HTWK

IM Fakultät Informatik und Medien

Dezibot Swarm Logging & Monitoring

Vom Einzelbot-Debugging zur Schwarmplattform

Projektpräsentation, Februar 2026
Projektteam Dezibot: Niclas Jost, Marius Busalt

Fakultät Informatik und Medien
HTWK Leipzig

Gliederung

1. Motivation
2. Grundlegende Architek-tur
3. Funktionen
4. Probleme
5. Ausblick

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Ausgangslage vor diesem Projekt

- Der bestehende Debug-Server zeigte nur die Sensordaten eines einzelnen Dezibots.
- Monitoring mehrerer Bots bedeutete: manuell zwischen mehreren WLANs wechseln.
- Keine zentrale Ubersicht, keine gemeinsame Historie, keine Fernsteuerung.
- Fokus lag auf lokaler Diagnose statt auf koordiniertem Schwarmbetrieb.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Warum wir gestartet sind

- Ein Schwarm aus Lernrobotern braucht eine gemeinsame Sicht auf Zustand und Verhalten.
- In Praktika und Demos muss man schnell sehen: Wer ist online, wer sendet, wer reagiert.
- Wir wollten Diagnostik und Steuerung zusammenfuhren statt getrennte Werkzeuge zu nutzen.
- Das Projekt sollte vor allem robuste Hardware-Kommunikation fur mehrere Bots ermoglichen.

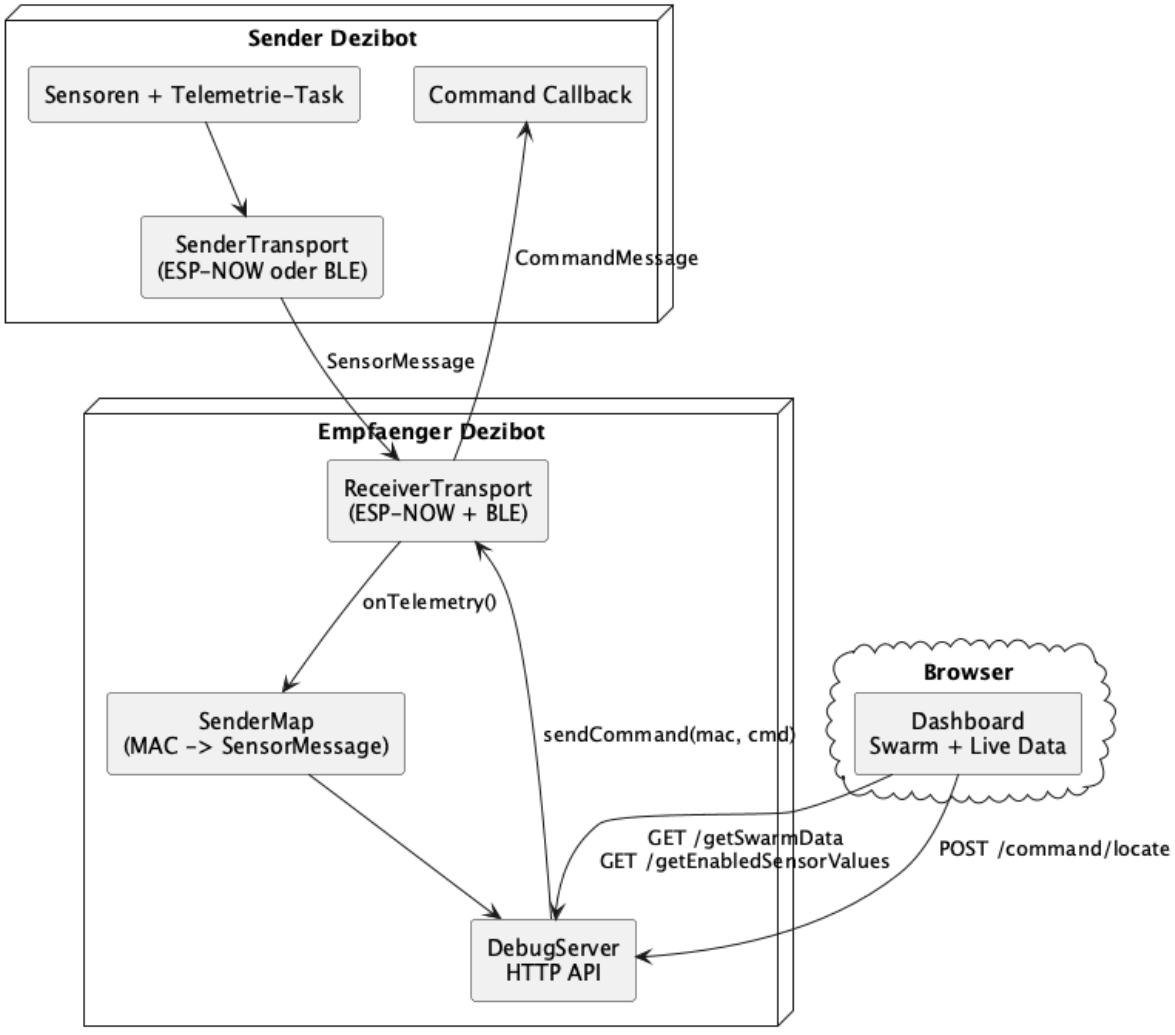
Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Projektziele

