



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Rico Flaegel

**Entwicklung eines drahtlosen Sensornetzwerkes mit mehreren  
Basisstationen für industrielle Anwendungen**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Rico Flaegel

**Entwicklung eines drahtlosen Sensornetzwerkes mit mehreren  
Basisstationen für industrielle Anwendungen**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof Dr. rer. nat. Hans Heinrich Heitmann  
Zweitgutachter: Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 16. November 2012

**Rico Flaegel**

**Thema der Arbeit**

Entwicklung eines drahtlosen Sensornetzwerkes mit mehreren Basisstationen für industrielle Anwendungen

**Stichworte**

Drahtlose Sensornetzwerke, Sensornetzwerk, ZigBee, IEEE 802.15.4, Dijkstra, ATMEGA128RFA1, LPC 1343, AVR, ARM, One Wire, DS18B20

**Kurzzusammenfassung**

Diese Arbeit beschäftigt sich mit der Entwicklung eines Sensornetzwerkes zur Erfassung einer großen Menge von umweltbezogenen Messdaten. Die Datenübertragung wird durch das Sensornetzwerk organisiert und energiesparend ausgeführt. Es werden mehrere Basisstationen unterstützt. Bei einem Ausfall einer Basisstation wird diese automatisch durch eine andere Basisstation ersetzt. Die Zuordnung einer Funkeinheit zu einer Basisstation erfolgt durch eine Kostenfunktion. Diese ist so konzipiert, dass unnötiges Weiterleiten über Geräte vermieden werden und bei beanspruchten Funkteilnehmern Alternativ-Pfade gewählt werden.

**Rico Flaegel**

**Title of the paper**

Development of a wireless sensor network with multiple base stations for industrial applications

**Keywords**

WSN, wireless sensor network, ZigBee, IEEE 802.15.4, Dijkstra, ATMEGA128RFA1, LPC 1343, AVR, ARM, One Wire, DS18B20

**Abstract**

This bachelor thesis deals with mass gathering of data from the environment. For this purpose it will be developed a sensor network, which the data transmission organized and the transmission is executed power saving. It will supports several base stations, in case of failure of a base station an alternative station will take the place of the lost device. The assignment of a wireless sensor to a base station will be calculated by a cost function. This function provides avoiding of unnecessary data forwarding over devices and it chooses alternative routes to bypass sensors with low energy.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Ziel dieser Arbeit . . . . .	1
1.2. Gliederung . . . . .	4
<b>2. Einführung Nahfunktechnologien</b>	<b>5</b>
2.1. Funktechnologien . . . . .	5
2.1.1. WLAN . . . . .	5
2.1.2. Bluetooth . . . . .	6
2.1.3. 802.15.4 . . . . .	7
2.2. Grundlagen Sensornetzwerke . . . . .	8
<b>3. Analyse Nahfunktechnologien</b>	<b>13</b>
3.1. Anforderungen an das System . . . . .	13
3.2. Vergleich Nahfunktechnologien . . . . .	15
3.2.1. Bluetooth 4.0 . . . . .	15
3.2.2. 802.15.4 . . . . .	16
3.2.3. Zigbee . . . . .	18
3.3. Bewertung der Technologien . . . . .	19
<b>4. Entwurf von Entwicklungsumgebung und Funksystem</b>	<b>21</b>
4.1. Planung Entwicklungsprozess . . . . .	21
4.2. Entwicklungsumgebung . . . . .	22
4.2.1. Bestandteile . . . . .	22
4.2.2. Funktionen . . . . .	24
4.2.3. Kommunikation . . . . .	25
4.3. Funkgeräteklassen . . . . .	29
4.4. Übertragung der Nutzdaten . . . . .	29
4.5. Kostenfunktion . . . . .	34
4.6. Routingverfahren . . . . .	36
4.7. Basisstation . . . . .	37
4.7.1. Koordinatorwahl . . . . .	37
4.7.2. Verhalten der Basisstation . . . . .	39
4.7.3. Nachbarsfindung . . . . .	43
4.7.4. Erkundungsvorgang . . . . .	45
4.7.5. Scheduling . . . . .	48
4.7.6. Zeitsynchronisierung . . . . .	50

4.8.	Funkeinheit . . . . .	52
4.8.1.	Verhalten der Funkeinheit . . . . .	52
4.8.2.	Synchronisierung . . . . .	53
4.9.	Gruppenkommunikation . . . . .	54
4.10.	Funkkommunikation . . . . .	56
<b>5.</b>	<b>Realisierung</b>	<b>60</b>
5.1.	Hardwareauswahl . . . . .	60
5.1.1.	Zusammenspiel der Komponenten . . . . .	62
5.2.	Entwicklungsumgebung . . . . .	63
5.2.1.	Bootloader . . . . .	63
5.2.2.	Geräteagent . . . . .	64
5.2.3.	Zentrale Benutzeroberfläche . . . . .	67
5.2.4.	Schnittstellensoftware . . . . .	68
5.3.	Funksystem . . . . .	70
5.3.1.	Softwareentwicklung . . . . .	70
5.3.2.	Sensorik . . . . .	72
5.3.3.	Maximale Anzahl der Knoten . . . . .	72
5.3.4.	Energiebedarf . . . . .	73
<b>6.</b>	<b>Tests</b>	<b>76</b>
6.1.	Energieverbrauch . . . . .	76
6.2.	Zuverlässigkeit . . . . .	78
6.3.	Störeinwirkung . . . . .	78
<b>7.</b>	<b>Weiterentwicklungsmöglichkeiten</b>	<b>80</b>
<b>8.</b>	<b>Zusammenfassung</b>	<b>82</b>
<b>A.</b>	<b>Anhang</b>	<b>84</b>
A.1.	Zustandsautomaten . . . . .	85
A.2.	Inhalt DVD . . . . .	87
	<b>Literaturverzeichnis</b>	<b>88</b>

# Tabellenverzeichnis

4.1.	Nachrichtenaufbau Routing data . . . . .	37
4.2.	Überblick über die Gruppennachrichten . . . . .	55
4.3.	Allgemeiner Aufbau Gruppennachrichten . . . . .	55
4.4.	Aufbau Gefundene Funkeinheiten . . . . .	56
4.5.	Aufbau Synchronisierungsnachricht . . . . .	56
4.6.	Aufbau Superframevorbereitung . . . . .	56
4.7.	Überblick über die Funknachrichten . . . . .	57
4.8.	Headerstruktur . . . . .	57
4.9.	Nachrichtenaufbau Schedule plan . . . . .	58
4.10.	Nachrichtenaufbau Routing data . . . . .	58
4.11.	Nachrichtenaufbau Synchronising . . . . .	58
4.12.	Nachrichtenaufbau Hello Request . . . . .	59
4.13.	Nachrichtenaufbau Hello Response . . . . .	59
4.14.	Nachrichtenaufbau Kindknoteneintrag . . . . .	59
5.1.	Nachrichtenaufbau . . . . .	65
5.2.	Nachrichtentypen . . . . .	66
5.3.	Tabelle Nachrichtenverzögerung . . . . .	69
5.4.	Dynamisch wachsende Datenstrukturen . . . . .	73
5.5.	Stromverbrauch Funkeinheit . . . . .	74
6.1.	Spannungsverlauf während des Tests . . . . .	77

# Abbildungsverzeichnis

1.1.	Beispielanwendung im Freifeld . . . . .	3
2.1.	Piconetz und Scatterntz . . . . .	6
2.2.	Schichten von 802.15.4 und Zigbee . . . . .	7
2.3.	Genutzte Topologien in Sensornetzwerken . . . . .	9
2.4.	Störungsauswirkungen auf Topologien . . . . .	10
2.5.	Kanaleinteilung von 802.11 und 802.15.4 ( <b>NationalInstruments 2012</b> ) . . . . .	11
2.6.	Schematischer Aufbau Funkeinheit und Basisstation . . . . .	12
3.1.	Beispiel für Frequenzsprungverfahren . . . . .	16
3.2.	Aufbau eines 802.15.4 Frames . . . . .	17
4.1.	geplanter Entwicklungsprozess . . . . .	21
4.2.	Zusammenspiel Entwicklungsumgebung . . . . .	23
4.3.	Anwendungsfälle der Entwicklungsumgebung . . . . .	25
4.4.	Sequenzdiagramm zum Logeintrag . . . . .	25
4.5.	Sequenzdiagramm für ein Gerätereset . . . . .	26
4.6.	Sequenzdiagramm für ein Zeitangleich . . . . .	26
4.7.	Sequenzdiagramm zur Identifikation . . . . .	27
4.8.	Sequenzdiagramm zum Softwareupdate . . . . .	28
4.9.	Beispiel für Zeitslots . . . . .	30
4.10.	Superframe . . . . .	31
4.11.	Ungenauigkeiten im Slot . . . . .	31
4.12.	Kosten Empfangsstärke . . . . .	35
4.13.	Kosten Batteriezustand . . . . .	35
4.14.	Beispielverlauf Bully-Algorithmus . . . . .	38
4.15.	Beispielverlauf des angepassten Bully-Algorithmus . . . . .	39
4.16.	Systemverhalten in groben Abschnitten: Teil 1 . . . . .	40
4.17.	Systemverhalten in groben Abschnitten: Teil 2 . . . . .	41
4.18.	Beispiel einer Hidden Station . . . . .	44
4.19.	Beispielfall für das Verschicken von Routinginformationen . . . . .	46
4.20.	Flussplan Funknetzwerkerkundung . . . . .	47
4.21.	Beispielsituation Erkundung . . . . .	48
4.22.	Beispiel Sendeslotzuordnung . . . . .	49
4.23.	Flussplan Scheduling . . . . .	50
4.24.	Beispiel Zeitsynchronisierung . . . . .	52

4.25.	Flussplan Synchronisierung Funkeinheit . . . . .	54
5.1.	deRFmega128-22Ao2 und Board, (vgl. <a href="#">DresdenElektronik 2012</a> ) . . . . .	60
5.2.	XBee DigiMesh 2.4, (vgl. <a href="#">DigiwirelessSolutions 2012</a> ) . . . . .	61
5.3.	LPC1343, (vgl. <a href="#">embeddedArtists 2012</a> ) . . . . .	62
5.4.	Zusammenspiel der Geräte . . . . .	63
5.5.	Flussplan des Geräteagents . . . . .	65
5.6.	Sequenzdiagramm Nachrichtenbehandlung . . . . .	66
5.7.	Zentrale Benutzeroberfläche . . . . .	67
5.8.	Logfenster . . . . .	68
5.9.	Threadkoordinierung . . . . .	69
5.10.	Diagramm Nachrichtenverzögerung . . . . .	70
5.11.	Allgemeines Verhalten Zustandsautomat . . . . .	71
5.12.	Schaltplan One Wire - Parasitär . . . . .	72
5.13.	verschiedene Energiesparmodi während des Betriebs . . . . .	74
6.1.	Testergebnis Störeinwirkung . . . . .	79
A.1.	Zustandsautomat der Funkeinheit . . . . .	85
A.2.	Zustandsautomat der Basisstation . . . . .	86
A.3.	Verzeichnisstruktur DVD . . . . .	87

# 1. Einleitung

Heizkörper geben ihren Verbrauch drahtlos weiter. In Gewächshäusern werden Feuchtigkeitssensoren genutzt, um optimales Wachstum zu ermöglichen. Maschinen werden mit Vibrationsensoren ausgestattet, um kleinste Fehler in der Mechanik festzustellen. Diese Aufgabe kann mit Sensornetzwerken gelöst werden. Ein Sensornetzwerk ist per Definition ein langlebiges System, welches aus einer Vielzahl von dicht beieinander positionierten Sensorknoten besteht und im Betrieb physikalische Größen misst (vgl. [Lan u. a. 2008](#), S. 579-582). Ein solches System ermöglicht die unterschiedlichsten Szenarien, sowohl im privaten Bereich (z.B. Hausautomatisierung), als auch im Industriebereich. Die aufgefangenen Daten der Sensorknoten können dann z.B. visuell dargestellt werden oder liefern für gekoppelte Systeme hilfreiche Informationen.

Sensorknoten in einem drahtlosen Sensornetzwerk werden in der Regel durch eine oder mehrere Batterien gespeist. Je nach Anwendungsgebiet kann es durchaus passieren, dass die Batterien bereits nach einer kurzen Betriebszeit aufgebraucht sind. Ein Forschungsgebiet, dass sich mit dem Entziehen von Energie aus der Umwelt beschäftigt, nennt sich „Energie Harvesting“. So wird zum Beispiel die Energie einer befahrener Straße genutzt, indem die Vibrationen aufgefangen und in elektrische Spannung umgewandelt wird (vgl. [Bogue 2009](#), S. 15).

Eine weitere Technik für eine möglichst lange Betriebszeit ist das „Schlafenlegen“ des Mikrocontrollers. In diesem Schlafzustand sind so gut wie fast alle Hardware-Bausteine abgeschaltet. Damit verbraucht der Mikrocontroller nur einen Bruchteil seiner vorhandenen Energie als im normalen Betriebszustand. Dieser Schlafzustand steht direkt in Beziehung zur Betriebsdauer, so würde die Betriebszeit eines Gerätes in dauerhafter Betriebsbereitschaft bei heutigen Batterien recht kurz ausfallen (vgl. [Sonavane u. a. 2008](#), S. 3).

## 1.1. Ziel dieser Arbeit

Die Zielsetzung dieser Arbeit ist der Entwurf und die Entwicklung eines drahtlosen Sensornetzwerkes. Auf folgende Eigenschaften wird besonders Wert gelegt: Zum einen sollen mehrere

## 1. Einleitung

---

Basisstationen an der Kommunikation beteiligt sein und je nach Situation an Funkeinheiten für die Datenübertragung gebunden sein. Dies bietet z.B. gerade bei vielen Netzteilnehmern den Vorteil, dass der Kommunikationsaufwand nicht zu einer Senke hin immer stärker wird. So verteilt sich der Datenfluss in mehrere Richtungen. Energie bei Funkeinheiten kann so gespart werden, da diese dann weniger Daten weiterleiten muss.

In einem Sensornetzwerk gibt es zwei Geräteklassen: Basisstationen und Funkeinheiten. Basisstationen organisieren das Netzwerk und leiten Messwerte an anderen Systemen weiter bzw. speichern diese. Funkeinheiten sind zur Aufnahme von Messdaten mit Sensoren ausgestattet. Die aufgenommenen Messdaten werden direkt oder indirekt an eine Basisstation übermittelt. Funkeinheiten sind in der Regel batteriebetrieben, Basisstationen haben eine dauerhafte Spannungsversorgung.

Bei Ausfall einer Basisstation übernimmt eine andere Basisstation dessen Funkeinheiten, so dass die Messwerte der Funkeinheiten trotz Ausfall weiterhin gesendet werden können. Dabei spielt die Art der aufgenommenen Messwerte keine Rolle. Es wird darauf geachtet, dass die Funkeinheiten genügend Anbindungsmöglichkeiten zu Sensoren bieten.

Eine Beispielanwendung ist in [Abbildung 1.1](#) zu sehen. In dem Beispiel werden Daten im Freifeld gesammelt. Die Basisstationen sind jeweils am Rand positioniert und sind durch ein zusätzliches Medium miteinander verbunden. Die Auswertung der Daten erfolgt in einem Gebäude.

Hier könnte z.B. Temperatur oder Feuchtigkeit gemessen werden. Die optimalen Kommunikationsstrecken zwischen den einzelnen Funkteilnehmern ist dabei zur Vereinfachung bereits festgelegt worden. In einer realen Anwendung hätten das Netzwerk einen höheren Verma- schungsgrad vorzuweisen.

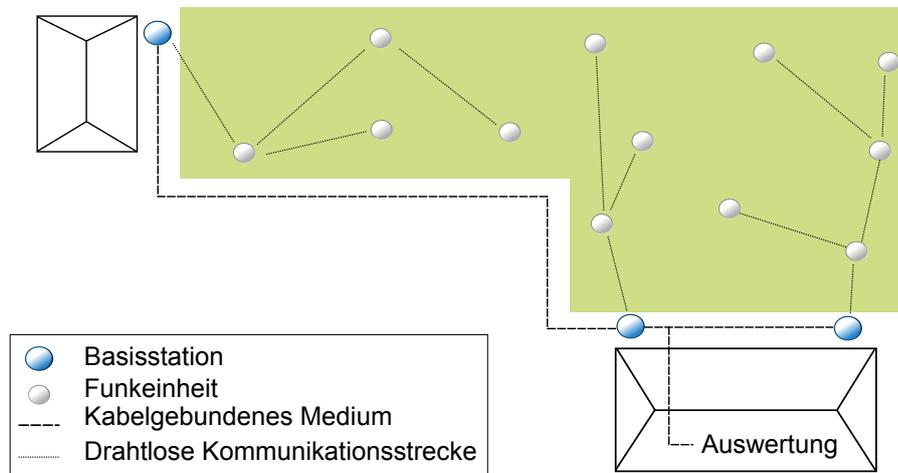


Abbildung 1.1.: Beispielanwendung im Freifeld

Eine grundsätzliche Eigenschaft von Sensornetzwerken ist deren Betriebsdauer. Eine Funkeinheit ist nur so lange einsatzfähig wie eine ausreichende Energieversorgung durch Batterien oder Akkus gewährleistet ist. Deshalb muss bereits beim Entwurf des Systems konsequent auf das Energiesparen geachtet werden.

Für die Netzwerkkommunikation wird eine möglichst optimale Wegeauswahl zwischen Basisstation und Funkeinheit gewählt. Optimal heißt in diesem Zusammenhang das Vermeiden von zusätzlichen Datenweiterleitungen über Funkeinheiten und der schonende Umgang mit der Energie. Es sollen dabei Geräte mit wenig Energieversorgung geschont werden, indem ein Alternativ-Weg um sie herum führt.

In der Entwicklungsphase dieses Systems sind mehrere Geräte involviert. Software wird erstellt und anschließend auf die Geräte übertragen. Dies muss im Normalfall stets mit einem Zusatzgerät erfolgen. Jedes Gerät muss dafür an das Zusatzgerät angeschlossen werden und die Software wird aufgespielt. So ein Vorgang ist in der Entwicklung zeitlich sehr hinderlich.

Jedes Gerät mehr im System kostet Zeit, da zum Testen immer die aktuelle Software benötigt wird. Ein Hilffsystem, wie eine zentrale Softwareverteilung, ist daher absolut notwendig. Daher wird zusätzlich zum Funksystem eine Entwicklungsumgebung eingeplant.

## 1.2. Gliederung

Die Hauptgliederung erstreckt sich über 6 Teilbereiche. Im ersten Teil **Einführung Nahfunktechnologien** werden die grundlegenden Begriffe und Technologien vermittelt. Darauf aufbauend wird im Abschnitt **Analyse Nahfunktechnologien** der Entwurf des Systems spezifiziert und die heutigen Nahfunktechnologien verglichen. Im Hauptteil, dem **Entwurf von Entwicklungsumgebung und Funksystem**, wird das Funksystem und benötigte Hilfsanwendungen konzipiert. Darin enthalten sind Verhaltensabläufe der Geräte, die Kommunikation untereinander sowie Strategien zum nachhaltigen Umgang mit der zur Verfügung stehenden Energie.

Im Schritt **Realisierung** wird das System dann konkret umgesetzt. Es werden Entscheidungen zur Hardware getroffen. Das Design wird in Software umgesetzt und die Geräte anschließend in das System eingebunden. Anschließend werden einige **Tests** zur Einschätzung der Qualität, Energieverbrauch und Robustheit gegenüber Störungen durchgeführt. In der **Zusammenfassung** werden schließlich alle Ergebnisse dieser Arbeit kompakt dargestellt.

## 2. Einführung Nahfunktechnologien

### 2.1. Funktechnologien

In unserem heutigen Alltag kommen wir sehr oft mit Funktechnologien in Berührung. Zu Hause sind viele am Computer drahtlos mit dem Internet verbunden. In Einkaufszentren sind Produkte mit Nahfunk-Chips versehen, um diese vor Diebstahl zu schützen. Mit Freunden und Kollegen können Daten von Handy zu Handy übertragen werden. Und wie im vorherigen Kapitel erwähnt kommunizieren Sensornetze zum Großteil mit Funktechnologie. Da die Funktechnologie die Basis jedes drahtlosen Sensornetzes sind, werden in diesem Kapitel oft genutzte Technologien vorgestellt.

#### 2.1.1. WLAN

WLAN ist die Abkürzung von Wireless Local Area Network und gehört zu der IEEE 802.11-Familie. Es ist ein Funknetzwerk, das räumlich lokal (z.B. in Gebäuden) beschränkt ist. Es findet in Privathaushalten sowie in Unternehmen Anwendung. Die genutzten Frequenzen liegen im 2,4GHz bzw. in den neueren Standards im 5GHz Bereich.

Für Betreiber solcher WLAN sind die Reichweite, Bandbreite und Sicherheit in solchen Netzwerken von besonderer Bedeutung. Für die Verschlüsselung in WLAN-Netzen muss eine hinreichend starke Verschlüsselungsmethode gewählt werden, aktuell wird der Verschlüsselungsstandard WPA2 vom BSI empfohlen (vgl. [BSI 2009](#)). Ansonsten besteht die Gefahr, dass private bzw. vertrauliche Daten ausgespäht werden können. Ältere Verschlüsselungsverfahren sollten daher nicht eingesetzt werden.

Gerade für größere Datenmengen ist eine hohe Bandbreite wichtig, so unterstützen die Protokollfamilie im Laufe der Standardisierung immer mehr Bandbreite, angefangen im 802.11 mit 2Mbit/s bis zum letzten Entwurf 802.11ac sind mehrere Gbit/s geplant. Eine möglichst große Funkabdeckung ist von vielen WLAN-Betreibern gewünscht. Eine Möglichkeit ist es mehrere

## 2. Einführung Nahfunktechnologien

---

Zugriffspunkte für das Funknetzwerk aufzustellen. Zusätzlich kann die Technologie MIMO (Multiple Input Multiple Output) genutzt werden, die bereits in vielen Geräten verbaut ist. Damit lassen sich nicht nur weitere Strecken überbrücken und höhere Geschwindigkeiten erreichen, sondern auch die Fehlerhäufigkeit reduzieren.

