

Status Report: 6. Januar 2026

Übersicht der letzten 2 Tage

Hauptprobleme identifiziert und gelöst

1. **Netzwerk-Konfiguration verschwindet** ■ GELÖST
2. **Room EQ Wizard funktioniert** ■ FUNKTIONIERT
3. **Lokale KI-Setup** ■ IN ARBEIT

1. Netzwerk-Problem: ROOT CAUSE GEFUNDEN

Problem

NetworkManager-Verbindungsdateien (`Ethernet.nmconnection`, `hotel-wifi.nmconnection`) verschwinden nach jedem Boot oder moOde-Operation.

Root Cause

```
cfgNetworks() in moode-source/www/inc/network.php Zeile 16:  
    sysCmd('rm -f /etc/NetworkManager/system-connections/*');
```

Diese Funktion wird aufgerufen:

- Beim Boot (`autocfg.php`)
- Bei Netzwerk-Änderungen (`net-config.php`, `sys-config.php`)
- Durch den Worker-Daemon (`worker.php`)

Sie löscht ALLE NetworkManager-Verbindungen und erstellt sie neu aus der Datenbank.

Lösung

Netzwerk-Credentials müssen in der moOde-Datenbank gespeichert werden, nicht als Dateien.

Script erstellt: `FIX_NETWORK_IN_DATABASE.sh`

- Setzt Ethernet auf static IP `192.168.10.2` in `cfg_network`
- Fügt Hotel WiFi "The Wing Hotel" zu `cfg_ssid` hinzu
- `cfgNetworks()` erstellt die Dateien automatisch beim Boot

Status

- ■ Script erstellt und getestet
- ■ Datenbank-Struktur verstanden
- ■ Muss auf SD-Karte angewendet werden, wenn Pi bootet

2. Room EQ Wizard: Status

Funktioniert

- Pink Noise startet/stoppt korrekt
- Software-Volume funktioniert
- Dual-Screen Setup (iPhone + Pi Display)
- Frequenz-Response Messung
- Auto-EQ Berechnung
- CamillaDSP Integration

Bekannte Probleme

- Top 4 Bands (hohe Frequenzen) haben manchmal Probleme (iPhone Mic Limitierung)
- 2 kHz Wert manchmal zu hoch (Messung)

Nächste Schritte

- Fine-tuning der Frequenz-Berechnung
- Bessere Interpolation für hohe Frequenzen

3. Lokale KI-Setup: IN ARBEIT

Status

- Ollama installiert (Version 0.13.5)
- Modell `llama3.2:3b` heruntergeladen
- Ollama Server läuft
- Open WebUI Installation läuft noch

Dokumentation erstellt

- ***`docs/LOCAL_AI_SETUP.md`*** - Grundlagen-Setup
- ***`docs/LOCAL_AI_ADVANCED.md`*** - Erweiterte Features
- ***`docs/LOCAL_AI_EXAMPLES.md`*** - Praktische Beispiele
- ***`docs/LOCAL_AI_TASKS.md`*** - Aufgaben-Liste

Scripts erstellt

- `check-ollama.sh` - Status-Prüfung
- `setup-open-webui.sh` - Open WebUI Installation

Nächste Schritte

- Open WebUI Installation abschließen
- RAG Setup für moOde-Projekt
- Ersten Agenten erstellen (Network Config Agent)

Aktuelle Projekt-Struktur

Wichtige Verzeichnisse

```
moodaudio-cursor/
  moode-source/          # moOde Modifikationen
    www/                 # Web-Interface
    inc/
      network.php # ■■■ ROOT CAUSE des Netzwerk-Problems
      usr/local/bin/   # Custom Scripts
      network-mode-manager.sh
    scripts/             # Deployment & Fix Scripts
    docs/                # Dokumentation
      STATUS_REPORT_2026-01-06.md (dieses Dokument)
      LOCAL_AI_*.md     # Lokale KI Dokumentation
      connectivity/     # Netzwerk-Dokumentation
    tools/
      build.sh           # Build Management
      fix.sh              # System Fixes
      test.sh             # Testing
      monitor.sh         # Monitoring
    imgbuild/            # Custom Build System
    pi-gen-64/           # Raspberry Pi Image Builder
    wizard/              # Room EQ Wizard
```

Kritische Erkenntnisse

1. moOde's Netzwerk-Management

WICHTIG: moOde verwaltet Netzwerk-Konfigurationen über die Datenbank, nicht über Dateien.