- Zentrale Swarm-Ubersicht mit Live-Status aller verbundenen Dezibots.
- Bidirektionale Kommunikation: Telemetrie empfangen und Kommandos zurucksenden.
- Zwei Transportwege unter einer Abstraktion: ESP-NOW und Bluetooth Low Energy.
- Stabile Laufzeit durch entkoppelte Telemetrie im Hintergrund statt blockierender Hauptschleife.

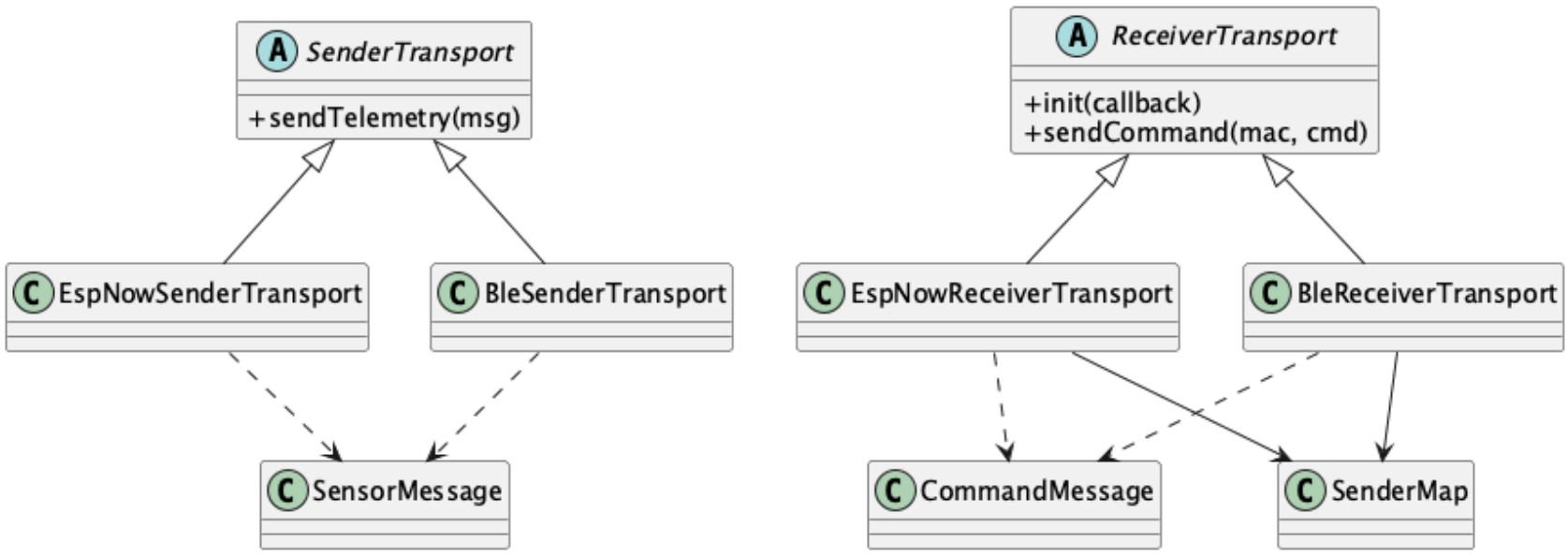
Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

PlantUML Grobarchitektur (Systemkontext)



Projektteam Dezibot: Niclas Jost, Marius Bucalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

PlantUML Klassendiagramm (Transport-Layer)



Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Dev Experience mit PlatformIO

- Einheitliche Build-Umgebung fur Sender und Empfanger als getrennte Environments.
- Reproduzierbarer Ablauf: `pio run`, `upload`, `uploadfs`, `device monitor`.
- Frontend-Artefakte konnen optional in `data/` gebaut und per SPIFFS hochgeladen werden.
- Kernnutzen bleibt die schnelle Iteration an Firmware und Kommunikationslogik.

Build- und Flash-Workflow

```
cd web
npm install
npm run build

pio run -e esp32s3_receiver -t uploadfs
pio run -e esp32s3_receiver -t upload
pio run -e esp32s3_sender -t upload
```

- Ein Build erzeugt sofort testbare Firmware und die passende Weboberflache.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Swarm-Ubersicht im Dashboard

- Pro Bot sichtbar: Online/Offline, MAC, Message Counter, Uptime, Last Seen, Power.
- Online-Status basiert auf `lastSeen` mit Timeout-Logik fur schnelle Diagnose.
- Klick auf einen Bot fuhrt direkt in die Live-Sensoransicht fur dieses Geraet.
- Locate-Action ist in derselben Tabelle integriert und ohne Kontextwechsel nutzbar.