### 2.1.2. Bluetooth

Bluetooth ist ein Standard gemäß IEEE 802.15.1, der Datenaustausch über kurze Strecken ermöglicht. Durch Neuentwicklungen wurde der Versionsstand alle paar Jahre inkrementiert und liegt aktuell bei Version vier. Der genutzte Frequenzbereich liegt wie WLAN im 2,4GHz-Bereich. Je nach gewählter Betriebsart (Profil) können bis zu 1Mbit/s in BR (Basic Rate) erreicht werden, in EDR bis zu 3Mbit/s(Enhanced Data Rate). Die Reichweite ist in der vierten Version bis zu 100m spezifiziert Bei der letzten Versionsaktualisierung wurde vor allem auf Energie sparende Aspekte geachtet und bietet dazu das Profil „low energy“ an. Heutzutage haben viele technischen Geräte so eine Schnittstelle. Notebooks, Mobiltelefone, Kameras, Drucker, Freisprecheinrichtungen und viele andere Geräte können mittels Bluetooth kommunizieren.

Dabei sind nicht nur Verbindungen zwischen zwei Geräten möglich. In so genannten Piconetzen können bis zu 255 Geräte existieren, wobei bis zu 8 Geräten gleichzeitig aktiv sein können (vgl. [Osterhage 2010](#), S. 133-148). Für die Datenübertragung wird ein gemeinsamer Kanal genutzt. Ein Piconetz erfordert immer einen Master, es können also bis zu 7 Slaves aktiv sein. Ein Slave kann auch parallel Master eines anderen Piconetzes sein. So eine Verbindung zweier Netze wird dann Scatternetz genannt. Ein Beispiel dazu ist in [Abbildung 2.1](#) gezeigt.

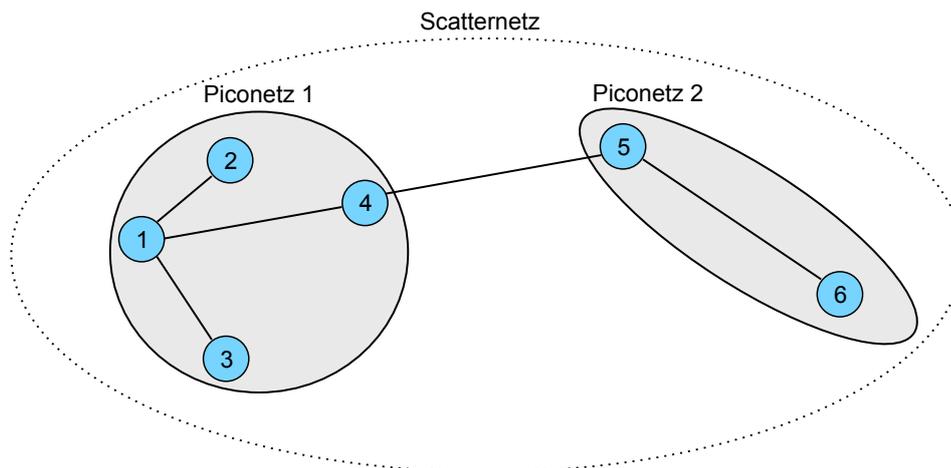


Abbildung 2.1.: Piconetz und Scatterntz

## 2. Einführung Nahfunktechnologien

---

Im Profil low energy gab es einige Änderungen im Vergleich zu den anderen Profilen BR und EDR: „unlike BR/EDR the slaves do not share a common physical channel with the master. Each slave communicates on a separate physical channel with the master.“ (BluetoothSIG 2010, S. 181). Es gibt also für jede Verbindung zum Master einen eigenen Kanal, in den beiden anderen Betriebsarten teilen sich die Slaves und der Master den Kanal.

### 2.1.3. 802.15.4

Dieser Standard von IEEE wird auch sehr oft ZigBee genannt, da dieser auf den Standard 802.15.4 aufsetzt. ZigBee selbst entstand aus dem Gedanken ein herstellerübergreifendes Protokoll zu entwickeln. Die dazugehörige Organisation, die ZigBee Alliance, zählt bereits mehr als 200 Unternehmen. Jedoch kann 802.15.4 nicht nur für ZigBee genutzt werden, da dieser Standard bis zur MAC-Schicht spezifiziert ist. Darauf aufbauend können viele Protokolle entwickelt werden. In Abbildung 2.2 ist der Schichtenaufbau gezeigt. Im Folgenden werden einige Eckdaten zu dem Standard 802.15.4 genannt.

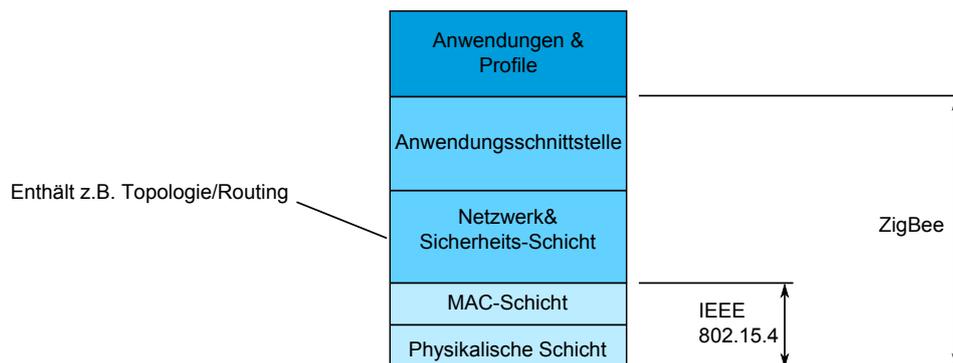


Abbildung 2.2.: Schichten von 802.15.4 und ZigBee

802.15.4 nutzt mehrere Frequenzbänder. Es kann im 800MHz-Bereich (Europa), den 900MHz-Bereich (USA), also auch wie Bluetooth und WLAN das 2,4GHz-Band nutzen. Damit sind also Kollisionen mit diesen Funktechnologien möglich. Die Übertragungsgeschwindigkeit beträgt 250kbit/s. Eigene Anwendungen können wie bei anderen Standards der Familie direkt auf der MAC-Ebene aufsetzen und eigene Kommunikationsprotokolle entwickeln. Routingalgorithmen müssen also in der höheren Schicht selbst implementiert werden. Die Topologie für das Netzwerk ist demnach nicht festgelegt und es sind voll vermaschte Netzwerke möglich.

## 2. Einführung Nahfunktechnologien

---

Es sind zwei Arten von Netzwerkteilnehmer im Standard 802.15.4 vorgesehen. Ein RFD (Reduced Function Device) ist ein Gerät mit weniger Funktionalitäten, da dieser einfacher zu entwickeln ist. Ein FFD (Full Function Device) ist ein Gerät mit vollen Funktionsumfang und kann auch als Router eingesetzt werden, ein RFD ist dazu nicht in der Lage. Diese Klassifizierung erlaubt bei einem RFD die Kosten für ein Gerät zu senken, da dieser weniger System- und Hardwareressourcen benötigt. Üblicherweise ist ein FFD nach Standard mit einer externen Spannungsquelle verbunden, ein RFD nutzt eigene Energiereserven. Um die Energiereserven zu schonen ist dieser Standard speziell auf geringen Energieverbrauch ausgelegt. Für zusätzliche Robustheit beim Senden wird CSMA/CA eingesetzt. Dabei wird die Funkschnittstelle erst auf bereits laufende Übertragungen überprüft, bevor die anstehenden Daten verschickt werden.

Zigbee kann bereits für viele Anwendungen genutzt werden. Es gibt ein spezielles Gerät der FFD-Klasse, der Koordinator. Dieser besitzt oft eine Schnittstelle zu anderen Netzwerken und ist fest als Koordinator konfiguriert. Er verwaltet das Netzwerk, stellt Informationen bereit und ist für den Aufbau des Netzes zuständig. Für den Fehlerfall sollte ein zweites Gerät als Koordinator ausgewählt sein, denn sonst kann ein Funknetzwerk nicht weiter verwaltet werden bzw. das Netzwerk kann nicht aufgebaut werden. So genannte Router(FFD) dienen als Kommunikationspartner für die RFD's und leiten optional Daten auch über weitere Router bis zum Koordinator weiter.

### 2.2. Grundlagen Sensornetzwerke

Eine kurze Erläuterung zu Sensornetzwerke gab es bereits in der [Einleitung](#). Dieser Abschnitt soll einen tieferen Einblick in den Bereich geben. Es werden die grundlegenden Begriffe und Probleme beschrieben, die es bei Sensornetzwerken zu bewältigen gilt.

Das Netzwerk besteht aus den Funkeinheiten sowie einer oder mehreren Basisstationen, die physikalisch miteinander verbunden sind. Die Struktur der Verbindungen wird Topologie genannt. Aus der Topologie ergeben sich verschiedene Vor- und Nachteile. Es werden hierbei aber nicht alle Topologien abgehandelt, sondern nur die relevanten für Sensornetzwerke. [Abbildung 2.3](#) zeigt hierbei einen kleinen Überblick über die genutzten Topologien.

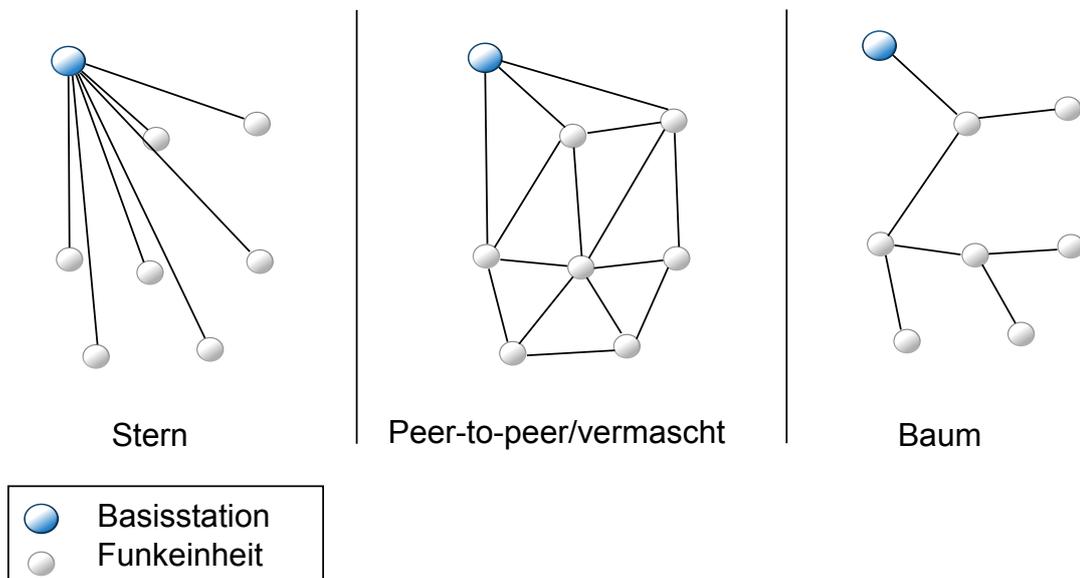


Abbildung 2.3.: Genutzte Topologien in Sensornetzwerken

Die Stern-Topologie ist eine simple Verbindung zwischen den Geräten, da sie einfach zu implementieren ist und Fehler schnell auffindbar sind. Nachteilig ist aber die direkte Verbindung, es gibt keine weiteren Wege zu einer Funkeinheit.

Bei der Peer-to-peer-Topologie bzw. auch vermaschtes Netzwerk genannt, können mehrere Wege zwischen den Teilnehmern existieren. Durch die vielen Verbindungspunkte können auch Wege ausfallen, ohne dass es zu einer Nichterreichbarkeit einer Funkeinheit kommt. Die Organisation der Datenübertragung ist im Vergleich zu z.B. der Stern-Topologie komplexer. Da es mehrere Wege zu einem Ziel gibt, muss ein Routingalgorithmus verwendet werden.

In der Baum-Topologie ist bereits eine gewisse Ordnung vorhanden und es gibt von einem Blattknoten aus nur einen Weg zur Wurzel. Ein spezieller Algorithmus ist dafür nicht erforderlich. Ein Ausfall eines Blattknotens hat keine Auswirkung auf andere Geräte. Findet der Ausfall mitten in der Baumstruktur statt, so können die Knoten, die unter dem ausgefallenen Knoten liegen, keine Daten mehr zur Wurzel schicken. Ein Ausfall ist in [Abbildung 2.4](#) dargestellt.

## 2. Einführung Nahfunktechnologien

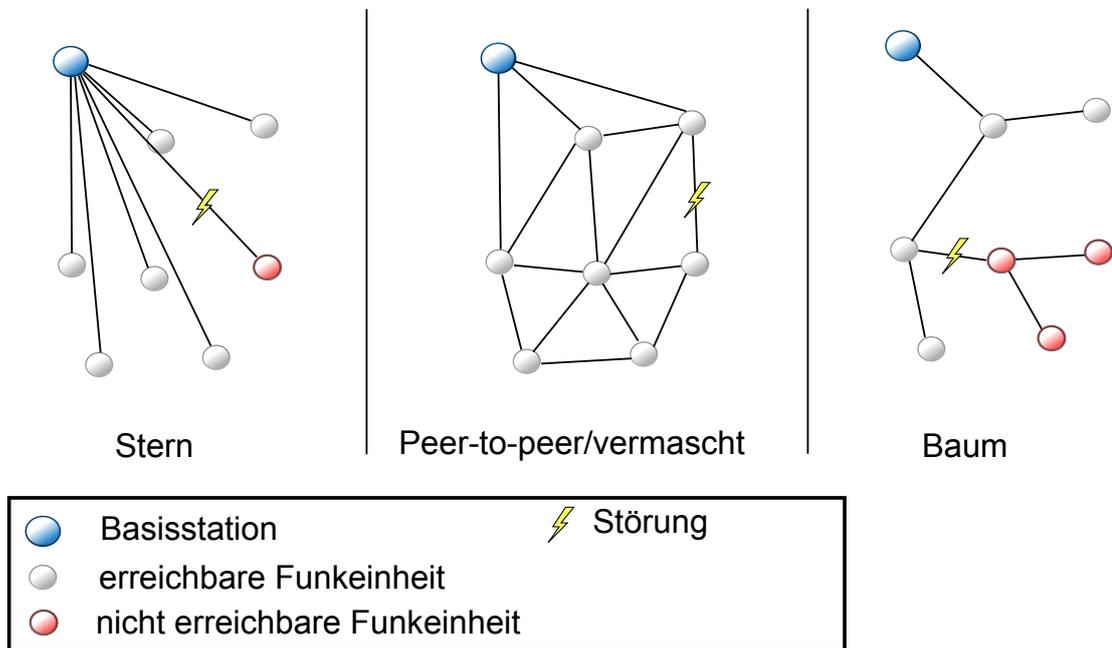


Abbildung 2.4.: Störungsauswirkungen auf Topologien

Wie im Abschnitt [Einführung Nahfunktechnologien](#) erklärt, nutzt eine Funktechnologie einen Frequenzbereich um Daten zu übertragen. Je nach Land sind bestimmte Frequenzbereiche durch Behörden freigegeben oder dürfen nicht benutzt werden. Alle vorgestellten Technologien nutzen das ISM-Band. Die Abkürzung steht für Industrial Scientific Medical Band. Es war ursprünglich für industrielle, medizinische und wissenschaftliche Zwecke gedacht, das aber heutzutage auch für allgemeine Datenübertragung genutzt wird. Das Frequenzband kann von jedem ohne Genehmigung genutzt werden, es gibt aber Regelungen zur Sendeleistung und Störungsvermeidung von benachbarten Frequenzbereichen. Diesen Frequenzbereich kann international genutzt werden (vgl. [ITU 2007](#)).

Aus dem Medium Luft ergeben sich Besonderheiten bei der Kommunikation. Da das Medium von allen Netzwerkteilnehmern genutzt wird, muss die vorhandene Bandbreite aufgeteilt werden. Das ist durch Multiplexing-Verfahren möglich. Diese Verfahren ermöglichen die wirtschaftliche Nutzung der Funkressourcen. In drahtlosen Netzwerken wie z.B. WLAN, Bluetooth und 802.15.4 wird das Frequenzmultiplexverfahren eingesetzt. Dabei wird das vorhandene Frequenzspektrum in Bändern unterteilt, jedes Band besitzt dabei eine eigene Trägerfrequenz. Die zu übertragenden Informationen werden zum Versand dann auf die jeweilige Trägerfrequenz moduliert (vgl. [Schmidt 2011](#), Seite 27). So wird z.B. der Bereich im 2,4GHz-Band in Kanäle

## 2. Einführung Nahfunktechnologien

---

eingeteilt, die dann parallel betrieben werden können. Abbildung 2.5 zeigt die Einteilung des Bandes von 802.11 und 802.15.4.

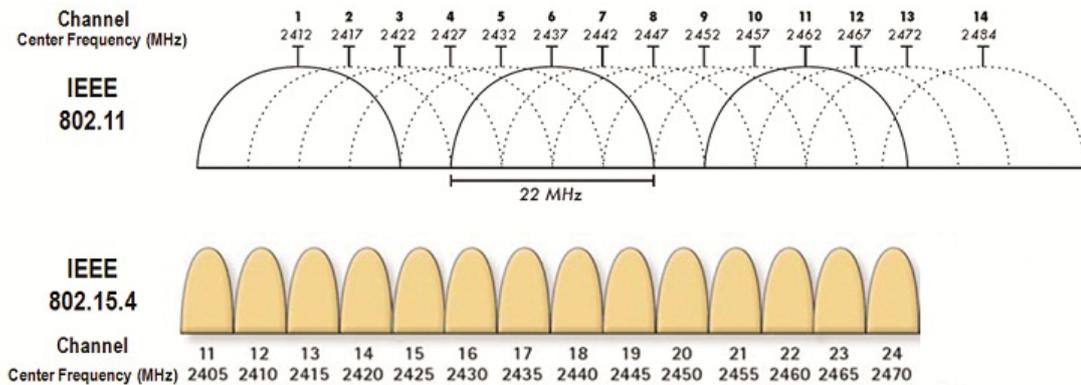


Abbildung 2.5.: Kanaleinteilung von 802.11 und 802.15.4 (NationalInstruments 2012)

Jedoch muss nun noch sichergestellt werden, dass innerhalb des Kanals die Daten organisiert übertragen werden. So können die Funkeinheiten nicht einfach unkontrolliert Daten auf dem Kanal verschicken. Wenn mehrere Sender zur gleichen Zeit Daten verschicken, so können die Nachrichten nicht mehr bei dem Empfänger auseinandergelassen werden. Es muss also noch ein Zugriffsverfahren genutzt werden.

Eine Möglichkeit ist CSMA/CA, eine Abkürzung für Carrier Sense Multiple Access/Collision Avoidance. Es handelt sich um ein Verfahren, das Kollisionen vermeidet, indem ein Netzwerkteilnehmer das Medium auf laufende Datenübertragungen lauscht und bei einem besetzten Medium auf seine eigene Datenübertragung wartet. Im Wartezustand läuft ein Back-Off-Timer, der bei Ablauf das Medium erneut prüft. Bei wiederholenden Warten kann sich die Zeit des Back-Off-Timers erhöhen. Die genauen Zeiten sind technologie- und hardwareabhängig.

Ist der Kanal frei, so kann er seine Daten senden. Bei diesem Verfahren stehen die Netzwerkteilnehmer im Wettbewerb und es muss Zeit für das Lauschen aufgewendet werden. Bei einer konkurrenzlosen Variante wird für eine bestimmte Zeit einem Teilnehmer der Kanal zugesichert. Er kann also ohne zusätzliches Lauschen seine Daten verschicken. Dies setzt aber eine Organisation der Sendezeitpunkte voraus, was im Vergleich zum konkurrierenden Senden ein Nachteil ist.

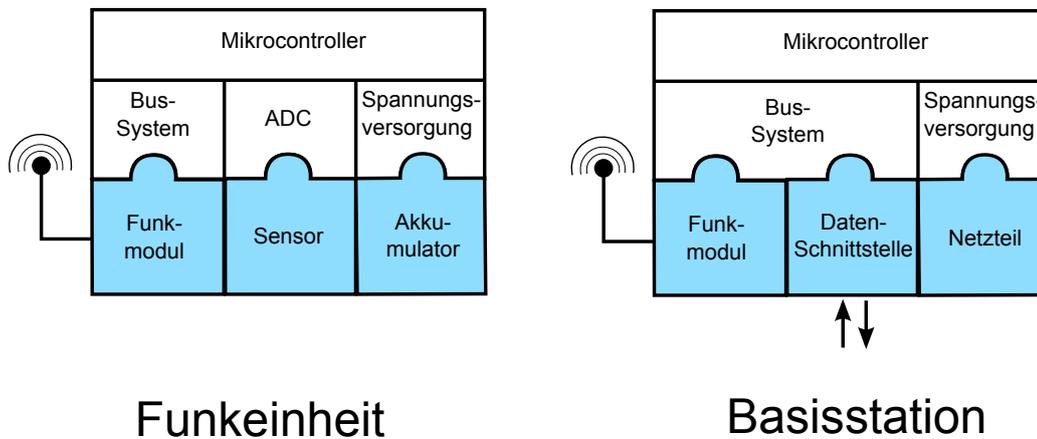


Abbildung 2.6.: Schematischer Aufbau Funkeinheit und Basisstation

Eine Funkeinheit misst Daten, speichert diese und verschickt sie anschließend zur Basisstation. Eine Funkeinheit muss also ein Sensor für die physikalische Größe haben, sowie die passende Schnittstelle zum Sensor vorweisen. Der Mikrocontroller ist dabei die Basis und nutzt für die Kommunikation ein Funkmodul. Eine Basisstation hat prinzipiell den gleichen Aufbau, besitzt aber zusätzlich eine weitere Schnittstelle für die Kommunikation zu anderen Basisstationen bzw. zur Datenweitergabe an anderen Systemen. In [Abbildung 2.6](#) sind beide dargestellt. Die Funkeinheit hat in diesem Beispiel eine Anbindung an einem Sensor via ADC (Analog Digital Converter). Hierbei steht die Spannungshöhe in einer Relation zum Messwert und kann so berechnet werden. Die Basisstation hat eine Datenschnittstelle, die für die Gruppenkommunikation und Datenweitergabe gedacht ist.