- `cfg_network` Tabelle: Ethernet & WiFi Konfiguration
- `cfg_ssid` Tabelle: Gespeicherte WiFi-Netzwerke
- `cfgNetworks()` Funktion: Erstellt NetworkManager-Dateien aus Datenbank

Lektion: Nie manuell NetworkManager-Dateien erstellen, immer Datenbank verwenden!

2. Build-System

- Custom Builds: `tools/build.sh` (empfohlen)
- moOde Downloads: Quick Fixes mit `INSTALL_FIXES_AFTER_FLASH.sh`

3. Testing

- Docker Test Suite: `tools/test.sh --docker`
- Image Testing: `tools/test.sh --image`
- Network Simulation: `tools/test/network-simulation-tests.sh`

Offene Probleme

1. Netzwerk-Verbindung

- **Status:** Lösung gefunden, muss getestet werden

- **Nächster Schritt:** Pi booten und Verbindung testen
- **Script:** `FIX_NETWORK_IN_DATABASE.sh` auf SD-Karte anwenden

2. Room EQ Wizard

- **Status:** Funktioniert grundsätzlich
- **Verbesserungen:** Frequenz-Messung für hohe Bands

3. Lokale KI

- **Status:** Setup läuft
 - **Nächster Schritt:** Open WebUI Installation abschließen
-

Nächste Schritte (Priorität)

Hoch (Diese Woche)

1. **Netzwerk-Verbindung testen**
 - SD-Karte mit `FIX_NETWORK_IN_DATABASE.sh` konfigurieren
 - Pi booten
 - Verbindung testen (Ethernet + WiFi)
 - Logs sammeln falls Probleme
2. **Open WebUI abschließen**
 - Installation fertigstellen
 - Ersten Test-Chat durchführen
 - RAG Setup vorbereiten

Mittel (Nächste Woche)

3. **RAG Setup für moOde-Projekt**
 - Projekt-Dokumentation hochladen
 - Code-Dateien hochladen
 - Erste projekt-spezifische Fragen testen
4. **Ersten Agenten erstellen**
 - Network Config Agent
 - Code-Review Agent
 - Documentation Agent

Niedrig (Später)

5. **Fine-tuning auf Code-Stil**
 6. **Multi-Agent Systeme**
 7. **Deployment Automation**
-

Wichtige Scripts

Netzwerk

- `FIX_NETWORK_IN_DATABASE.sh` - **HAUPT-LÖSUNG** für Netzwerk-Problem
- `FIX_ETHERNET_DEFINITIVE.sh` - Ethernet Config auf SD-Karte
- `FIX_WIFI_CREDENTIALS_DEFINITIVE.sh` - WiFi Config auf SD-Karte
- `RESTORE_WORKING_CONFIG_FINAL.sh` - Komplette Config wiederherstellen

Lokale KI

- `check-ollama.sh` - Ollama Status prüfen
- `setup-open-webui.sh` - Open WebUI installieren

Build & Deploy

- `tools/build.sh` - Build Management
 - `tools/fix.sh` - System Fixes
 - `tools/test.sh` - Testing
-

Lessons Learned

1. moOde's Architektur verstehen

- moOde verwaltet viele Dinge über Datenbank, nicht Dateien
- `cfgNetworks()` ist kritisch für Netzwerk-Management
- Worker-Daemon kann Konfigurationen überschreiben

2. Debugging-Strategie

- Root Cause Analysis ist wichtig
- Runtime-Logs sind essentiell
- Systematisches Vorgehen statt Ad-Hoc-Fixes

3. Dokumentation

- Wichtige Erkenntnisse dokumentieren
 - Scripts kommentieren
 - Status-Reports regelmäßig aktualisieren
-

Zusammenfassung

Was funktioniert ■

- Room EQ Wizard (grundsätzlich)
- Build-System
- Testing-Suite
- Lokale KI-Setup (läuft)

Was gefixt wurde ■

- Netzwerk-Problem: Root Cause gefunden und Lösung erstellt
- Dokumentation: Umfassende Guides erstellt

Was noch zu tun ist ■

- Netzwerk-Lösung testen
- Open WebUI abschließen
- RAG Setup
- Agenten erstellen

Letzte Aktualisierung: 6. Januar 2026, 13:00 Uhr

Nächste Review: Nach Pi-Boot und Netzwerk-Test