ZapBox White Paper

Bitcoin Lightning Network-gesteuerte IoT-Gerätesteuerung

Version wp930588 | Januar 2026

Abstract

ZapBox ist eine Open-Source-Hardware- und Softwarelösung, die es ermöglicht, physische Geräte über das Bitcoin Lightning Network zu steuern. Durch die Kombination eines ESP32-Mikrocontrollers mit einem Display, Relais und einer WebSocket-basierten Backend-Infrastruktur schafft ZapBox eine Brücke zwischen Kryptowährungszahlungen und realer IoT-Gerätesteuerung. Dieses White Paper beschreibt die technische Architektur, Implementierungsdetails, Anwendungsfälle und die Vision hinter dem Projekt.

Inhaltsverzeichnis

1. Einleitung

Die Digitalisierung von Zahlungsprozessen hat in den letzten Jahren erhebliche Fortschritte gemacht. Das Bitcoin Lightning Network bietet als Layer-2-Lösung auf der Bitcoin-Blockchain die Möglichkeit, nahezu kostenfrei und in Echtzeit Mikrotransaktionen durchzuführen. ZapBox nutzt diese Technologie, um eine dezentrale, programmierbare Schnittstelle zwischen digitalen Zahlungen und physischer Gerätesteuerung zu schaffen.

Vision

"Wir möchten die Bitcoin-Lightning-Technologie ⚡ für alle zugänglich machen – weltweit. 🌍"

ZapBox senkt die Einstiegshürden für die Nutzung von Bitcoin Lightning im Internet of Things (IoT) und ermöglicht es jedem – vom Hobbybastler bis zum professionellen Anwender – eigene Lightning-gesteuerte Anwendungen zu realisieren.

---

2. Problemstellung

Trotz des enormen Potenzials des Lightning Networks existieren mehrere Barrieren für dessen breite Adoption im IoT-Bereich:

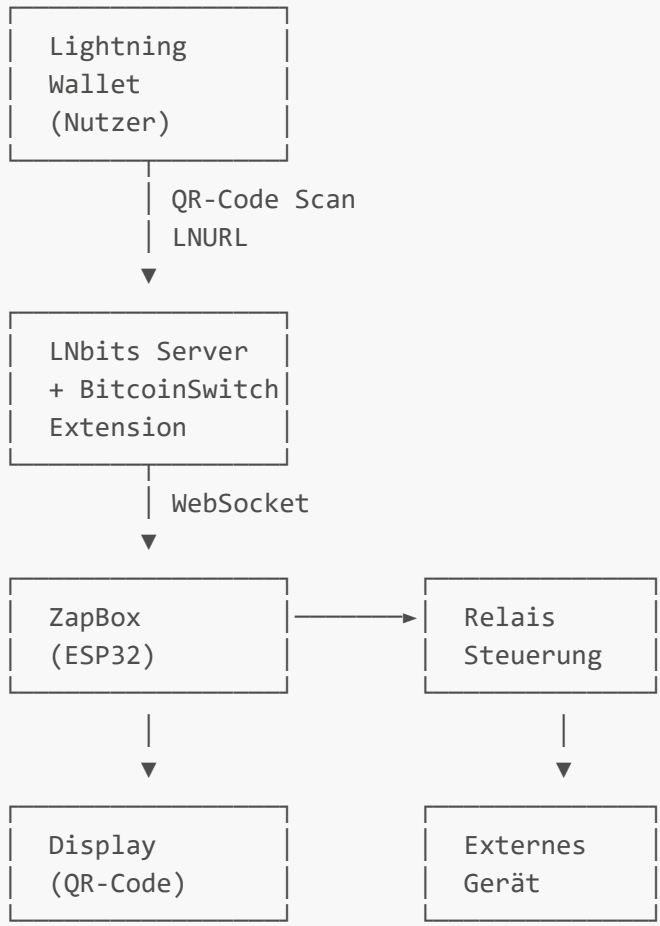
- **Technische Komplexität**: Die Integration von Lightning-Zahlungen in Hardware-Projekte erfordert tiefgreifendes technisches Wissen
- **Fehlende Standardlösungen**: Es gibt wenige fertige Lösungen, die Hardware, Software und Backend integrieren
- **Zugänglichkeit**: Nicht-technische Nutzer können die Technologie nicht ohne weiteres nutzen
- **Hardware-Integration**: Die Verbindung zwischen Zahlungseingang und physischer Aktion ist komplex

ZapBox adressiert diese Herausforderungen durch eine vollständig integrierte, benutzerfreundliche Lösung.