## 3. Analyse Nahfunktechnologien

### 3.1. Anforderungen an das System

Im vorigem Kapitel wurden die Ziele für das funkbasierte Sensornetzwerk beschrieben. Diese werden im Folgenden in Anforderungen formuliert, um eine klare Übersicht der zu erfüllenden Funktionalitäten zu erhalten. Zusätzlich gibt es zu jedem Punkt eine kurze Erläuterung.

- Für die reine Nutzdatenübertragung wird jeder Funkeinheit eine gewisse Datenmenge während der Phase der Messdatenübertragung zugesichert. Mindestens aber kann eine Funkeinheit in dieser Phase 10 Bytes senden
  - Die Nutzdatenmenge richtet sich grundlegend an die verwendete Hardware und die Anzahl der Funkeinheiten im Netzwerk. Die Speicherkapazität der Funkeinheiten ist begrenzt und es kann nur eine bestimmte Datenmenge von anderen Funkteilnehmern zwischengespeichert werden. Bei der Teilnehmeranzahl im Netzwerk gibt es ab einer Größenordnung Synchronisierungsprobleme. Je mehr Zeit seit der letzten Synchronisierung vergangen ist, desto höher wird die Ungenauigkeit einer Uhr in jeder Funkeinheit. Vergeht zu viel Zeit besteht die Gefahr, dass zwei Funkeinheiten sich bei einer Übertragung verpassen
- Das Funksystem unterstützt mehrere Basisstationen, mindestens eine ist für den Betrieb erforderlich
  - Eines der wichtigsten Ziele ist es, den Einsatz mehrerer Basisstationen zu gewährleisten. Deshalb verfügen die Basisstationen über ein eigenes Übertragungsmedium für die Gruppenkommunikation. Sollte eine Basisstation nicht mehr erreichbar sein, so übernimmt eine andere Basisstation dessen Platz (sofern mehrere Basisstationen vorhanden sind)
- Die Wegeauswahl erfolgt nach einer Kostenfunktion
  - Diese Kostenfunktion beurteilt eine Verbindung zu einer entfernten Funkeinheit anhand Empfangsstärke und Betriebsinformationen zu einer Funkeinheit. Eine

### 3. Analyse Nahfunktechnologien

---

Funkeinheit wird während des Betriebes z.B. seine Systemlaufzeit und/oder die Anzahl verschickter und empfangener Nachrichten aufsummieren und bei Anfrage weiterreichen. Das Ziel ist es, anhand dieser Informationen Hops zu Vermeiden. Hops sind Zwischenstationen einer Nachricht zum Ziel. So ist eine Funkeinheit ein Hop, da dieser Nachrichten weiterleitet. Damit sollen Hops bei zu dicht beieinander stehenden Funkeinheiten vermieden werden. Auch werden Alternativ-Routen gewählt, wenn die Energiereserven der Funkeinheiten bereits durch hohen Kommunikationsaufwand (Datenweiterleitungen) strapaziert werden

- Die Betriebszeiten der Funkeinheiten soll möglichst lang sein, mindestens ein halbes Jahr soll eine Funkeinheit ohne Batteriewechsel auskommen
  - Im Normalfall befindet sich der Mikrocontroller einer Funkeinheit in einer Hauptschleife und wartet bis ein Ereignis eintritt. Für diesen Wartezustand benötigt der Mikrocontroller genauso so viel Energie, wie für die Ausführung von Berechnungen. Damit Energie gespart werden kann, werden Energiesparmodi genutzt. Wie und in welchem Umfang Energie gespart wird, richtet sich nach der ausgewählten Hardware
- Das Funksystem muss auf Veränderungen und Störungen reagieren
  - Während des Betriebes können verschiedene Ereignisse auftreten. Funkeinheiten kommen zum System dazu, werden entfernt oder werden an einem anderem Ort platziert. Es muss also in gewissen Abständen das Netzwerk neu erkundet werden. Auch können Basisstationen hinzukommen oder das Netzwerk verlassen. Es gilt hier auch, dass Basisstationen in Abständen neu erfasst werden. Da die Luftschnittstelle für die Datenübertragung genutzt wird, muss mit Störungen gerechnet werden. Diese Störungen könnten z.B. durch andere in der Nähe platzierte Geräte verursachen, die im gleichen Frequenzbereich Daten übertragen
- Für den Entwicklungsprozess wird eine Anwendung entwickelt, mit dessen Hilfe Funkeinheiten und Basisstationen aus der Ferne steuerbar sind und Log-Informationen der Geräte aufnimmt
  - Gerade bei Entwicklung von Systemen, die viele Geräte beinhalten, entsteht ein hoher Aufwand die Software einzeln auf die Geräte zu übertragen und diese anschließend zu testen. Deshalb ist so ein Hilfsprogramm unerlässlich. Die Geräte können so in kürzester Zeit mit neuer Software versorgt werden. Zusätzlicher Komfort entsteht durch Geräteneustart und die Möglichkeit das Gerät komplett

### 3. Analyse Nahfunktechnologien

---

anzuhalten. Durch eine Loggingfunktion können Geräte Nachrichten verschicken, so kann unter anderen der Protokollablauf mitverfolgt werden. Diese Anforderung ist unabhängig von der gewählten Funktechnologie

## 3.2. Vergleich Nahfunktechnologien

Die Anforderungen an das System sind nun bekannt. Jetzt werden die verschiedenen Nahfunktechnologien miteinander verglichen. Diejenige Technologie, die alle Anforderungen erfüllt, wird für das Projekt genutzt. Da für die Auswahl nur energiesparende Technologien in Frage kommen, wird das im Vorfeld angesprochene WLAN nicht in die engere Auswahl genommen.

### 3.2.1. Bluetooth 4.0

Da in Bluetooth 4.0 vor allem Energiesparmechanismen hinzukamen, ist dieser Standard für drahtlose Sensornetzwerke interessant geworden. In vorherigen Versionen ist dies nicht der Fall. Ein Verbindungsaufbau dauerte bis zu einigen Sekunden. Aktuell wird nun einer Verbindung binnen weniger Millisekunden aufgebaut. Anschließend kann der Teilnehmer bis zu 27 Bytes senden und dann, je nach Protokoll, sich in einem Tiefschlaf begeben.

Der Aufbau eines Netzwerkes über mehrere Funkeinheiten hinweg gestaltet sich als schwierig, da Bluetooth in seinem Piconetz normalerweise nur Punkt-zu-Punkt bzw. Punkt-zu-Mehrpunkt vorgesehen ist. Auch kann ein Master nur 7 Slaves verwalten. In einer Situation mit mehr als 7 Slaves muss ein Teil der Slaves in den passiven Zustand überführt werden. Dies ist recht umständlich und kostet zusätzlichen Kommunikationsaufwand.

Der Betrieb mit mehreren Basisstationen ist möglich. Mehrere Piconetze können zu einem Scatternetz zusammengeschlossen werden. Allerdings müssen Datenweiterleitungen über mehrere Bluetooth-Teilnehmer hinweg angepasst werden, da eine vermaschtes Netzwerk nicht vorgesehen ist. Laut Anforderungen soll die Wegeauswahl selbst bestimmt werden, dafür ist wie bei der Datenweiterleitung eine zusätzlicher Anpassung im Design bzw. in der Realisierung nötig.

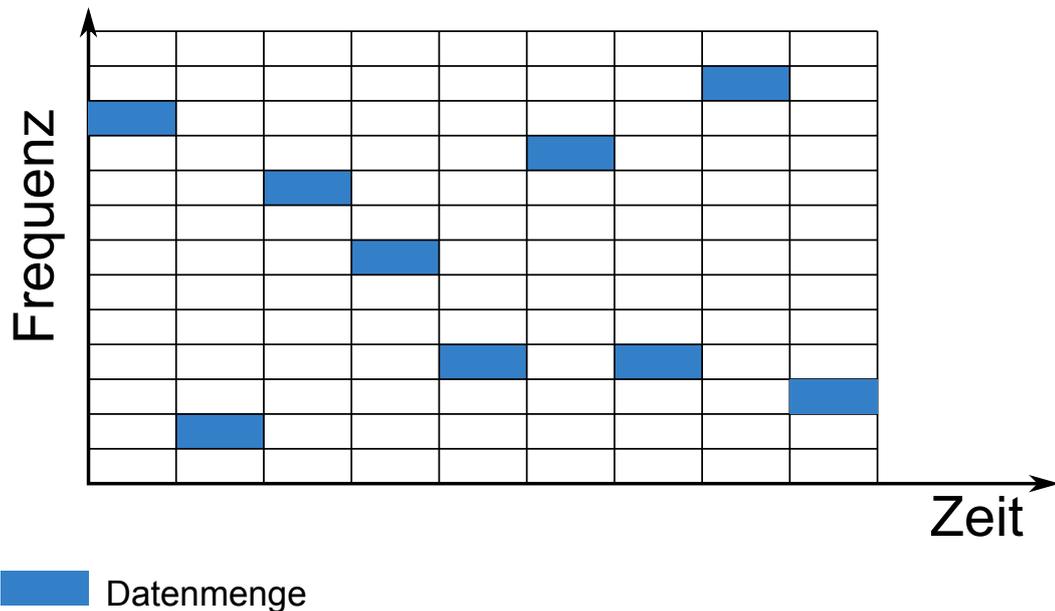


Abbildung 3.1.: Beispiel für Frequenzsprungverfahren

Störungen werden bei Bluetooth durch das Frequenzsprungverfahren FHSS (Frequency Hopping Spread Spectrum) minimiert. „FHSS verwendet 79 Kanäle innerhalb des 2,4-GHz-ISM-Bandes von denen jeder 1 MHz breit ist. Ein Generator für Pseudozufallszahlen erzeugt die Folge von Frequenzen, auf die gewechselt wird. Verwenden alle Geräte den gleichen Startparameter und sind die gesamte Zeit über synchronisiert, wechseln die Geräte gleichzeitig auf die Frequenzen. Jede Frequenz ist nur für eine gewisse Zeitspanne aktiv [...] .Daneben bietet es auch ein Minimum an Sicherheit, da ein Eindringling nicht weiß, zu welcher Frequenz gewechselt wird und auch nicht die Zeitspanne kennt, für die eine Frequenz gültig ist. Das Abhören einer Verbindung wird dadurch erschwert“ (Holtkamp 2003, Seite 4). Eine Informationsfolge bei Bluetooth wird also über mehrerer Kanäle hinweg „verstreut“ um Kollisionen zu entgehen. Abbildung 3.1 veranschaulicht dieses Verfahren. Wenn eine Kollision auftritt, so ist nur ein kleiner Teil der Übertragung fehlerhaft und wird nach der Kollisionserkennung erneut gesendet.

### 3.2.2. 802.15.4

Wie bei dem Bluetooth 4.0-Standard wurde beim Design von 802.15.4 auf das Energiesparen geachtet. Funkmodule können kurz vor Sendebeginn aus dem Tiefschlaf geholt werden, senden Daten und versetzen sich danach wieder in einem Energiesparmodus. Es sind also je nach

### 3. Analyse Nahfunktechnologien

verwendeten Kommunikationsdesign sehr langlebige Sensornetzwerke technisch realisierbar. Da dieser Standard direkt auf der MAC-Schicht aufbaut, kann ein Verbindungsaufbau oberhalb dieser Schicht selbst definiert werden, also z.B. die Daten ohne Rückmeldung zum Teilnehmer direkt schicken oder es wird mit Quittierungsmeldungen kommuniziert.

Die maximale Größe eines Frames beträgt 133 Bytes, davon sind 127 Bytes der MAC-Schicht und 6 Bytes sind der physikalischen Schicht zugeordnet. Von den 127 Bytes werden bis zu 25 Bytes für Adresszuordnung, Frameeigenschaften und Fehlerüberprüfung genutzt. Einsparungen sind bei der Adressierung möglich, hier stehen MAC-Adressen mit 8 Bytes Länge oder Kurzadressen mit 2 Bytes zur Verfügung. Wenn zur Adressierung 2 Bytes genutzt werden stehen demnach  $2^{16}$  Adressen zur Verfügung. Dies ist absolut ausreichend für industrielle Aufgaben, bei dem die Teilnehmeranzahl wesentlich kleiner ist als die zu Verfügung stehenden Adressen. Für die eigene Kommunikation stehen im schlechtesten Fall 102 Nutzbytes zur Verfügung, bzw. im besten Fall 118 Bytes. In der Abbildung 3.2 ist der Frameaufbau dargestellt.

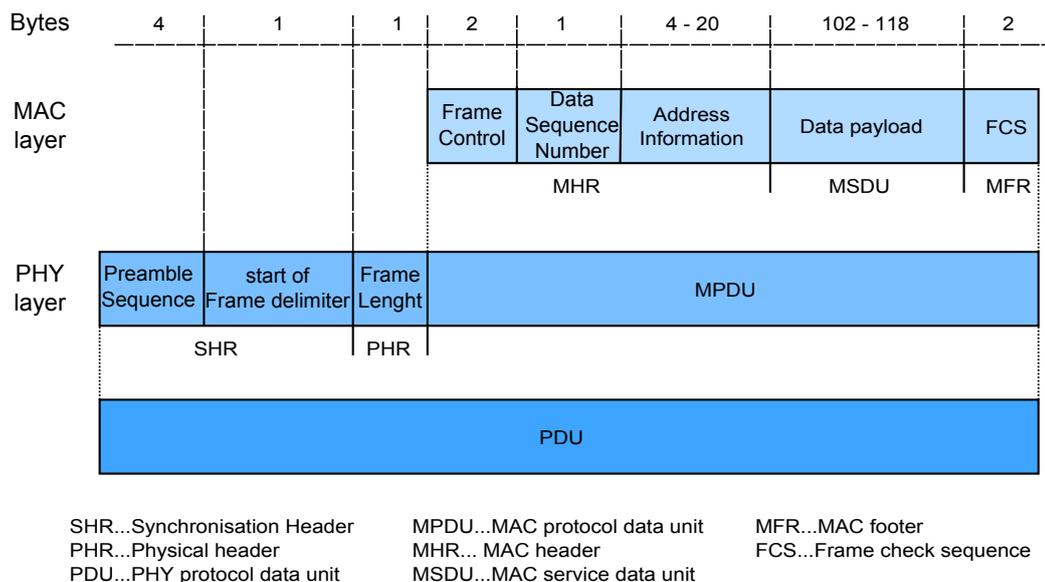


Abbildung 3.2.: Aufbau eines 802.15.4 Frames  
 (vgl. Kupris und Sikora 2007, S. 66, mit wesentlichen Änderungen)

Wie bei dem Verbindungsaufbau ist die Topologie nicht Bestandteil der Spezifikation und kann frei nach Anwendungsfall gewählt werden. Es sind damit verschiedene Szenarien möglich.

### 3. Analyse Nahfunktechnologien

---

Funkeinheiten können sich bei einer Basisstation registrieren oder die Teilnehmer erkunden selbstständig das Netzwerk. Dadurch ist eine eigene Wegeauswahl für ein möglichst optimalen Pfad möglich.

Mehrere Basisstationen sind technisch realisierbar, weil es keine Einschränkung zur Geräteklasse innerhalb eines Netzwerkes gibt. Wie bereits im Kapitel [Einführung Nahfunktechnologien](#) beschrieben, gibt es Geräte mit vollen Funktionsumfang, bzw. Geräte mit Funktionseinschränkungen. Diese dienen aber eher der Kostenreduktion, um Endgeräte einfach zu halten.

Störungen während der Kommunikation werden nicht durch spezielle Technologien vermieden. Das Frequenzband ist in Kanälen aufgeteilt, auf dem parallel gesendet werden kann. Sollte es zu Kollisionen kommen, müssen höhere Schichten für einen weiteren Sendeversuch sorgen bzw. das Gerät als nicht erreichbar einstufen (da das Gerät z.B. zu weit entfernt ist).

#### 3.2.3. Zigbee

Zigbee baut auf 802.15.4 auf und hat damit alle Vor- und Nachteile, die dieser Standard bietet. Alle Funktionen beginnen ab Schicht 3, für die darunter liegende Schichten ist 802.15.4 zuständig. Nach einer Anmeldung einer Funkeinheit an dem Netzwerkkoordinator oder einem Router können Nutzdaten übertragen werden. Vorgesehen sind dabei das direkte Senden mit Mitbewerbern oder das Senden innerhalb eines zugeteilten Zeitschlitzes. Das Senden mit Zeitschlitz ist aber nur in Beacon-Enabled Netzwerken möglich (vgl. [Sikora 2004](#), Seite 18). Beacon-Enabled bedeutet die Nutzung von „Leuchtfener“ für die Signalisierung. Erhält eine Funkeinheit (RFD, Reduced Function Device) ein solches Leuchtfener, kann sich die Funkeinheit in einem Energiesparmodi begeben, bis sein zugeteilter Zeitraum für das Senden erreicht ist.

Ein großes Energiesparpotenzial ist vor allem beim Senden in Zeitschlitz möglich. Neben der Kommunikation zur Organisation der Sendezeitpunkte und die Nutzdatenübertragung befindet sich das Endgerät im Tiefschlaf. Eine lange Betriebsdauer ist so möglich. Geräte in ZigBee-Netzwerken bieten Dienste in so genannten Endpunkte an. Dabei kann ein Geräte bis zu 240 Endpunkte besitzen. Der Endpunkt Null hat eine spezielle Aufgabe. Er beinhaltet das Gerätemanagement, Sicherheitsfunktionen, Management für Routerzuordnung und das Discoverymanagement (vgl. [Kupris und Sikora 2007](#), Seite 89).

Nutzdaten werden in der höheren Schicht „application support sublayer“ generiert bzw. verarbeitet. Dem Datenframe wurde dafür eine extra Header hinzugefügt und enthält zusätzliche Angaben wie die Endpunktnummer der Quelle und die des Ziels (vgl. [Kupris und Sikora 2007](#),

Seite 99). Dadurch wird die maximale Größe der Nutzdaten etwas eingeschränkt.

Die Wegeauswahl erfolgt anhand eines Routingverfahrens. Hierbei wird AODV genutzt. AODV ist die Abkürzung für Ad-hoc On-Demand Distance Vector. Dabei werden Routen nur generiert, wenn eine Nachricht an ein Ziel verschickt werden soll, wobei die genaue Route noch nicht bekannt ist. „Wenn eine Nachricht von einem Knoten zu einem anderen Knoten gesendet werden soll, wird zuerst nachgesehen, ob sich der Zielknoten in der unmittelbaren Nachbarschaft befindet, also ob ein Eintrag in der List der Nachbarknoten(Neighbor-Table) vorhanden ist.

Wenn das nicht der Fall ist, dann wird es mit vermaschten Routing versucht, es wird nachgesehen, ob sich ein Eintrag über den Zielknoten in der Weiterleitung-Tabelle(Routing-Table) befindet. Wenn das auch nicht der Fall ist, dann wird als Ausweidlösung auf das Routing entlang der Baumstruktur zurückgegriffen“ (Kupris und Sikora 2007, Seite 98). Eigene Wege können demnach nicht gewählt werden, sondern es wird dem Routingverfahren AODV überlassen. Das Verfahren nimmt dabei die Strecke mit den wenigsten Zwischenschritten zwischen Start- und Endknoten.

ZigBee unterstützt als Netzwerkstruktur die Baum-Topologie sowie ein vermaschtes Netz. Bei der Baum-Topologie wird streng entlang der Baumhierarchie geroutet, eine Routingverfahren müsste hier nicht verwendet werden. Im vermaschten Netz kommt das voran erklärte Routingverfahren AODV zum Einsatz. Ein großer Nachteil von ZigBee ist die Begrenzung auf einer Basisstation bzw. Koordinator. Ein anderes FFD kann aber als Ersatzkoordinator dienen, falls der Koordinator ausfallen sollte.

Zur Störungsvermeidung wird am Anfang der Netzwerkaufbauphase zuerst alle Kanäle auf andere Funkteilnehmer gelauscht. Abschließend fällt die Entscheidung auf dem Kanal mit dem geringsten Störungen. Eine spätere Änderung des Kanals ist nicht vorgesehen.

### **3.3. Bewertung der Technologien**

Da nun die verschiedenen Technologien anhand der Anforderungen des Funksystems verglichen wurden, werden die Ergebnissen kompakt dargestellt und es erfolgt anschließend eine Bewertung. Die Anforderung für mindestens 10 Bytes pro Funkeinheit während der Nutzdatenübertragung wird von allen Technologien unterstützt und es wird daher nicht näher darauf

### 3. Analyse Nahfunktechnologien

---

eingegangen.

Bluetooth erfüllt durch die neue Version 4.0 die Anforderungen für lange Betriebszeiten der Funkeinheiten. Es werden auch Störungen erkannt und es erfolgen Sendewiederholungen. Allerdings ist eine dynamische Wegewahl mit einer Kostenfunktion nur schwer realisierbar. Auch ist ein vermaschtes Netzwerk nicht vorgesehen und eine beliebige Anordnung der Geräte ist nicht immer möglich.

802.15.4 unterstützt von Haus aus bereits energiesparende Eigenschaften. Funkeinheiten können in einem langen Zeitraum Daten erfassen und senden. Je nach gewählten Protokolldesign können aufgetretene Fehlübertragungen durch ein wiederholtes Senden ausgeglichen werden. Auf dem Standard aufbauend ist eine eigene Wegewahl möglich. Auch ist der Betrieb von mehreren Basisstationen machbar.

Zigbee besitzt einen Protokollstapel, der bis zur Anwendungsschicht reicht. Dabei sind alle Vorgänge zur Verwaltung und zum Betrieb des Sensornetzwerkes wie z.B. die Pfadauswahl durch das Routingverfahren AODV vorgegeben. Es sind für eigene Anpassungen engere Grenzen gesetzt. Auch ist ein expliziter Betrieb mit mehr als einer Basisstation nicht möglich. Eine Fehlererkennung wird jedoch standardmäßig durch Quittieren des ankommenden Datenpakets sichergestellt.

Es ist also der Standard 802.15.4, der alle Anforderungen erfüllt und damit als Technologiegrundlage des Sensornetzwerkes genutzt wird.

