



# **almond**

**autonomous logging and  
management of networked devices**

**Lastenheft**  
und Projektübersicht

# Inhaltsverzeichnis

<b>1</b>	<b>Übersicht</b>	<b>2</b>
<b>2</b>	<b>Ziele</b>	<b>2</b>
2.1	Endanwender . . . . .	2
2.2	Entwickler . . . . .	2
<b>3</b>	<b>Teilnehmer und Aufgabenverteilung</b>	<b>2</b>
<b>4</b>	<b>Controller</b>	<b>3</b>
4.1	Aufbau . . . . .	3
4.2	Features . . . . .	3
4.3	Display . . . . .	3
4.4	Speicherung (SD-Karte) . . . . .	4
4.4.1	mögliche Erweiterungen . . . . .	4
<b>5</b>	<b>Aktor/Sensor Hardware</b>	<b>4</b>
5.1	Aufbau . . . . .	4
5.2	Anwendungsbeispiel: Wetterstation . . . . .	4
5.2.1	mögliche Erweiterungen der Wetterstation . . . . .	4
<b>6</b>	<b>Protokolle</b>	<b>5</b>
6.1	Uplink . . . . .	5
6.2	Downlink . . . . .	5
6.3	Bluetooth . . . . .	5
<b>7</b>	<b>Backend</b>	<b>5</b>
7.1	Aufbau . . . . .	5
7.2	Backend Bluetooth Interface . . . . .	5
7.2.1	mögliche Erweiterungen . . . . .	6

# 1 Übersicht

Almond (**A**utonomous **L**ogging and **M**anagement of **N**etworked **D**eVICES) ist ein System aus vernetzten Aktoren und Sensoren: Nuts (**N**etworked **U**tilities and **S**ensors), die über einen zentralen Controller (Squirrel) gesteuert werden. Zusätzlich gibt es ein Backend für PCs, welches über eine Weboberfläche Zugriff auf Gerätedaten gibt und die Steuerung der Aktoren sowie Konfigurationsänderungen erlaubt.

Die einzelnen Geräte werden über ein dafür entwickeltes Protokoll via Bluetooth mit dem Controller verbunden. Der Controller selbst wird dann ebenfalls über Bluetooth mit dem auf einem Linux/Windows/Mac laufenden Backend verbunden. Dabei werden auch Logdaten (Verlauf der Sensor-Werte) mit übermittelt, die dann später auf dem Rechner ausgewertet werden können.

## 2 Ziele

### 2.1 Endanwender

Für Endanwender bietet Almond eine einfache Möglichkeit zur Überwachung und Automatisierung von Vorgängen mithilfe eines drahtlosen Netzes aus Geräten.

Dabei ist Almond im Gegensatz zu bestehenden Systemen

- kostengünstig
- wartungsarm
- einfach zu installieren und zu benutzen

### 2.2 Entwickler

Für Entwickler bietet Almond ein Framework um drahtloses Steuern und Messen zu realisieren. Dabei wird auf folgende Kriterien Wert gelegt

- kostengünstig im Vergleich zu bestehenden Systemen
- erweiterbar
- einfache Benutzung
- wartungsarm
- einfache Installation

## 3 Teilnehmer und Aufgabenverteilung

Den 9 Teilnehmern sind folgende Aufgabenbereiche im Projekt zugeteilt:

- **Salomon Sickert:** Teamleiter, Downlink-/Uplink Protokoll Design, Controller

- **Pascal Schnurr:** Backend, Kommunikation mit Webinterface, Bluetooth Interface
- **Matthias Schwab:** Backend, Webinterface
- **Christian Rupprecht:** I/O Interface für Controller (SD Karte)
- **Seán Labastille:** Controller, Up-/Downlink Protokolldesign
- **Stefan Profanter:** Bluetooth Treiber für Aktoren/Sensoren, Backend
- **Linus Lotz:** Hardwarelayout Aktoren/Sensoren, Controller
- **Thomas Parsch:** Displaysteuerung des Controllers
- **Maximilian Karl:** Hardwarelayout Aktoren/Sensoren, Controller

Die einzelnen Projektteile werden im Folgenden genauer erläutert.

## 4 Controller

### 4.1 Aufbau

Der Controller (“Squirrel”) basiert auf einem Atmel Xmega 128a1 AVR Mikroprozessor. Er dient als Zentrale für das System, welches autark vom Rechner arbeitet und auch als Schnittstelle zum Rechner.

Am Controller selbst befindet sich zusätzlich ein SD-Kartenleser für die Speicherung von Logdaten, ein Display mit Tasten für Statusüberprüfung und ggf. Konfigurationsänderungen. Wie alle Client-Geräte (“Nuts”) besitzt der Controller ein Bluetooth Modul.

### 4.2 Features

Mit dem Downlink-Protokoll spricht der Controller die Client-Geräte an und empfängt Daten von diesen. Der Controller führt Buch über die ihm bekannten Geräte und speichert diese intern zwecks Identifizierung, Displayausgabe und Logverwaltung.