---

3. Technische Architektur

3.1 Systemübersicht



3.2 Komponenten

Frontend (Hardware)

- ESP32 Mikrocontroller
- Display (TFT)
- Touchscreen (CST816S)
- Relais-Modul
- Optionale NFC-Unterstützung (PN532)

Backend (Software)

- LNbits Server (Wallet & Account System)
- BitcoinSwitch Extension
- WebSocket-Server für Echtzeitkommunikation

Kommunikation

- WiFi (2.4 GHz)
- WebSocket-Protokoll
- LNURL-Standard

---

4. Hardware-Spezifikationen

4.1 Kern-Komponenten

Mikrocontroller: ESP32

- Dual-Core Prozessor (240 MHz)
- 520 KB SRAM
- WiFi 802.11 b/g/n
- Bluetooth (optional nutzbar)
- Geringe Stromaufnahme
- Deep-Sleep-Modus verfügbar

Display

- TFT-Display (verschiedene Größen)
- Kapazitiver Touchscreen
- Hochauflösende QR-Code-Darstellung
- Konfigurierbare Farbschemata

Relais-Modul

- Schaltspannung: bis 250V AC / 30V DC
- Schaltstrom: bis 10A
- Optionale USB-Buchse (5V)
- Galvanische Trennung

4.2 Physisches Design

Die Hardware ist in drei Hauptvarianten verfügbar:

- **ZapBox Compact**: Einzelrelais, kompakte Bauweise
- **ZapBox Duo**: Zwei Relais, davon ein Leistungsrelais mit externen Klemmen
- **ZapBox Quattro**: Vier Relais mit externen Anschlussklemmen für professionelle Anwendungen

Alle Varianten verfügen über:

- 3D-druckbare Gehäuse (FreeCAD-Dateien verfügbar)
- Elektrische Konstruktionsunterlagen
- Stücklisten mit Bezugsquellen

---

5. Backend-Infrastruktur

5.1 LNbits Server

LNbits ist ein Open-Source Lightning Network Wallet und Account System. Es bietet:

- Multi-User-Wallet-Verwaltung
- Erweiterbare Plugin-Architektur
- REST API
- WebSocket-Support
- Lightning Node Abstraktion

5.2 BitcoinSwitch Extension

Die BitcoinSwitch Extension erweitert LNbits um IoT-Steuerungsfunktionalitäten:

- **Device Management**: Verwaltung mehrerer ZapBox-Geräte
- **Invoice Generation**: Automatische Rechnungserstellung
- **WebSocket Bridge**: Echtzeitkommunikation mit Geräten
- **Event Tracking**: Protokollierung aller Schaltvorgänge
- **Multi-Channel Support**: Verwaltung mehrerer Ausgänge pro Gerät

5.3 Device String

Jede ZapBox wird durch einen eindeutigen "Device String" identifiziert, der:

- Die Verbindung zum LNbits-Server authentifiziert
- Das spezifische Gerät identifiziert
- Den zu steuernden Ausgang spezifiziert
- In der Firmware konfiguriert wird

---

6. Zahlungsablauf

6.1 LNURL-Protokoll

Der auf dem Display angezeigte QR-Code verwendet das LNURL-Protokoll (Lightning URL), einen Standard für Lightning Network Interaktionen. Der QR-Code enthält:

1. Server-URL des LNbits-Backends
2. Device-Identifikation
3. Kanal/Ausgangs-Spezifikation
4. Optionale Metadata (Produktname, Preis)