## 4. Entwurf von Entwicklungsumgebung und Funksystem

Dieses Kapitel widmet sich dem Design des Funksystems. Zusätzlich wird laut Anforderung dieses Projektes ein Hilfssystem einwickelt, mit dem die Funkeinheiten während des Entwicklungsprozesses aus der Ferne steuerbar sind. Das Testen neuer Software wird somit erheblich vereinfacht und das Funksystem kann schneller in der Realisierungsphase entstehen.

### 4.1. Planung Entwicklungsprozess

Im Folgenden werden die Schritte erklärt, wie der Umgang mit dem Hilfssystem zur zentralen Softwareverteilung erfolgen soll. Dabei wird wiederholend die Software angepasst und getestet. In Abbildung 4.1 ist der Prozess dargestellt. Zur Bezeichnung wird die Hardware, die zur Entwicklungsumgebung gehört, als Geräteagent-Hardware benannt. Die Software, die auf der Geräteagent-Hardware läuft, wird Geräteagent genannt.

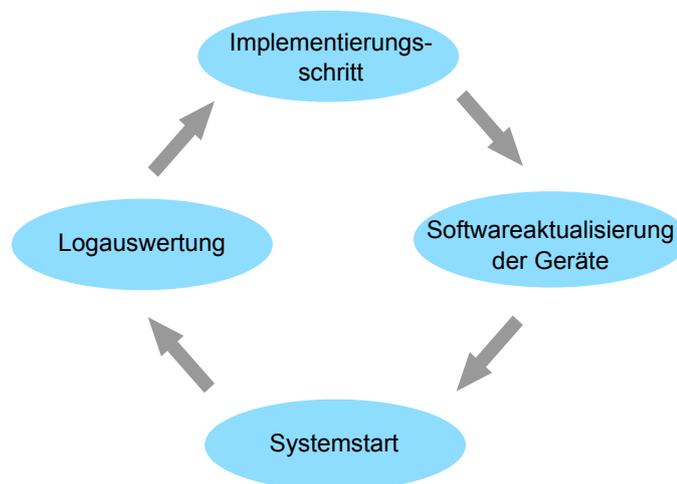


Abbildung 4.1.: geplanter Entwicklungsprozess

## 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Die Software wird schrittweise entwickelt. Eine zu testendes Zwischenergebnis wird auf die Geräte übertragen. Das System wird dann im Ganzen gestartet. Während des Testbetriebes schicken die Geräte Logeinträge, die den Systemstatus, Fehler und Ereignisse beinhalten.

Diese Einträge werden zentral gesammelt, so dass Ereignisse zeitlich eingeordnet werden können. Damit sollen Fehler erkannt werden. Beispielsweise verschickt ein Gerät ein Frame an ein anderes Gerät, das jedoch kein Empfangsereignis auslöst. Die Software wird nach einer Logauswertung angepasst und wieder auf die Geräte gespielt, das System kann wieder getestet werden.

### 4.2. Entwicklungsumgebung

In diesem Abschnitt werden die einzelnen Komponenten der Entwicklungsumgebung näher erläutert. Danach folgt eine Auflistung der geplanten Funktionalitäten und das Kommunikationverhalten der Komponenten untereinander. Die Funkeinheit und die Basisstation werden hier einheitlich als Target bezeichnet.

#### 4.2.1. Bestandteile

Die Entwicklungsumgebung beinhaltet 4 Komponenten: Die zentrale Benutzeroberfläche, eine Schnittstellensoftware, ein Softwarebestandteil im Target (nimmt Daten entgegen und schreibt diese Daten ins Flash) und Hardware, die die Schnittstellensoftware mit dem Target verbindet (Geräteagent). Die zentrale Benutzeroberfläche ist die Schnittstelle zwischen Mensch und Steuerung. Es werden Befehle entgegengenommen und weitergeleitet. Die Ausführung geschieht dann in der nachfolgenden Verarbeitung. Die Schnittstellensoftware wird auf der Maschine ausgeführt, an der die Geräteagent-Hardware und dahinter die Funkeinheit bzw. Basisstation angeschlossen ist.

Die Geräteagent-Hardware wird benötigt, um das dahinter liegende Target direkt zu steuern. Diese Aufgaben sind z.B. das Gerät zurückzusetzen oder es in ein Zustand zu bringen, in welche neue Software aufgespielt werden kann. Ein Softwaremodul im Target sorgt für das Speichern der ankommenden Daten in den Flashspeicher.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

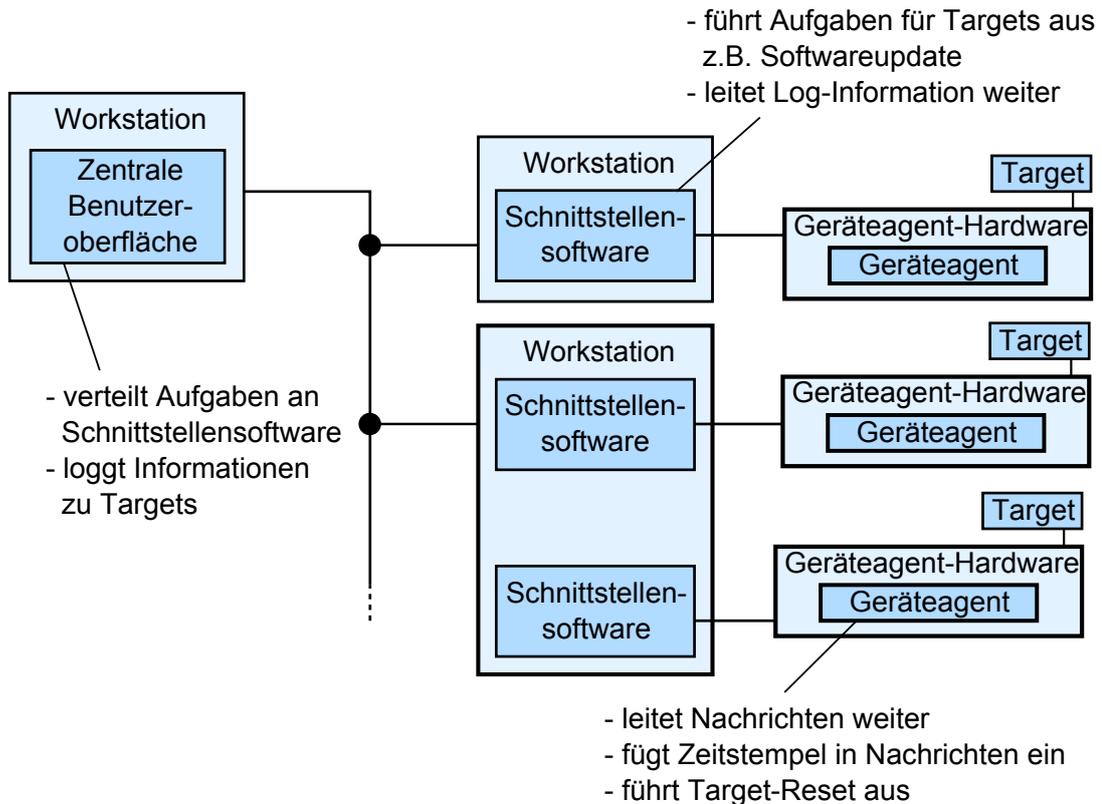


Abbildung 4.2.: Zusammenspiel Entwicklungsumgebung

Das Zusammenspiel der Komponenten ist in Abbildung 4.2 veranschaulicht. Auf einer Workstation läuft die zentrale Benutzeroberfläche. Die Workstations sind per Netzwerk miteinander verbunden, auf anderen Rechnern läuft jeweils pro Target eine Instanz der Schnittstellensoftware.

Mit dieser Verteilung der Aufgaben können an verschiedenen Orten die Hardware angebracht werden. Eine Steuerung erfolgt zentral über ein Benutzerinterface. Somit lassen sich Testszenarien entwickeln, bei denen ein größerer Abstand zwischen den Geräten existiert. Realistische Testergebnisse sind somit möglich.

### 4.2.2. Funktionen

In diesem Abschnitt werden die Funktionen der Entwicklungsumgebung (zentrale Benutzeroberfläche + Schnittstellensoftware + Geräteagent + Softwaremodul im Target) konkret erläutert.

- Aufspielen neuer Software auf die Targets: Hierbei kann die erstellte Software direkt auf das Target übertragen werden. Als Eingabeparameter wird die Binärdatei und das Target genutzt. Als Zusatzkomfort werden Geräte gleicher Klasse mit derselben Software bespielt. Das geschieht parallel, um noch mehr Zeit zu sparen. In Hardware wird dies mit einem Bootloader realisiert. Der Bootloader hat die Aufgabe ein neues Hauptprogramm in den Flashspeicher zu schreiben
- Geräte werden in den Urzustand versetzt: Dazu wird die Geräteagent-Hardware verwendet. Der Befehl wird von der zentralen Benutzeroberfläche über die Schnittstellensoftware geschickt und wird in der Geräteagent-Hardware ausgewertet. Von dort aus wird beim Target ein Reset ausgelöst. Eine zusätzliche Verdrahtung an das Target ist vonnöten. Eingabeparameter ist das Target
- Identifikationsdaten werden gesetzt: Um den Komfort der parallelen Softwareaufspielung zu realisieren, müssen Geräte z.B. einer Geräteklasse zugeordnet werden. Auch ist ein Namensfeld für die Logausgabe vorgesehen. Falls ein Hersteller aus einem Grund keine MAC-Adresse in das Gerät gebrannt hat, wird dieses Identitätsmerkmal auch hinzugefügt. Eingabeparameter sind die Identifikationsmerkmale Name, MAC-Adresse, Gerätetyp und Zielgerät. Dafür wird der Bootloader mit verwendet. Es ist somit immer möglich, diese Eigenschaften zu setzen, unabhängig von der aktuell aufgespielten Software
- Logdateien beziehen: Informationen werden zur Fehlersuche gebraucht. Deshalb gibt es eine Funktion Logdateien für ein Gerät darzustellen oder alle Logdateien in einer Ausgabe zusammenzufassen. Ein Target verschickt als Eingabeparameter die Lognachricht
- Zeitangleich vornehmen: Für die Entwicklung eines funkbasierten Protokolls ist das Einhalten von Zeiten sehr wichtig. Um die Uhrenungenauigkeiten auszugleichen, werden die Uhren der Geräte in definierten Abständen mit der Workstation des Anwenders synchronisiert

Diese Funktionen wurden in der Abbildung 4.3 als Anwendungsfall zusammengefasst.

## 4. Entwurf von Entwicklungsumgebung und Funksystem

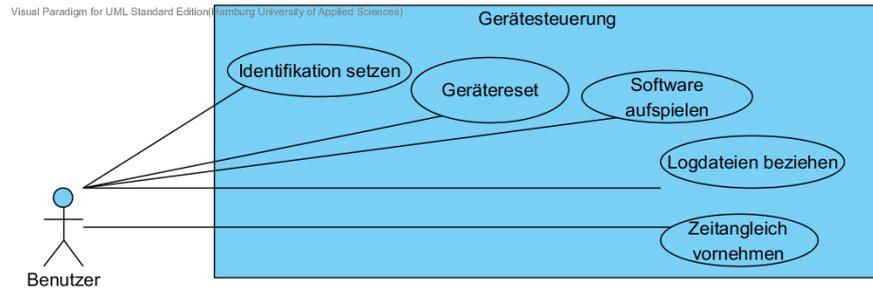


Abbildung 4.3.: Anwendungsfälle der Entwicklungsumgebung

### 4.2.3. Kommunikation

Für die Kommunikation wurden Sequenzdiagramme erstellt. Diese zeigen die nun die einzelnen Bestandteile der Entwicklungsumgebung und den Nachrichtenverlauf für verschiedene Aufgaben.

- Logeintrag: Dieses Ereignis wird bei dem Target ausgelöst. Der generierte Eintrag wird zum Schnittstellenprogramm übertragen. Wenn eine Verbindung zur zentralen Benutzeroberfläche existiert, wird die Nachricht weiter verschickt. Wenn nicht, wird die Nachricht verworfen. Dargestellt ist dies in [Abbildung 4.4](#)

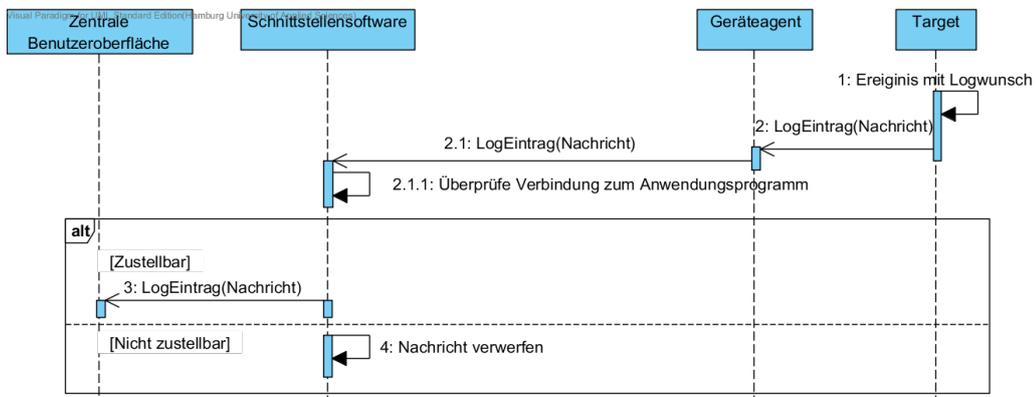


Abbildung 4.4.: Sequenzdiagramm zum Logeintrag

- Gerätereset: Das Absetzen des Befehls erfolgt per Unicast, da nicht immer für alle Geräte ein Reset benötigt wird. Der Befehl wird bis zum Geräteagent übertragen. Durch eine direkte Verdrahtung zum Target, wird dieses nun resettet. Das Target gibt eine Rückmeldung nach einem Reset. Erfolgt die Rückmeldung nicht, ist der Reset nicht sicher

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

durchgeführt worden. Jeder Reset führt zuerst in den Bootloader. Erfolgt innerhalb eines kurzen Zeitraums keine weiteren Anweisungen, so wird das Hauptprogramm gestartet. Das Sequenzdiagramm ist in 4.5 zu sehen

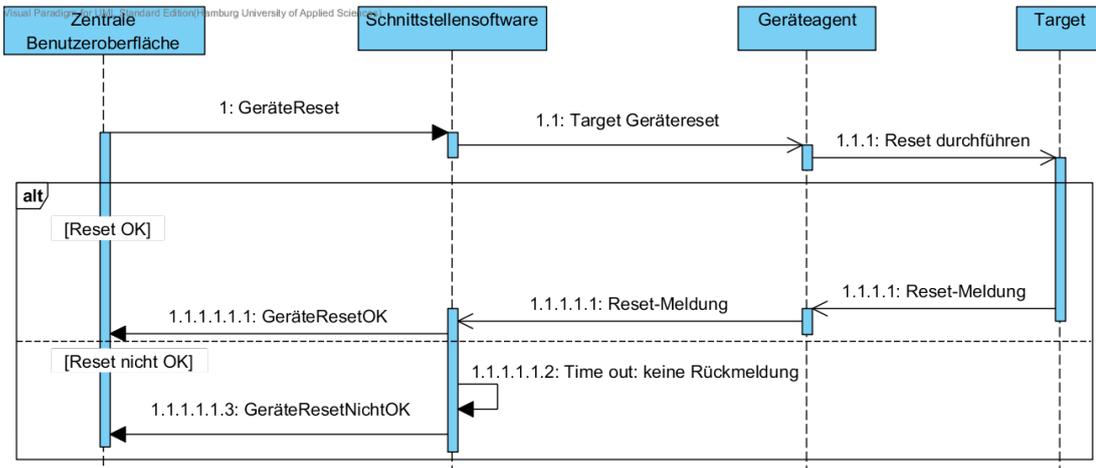


Abbildung 4.5.: Sequenzdiagramm für ein Geräteset

- Zeitangleich: Die Nachricht zum Zeitangleich erfolgt als Gruppennachricht an die Schnittstellensoftware. Ein Abgleich der Uhren mittels Unicast würde länger dauern und kann durch verschiedene nicht einschätzbare Verzögerungszeiten zusätzlich verfälscht werden. Wenn es zur Zeit eine laufende Übertragung zum Target gibt, wird der Vorgang abgebrochen. Es wird ein Logeintrag für das Gerät erstellt, indem der Erfolg oder der Fehlschlag des Befehls protokolliert wird. In 4.6 ist die Kommunikation visuell dargestellt

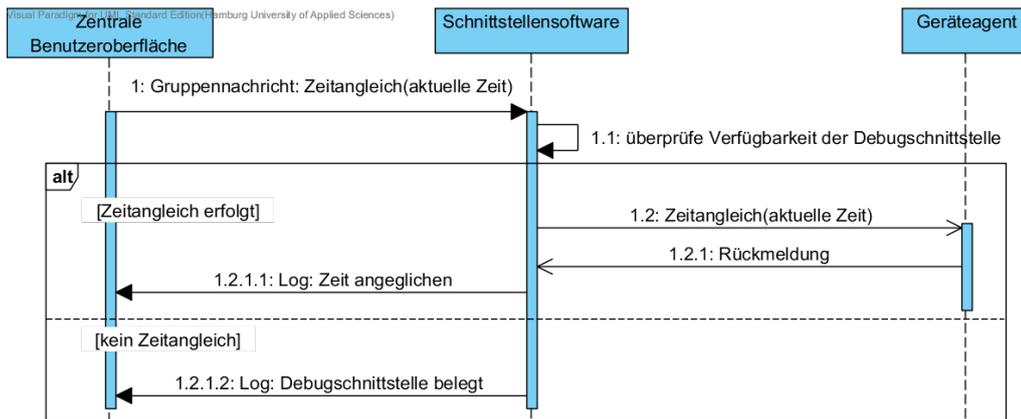


Abbildung 4.6.: Sequenzdiagramm für ein Zeitangleich

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

- Identifikation setzen: Zuerst erfolgt der Wechsel des Targets in den Bootloadermodus. Anschließend werden die Identifikationsdaten übertragen. Bei einer erfolgreichen Speicherung gibt das Target eine Rückmeldung. Erfolgt diese nicht, wird der Vorgang abgebrochen und die zentrale Benutzeroberfläche wird darüber informiert. Das dazugehörige Sequenzdiagramm ist in 4.7 dargestellt

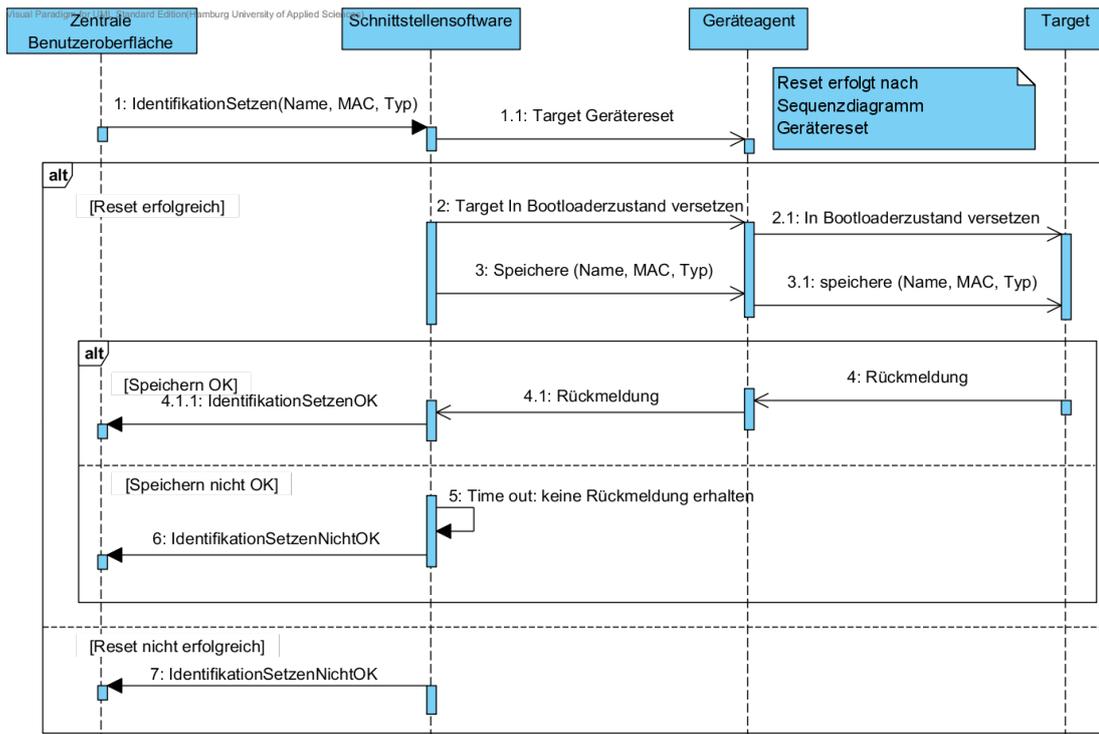


Abbildung 4.7.: Sequenzdiagramm zur Identifikation

- Softwareaktualisierung: Das Target wird zuerst in den Bootloadermodus versetzt, dafür kommt ein Reset zum Einsatz. Danach wird die Software Stück für Stück zum Target übertragen. Zur Sicherheit berechnet die Schnittstellensoftware sowie das Target einen Hash über den jeweiligen Teil der Daten. Ein Vergleich beider Berechnungen deckt Übertragungsfehler auf. Diese Kommunikation ist in Sequenzdiagramm 4.8 zu sehen

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

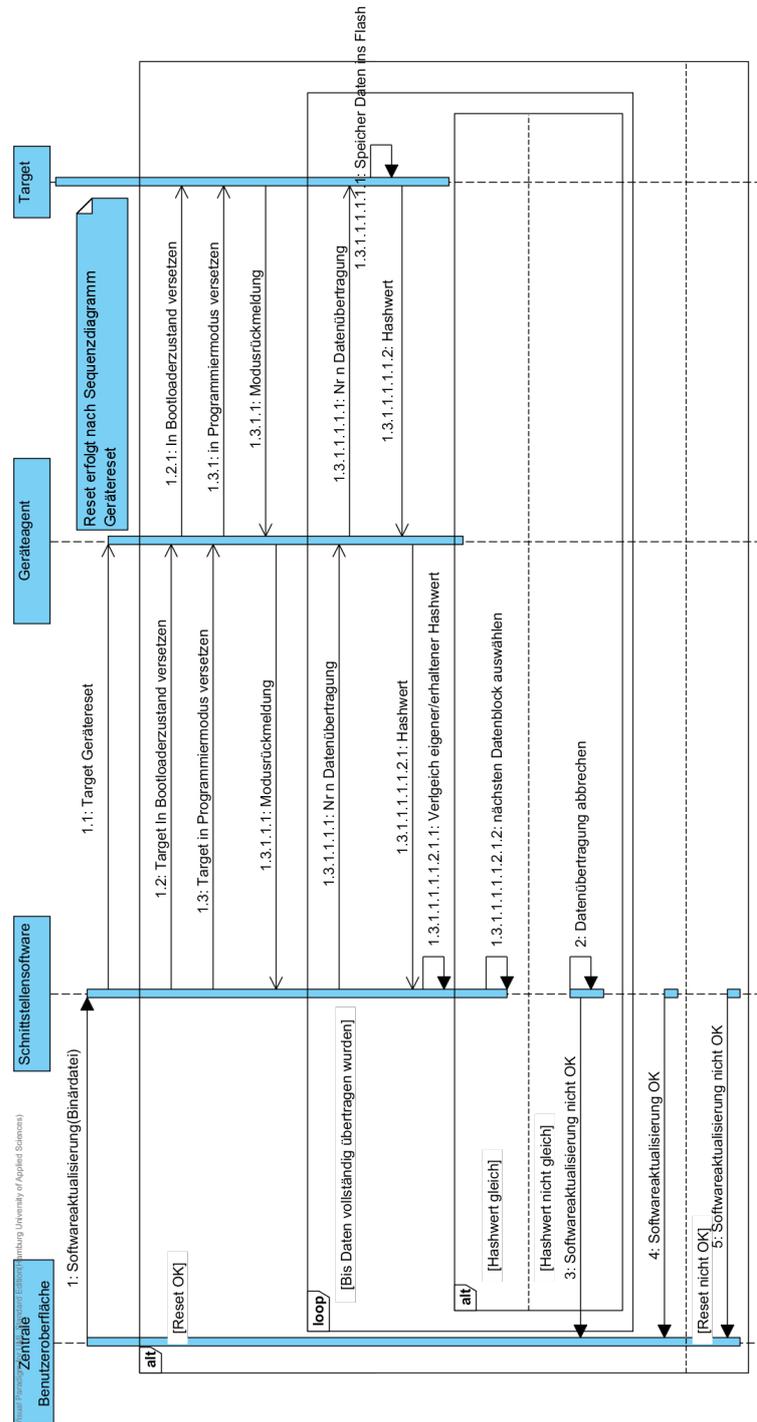


Abbildung 4.8.: Sequenzdiagramm zum Softwareupdate

### 4.3. Funkgeräteklassen

Das Funksystem wird durch Basisstationen und Funkeinheiten aufgebaut. Hier werden die näheren Eigenschaften und Aufgaben der beiden Geräteklassen erklärt.

