Frontend-Dokumentation

# Überblick

Das Frontend dient als interaktive Benutzeroberfläche für die Systeminteraktion und -überwachung. Es bietet Echtzeit-Visualisierung von Sensordaten, Konfigurationsverwaltung und Vorfallsüberwachung. Durch ein responsives Design ermöglicht es die sofortige Datenanzeige vom Mikrokontroller und die nahtlose Übertragung von Benutzeraktionen an das Backend.

# Technologie-Stack

|  |  |
| --- | --- |
| Komponente | Technologie |
| Framework | Angular 19 |
| Datenvisualisierung | Chart.js |
| Icons | Lucide Icons |
| Backend-Kommunikation | REST API + WebSockets |
| Authentifizierung | JWT Token (über HTTP-Header) |
| Design | Responsive Angular-Komponentenarchitektur |

# Anwendungsarchitektur

Aufbauend auf Angulars komponentenbasierter Architektur folgt das Frontend einem modularen Design, bei dem jeder Funktionsbereich in dedizierten Modulen gekapselt ist. Die Backend-Kommunikation wird durch spezialisierte Services abgewickelt, die sowohl HTTP-Anfragen als auch WebSocket-Verbindungen verwalten.

## Kernmodule

### App-Modul

* Bildet das Grundgerüst der Anwendung einschließlich Routing
* Enthält globale Services und Layout-Komponenten
* Verwaltet Navigationselemente und Seitenleiste

### Authentifizierungsmodul

* Behandelt Benutzeranmeldung und Token-Verwaltung
* Sichert Routen mit Guards ab, um unbefugten Zugriff zu verhindern
* Erhält den Authentifizierungsstatus über die gesamte Anwendung hinweg

### Dashboard-Modul

* Visualisiert Systemdaten durch interaktive Chart.js-Diagramme
* Zeigt Schlüsselmetriken wie Temperaturverläufe und Ereignishäufigkeit an
* Kombiniert REST-Datenabruf mit WebSocket-Updates für Echtzeitanzeige

### Konfigurations-Modul

* Bietet Schnittstellen zur Anzeige und Änderung von Systemeinstellungen
* Speichert Änderungen über REST-Endpunkte
* Implementiert Validierung für Konfigurationsparameter

### Vorfälle-Modul

* Zeigt filterbare Vorfallslisten mit zeit- und typbasierter Filterung
* Implementiert Paginierung für effiziente Datenabrufe
* Bietet Sortier- und Suchfunktionen

## WebSocket-Service

Fungiert als zentrale Schnittstelle für die bidirektionale Backend-Kommunikation:

* Empfängt Live-Datenströme vom Mikrokontroller
* Überträgt Steuerbefehle (z.B. Neustartoperationen)
* Unterhält eine persistente Verbindung während der gesamten Benutzersitzung

# Benutzeroberflächen-Design

* Klare, minimalistische Oberfläche mit Fokus auf Lesbarkeit und Echtzeitinformationen
* Konsistente visuelle Sprache durch Lucide Icons
* Vollständig responsives Layout, optimiert für Desktop und mobile Geräte
* Interaktive, skalierbare Diagramme mit Echtzeit-Aktualisierungsfunktionen

# Echtzeit-Kommunikation

Die beim Anwendungsstart hergestellte WebSocket-Verbindung bleibt während der gesamten Sitzung aktiv und ermöglicht:

* Sofortige Visualisierung von Mikrokontroller-Daten
* Echtzeit-Statusaktualisierungen (Warnungen, Fehler, Alarme)
* Beidseitige Interaktion für Konfigurationsänderungen und manuelle Steuerbefehle ohne Seitenneuladen

# Sicherheitsimplementierung

* Geschützte Ansichten sind nur mit gültiger JWT-Authentifizierung zugänglich
* Token-Verwaltung über LocalStorage mit automatischer Einbindung in Anfragen
* Rollenbasierte Navigationsbeschränkungen (z.B. Konfigurationsänderungen beschränkt auf Admin-Benutzer)
* Automatische Sitzungsverwaltung und Token-Aktualisierungsmechanismen