Logging

- Eigene Logging-Seite mit Filterung nach Level (ALL, INFO, WARNING, ERROR, DEBUG, TRACE).
- Polling-basierter Nachlade-Mechanismus fur neue Eintrage in nahezu Echtzeit.
- Ringpuffer im Backend verhindert unkontrolliertes Speicherwachstum.
- Hilft bei Integrationstests zwischen Firmware, Transport und Web-UI.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Charts mit Live-Sensordaten

- Pro Sensorwert ein eigenes Echtzeit-Liniendiagramm (Chart.js im SolidJS-Frontend).
- Anzeige umfasst Sensordaten und Systemmetriken wie Heap, Taskzahl, Chip-Temperatur.
- Zeitfenster ist konfigurierbar (50 bis 2000 Datenpunkte) fur Debugging und Demo.
- Fokus auf Lesbarkeit bei mehreren Bots statt auf historische Langzeitarchivierung.
- Datenexport als CSV/JSON ist als nachster Ausbauschritt in der Roadmap vorgesehen.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

ESP-NOW: Telemetrie und Kommandos

- Sender broadcastet `SensorMessage` im Sekundentakt, Empfanger sammelt zentral.
- Kommandos gehen als Unicast zuruck an ein konkretes Geraet.
- `Magic Numbers` trennen Sensor- und Kommando-Pakete robust.
- Beispielbefehl `CMD_LOCATE`: Bot blinkt 5 mal grun zur schnellen Lokalisierung.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Bluetooth (BLE GATT)

- Sender als Peripheral/GATT-Server, Empfanger als Central/GATT-Client.
- Sensordaten kommen per Notification, Kommandos per Write-Characteristic.
- Ermoglicht gemischte Setups mit ESP-NOW und BLE innerhalb derselben Plattform.
- Besonders nutzlich fur Testszenarien, in denen BLE-Interoperabilitat wichtig ist.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Web-Build-Chain (Randaspekt)

- UI basiert auf SolidJS/Vite und wird als statisches Paket auf den Empfanger gelegt.
- Dieser Teil war nicht das Hauptziel, sondern hat das Monitoring nutzerfreundlicher gemacht.
- Nebeneffekt: JavaScript-Bundle sank von 507441 Bytes auf ca. 376 KB.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Technische Herausforderungen im Betrieb

- BLE-Skalierung: gleichzeitige Verbindungen auf ESP32-S3 sind begrenzt.
- BLE-Onboarding: Scan- und Verbindungsaufbau verursachen spurbare Verzogerung.
- Kanalabhangigkeit bei ESP-NOW: Sender und Empfanger mussen denselben Kanal nutzen.
- Blockierender Locate-Handler kann den Empfangspfad fur einige Sekunden storen.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Qualitat und Grenzen der Messwerte

- Power-Wert ist bewusst eine Schatzung und keine elektrische Messung.
- Chip-Temperatur ist intern und nur begrenzt als Umgebungsindikator nutzbar.
- Thread-Safety in Teilen des Logging-Zugriffs ist als Verbesserungsaufgabe dokumentiert.
- Diese Punkte sind dokumentiert und priorisiert fur die nachsten Iterationen.

Projektteam Dezibot: Niclas Jost, Marius Busalt, Fakultät Informatik und Medien
Hochschule für Technik, Wirtschaft und Kultur Leipzig

Roadmap

- Sensor-Export als CSV/JSON fur Analyse in Python oder Tabellenkalkulation.
- Device Naming fur besser lesbare Bot-Identifikation statt reiner MAC-Adressen.
- Erweiterte Remote-Commands fur Motoren, LEDs und gruppierte Aktionen.
- Alerting bei Schwellwerten, z. B. Heap, Temperatur oder Verbindungsverlust.

Fazit

- Wir haben aus einem Einzelbot-Debugging eine zentrale Schwarmplattform entwickelt.
- Architektur und Toolchain sind so aufgebaut, dass weitere Protokolle und Features anschlussfahig sind.
- Der Mehrwert liegt in der Kombination aus Live-Monitoring, Kommandokanal und reproduzierbarer Dev-Experience.

Fragen?

Quellen und Referenzen

- Projekt-Repository und README:
  `https://github.com/dezibot/dezibot-swarm-logging`
- Dezibot Logging Vorgangerprojekt:
  `https://github.com/Tim-Dietrich/dezibot-logging`
- Dezibot4 Bibliothek: `https://github.com/dezibot/dezibot`
- ESP-NOW Dokumentation (Espressif):
  `https://docs.espressif.com/`
- PlatformIO, SolidJS, Vite, Chart.js Projektdokumentationen