Mit dem Uplink-Protokoll kommuniziert der Controller mit dem Backend. Hiermit werden Daten (zum Beispiel zeitlicher Verlauf von Sensorwerten) zum Rechner geladen.

### 4.3 Display

Die grafische Benutzeroberfläche (GUI) des Controllers wird durch einen zweiten Mikroprozessor realisiert.

Das Display wird durch einen Mikroprozessor angesteuert, sodass hier eine zusätzliche Abstraktionsschicht zwischen Display-Controller und Controller eingeführt wird. Dies trennt die GUI Funktion von der Grundfunktion des Controllers.

## 4.4 Speicherung (SD-Karte)

Der Controller besitzt folgende Zugriffsmöglichkeiten auf SD-Karten:

1. SD-Karte mit FAT-32 formatieren
2. Log-Datei schreiben/lesen
3. Konfigurationsdatei schreiben/lesen
4. beliebige Textdatei schreiben/lesen

So ist es möglich auf die Daten auch außerhalb des Controllers, z.B. vom PC aus zugreifen zu können.

### 4.4.1 mögliche Erweiterungen

Eine wünschenswerte Erweiterung wäre die Möglichkeit zeitgesteuert Aktoren auszulösen oder auf Zustandsänderungen bei einem Sensor zu reagieren.

## 5 Aktor/Sensor Hardware

### 5.1 Aufbau

Die Aktoren/Sensoren basieren auf dem Atmel AVR AtMega8535. Sie besitzen ein Bluetooth-Modul zur Kommunikation mit dem Controller.

Ein Designkriterium der Aktoren/Sensoren ist, dass sie einfach gehalten sind und wenig eigene Funktionalität bieten um kosten- und energiesparend zu sein.

### 5.2 Anwendungsbeispiel: Wetterstation

Die Wetterstation basiert auf der Aktor/Sensor Grundplatine mit zusätzlichen folgenden Sensoren:

- Drucksensor (BMP085)
- Temperatursensor (BMP085)
- Windrichtungssensor (Eigenbau, optische Dekodierung)
- Windgeschwindigkeitssensor (Eigenbau, Dynamo)
- Lichtstärke (Fotowiderstand)

Sie besitzt keine Aktoren und ist batteriebetrieben.

#### 5.2.1 mögliche Erweiterungen der Wetterstation

- weitere Sensoren wie Niederschlagssensor/Niederschlagsmenge
- erweiterte Unabhängigkeit der Wetterstation durch Windenergie/Sonnenenergie

## 6 Protokolle

Den Up-/Downlink Protokollen ist gemeinsam, dass Pakete fester Größe ausgetauscht werden.

### 6.1 Uplink

Dient der Kommunikation zwischen Backend und Controller, bei dem Logdaten und Konfiguration ausgetauscht werden.

### 6.2 Downlink

Dient der Kommunikation zwischen Aktor/Sensor Hardware und Controller.

Es werden Eigenschaften und Daten von Aktoren/Sensoren abgerufen (z.B. Sensordaten, Geräteinformation sowie Aktorsteuerung).

### 6.3 Bluetooth

Alle Geräte besitzen ein BTM-222 Bluetooth Modul zur Kommunikation untereinander sowie mit dem Computer (Backend). Zur Verbindung mit dem Computer wird ein bereits existierendes Bluetooth Modul am PC vorausgesetzt.

Sobald der Controller mit einem Aktor/Sensor oder dem PC verbunden ist, wird das Up- bzw. Downlink Protokoll genutzt.

## 7 Backend

### 7.1 Aufbau

Das Backend ist ein Softwarepaket für Linux, Windows oder Mac OS X basierte Computer. Mit dem Backend ist es möglich über eine webbasierte Oberfläche auf die Sensordaten der vernetzten Geräte zuzugreifen.

Der Vorteil der Weboberfläche ist, dass der Backend Server so von jedem Browser aus bedient werden kann (inklusive mobiler Endgeräte) und so eine bestmögliche Integration des Systems ermöglicht wird.

Als Webserver kommt der **lighttpd** zum Einsatz. Zusätzlich wird eine lokale **MySQL** Datenbank zur Speicherung von Einstellungen und Daten auf der Backend-Seite verwendet.

Die Software zur Kommunikation mit dem Backend wird in C++ entwickelt und schreibt empfangene Daten direkt in die Datenbank.

### 7.2 Backend Bluetooth Interface

Das Backend Bluetooth Interface lädt regelmäßig neue Logdaten vom Controller und speichert diese in die MySQL Datenbank.

Außerdem werden geänderten Einstellungen oder Aktionen für Aktoren sofort wieder per Bluetooth in das Netz eingespielt.

**7.2.1 mögliche Erweiterungen**

1. Benutzerverwaltung für das Webinterface
2. Konfigurationseditor im Web
3. evtl. verschlüsselte Kommunikation mit dem Controller