
 - Check that a firewall is not blocking local traffic
 - Not available on iOS (use the [Safari Web Extension](#) instead)
-

Copyright © 2026 Haudi Crypto, Inc. All rights reserved.