Eine Basisstation organisiert den Ablauf und erhält Nutzdaten der Funkeinheiten. Dazu kommuniziert sie mit anderen Basisstationen. Dabei müssen verschiedene Daten ausgetauscht werden: Welche Funkeinheiten wurden bereits erkannt, welche Adressen im Netzwerk gibt es und welche Funkeinheit wird welcher Basisstation zugeordnet. Um das Senden in Zeitschlitz zu ermöglichen, müssen Basisstationen und Funkeinheiten synchronisiert werden.

Anhand einer gemeinsamen Zeit können dann Sende- und Empfangszeitpunkte vergeben werden. Diese Synchronisierung wird von einer ausgezeichneten Basisstation durchgeführt und wird in nachfolgenden Kapiteln als Koordinator bezeichnet. Ein Koordinator wird als erste Station das Netzwerk erkunden und besitzt bei dem Synchronisieren die führende Rolle. Ein Ausfall des Koordinators muss erkannt werden.

Eine Funkeinheit ist ein passiver Teilnehmer des Netzwerkes. Es werden Operationen nur durch eingehende Befehle ausgeführt. Zu diesen Operationen gehören die Erkundung des Netzwerkes, um Nachbarn zu finden und die Weitergabe von Informationen. Wie die Weiterleitung genau abläuft, wird im Abschnitt **Routingverfahren** erläutert. Durch den Betrieb mit einer Batterie bzw. Akku muss das Design ganz besonders energiesparend ausgelegt sein. Für einen Realeinsatz wird an einem Bussystem bzw. direkt an die GPIO (General Purpose Input Output) ein Sensor angebracht. Hierbei ist der Energieverbrauch des Sensors im Standby und im Aktivzustand zu beachten.

### 4.4. Übertragung der Nutzdaten

Die Übertragung der Messdaten im Funksystem muss zuerst organisiert werden. Sind alle Basisstationen und Funkeinheiten bekannt, gibt es verschiedene Möglichkeiten, wie die Messdaten verschickt werden können.

Die erste Möglichkeit ist das Senden in einer CAP (Contention-Access-Period). Dabei schickt eine Funkeinheit von sich aus die Messdaten. Dies hat den Vorteil, dass die Übertragung nicht speziell organisiert werden muss. Die Funkeinheiten müssen dabei zuerst auf ein freies Medium

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

warten, bevor sie ihre Messdaten verschicken.

Voraussetzung dafür ist eine ständige Bereitschaft aller Funkeinheiten, die zur Basisstation führen. Dafür müssten die dazwischenliegenden Funkeinheiten dauerhaft mit einer externen Spannungsversorgung verbunden sein, da eine anhaltende Empfangsbereitschaft mit Akkus bzw. Batterien unwirtschaftlich ist. Diese müssten in kurzen Abständen ausgetauscht werden.

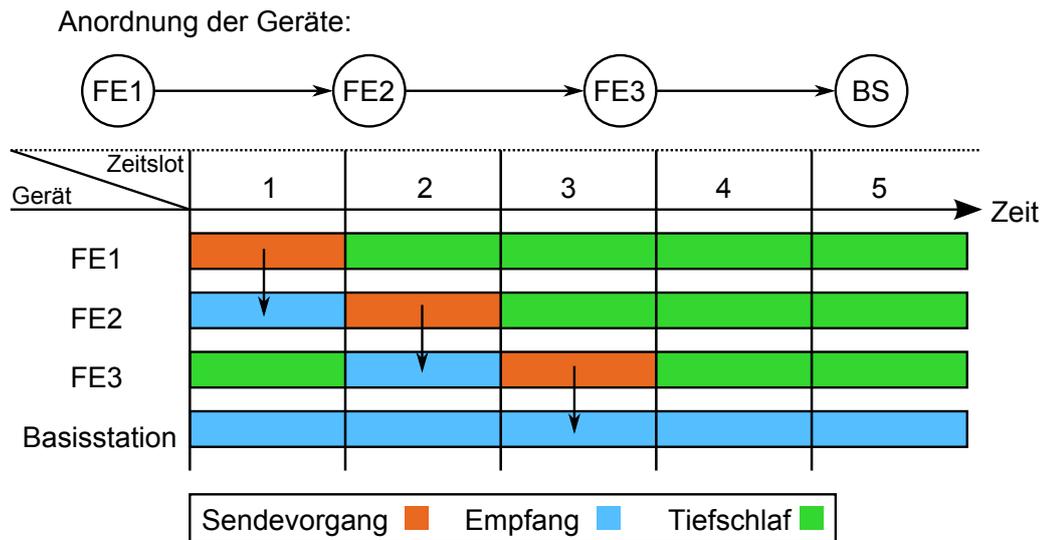


Abbildung 4.9.: Beispiel für Zeitslots

In einer zweiten Möglichkeit erfolgt eine zeitliche Teilung des Mediums für die Übertragungsphase. Jede Funkeinheit erhält dabei einen Sendezeitraum. Ob eine Funkeinheit ein Empfangszeitraum erhält, hängt davon ab, ob eine Funkeinheit Messdaten weiterleiten muss. Ein Zeitschlitz bzw. eine Slot wird jeweils einer Funkeinheit für das Senden und für das Empfangen zugewiesen. Alle Zeitschlitz sind gleiche lang. Benötigt eine Funkeinheit mehr als einen Zeitschlitz, so werden ihr mehrere zugeteilt. Das heißt, dass eine Funkeinheit nur in den zugewiesenen Zeitslots aktiv sein muss. Ein Beispiel dafür ist in [Abbildung 4.9](#) gegeben.

Ist eine Funkeinheit nicht im aktuellen Zeitslot involviert, so kann sie sich in einem Energiesparmodus begeben. Diese Art der Übertragung erfolgt in einer CFP (Contention-Free-Period). Kollisionen sind somit theoretisch ausgeschlossen. Die Zuteilung errechnet eine Basisstation anhand eines Schedule-Verfahrens und verteilt die Sende/Empfangspläne an die Funkeinheiten. Der Zeitraum für das Senden in Sendeschlitzen wird auch Superframe genannt. In [Abbildung](#)

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

4.10 ist ein Superframe dargestellt. Es besteht aus einer aktiven und inaktiven Phase. In der aktiven Phase wird das Netzwerk organisiert und synchronisiert. Anschließend erfolgt das Senden in Zeitslots. Bevor eine neue Synchronisierung durchgeführt wird, erfolgt eine inaktive Phase. In dieser Phase befinden die Funkeinheiten im Tiefschlaf.

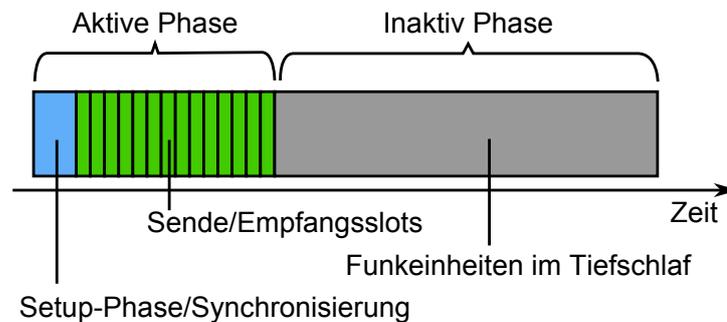


Abbildung 4.10.: Superframe

Da die Übertragung in erster Linie energiesparend sein soll, wird die zweite Möglichkeit für das Funksystem ausgewählt. Die Messdaten werden in Zeitslots gesendet.

Um eine Slotgröße zu berechnen, müssen bestimmte Voraussetzungen gegeben sein. Da in einem Tiefschlaf eines Mikrocontrollers nur noch asynchrone getaktete Uhren laufen, richtet sich die Berechnung auf die Kenngrößen dieser Uhr. Sehr oft wird diese Uhr mit 32768 Hertz betrieben. Mit dieser Taktung kann man einfach Sekunden zählen, da ein Überlauf der Zähler bei  $2^{15}$  stattfindet und dies genau einer Sekunde entspricht.

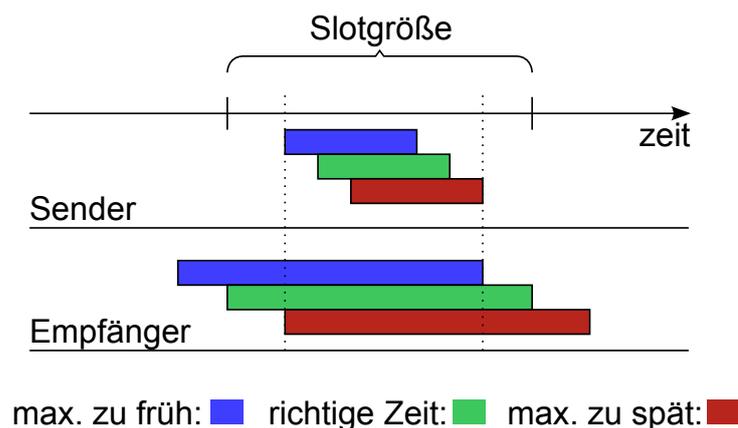


Abbildung 4.11.: Ungenauigkeiten im Slot

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Die Abbildung 4.11 zeigt ein Zeitslot mit einem Sender und Empfänger. Die verschiedenfarbigen Balken kennzeichnen jeweils den frühesten, den optimalen und den spätesten Zeitraum des Senders und Empfängers. Der Zeitraum, der gepunktet dargestellt ist, zeigt den gesamt möglichen Sendezeitraum. In diesem Zeitraum muss auch der Empfang garantiert werden.

Eine Uhr in einem Mikrocontroller besitzt immer eine Ungenauigkeit. Diese Uhr kann dabei entweder zu langsam oder zu schnell im Vergleich zur wirklichen Zeit laufen. Und je länger eine Synchronisierung mit anderen Uhren her ist, desto größer wird die absolute Ungenauigkeit. Wenn zwei Uhren verglichen werden, so können beide Uhren in die „entgegengesetzte“ Richtung laufen. Dabei ist eine Uhr langsamer und die andere schneller im Vergleich zu wirklichen Zeit. In einem Zeitraum können die Uhren also erhebliche Unterschiede aufweisen. Deshalb dürfen nicht zu viele Zeitslots verteilt werden, es besteht die Gefahr, dass sich Sender und Empfänger komplett verpassen.

Die Datenübertragungsrate beträgt nach Standard 250Kbit/s. Bei einem vollen Frame von 133 Bytes ergibt sich eine Übertragungsdauer von 4,256 Millisekunden. Mit keinem Vorteiler bei 32768 Hz und einem 8Bit-Zähler erfolgt in einer Sekunde 128 mal ein Überlauf dieser 8Bit-Variable, dies entspricht einer Slotgröße von 7,8125 Millisekunden.

Gesucht ist nun die vorhandene Reserve für die Uhrenungenauigkeiten während des Superframes. Es werden für die Übertragung 4,256ms sowie 2ms für die Ungenauigkeiten der Synchronisierung angenommen. Es verbleiben insgesamt ca. 1,55ms. Diese Reserve ist zu wenig. Am Rand (vor/nach Sendevorgang) stehen nur noch 0,78ms zur Verfügung, wenn beide Uhren entgegen laufen nur noch 0,39ms. Dieser Zeitraum ist zu kurz, weil 2 Uhren sehr schnell um diese Zeit bereits auseinander laufen.

$$t_{Reserve7,8ms} = 7,8125ms - 4,256ms - 2ms = 1,5565ms$$

$$t_{Randreserve7,8ms} = \frac{t_{reserve7,8125ms}}{2} = 0,77825ms$$

$$t_{Max.Ungenauigkeit7,8ms} = \frac{t_{Randreserve7,8ms}}{2} = 0,389125ms$$

Bei einer Verdopplung der Slotgröße auf 15,6 ms sollten bereits genug Reserven für Ungenauigkeiten vorhanden sein. Nun erfolgt die Berechnung der maximal übertragbaren Messdaten in einem Superframe. Die Reserven vor bzw. nach einem Sendevorgang betragen  $(15,6ms - 4,256ms(\text{Übertragungsdauer}) - 2ms(\text{Synchronisierungsungenauigkeit})) / 2 = 4,672ms$ . Da die

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Uhren im schlimmsten Fall komplett entgegenlaufen, muss diese Zeit noch halbiert werden. Demnach können die Uhren des Senders bzw. Empfänger bis zu 2,336 ms Ungenauigkeit aufweisen.

$$t_{Reserve15,6ms} = 15,6ms - 4,256ms - 2ms = 9,344ms$$

$$t_{Randreserve15,6ms} = \frac{t_{Reserve15,6ms}}{2} = 4,672ms$$

$$t_{Max.Ungenauigkeit15,6ms} = \frac{t_{Randreserve15,6ms}}{2} = 2,336ms$$

Bei Uhren wird die Genauigkeit in ppm (parts per million) angegeben. Hat eine Uhr eine Genauigkeit von 1 ppm, so besitzt sie in einer Minute eine Genauigkeit von:

$$t_{genauigkeit60s/1ppm} = \frac{60 \cdot 1000ms}{1 \cdot 10^{-6}} = 0,06ms$$

Bei einer Genauigkeit von 100 ppm sind dies bereits 6 ms. In einem Funksystem mit langer Einsatzdauer und Umwelteinflüssen wie Sonneneinstrahlung ist ein Wert von 100 ppm angemessen. Wenn in einem System sogar die Gangart einer Uhr bestimmt werden kann, so wäre ein Berechnung mit drastisch weniger ppm möglich. Diese Weiterentwicklung kann in Folgenden Bachelorarbeiten erfolgen, diese Arbeit widmet sich im Kerngebiet dem Funkprotokoll.

Mit den nun vorhandenen Informationen zur maximal zulässigen Uhrenabweichung und einer Annäherung der Uhrengenauigkeit lässt sich die maximale Dauer der Zeitslots errechnen.

$$t_{Zeitslots} = 2,336 \cdot 10^{-3}s \cdot \frac{1 \cdot 10^{-6}}{100} = 23,36s$$

Da eine Sekunde 64 Zeitslots beinhaltet, können bis zu 1494 Frames übertragen werden. Geht man dabei von bis zu 10 Messdaten in einem Frame aus, so könnten bis zu 14.950 Sensordaten übertragen werden. Dieser Wert entspricht aber nicht der maximalen Anzahl von Funkeinheiten. Sensordaten müssen sehr oft über andere Funkeinheiten hinweg verschickt werden, damit die Daten an der Basisstation eintreffen. Der schlimmste Fall wäre demnach eine Verkettung der Funkeinheiten an einer Linie. Die Sensordaten des entferntesten Funkeinheit müsste so über jede andere Funkeinheit vor ihr weitergeleitet werden.

Wenn die lineare Anordnung der Funkeinheiten angenommen wird, so lässt sich daraus die maximale Anzahl der Teilnehmer bestimmen. Die erste Funkeinheit sendet ihre Messdaten, die nächste sendet ihre Messdaten und die Daten der vorigen Funkeinheit. Für die Berechnung

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

kann daher die gaußsche Summenformel  $Summe = \frac{n*(n+1)}{2}$  hergenommen werden, da die Daten immer um eins wachsen. Umgestellt nach n und eingesetzt Summe ergibt sich die Gleichung  $n^2 + n - 29900 = 0$ . Die Lösung ist  $n_1 = 172,41, n_2 = -173,41$ . Die negative Lösung fällt weg.

Es können demnach bis zu 172 Funkteilnehmer an einer Kette angeordnet werden, bevor die Uhren bei der Datenübertragung zu stark abweichen. Andere Anordnungen, wie etwa ein vermaschtes Netz, muss insgesamt weniger Sensordaten weiterleiten und kann mehr Funkteilnehmer aufnehmen. Die untere Grenze im schlimmsten Fall aber beträgt 172 Funkteilnehmer und kann für alle Anordnungen der Funkeinheiten garantiert werden.

### 4.5. Kostenfunktion

Die Kostenfunktion setzt sich aus zwei Teilen zusammen. Es werden dafür die Empfangsstärke und den Ladezustand der Batterie bzw. des Akkus hergenommen.

Die Empfangsstärke wird von einem Funkmodul als dbm ermittelt. Dbm ist eine Leistungsangabe, die in Relation zu einem Milliwatt steht. Sie ist folgendermaßen definiert:  $P_{dbm} = 10 \cdot \log_{10} \cdot \left(\frac{P_{Watt}}{1mW}\right)$ . 0 dbm entspricht dabei einer Leistung von 1 mW. Eine Addition bzw. Subtraktion z.B. von 10 verändert die Leistungsangabe in Watt um den Faktor 10.

Die Kosten bei der Empfangsstärke sollen so verteilt werden, dass die Kosten einer Verbindung zwei eng benachbarter Funkeinheiten hoch sind. Ist die Entfernung größer, so sinken die Kosten. Wenn jedoch die Entfernung zu hoch ist, wird der Verbindung hohe Kosten zugewiesen. Bei vielen Funkchips gilt eine Verbindung von weniger als -98dbm als unzuverlässig, da ab -98dbm sinkend die Fehlerhäufigkeit zunimmt (vgl. [AtmelCorporation 2012](#), S. 531).

Sollte es eine Situation geben, in welche zu einer Funkeinheit eine relativ schlechte und eine sehr gute existieren, so hat die gute Verbindung Vorrang. Es gibt für eine Verbindung immer Grundkosten. Diese wurden mit 5 festgelegt. Die Verbindung ist also am besten, wenn zwischen zwei Geräte ein bestimmter Abstand existiert. Dieser wurde mit -60dbm festgelegt und kann sich durch gesammelte Erfahrung innerhalb der Realisierung noch verändern. Diese Funktion ist in [Abbildung 4.12](#) dargestellt.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Im Vergleich zu dem Routingverfahren, das Zigbee nutzt, wird eine Route nicht nur anhand der geringsten Hop-Anzahl ausgewählt, Geräte mit wenige Energiereserven und zu eng benachbarte Funkeinheiten werden dort nicht beachtet.

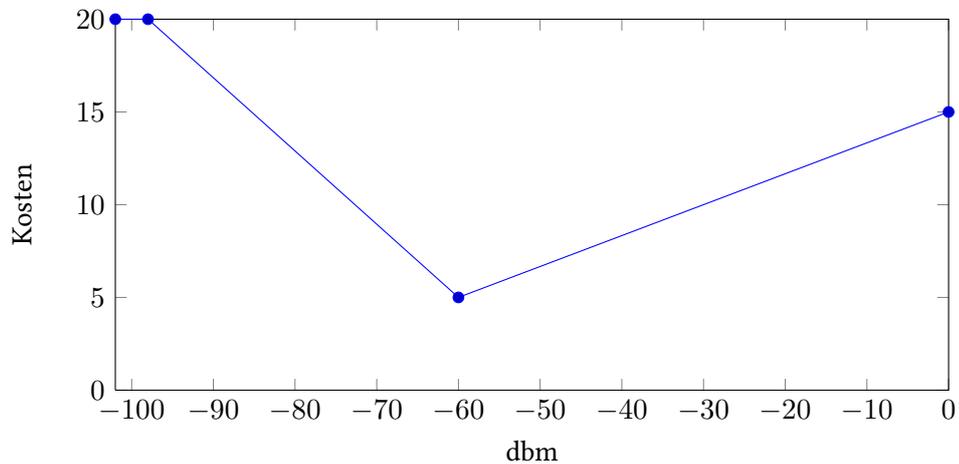


Abbildung 4.12.: Kosten Empfangsstärke

Die Kostenbewertung für den Batteriezustand ist linear gehalten. Steht eine neue Batterie oder neuer Akku bereit, so fallen keine Kosten an. Geht der Energievorrat zu Neige, kostet eine Verbindung über so eine Funkeinheit sehr viel. Die Funktion ist in [Abbildung 4.13](#) zu finden. Die genaue Berechnung erfolgt im der Realisierungsphase.