6.2 Transaktionsablauf

```
1. ZapBox zeigt QR-Code (LNURL)
   └─> Enthält: Server-URL + Device-ID + Channel

2. Nutzer scannt QR-Code mit Lightning Wallet
   └─> Wallet kontaktiert LNbits Server

3. LNbits prüft WebSocket-Verbindung zur ZapBox
   └─> Bestätigt Geräteverfügbarkeit

4. LNbits generiert Invoice
   └─> Sendet an Wallet mit Betrag & Beschreibung

5. Nutzer bestätigt Zahlung im Wallet
   └─> Lightning-Zahlung wird ausgeführt

6. LNbits bestätigt Zahlungseingang
   └─> Sendet Event über WebSocket an ZapBox

7. ZapBox empfängt Event
   └─> Schaltet Relais für definierte Dauer

8. Relais öffnet/schließt Stromkreis
   └─> Externes Gerät wird aktiviert
```

6.3 Timing und Performance

- **Zahlungsbestätigung**: < 1 Sekunde (Lightning)
- **Event-Übertragung**: < 100ms (WebSocket)
- **Relais-Reaktion**: < 50ms
- **Gesamtdauer (Scan bis Schaltung)**: ~1-2 Sekunden

---

7. Installation und Konfiguration

7.1 Web Installer

ZapBox verwendet einen browserbasierten Installer: **https://installer.zapbox.space/**

Vorteile:

- Keine Software-Installation notwendig
- Plattformunabhängig (Windows, macOS, Linux)
- Nutzt Web Serial API

- Visuelle Feedback während des Flash-Vorgangs

7.2 Installationsprozess

Schritt 1: Firmware-Flash

1. ZapBox via USB verbinden
2. Web Installer öffnen
3. Firmware-Version auswählen
4. "Flash" Button klicken
5. Warten auf Abschluss

Schritt 2: Konfiguration

Pflichtparameter:

- **SSID**: WiFi-Netzwerkname
- **Password**: WiFi-Passwort
- **Device String**: Von LNbits BitcoinSwitch Extension

Optionale Parameter:

- **Display Orientation**: 0°, 90°, 180°, 270°
- **Color Scheme**: Hintergrund- und Textfarben
- **Screensaver**: Aktivierung und Timeout
- **Deep Sleep**: Energiesparmodus
- **Special Modes**: Multi-Channel, Threshold, BTC-Ticker

7.3 Ersteinrichtung

Nach dem ersten Boot:

1. ZapBox startet im AP-Modus (falls keine WiFi-Konfiguration)
2. Nutzer verbindet sich mit ZapBox-Access-Point
3. Captive Portal öffnet sich automatisch
4. WiFi und Device-Einstellungen werden eingegeben
5. ZapBox speichert Konfiguration und startet neu
6. Verbindung zum konfigurierten WiFi und LNbits-Server

---

8. Betriebsmodi

8.1 Standard-Modus

Klassischer Bitcoin-Switch:

- Ein Relais
- Ein Preis pro Zahlung
- Feste Schaltdauer

8.2 Multi-Channel-Modus

Mehrere unabhängige Zahlungskanäle:

- Bis zu 4 verschiedene QR-Codes
- Unterschiedliche Preise
- Verschiedene Schaltdauern
- Individuelle Ausgänge

Anwendung: Verkaufsautomat mit mehreren Produkten

8.3 Threshold-Modus

Akkumulative Zahlungen:

- Sammelt Zahlungen bis Schwellwert erreicht
- Schaltet erst bei Erreichen des Threshold
- Anzeige des aktuellen Saldos
- Reset nach Schaltung

Anwendung: Crowdfunding-basierte Aktivierung

8.4 BTC-Ticker-Modus

Bitcoin-Preis-Display:

- Echtzeit BTC/USD (oder andere Fiat-Währungen)
- Optionale Zahlungsfunktion
- Screensaver mit wechselnden Informationen

Anwendung: Informationsdisplay mit optionaler Spendenfunktion

8.5 Special-Modus

Erweiterte Funktionen:

- NFC-Integration (LNURL-NFC-Karten)
- Custom Display-Layouts
- API-Endpunkte für externe Systeme
- Event-Logging

---

9. Sicherheit und Zuverlässigkeit

9.1 Sicherheitsaspekte

Netzwerksicherheit

- TLS/SSL-verschlüsselte WebSocket-Verbindung
- WPA2-geschütztes WiFi
- Keine privaten Schlüssel auf dem Gerät
- Device String als einzige Authentifizierung