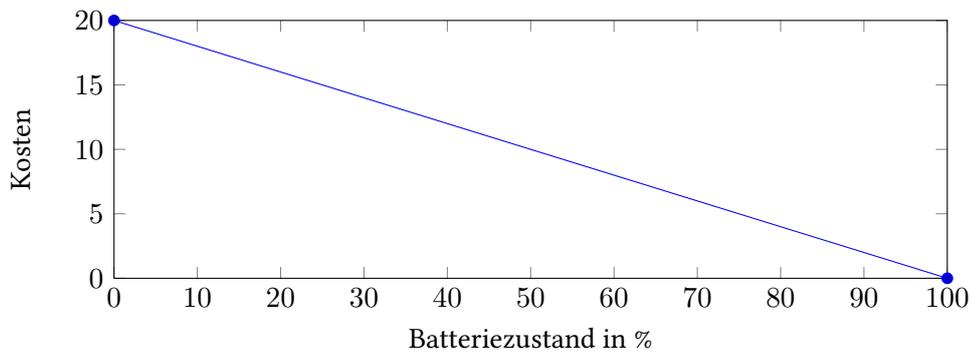


Abbildung 4.13.: Kosten Batteriezustand

Der Batteriezustand wird berechnet indem die bisherige Wachzeit in Relation zu der maximal theoretisch möglichen Wachzeit gesetzt wird:

$$Ladezustand_{Prozent} = \left( \frac{t_{Wachzeit_{seitStart}}}{t_{Wachzeit_{Max.moeglich}}} \right) * 100$$

### 4.6. Routingverfahren

Das Routing hat das Ziel den Pfad zwischen einer Basisstation und einer Funkeinheit für Nachrichten festzulegen. Ein Funkteilnehmer bekommt die nötigen Informationen, um eine Nachricht an einen entfernten Teilnehmer zu schicken. Dabei gibt es verschiedene Ansätze.

Bei dem Source-Routing enthält das Datenpaket die geplante Route. Zwischenstationen müssen so keine eigene Routingtabellen halten und pflegen. Es bedeutet aber auch eine großen Overhead für jede zu übermittelnde Nachricht. Ein Frame kann somit nur eine bestimmte Hopanzahl erreichen, bevor ein weiteres Datenpaket für weitere Sprungziele und die Nutzdaten erforderlich ist. Je größer die Anzahl an Hops ist, die überwindet werden müssen und je kleiner die Nutzdaten sind, desto höher wird der Verwaltungsaufwand. In solchen Fällen kann eine Nachricht nicht energieeffizient übertragen werden.

Weiterhin lässt sich das Routing in Link-State und Distance-Vector unterteilen. In beiden Verfahren werden Informationen über das Netzwerk getauscht. Dabei findet die Informationsweitergabe meist zu den Nachbarn statt. Das Distance-Vector-Routing tauscht nur einfache Informationen aus. Dagegegen nutzt das Link-State-Routing mehr Informationen, um eine komplexere Ansicht zu erstellen. Mit diesem Aufwand können optimale Wege ermittelt werden. Jedoch wird das zu entwickelnde Funksystem von Basisstationen aufgebaut und zentral gesteuert. Eine zentralere Lösung ist gesucht.

Für das Funksystem eignet sich am besten das zentralisierte Routing. Routen werden bei der Basisstation errechnet und an die Funkeinheiten weitergegeben. So eine Routinginformation ist in Tabelle 4.1 zu sehen. Diese Information ist ausreichend, um Daten zur nächsten Funkeinheit zu übertragen. Der Vorgang wird so lange fortgeführt, bis der Empfänger die Nachricht erhalten hat.

## 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Bytegröße	Bezeichnung	Funktion
1	hop_id	ID Funkeinheit
1	destination_id	ID Endziel
1	nextHop_id	ID nächste Funkeinheit

Tabelle 4.1.: Nachrichtenaufbau Routing data

Die Funkeinheiten werden bei diesem Verfahren entlastet und geben nur die Informationen zu ihrem direkten Umfeld weiter. Eine Basisstation kann so beim Aufbau des Netzwerkes entscheiden, über welche Pfade der Erkundungsvorgang fortgesetzt wird. Die Wege für das Weiterleiten der Messdaten errechnet die Basisstation nach dem Erkundungsvorgang und gibt die Routeninformationen anschließend an die Funkeinheiten weiter.

### 4.7. Basisstation

Dieser Teil beschäftigt sich mit dem Design der Basisstation. Es werden Strategien diskutiert, die für die Organisation nötig sind. Das gewünschte Verhalten wird modelliert und dargestellt. Auch wird ein Kommunikationsprotokoll für die Basisstationen untereinander entworfen.

#### 4.7.1. Koordinatorwahl

die Wahl eines Koordinators erfolgt mithilfe einer abgeänderten Version des Bully-Algorithmus. Bei dem Originalverfahren werden unter einer Gruppe von Stationen ein Koordinator bestimmt. Folgende Voraussetzungen sind für diese Wahl erforderlich:

- Nachrichten treffen immer in der gleichen Reihenfolge ein
- Das Empfangsereignis einer Nachricht findet nach dem Sendeereignis statt
- Es ist eine zuverlässige Kommunikation erforderlich
- Jeder Teilnehmer besitzt eine eindeutige ID
- Ein Teilnehmer reagiert immer bei einer Nachricht, sollte der Teilnehmer ausgefallen und damit unerreichbar sein, kann der Teilnehmer nicht mehr auf Nachrichten reagieren

Wenn eine Wahl in dem Originalverfahren ausgelöst wird, schickt der Initiator eine Nachricht mit seiner Nummer. Gruppenmitglieder, mit höherer Nummer reagieren bei dieser Mitteilung und schicken ihrerseits eine neue Nachricht ihrer Ordnungsnummer. Dieser Vorgang findet

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

so lange statt, bis die Station mit der höchsten Nummer die Nachricht sendet. Nach einem Timeout erklärt sie sich zum Koordinator und benachrichtigt die übrigen Stationen.

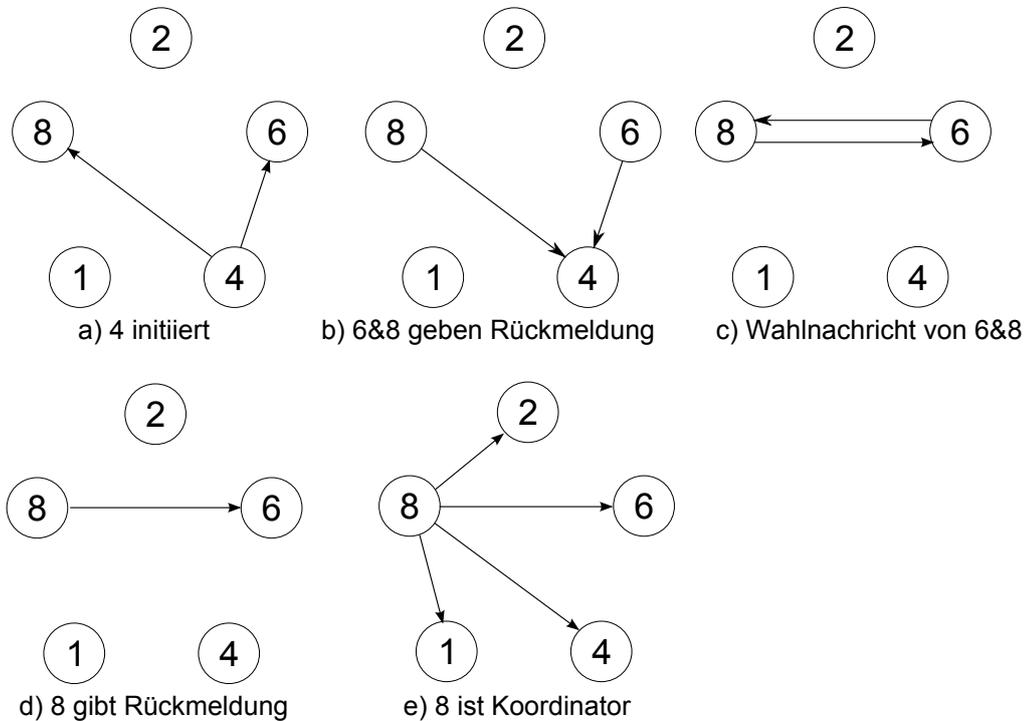


Abbildung 4.14.: Beispielverlauf Bully-Algorithmus

In dem angepassten Algorithmus werden Nachrichten anhand ihrer Ordnungsnummer verzögert. Kleinere Nummern werden schneller versandt als größere. Dabei wird die Station mit der kleinsten Ordnungsnummer als Koordinator gewählt.

Somit werden die Anzahl der Nachrichten verringert, da die Station mit der kleinste Nummer zuerst sendet und alle anderen Stationen mit einer größeren Ordnungsnummer daraufhin ihr Sendevorhaben abbrechen. Nach einem Timeout erklärt sich die Basisstation zum Koordinator und benachrichtigt die Gruppe über das Ende der Wahl.

Sollte eine Benachrichtigung über das Ende des Verfahrens oder eine Nachricht mit der Rückmeldung bei Teilnehmern nicht eintreffen, können Fehler eintreten. So können sich bei dem Algorithmus 2 Teilnehmer als Koordinatoren erklären. Deshalb wird in einem Fehlerfall die Wahl erneut angestoßen, wenn eine doppelte Koordinator-Ernennung erkannt wurde.

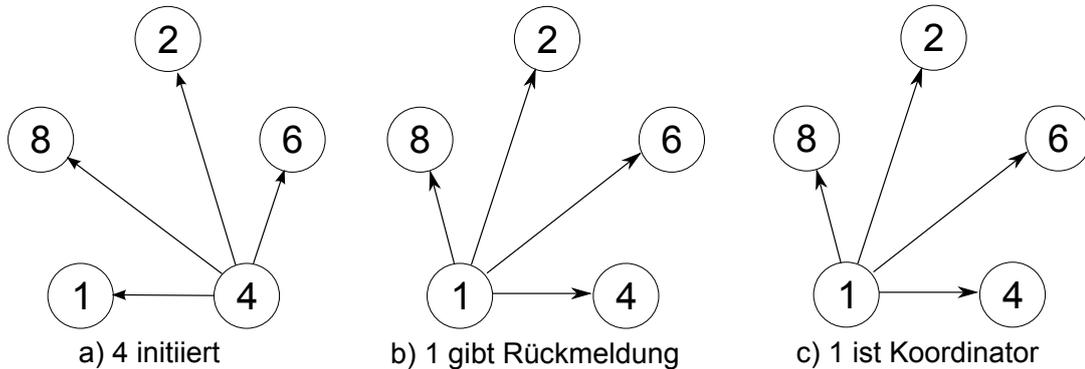


Abbildung 4.15.: Beispielverlauf des angepassten Bully-Algorithmus

#### 4.7.2. Verhalten der Basisstation

Das grobe Systemverhalten ist als Beispiel in Abbildung 4.16 und 4.17 gezeigt. Das Funknetzwerk beinhaltet zwei Basisstationen und 3 Funkeinheiten. Funkeinheit eins ist in Reichweite beider Basisstationen. Die anderen zwei Funkeinheiten sind nur in Reichweite jeweils einer Basisstation.

Basisstation null wurde zum Koordinator gewählt, anschließend erkundet sie das Funknetzwerk (a). Dabei wird Funkeinheit null und eins sowie die Basisstation eins erkannt. Nach der Erkundung werden die besten Wege errechnet. In diesem Beispiel wird Funkeinheit null ihre Messdaten an Basisstation null versenden und Funkeinheit eins an Basisstation eins. Zu welchen Zeiten die Messdaten gesendet werden müssen, ergibt sich aus dem individuellen Sende/Empfangsplan, der zu den beiden Funkeinheiten geschickt wird (b).

Danach erfolgt die Erkundung von der Basisstation eins aus. Funkeinheit eins wurde bereits erkannt. Diese Funkeinheit wird ignoriert (c). Funkeinheit zwei wird erkannt. Anschließend erfolgt die Errechnung der besten Wege sowie die Erstellung des Scheduling. Funkeinheit zwei wird dann ein Sende/Empfangsplan geschickt (d).

Der Koordinator (Basisstation null) leitet dann die Synchronisierung ein. Dazu verschickt der Koordinator eine Synchronisierungsnachricht an die andere Basisstation, die daraufhin Synchronisierungsnachrichten an ihre Funkeinheiten schickt. Der Koordinator sendet danach Synchronisierungsnachrichten an seine zugewiesenen Funkeinheiten (e).

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

Im nächsten Abschnitt senden die Funkeinheiten ihre Messdaten in Richtung der ihr zugeteilten Basisstation (f). Nachdem der Sendevorgang abgeschlossen ist, versetzt sich jede Funkeinheit in einem Energiesparmodus, bis der Superframe beendet ist (g).

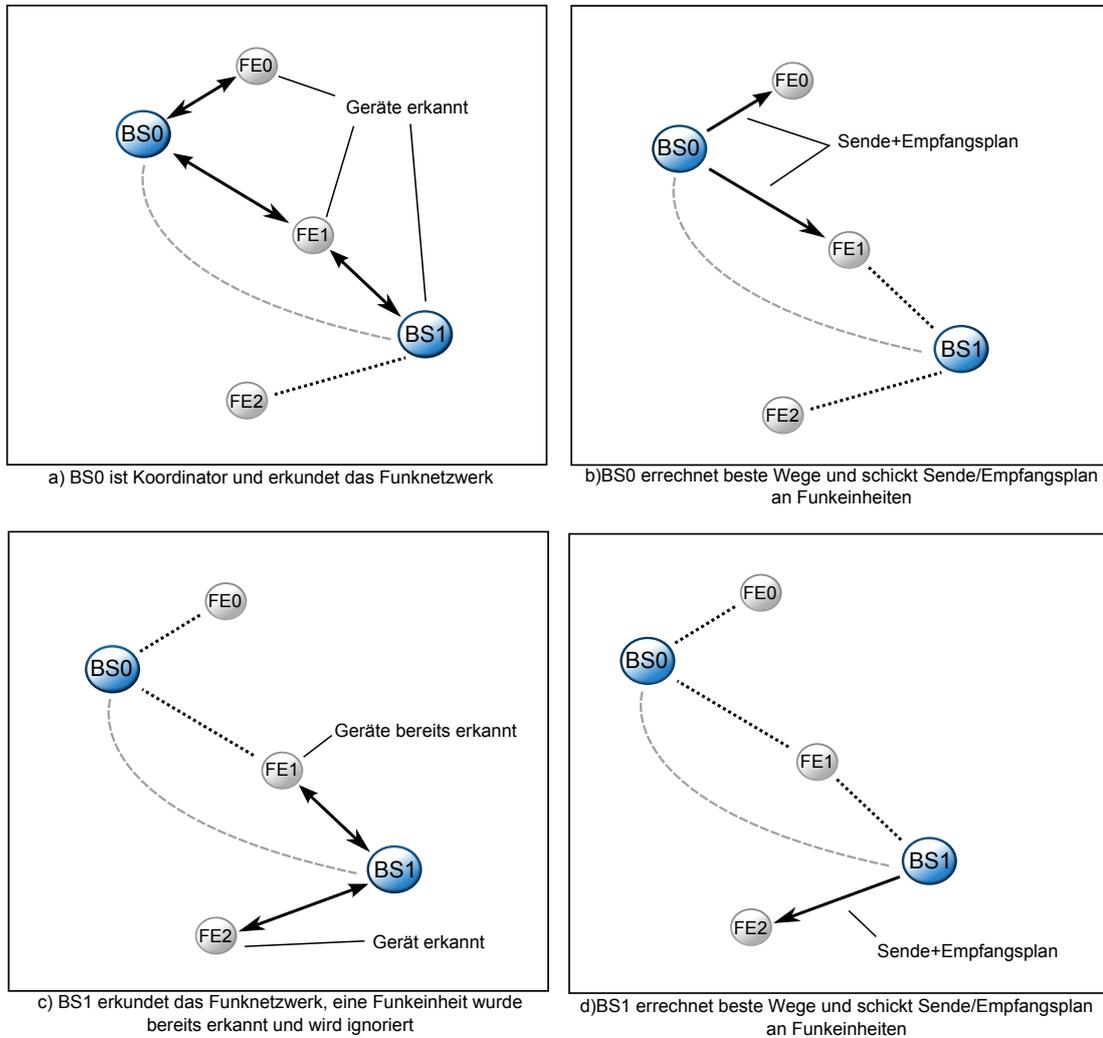


Abbildung 4.16.: Systemverhalten in groben Abschnitten: Teil 1

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

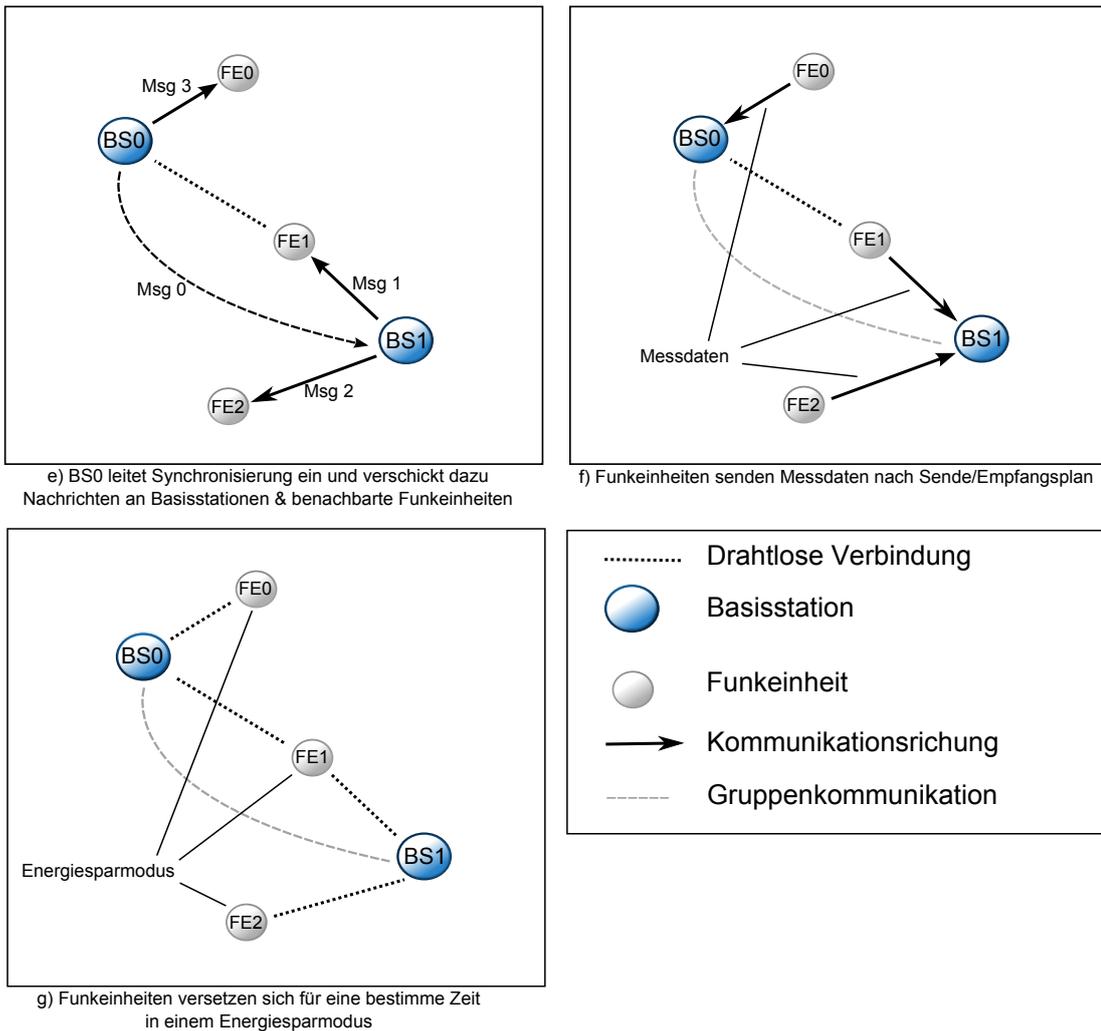


Abbildung 4.17.: Systemverhalten in groben Abschnitten: Teil 2

Nun erfolgt die genaue Verhaltensbeschreibung einer Basisstation. Nach dem Systemstart erfolgt die Wahl des Koordinators nach dem angepassten Bully-Algorithmus. Der Koordinator erstellt daraufhin eine Liste aller Basisstationen und schickt eine Meldeaufforderung an die Gruppe. Die anderen Stationen, die sich in einem Wartezustand befinden, reagieren darauf und schicken eine Nachricht mit ihrer Ordnungsnummer.

Diese Liste über Basisstationen werden nach Abarbeitung und Scheduling des Koordinators benötigt. Die aktive Basisstation wird nun zuerst sein direktes Umfeld auf Funkeinheiten

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

erkunden. Der genaue Vorgang ist im Abschnitt **Erkundungsvorgang** erklärt.

Falls Funkeinheiten gefunden wurden, werden diese zur lokalen Liste hinzugefügt. Nun werden Erkundungsbefehle sequentiell an die Funkeinheiten geschickt. Die Reihenfolge dieser Abarbeitung richtet sich an eine Kostenfunktion und ist im **Erkundungsvorgang** näher erläutert. In dieser Phase erfolgt die Reihenfolge anhand der qualitativsten Verbindung. Bei jeder dazukommenden Funkeinheit vergrößert sich der Graph. Andere Basisstationen sind in diesem Graphen auch enthalten, somit kann die Zugehörigkeit von Funkeinheit zu Basisstation sofort vorgenommen werden. Spätere Erkundungen von bereits zugewiesene Funkeinheiten werden verhindert.