Zahlungssicherheit

- Lightning Network native Sicherheit
- Atomare Transaktionen
- Keine Rückbuchungen
- Invoice-basiertes System

Physische Sicherheit

- Galvanisch getrennte Relais
- Überstromschutz (optional)
- Gehäuse-Design verhindert versehentlichen Zugriff

9.2 Zuverlässigkeit

Fehlerbehandlung

- Automatische WiFi-Reconnection
- WebSocket-Reconnection mit Backoff
- Watchdog-Timer
- Fehler-Logging auf Display

Monitoring

- Verbindungsstatus auf Display
- Letzte Transaktion sichtbar
- Fehler-Codes bei Problemen
- Optional: Remote-Monitoring über LNbits

Wartung

- OTA-Updates möglich (zukünftig)
- Config-Reset via Button
- Factory-Reset-Funktion
- Diagnose-Modus

---

10. Anwendungsfälle

10.1 Kommerzielle Anwendungen

Verkaufsautomaten

- Getränkeautomaten
- Snackautomaten
- Waschmaschinen/Trockner in Waschsalons
- Schließfächer

Zugangskontrollen

- Türöffner
- Parkplatzschranken
- Ladestationen (E-Bike, E-Scooter)

- Co-Working-Spaces

Dienstleistungen

- Luftpumpen an Tankstellen
- Staubsauger an Waschplätzen
- Massagesessel
- Duschen (Campingplätze, Strandbäder)

10.2 Bildung und Demonstration

Workshops

- Bitcoin-Lightning-Schulungen
- IoT-Entwicklung mit Micropayments
- Maker-Spaces und Hackerspaces

Ausstellungen

- Hands-on Lightning-Demonstrationen
- Bitcoin-Konferenzen
- Tech-Messen

10.3 Private Nutzung

Smart Home

- Lichtsteuerung (Gamification)
- Garagentoröffner für Gäste
- Heimautomation mit Bitcoin-Trigger

Spiele und Challenges

- Escape-Room-Rätsel
- Bitcoin-Trivia-Automaten
- Belohnungssysteme

10.4 Soziale Projekte

Crowdfunding

- Threshold-Modus für Gemeinschaftsprojekte
- Öffentliche Kunstinstallationen
- Charity-Events

Dezentralisierung

- P2P-Ressourcenteilung ohne zentrale Verwaltung
- Community-betriebene Infrastruktur

---

11. Produktvarianten

11.1 ZapBox Compact

Zielgruppe: Einsteiger, Hobbyisten

Spezifikationen:

- 1 Relais (10A)
- USB-Ausgang (5V/2A)
- Kompaktes Gehäuse
- Eingebaute Antenne
- Touch-Display

Anwendungen:

- USB-Gerätesteuerung
- Einzelne Schaltfunktion
- Demonstrationsprojekte

11.2 ZapBox Duo

Zielgruppe: Ambitionierte Anwender

Spezifikationen:

- 2 Relais
- 1 Standard-Relais (10A)
- 1 Leistungsrelais (16A) mit externen Klemmen
- Größeres Gehäuse
- Potentialfreie Kontakte

Anwendungen:

- Zwei unabhängige Schaltungen
- Direkte 230V-Schaltung (Leistungsrelais)
- Kombinierte Anwendungen

11.3 ZapBox Quattro

Zielgruppe: Professionelle Anwendungen

Spezifikationen:

- 4 Relais (je 10A)
- Alle Kontakte nach außen geführt
- Professionelle Schraubklemmen
- Robustes Gehäuse
- Optionale DIN-Rail-Montage

Anwendungen:

- Verkaufsautomaten
- Komplexe Steuerungen

- Industrielle Integration
- Multi-Produkt-Systeme

11.4 Vergleichstabelle

Eigenschaft	Compact	Duo	Quattro
Relais	1	2	4
Externe Klemmen	Nein	Ja (1x)	Ja (4x)
USB-Ausgang	Ja	Optional	Nein
Gehäusegröße	Klein	Mittel	Groß
Max. Schaltleistung	2300W	3680W	9200W
Preissegment	Einstieg	Mittel	Profi

---

12. Open-Source-Ökosystem

12.1 Lizenzierung

Alle Komponenten der ZapBox sind vollständig Open Source:

Hardware

- Elektrische Schaltpläne: CC BY-SA 4.0
- 3D-Modelle (FreeCAD): CC BY-SA 4.0
- PCB-Layouts: CERN OHL v2

Software

- Firmware: MIT License
- Web Installer: MIT License
- Dokumentation: CC BY-SA 4.0

12.2 Repository-Struktur

GitHub Repository: https://github.com/AxelHamburch/ZapBox

```
ZapBox/
├── src/                    # Firmware-Quellcode
├── assets/                 # Hardware-Designs
│   ├── electric/          # Schaltpläne
│   └── housing/           # Gehäuse-3D-Modelle
├── installer/             # Web Installer
├── lib/                   # Bibliotheken
├── docs/                  # Dokumentation
└── test/                  # Tests
```

12.3 Verwendete Open-Source-Projekte

Backend:

- **LNbits** (MIT): https://github.com/lnbits/lnbits
- **BitcoinSwitch Extension** (MIT): https://github.com/lnbits/bitcoinswitch_extension

Firmware-Bibliotheken:

- ESP32 Arduino Core
- TFT_eSPI (Display)
- ArduinoJson
- WebSocketsClient
- WiFiManager

12.4 Community-Beiträge

Das Projekt lebt von Community-Beiträgen:

- Bug Reports und Feature Requests über GitHub Issues
- Pull Requests für Verbesserungen
- Hardware-Varianten und Anpassungen
- Übersetzungen
- Dokumentation

---

13. Roadmap und Zukunftsperspektiven

13.1 Kurzfristige Ziele (Q1-Q2 2026)

Firmware

- OTA-Update-Funktionalität
- Erweiterte NFC-Unterstützung
- Verbesserte Energieverwaltung
- Lokale API für externe Integration

Hardware

- ZapBox Quattro Serienproduktion
- Industrielle Variante mit DIN-Rail
- Outdoor-Gehäuse (IP65)

Software

- Web Installer v2.0 mit erweiterten Features
- Mobile App für Geräteverwaltung
- Monitoring-Dashboard

13.2 Mittelfristige Ziele (Q3-Q4 2026)

Erweiterungen

- ZapBox Mini (ohne Display, nur NFC)
- ZapBox Pro (mit integriertem Lightning Node)
- Modular-System für individuelle Konfigurationen

Integration

- Home Assistant Integration
- MQTT-Broker-Support
- RESTful API für Third-Party-Apps

Ökosystem

- Marketplace für Custom Firmware
- Template-Bibliothek für Anwendungen
- Zertifizierungsprogramm für Partner

13.3 Langfristige Vision

Globale Adoption

- Entwicklungsländer: Micropayment-Infrastruktur
- Unbanked Populations: Finanzielle Inklusion
- Dezentrale Energiemärkte: P2P-Stromhandel

Technische Innovation

- Integration mit RGB Protocol
- Taproot Assets Support
- Lightning-native Smart Contracts

Standardisierung

- LNURL-Device-Protocol-Standard
- IoT-Payment-Framework
- Interoperabilität mit anderen Lightning-IoT-Geräten

---

14. Fazit

ZapBox repräsentiert einen bedeutenden Schritt in der Verbindung von Bitcoin Lightning Network und dem Internet of Things. Durch die Kombination von benutzerfreundlicher Hardware, robuster Software und einem vollständig Open-Source-Ansatz macht das Projekt Lightning-Zahlungen für physische Anwendungen zugänglich.

Kernvorteile

1. **Zugänglichkeit**: Senkt technische Hürden drastisch
2. **Dezentralisierung**: Keine zentralen Intermediäre notwendig
3. **Kosten**: Mikrotransaktionen ohne signifikante Gebühren
4. **Geschwindigkeit**: Echtzeit-Zahlungen und -Reaktionen
5. **Open Source**: Vollständige Transparenz und Anpassbarkeit

6. **Skalierbarkeit**: Von Einzelgeräten bis zu großen Installationen

Bedeutung für das Lightning-Ökosystem

ZapBox demonstriert das Potenzial von Lightning über reine Peer-to-Peer-Zahlungen hinaus. Die Ermöglichung von Machine-to-Machine-Payments und IoT-Integration öffnet neue Anwendungsfelder:

- **Circular Economy**: Bitcoin als natives Zahlungsmittel in automatisierten Systemen
- **Programmable Money**: Zahlungen, die automatisch Aktionen auslösen
- **Micropayment Infrastructure**: Grundlage für pay-per-use Geschäftsmodelle

Aufruf zur Beteiligung

Das Projekt lädt jeden ein, Teil der Entwicklung zu werden:

- **Entwickler**: Tragen Sie zur Firmware und Software bei
- **Hardware-Designer**: Entwickeln Sie neue Varianten und Gehäuse
- **Anwender**: Teilen Sie Ihre Use Cases und Feedback
- **Unternehmen**: Integrieren Sie ZapBox in Ihre Produkte

"Jeder soll die ZapBox selbst bauen können. Aber viele werden das gar nicht können oder wollen. Wir möchten die Bitcoin-Lightning-Technologie ⚡ für alle zugänglich machen – weltweit. 🌍"

---

15. Referenzen

Technische Spezifikationen

- **Lightning Network**: https://lightning.network/
- **LNURL Protocol**: https://github.com/lnurl/luds
- **ESP32 Datasheet**: https://www.espressif.com/en/products/socs/esp32
- **Bitcoin Switch (Ben Arc)**: Original Konzept und Inspiration

Open-Source-Projekte

- **ZapBox GitHub**: https://github.com/AxelHamburch/ZapBox
- **LNbits**: https://github.com/lnbits/lnbits
- **BitcoinSwitch Extension**: https://github.com/lnbits/bitcoinswitch_extension

Weiterführende Informationen

- **ZapBox Website**: https://zapbox.space/
- **Web Installer**: https://installer.zapbox.space/
- **Community Forum**: (In Entwicklung)
- **Documentation**: https://github.com/AxelHamburch/ZapBox/wiki

Kontakt

- **GitHub**: https://github.com/AxelHamburch
- **Repository Issues**: https://github.com/AxelHamburch/ZapBox/issues
- **Email**: (Über GitHub-Profile)

Anhang A: Technische Glossar

- **ESP32**: Mikrocontroller von Espressif mit WiFi und Bluetooth
- **Lightning Network**: Layer-2-Protokoll für schnelle Bitcoin-Transaktionen
- **LNURL**: Lightning URL Protocol für vereinfachte Interaktionen
- **Invoice**: Lightning-Rechnung für eine Zahlung
- **WebSocket**: Bidirektionales Kommunikationsprotokoll
- **Relais**: Elektromechanischer Schalter
- **OTA**: Over-The-Air (drahtlose Firmware-Updates)
- **FOSS**: Free and Open Source Software

Anhang B: Elektrische Spezifikationen

Relais-Spezifikationen

Standard-Relais (Compact, Duo, Quattro)

- Kontakttyp: SPDT (Single Pole Double Throw)
- Nennspannung: 250V AC / 30V DC
- Nennstrom: 10A
- Schaltleistung: 2500VA / 300W
- Lebensdauer: >100.000 Zyklen

Leistungsrelais (Duo)

- Kontakttyp: SPDT
- Nennspannung: 250V AC / 30V DC
- Nennstrom: 16A
- Schaltleistung: 4000VA / 480W
- Lebensdauer: >50.000 Zyklen

Stromversorgung

- Eingangsspannung: 5V DC via USB-C
- Stromaufnahme: 150-500mA (je nach Display-Helligkeit)
- Deep Sleep: <10mA
- Empfohlenes Netzteil: 5V/2A

Anhang C: Versionierung

Firmware-Versionen:

- v9.x.x: Aktuelle Hauptversion
- Semantische Versionierung (MAJOR.MINOR.PATCH)
- Release Notes im GitHub Repository

Hardware-Revisionen:

- Elektrische Schemata: eXXXXXX-compact
- Gehäuse: bXXXXXX-compact
- Vollständige History in Git

Dieses Dokument wurde am 2. Januar 2026 erstellt und wird regelmäßig aktualisiert. Die neueste Version ist stets im GitHub-Repository verfügbar.

ZapBox - Open Source Bitcoin Lightning IoT

Powered by the Lightning Network | Available Worldwide | Fully Open Source