Nachdem die Basisstation das Funknetzwerk erforscht hat, werden nun die optimalen Wege für die Datenübertragung berechnet. Dabei wird von jeder Basisstation im Graphen aus ein Dijkstra-Algorithmus durchgeführt. Die berechneten Ergebnisse werden anschließend für das Scheduling verwendet, Details dazu sind im Abschnitt **Scheduling** verfügbar.

Die erstellten Scheduleinträge schickt die Basisstation dann an die Funkeinheiten. Die Scheduleinträge beinhalten jeweils die Nummer des Zeitschlitzes für das Senden/Empfangen sowie die Anzahl der Zeitschlitze, da eine Funkeinheit durchaus mehrere Intervalle zum Senden oder Empfangen benötigt.

Nach Versenden der Scheduleinträge wird die Basisstation die gesammelten Informationen an die Gruppe der Basisstationen weitergeben. Dazu gehören die nächst verwendbare Adresse für eine Funkeinheit und alle Einträge zu den gefunden Funkeinheiten inklusive Zuordnung der Basisstation. Es wird dann (nur im Falle des Koordinators) die Liste aller Basisstationen veröffentlicht. Nachfolgende Basisstationen können anhand dieser Liste ihren Nachfolger ein Startsignal schicken. Das Startsignal wird dann versendet, wenn die Erkundung sowie das Scheduling einer Basisstation abgeschlossen sind.

Diese Prozedur zieht sich bis zur letzten Basisstation hin. Ist die letzte Basisstation mit dem Erforschen und dem Verschicken der Scheduleinträge fertig, so wird nun dem Koordinator ein Signal geschickt. Es folgt nun die Phase der Zeitsynchronisierung. Der Koordinator berechnet die benötigte Vorbereitungszeit bis alle Funkeinheiten eine Synchronisierungsnachricht erhalten haben. Es wird für jede Basisstation und Funkeinheit eine Vorbereitungszeit eingerechnet. In dieser Zeit hat eine Basisstation oder Funkeinheit Zeit Synchronisierungsnachrichten weiter-

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

zuschicken. Da in jedem Falle eine gewisse Verzögerungszeit vorhanden ist, wird diese Zeit hinreichend groß gewählt. Die genaue Zeitspanne wird im Realisierungsschritt festgestellt, da z.B. die Verzögerungszeiten bei Gruppennachrichten noch nicht zu bestimmen sind.

Ab dem Zeitpunkt einer gemeinsamen Zeit versetzen sich die Funkeinheiten in Tiefschlaf und wecken sich erst ab ihren Empfangs- bzw. Sendezeitpunkt auf. Beim Schedulingverfahren wurde darauf geachtet, dass sich eine Funkeinheit nur einmal zum Empfang aufwecken muss. Nach dem Verschicken der Messdaten wird ein Energiesparmodus aktiviert, indem nur noch das Nötigste mit Energie versorgt wird.

Nach einer fest definierten Zeit sind die Funkeinheiten wieder aktiv ansprechbar und es kann je nach Anwendung sofort ein neues Datenübertragungsintervall beginnen oder das Netzwerk wird von den Basisstationen neu erfasst. Dieses Verhalten ist auch als Zustandsdiagramm [A.2](#) im [Anhang](#) dargestellt.

##### 4.7.3. Nachbarsfindung

Direkte Nachbarn zu finden ist beim Aufbau des Funknetzwerkes von grundlegender Bedeutung. Die Qualität des Verfahrens bestimmt wie schnell andere Funkteilnehmer erkannt werden und ob alle Teilnehmer immer erkannt werden.

Ein erster Ansatz ist es eine Rundmeldung zu versenden, woraufhin sich alle Funkteilnehmer melden. Dieser Ansatz würde aber Kollisionen verursachen. Denn trotz CSMA/CA kann es passieren, dass Nachrichten zur gleichen Zeit versendet werden. Andere Nachbarn müssen nun auf ihr Senden warten. Durch den Ansturm der Rückmeldungen erkennen viele Funkeinheiten ein blockiertes Medium und nach einer bestimmter Zeit wird der Sendeversuch abgebrochen. Das Ergebnis sind einige Kollisionen und nicht erfasste Teilnehmer in Reichweite.

Manche Kollisionen werden durch spezielle Situationen ausgelöst. Ein Beispiel dafür ist das „Hidden Station“-Problem. Zwei Stationen sind nicht direkt erreichbar, dazwischen ist eine Station, die in Reichweite der beiden Stationen liegt. Wollen die außen liegenden Station zeitgleich eine Nachricht an die mittlere Station schicken, so wird es zu einer Kollision kommen. Das CSMA/CA wird fehl schlagen, da beide Stationen nicht in unmittelbarer Reichweite liegen. Ein Beispiel dafür ist in [Abbildung 4.18](#) gegeben.

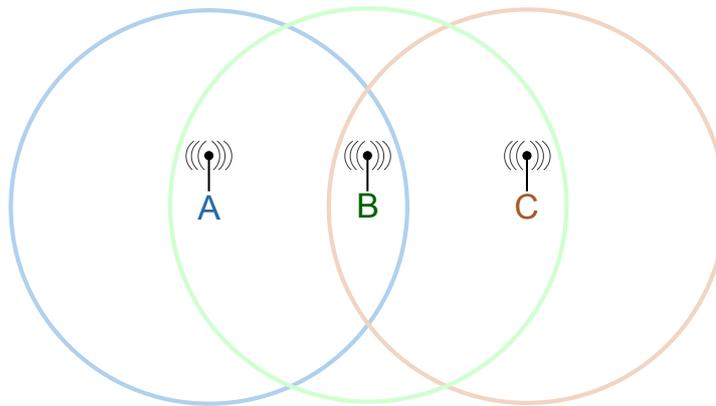


Abbildung 4.18.: Beispiel einer Hidden Station

In einem zweiten Versuch verschickt ein Funkteilnehmer nun viele Rundmeldungen. Es werden so viele Rundmeldungen verschickt, wie es Adressen im Funknetzwerk existieren. Dafür ist ein 8Bit-Wert vorgesehen. Auf welche Rundmeldung sich ein Teilnehmer nun meldet, hängt vom Vergleich einer Bitmaske ab. In Tabelle 4.12 ist die dazugehörige Nachricht abgebildet. Als Vergleich wird das letzte Byte der Mac-Adresse verwendet, da dieses Feld in einer Herstellerserie pro Gerät inkrementiert ist. somit sollte eine ungefähre Gleichverteilung gegeben sein. Mit dieser Änderung sollten Kollisionen deutlich seltener auftreten. Allerdings muss jeder Funkteilnehmer  $2^8$  Rundmeldungen versenden und auf Antwort warten. Das ist bei vielen Geräten im Funknetz sehr zeitintensiv.

Es fehlt also noch eine Verbesserung, um die Zeit zu verkürzen. In einem dritten Ansatz erfolgen nun gestaffelte Rundmeldungen von 16 Blöcken. Durch eine Anpassung der Vergleichsdaten in der „HelloResponse“-Nachricht wird nur ein kleiner Kreis von Funkteilnehmern ausgewählt. Um einer Vielzahl von zeitnahen Nachrichten zu entgehen, versenden die Geräte nach Erhalt der „HelloRequest“-Anfrage ihre Nachricht zeitversetzt.

Da Sensornetzwerke sich nur langsam verändern, muss ein erneuter Suchvorgang nicht vollständig ausgeführt werden. Hierbei genügt es alle bekannten Geräte auf eine noch bestehende Verbindung zu überprüfen. Anschließend wird eine Rundmeldung für eventuell neu hinzugekommene Funkeinheiten verschickt. Ab dem zweiten Suchdurchlauf kann Zeit gespart werden.

### 4.7.4. Erkundungsvorgang

Die Netzwerkerkundung einer Basisstation hat das Ziel, alle von ihm erreichbare Funkeinheiten und Basisstationen zu erfassen. Im ersten Schritt wird eine Basisstation seine Umgebung erkunden. Dies geschieht nach einem Vorgehen, wie es im Abschnitt 4.7.3 erklärt wurde.

Findet die Basisstation mindestens eine Funkeinheit, so wird die Erkundung fortgesetzt. Werden von allen Basisstationen keine Funkeinheiten gefunden, so endet das Verfahren und das Netz wird zu einem späteren Zeitpunkt wieder durchsucht. Die gefundenen Funkeinheiten werden jeweils einer Adresse zugewiesen und zu dem Graphen hinzugefügt.

Ausgangspunkt bei dem Graphen ist die Basisstation als erstes Element. Dort werden die im ersten Schritt gefundenen Nachbarn angefügt. Anschließend wird mit dem Dijkstra-Algorithmus neue beste Wege berechnet. Hierbei kommt eine Kostenfunktion zum tragen. Verbindungen zu Funkeinheiten mit besonders gute Empfangsqualität werden bevorzugt. Es könnte jetzt ein „günstigeren“ Weg zu einer Funkeinheit vorhanden sein, dem noch kein Suchauftrag geschickt wurde.

Nachdem die Funkeinheiten ihre Adresse erhalten haben, verschickt die Basisstation nun Suchaufträge zu den Funkeinheiten. Es werden zuerst die Funkeinheiten abgearbeitet, die bei dem Dijkstra-Algorithmus die günstigsten Kosten aufweisen.

Eine Funkeinheit durchsucht nach Auftragserhalt nach demselben Verfahren wie die Basisstation. Anschließend schickt die Funkeinheit ihre Suchergebnis zurück zur Basisstation. Die Ergebnisse werden dem Graphen hinzugefügt und wird durch den Dijkstra-Algorithmus wieder aktualisiert. Die Funkeinheit, die zuletzt den Suchauftrag erhalten hat, wird nun im Graphen markiert.

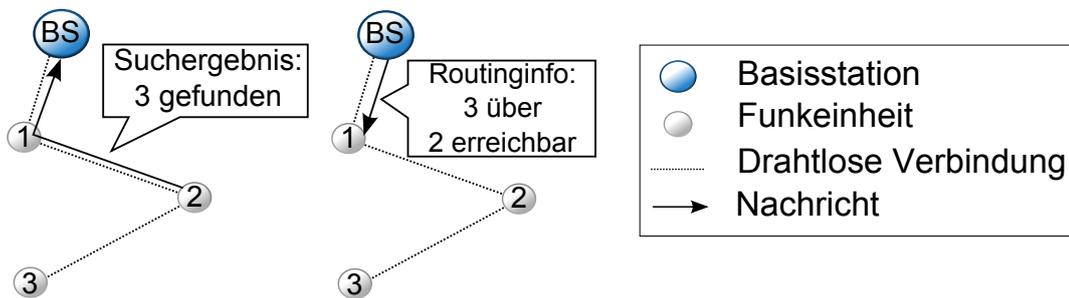


Abbildung 4.19.: Beispielfall für das Verschicken von Routinginformationen

In einigen Fällen müssen noch Routinginformationen weitergereicht werden. Dies ist dann nötig, wenn zwischen einer neu gefundenen Funkeinheit bzw. bekannten Funkeinheit mit besserem Weg und der Basisstation mehr als eine Funkeinheit existiert. Ohne diese Information kann eine Funkeinheit Daten nicht weiterleiten, da dieser kein Wissen über den nächsten Zwischenempfänger hat.

Ein Beispiel ist in [Abbildung 4.19](#) dargestellt. Funkeinheit zwei hat einen Suchvorgang abgeschlossen und dabei die Nummer Drei gefunden. Die Basisstation informiert daraufhin Funkeinheit eins, dass ab sofort die Funkeinheit drei über die Zwei erreichbar ist.

Weitere Funkeinheiten werden so Schritt für Schritt erfasst. Falls eine Basisstation in den Ergebnissen auftaucht, wird diese nicht weiter verarbeitet. Jedoch wird beim Scheduling die Informationen über weitere vorhandene Basisstationen im Funknetzwerk benötigt. Das Verfahren für das Erkunden ist als Flussplan in [Abbildung 4.20](#) dargestellt.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

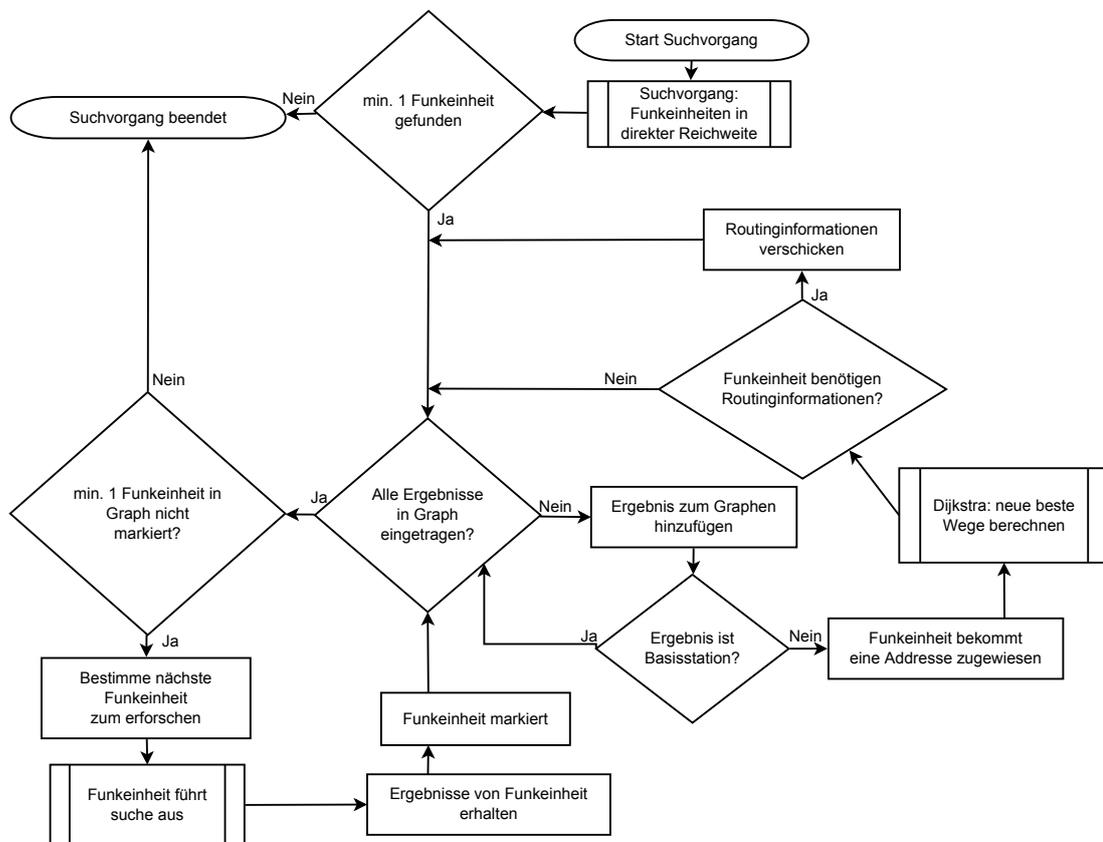


Abbildung 4.20.: Flussplan Funknetzwerkerkundung

Das Ende der Erkundung ist erreicht, wenn alle Funkeinheiten im Graphen markiert sind. Das Funknetzwerk ist jetzt von der Basisstation aus aufgebaut und es können weitere Schritte vorgenommen werden, um den Superframe zu koordinieren.

In Abbildung 4.21 ist eine Beispielsituation gegeben. Die Basisstation BS<sub>1</sub> hat ihre Erkundung gerade abgeschlossen. In der Abbildung ist zu erkennen, in welchem Teilschritt die Verbindungen zwischen den Funkteilnehmern erkannt wurde. Funkeinheit eins bis fünf sowie Basisstation BS<sub>2</sub> und BS<sub>3</sub> wurden von der Basisstation eins erkannt.

Diese Funkeinheiten und Basisstationen werden von der Basisstation BS<sub>1</sub> für das Scheduling verwendet. Dafür wird mittels dem Dijkstra-Algorithmus die optimalen Wege für die Nutzdatenübertragung berechnet. Von jeder erkannten Basisstation wird überprüft, zu welchen Funkeinheit ein optimaler Weg von ihr aus existiert. Zum Einsatz kommt eine Kostenfunktion,

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

die den optimalen Weg anhand der in den Anforderungen angegebenen Spezifikation errechnet. Demnach werden Hops bei wenig Abstand vermieden und Funkeinheiten mit geringen Energiereserven entlastet.

Wird eine Funkeinheit gefunden, so werden auch alle Pfade zu allen anderen Basisstationen gefunden, die die Funkeinheit erreichen können. Im Beispiel erreicht die Funkeinheit zwei Basisstationen BS2 und BS3. Der Pfad zu BS1 ist bekannt, da von dort der Erkundungsauftrag zu Funkeinheit zwei kam. Der Algorithmus zum Erkunden endet erst, wenn die letzte Funkeinheit entweder keine weiteren Nachbarn findet oder nur Basisstationen erkannt hat. Somit kann eine Basisstation sicherstellen, dass es keine besseren Wege zu der Funkeinheit existiert.

Nun kann ein Schedulingverfahren mit diesen Informationen Sendeslots und Empfangsslots berechnen. BS1 wird nach dem Scheduling die Ergebnisse an die Funkeinheiten weitergeben. Durch das Weitergeben der Erkundungsergebnisse an die Gruppe wird eine erneute Erkundung und Slotzuordnung verhindert.

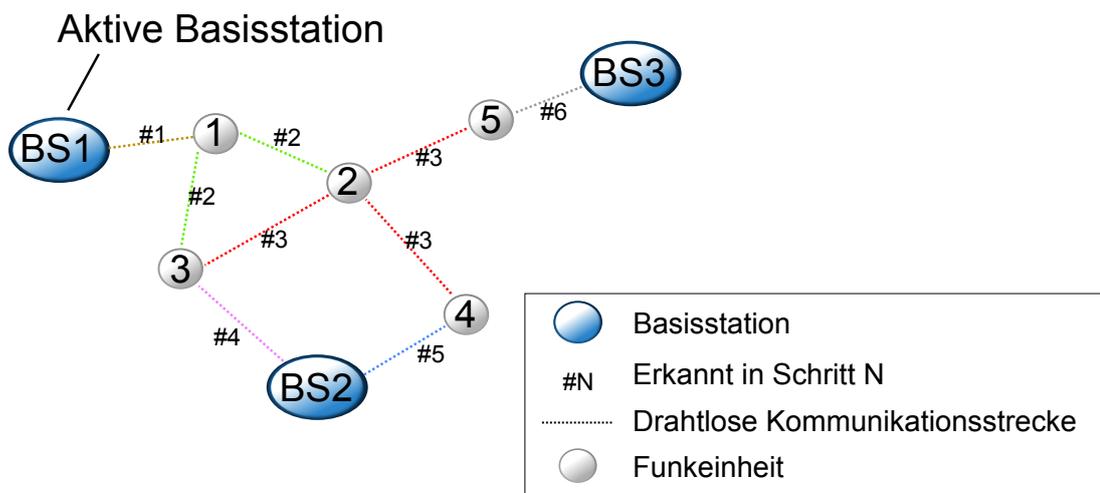


Abbildung 4.21.: Beispielsituation Erkundung

#### 4.7.5. Scheduling

Das Scheduling dient der Zuteilung von Sendeslots und Empfangsslots sowie zur Berechnung der benötigten Zeit für das Senden bzw. Empfangen von Daten.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Die Basisstationen im Graphen werden nacheinander abgearbeitet. Zuerst erfolgt die Zuordnung der entferntesten Funkeinheiten. Dies hat den Hintergrund, dass Funkeinheiten vor ihnen nur einmal im Superframe zum Empfang aufgeweckt werden müssen. Ein Beispiel dazu findet sich in Abbildung 4.22. In diesem Beispiel erhalten die beiden Funkeinheiten ganz rechts den Sendeslot Eins und Zwei. Die Funkeinheit vor ihnen mit dem Sendeslot 6 muss so nur einmal aufgeweckt werden und geht für zwei Zeitslots in den Empfang.

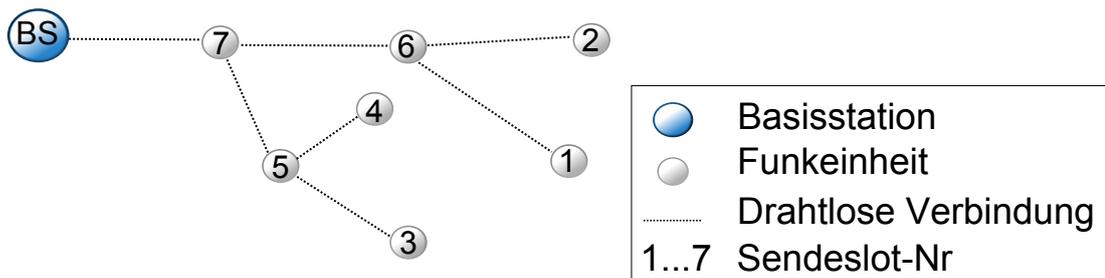


Abbildung 4.22.: Beispiel Sendeslotzuordnung

Ein Schedule-Eintrag für eine Funkeinheit beinhaltet neben dem Sende- und Empfangsplan eine Adresse. An diese Adresse schickt eine Funkeinheit seine Messdaten. Diese Adresse wurde aus dem Graphen entnommen. Damit eine Funkeinheit später andere Funkeinheiten synchronisieren kann, benötigt es die Angabe der Kindknoten seiner direkten Nachbarn.

Damit sind die nachfolgenden Funkeinheiten gemeint, die im späteren Verlauf im Superframe Daten zum Vaterknoten weiterleiten. Auch erhält eine Funkeinheit die Information, wie viele Messdaten zu erwarten sind bzw. wie viele Funkeinheiten nach ihr folgen. Diese benötigt die Funkeinheit in der Synchronisierungsphase.

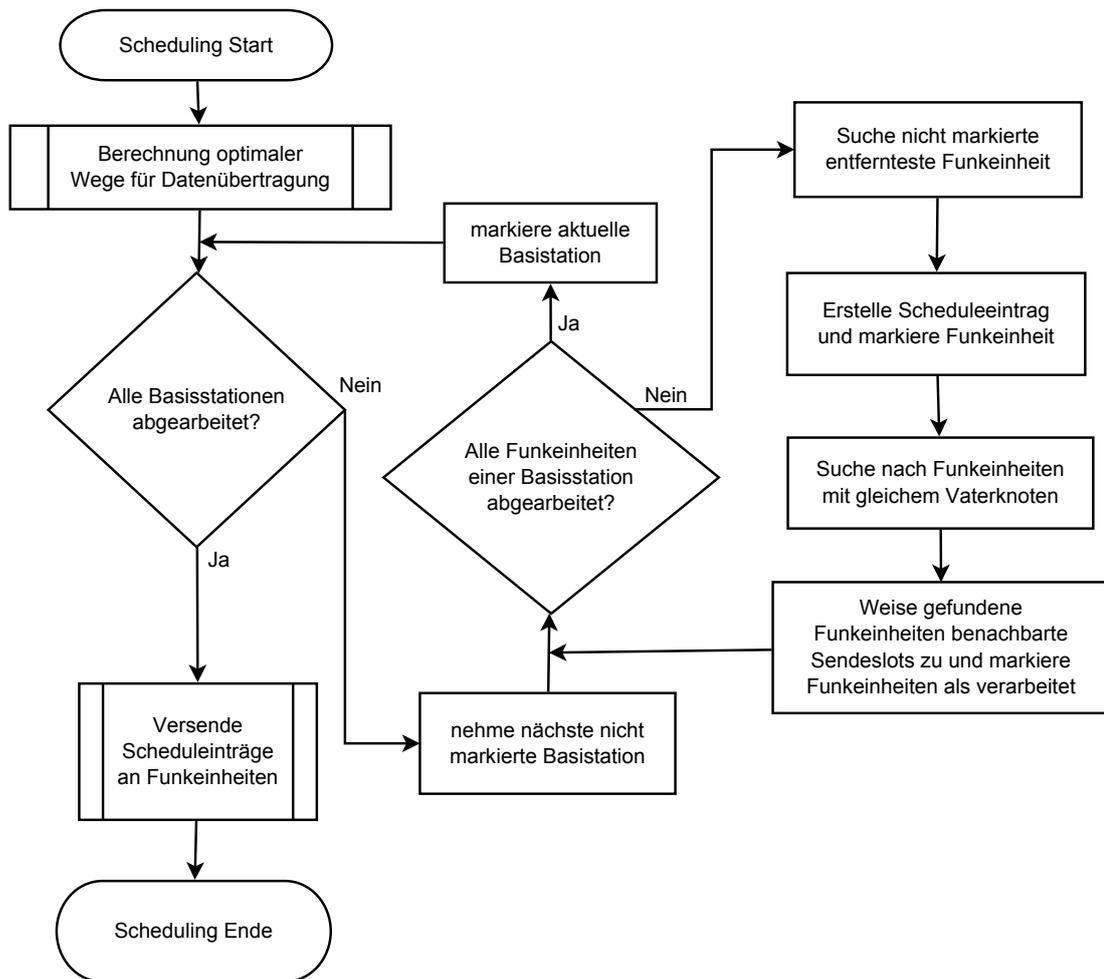


Abbildung 4.23.: Flussplan Scheduling

Nach Erstellung der Sende- und Empfangspläne einer Basisstation erfolgt das Scheduling eventuell weiterer Basisstationen, die gefunden wurden. Sind alle Basisstationen abgearbeitet, werden die Schedulingeinträge an die Funkeinheiten verschickt. Der gesamte Ablauf zum Scheduling ist in der Abbildung 4.23 dargestellt.

#### 4.7.6. Zeitsynchronisierung

Für eine gemeinsame Zeitbasis des Superframes wird eine Zeitsynchronisierung benötigt. Dabei werden Nachrichten mit Angabe der aktuellen Zeit verschickt. Allerdings muss beim Empfänger eine Korrektur der mitgeschickten Zeit erfolgen, da die Nachricht nach dem Sendevorgang eine bestimmte Laufzeit bis zum Empfänger benötigt. Eine solche Synchronisierung

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

wird nun im Detail erläutert.

Eingeleitet wird die Synchronisierung durch den Koordinator. Da bei jeder Netzwerkerkundung einer Basisstation alle Informationen veröffentlicht werden, ist die Gesamtanzahl der Funkeinheiten im Netzwerk bekannt. Der Koordinator kann durch Auswerten der Informationen feststellen wie viele Synchronisierungsnachrichten verschickt werden und berechnet den Startzeitpunkt des Superframes. Anschließend erhält jede Basisstation eine Nachricht, in welcher Zeit sie ihre Synchronisierung durchführen können.

Die Laufzeit der Nachrichten muss in der Realisierung gemessen werden, da die genaue Hardwarezusammensetzung noch nicht bekannt ist. Der Koordinator selber erhält den letzten Zeitpunkt für die Synchronisierung der Funkeinheiten.

Ist der Synchronisierungszeitpunkt einer Basisstation erreicht, wird mit Hilfe des lokalen Graphens errechnet, wie viele Nachrichten er insgesamt verschicken wird. Nachfolgend versendet die Basisstation sequentiell Synchronisierungsnachrichten an jeweils eine Funkeinheit. Der Abstand zwischen zwei Synchronisierungsnachrichten errechnet sich durch die Anzahl nachfolgender Funkeinheiten. Werden Synchronisierungsnachrichten nicht sequentiell abgearbeitet, besteht die Gefahr von Kollisionen durch zeitgleich abgeschickte Nachrichten. Hat eine Funkeinheit keine „Kindknoten“ so sendet die Basisstation die nächste Nachricht nach einem Zeitabstand. Dieser Zeitabstand wird auch im Realisierungsschritt festgelegt in wird im Millisekundenbereich liegen.

Besitzt eine Funkeinheit  $n$  Kinder, so wartet die Basisstation das  $(n+1)$ -Fache eines Zeitabstandes. Es wird so einer Funkeinheit genug Zeit gegeben seine Nachricht zu verschicken. Der genaue Synchronisierungsvorgang bei den Funkeinheiten ist im Abschnitt 4.8.2 erklärt.

Hat eine Basisstation alle benachbarten Funkeinheiten jeweils eine Synchronisierungsnachricht zukommen lassen, wartet diese auf den Start des Superframes. Danach wechselt die Basisstation für ankommende Messdaten in dem Empfangszustand.

Ein Beispiel ist in der Abbildung 4.24 gegeben. In dieser Beispielsituation sind drei Basisstationen und sechs Funkeinheiten vorhanden. Gezeigt wird die Reihenfolge der Synchronisierungsnachrichten und der Sendezeitpunkt der Nachrichten.

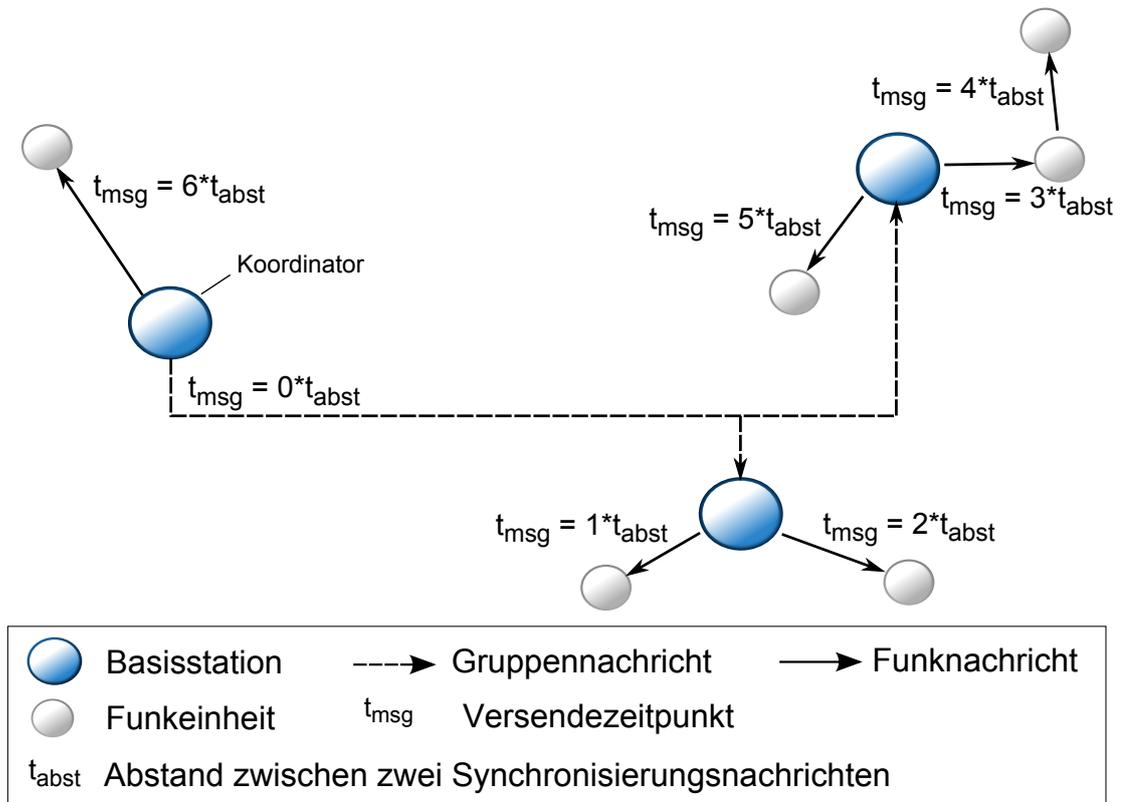


Abbildung 4.24.: Beispiel Zeitsynchronisierung

## 4.8. Funkeinheit

Eine Funkeinheit wartet allgemein immer auf Aufträge von Basisstationen. Sie sucht nach anderen Funkeinheiten in der Umgebung, leitet Daten weiter und sendet seine Messdaten in Richtung Basisstation. Einzelheiten zu diesen Aufgaben finden sich in diesem Abschnitt.

### 4.8.1. Verhalten der Funkeinheit

Nach Gerätestart befindet sich eine Funkeinheit im Empfangsmodus und wartet auf ankommende Daten. Dabei reagiert eine Funkeinheit auf Broadcastnachrichten und auf Nachrichten mit dessen Adresse. Da eine Funkeinheit nach Gerätestart noch keine Adresse besitzt, reagiert eine Funkeinheit nur auf Broadcastmeldungen. Wenn eine HelloRequest-Nachricht eintrifft, wird berechnet, ob sie auf diese Nachricht reagieren soll. Wie im Abschnitt 4.7.3 erklärt soll dies eine Massentrückmeldung verhindern. Nach einem positiven Vergleich wird eine

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

HelloResponse-Nachricht verschickt. Anschließend wird jede neu entdeckte Funkeinheit eine Adresse zugewiesen. Ab diesem Zeitpunkt kann eine Funkeinheit über ihre eigene Adresse angesprochen werden.

Daraufhin folgend erhält die Funkeinheit bald einen Auftrag von einer Basisstation seine Umgebung nach weiteren Funkteilnehmern abzusuchen. Ist dies abgearbeitet, wird das Ergebnis an der Basisstation zurückgesendet. Besitzt die Funkeinheit weitere Nachbarn, so kann die Funkeinheit als Zwischenstation zur Nachrichtenweiterleitung genutzt werden. Dies ist davon abhängig, wie die Basisstation die Routen zu neuen Funkeinheiten bewertet hat.

Sind alle erreichbaren Funkeinheiten von einer Basisstation aus erkundet, werden alle Funkeinheiten mit Informationen für den Superframe sowie dessen Synchronisierung versorgt. Eine Funkeinheit weiß nun Sende- und Empfangszeitpunkte, wohin Messdaten versendet werden und nähere Informationen zu Nachbarfunkeinheiten. Danach folgt der Synchronisierungsschritt, der im Abschnitt [4.8.2](#) erklärt wird.

Befindet sich eine Funkeinheit in einem zugeteilten Slot, so verlässt sie den Energiesparmodus, aktiviert den Transceiver und führt den Sendevorgang aus bzw. ist empfangsbereit. Falls ein benachbarter Sende/Empfangslot zugeteilt wurde, geht die Funkeinheit beim Übergang vom Empfangen zu Senden nicht in den Energiesparmodus. Nach erfolgtem Sendevorgang begibt sich eine Funkeinheit wieder in einen Tiefschlaf bis das Ende des Superframes erreicht wurde. Diese Zeitangabe ist vor dem Superframe festgelegt worden.

Nach dem Ende des Superframe ist eine Funkeinheit wieder empfangsbereit und wartet entweder auf einen erneuten Beginn des Superframes oder es bekommt den Auftrag seine Umgebung zu Erkundung. Dies wird vom aktuellen Koordinator bestimmt. Dieses Verhalten ist auch im [Anhang](#) in der Abbildung [A.1](#) als Zustandsautomat dargestellt.

##### **4.8.2. Synchronisierung**

Voraussetzung einer Synchronisierung ist eine Adresszuweisung und ein Sende/Empfangsplan. Zum Sende/Empfangsplan werden die Informationen weitergegeben, wie viele Kindknoten ein Nachbar hat. Eine Funkeinheit kann so einschätzen, wie lange eine benachbarte Funkeinheit braucht, um seine Nachbarn (inklusive weiterfolgende Kindknoten) zu synchronisieren. Da Funkeinheiten Synchronisierungsnachrichten nur an ihre Kindknoten weiter senden, wird

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

doppeltes Synchronisieren verhindert.

Jede Funkeinheit hat in der Schedulingphase eine Liste mit den Nachbarn erhalten, die von ihr synchronisiert werden soll. Der Vorgang zum Synchronisieren ist auch in Abbildung 4.25 ersichtlich. Wie bei der Basisstation wartet eine Funkeinheit bei Nachbarn ohne weitere Kindknoten den einfachen Zeitabstand bis zur nächsten Synchronisierungsnachricht und bei benachbarten mit  $n$  Kindknoten  $(n+1)$  Zeitabstände.

Hat eine Funkeinheit alle zugeteilten Funkeinheiten synchronisiert, wird auf den Anfang des Superframes gewartet. Eine Funkeinheit begibt sich daraufhin im Energiesparmodus, wenn sie nicht gerade im ersten Sende- bzw. Empfangsslot aktiv ist.

Ob eine Funkeinheit einem Empfangsslot zugewiesen wurde, hängt von der Position der Funkeinheit in der Struktur des Netzwerkes ab. Ist diese am „Rand“ des Netzwerkes, so muss diese nur ihre Daten senden. Daten wird die Funkeinheit dort nicht erhalten, da es keine Funkeinheiten hinter ihr existieren.

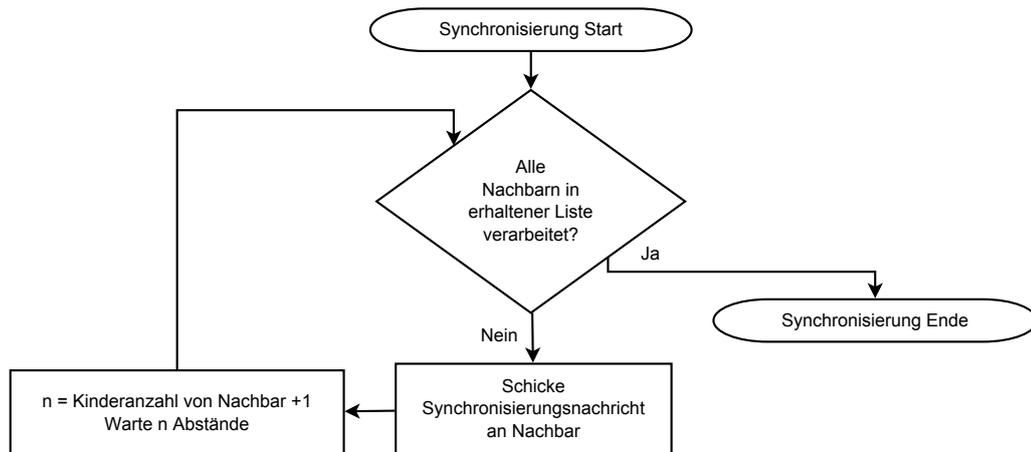


Abbildung 4.25.: Flussplan Synchronisierung Funkeinheit

### 4.9. Gruppenkommunikation

Wie im vorherigen Abschnitt erläutert, kommunizieren die Basisstationen mittels Gruppenkommunikation. In der Tabelle 4.2 sind die benötigten Nachrichtenarten dargestellt.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Nachrichtenart	Verwendungszweck
Systemstart	Leitet die Koordinatorwahl ein
Wahlnachricht	Wahlbekundung einer Basisstation
Wahlende	Neuer Koordinator gefunden
Meldeaufforderung	Listenerstellung aktiver Basisstationen
Rückmeldung	Antwort auf Meldeaufforderung
Gefundene Funkeinheiten	Funkeinheiten nicht doppelt verarbeiten
Nächste Adresse	Nächste nutzbare Adresse für eine Funkeinheit
Nächste Zeitslot-Nr	Nächste nutzbarer Slotnummer für das Scheduling
Verarbeitungsstart	Startbefehle an Basisstationen
Superframevorbereitung	Vorbereitungsinformationen für eine Basisstation
Synchronisierungsnachricht	Gemeinsame Zeitbasis im Netzwerk

Tabelle 4.2.: Überblick über die Gruppennachrichten

Für das Datenformat wurde eine einfache Lösung hergenommen. Die Angabe für die Größe der Daten sowie das Absenderfeld ist fester Bestandteil. Die danach folgenden Daten sind je nach Nachrichtenart individuell aufgebaut.

Bytegröße	Bezeichnung	Funktion
4	msg_dataSize	Gesamtgröße der Nachricht
1	msg_src_id	Absender-ID
1	msg_dataType	Nachrichtenart
n	msg_data	Nachrichtendaten

Tabelle 4.3.: Allgemeiner Aufbau Gruppennachrichten

Einige Nachrichtenarten benötigen keine zusätzlichen Angaben. Die Information zur Nachrichtenart mit Absenderkennung genügt. Dies ist der Fall bei Systemstart, Wahlnachricht, Wahlende, Meldeanforderung und Rückmeldung. Bei Verarbeitungsstart, „Nächste Adresse“ und „Nächste Zeitslot-Nr“ kommt jeweils ein 8Bit-Feld hinzu. Der Aufbau der Nachrichtenarten „Gefundene Funkeinheiten“, „Synchronisierungsnachricht“ und „Superframevorbereitung“ sind in den folgenden Tabellen ersichtlich.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Bytegröße	Bezeichnung	Funktion
8	macAddress	Mac-Adresse der Funkeinheit
1	id	Zugewiesene ID
1	baseRelation	Zugewiesene Basisstation
1	childCount	Anzahl nachfolgender Funkeinheiten
1	isNBofBase	Angabe, ob Funkeinheit Nachbar einer Basisstation ist

Tabelle 4.4.: Aufbau Gefundene Funkeinheiten

Bytegröße	Bezeichnung	Funktion
4	actualTime	Aktuelle Zeit beim Sender kurz vor dem Sendevorgang

Tabelle 4.5.: Aufbau Synchronisierungsnachricht

Bytegröße	Bezeichnung	Funktion
1	slotNr	Synchronisierungszeit für Basisstation
1	syncEntities	Anzahl zu verschickender Synchronisierungsnachrichten
2	superframeBegRel	Relative Zeitangabe zum Superframeanfang

Tabelle 4.6.: Aufbau Superframevorbereitung

### 4.10. Funkkommunikation

Nachrichtenaustausch zwischen einer Basisstation und einer Funkeinheit sowie zwischen zwei Funkeinheiten erfolgt mittels der Funkschnittstelle. Für die Organisation der Nutzdatenübertragung werden verschiedene Nachrichtenarten benötigt, diese sind in Tabelle 4.7 zusammengefasst.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Nachrichtenart	Verwendungszweck
Hello request	Anforderung zur Rückmeldung
Hello response	Rückmeldung zu Hello request
Ping request	Anforderung zur Rückmeldung bei gesetzter ID
Ping response	Rückmeldung zu Ping request
Discovery task	Funkeinheit soll seine Umgebung erforschen
Data acknowledge	Rückmeldung zum Datenerhalt
Routing data	enthält Routinginformationen
Routing data finished	Routinginformationen wurden abgearbeitet
Schedule plan	enthält Sende- und Empfangsplan
Kindknoteneintrag	Beinhaltet Anzahl Kindknoten einer Funkeinheit
Synchronising	Synchronisierungsmittelung für die gemeinsame Zeit
Sensor data	Messdaten einer Funkeinheit

Tabelle 4.7.: Überblick über die Funknachrichten

Im Header müssen zusätzliche Daten enthalten sein. Ein gerouteter Frame enthält neben der Adresse der nächsten Funkeinheit (die zur Weiterleitung der Daten dient) die Adresse des Empfängers. Da auch größere zusammenhängende Datenmengen übertragen werden, kommen Felder zur kompletten Größenangabe der Daten sowie Nummer des aktuellen Frames hinzu.

Bytegröße	Bezeichnung	Funktion
2	frame control	MAC header: Frameeigenschaften
1	sequence number	MAC header: Sequenznummer
2	Destination_pan_address	MAC header: Ziel-Pan
2	Destination_address	MAC header: Ziel
2	Source_address	MAC header: Quelle
1	data_type	Nachrichtenart
1	source_id	Absender der Nachricht
1	destination_id	Empfänger der Nachricht
1	frame_number	Nummer des aktuellen Frames
1	frame_amount	Anzahl Frames einer Datenübertragung
1	data_size	Anzahl Bytes einer Datenübertragung

Tabelle 4.8.: Headerstruktur

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Einige Nachrichtenarten benötigen neben dem Header keine weitere Informationen. Dies sind z.B. alle Pingnachrichten, „Discovery\_task“, „Data Acknowledge“ und „Routing data finished“. In „Sensor data“ sind die reinen Messdaten enthalten und werden individuell von der Basisstation bzw. vom weiterverarbeitenden System interpretiert.

Bytegröße	Bezeichnung	Funktion
1	sendToID	nächstes Ziel der Messdaten
1	id	ID der Funkeinheit
2	slotNr_rx	Nummer des Empfangsslots
2	slotSize_rx	Anzahl der Empfangsslots
2	slotNr_tx	Nummer des Sendeslots
2	slotNr_tx	Anzahl der Sendeslots

Tabelle 4.9.: Nachrichtenaufbau Schedule plan

Bytegröße	Bezeichnung	Funktion
1	hop_id	ID Funkeinheit
1	destination_id	ID Endziel
1	nextHop_id	ID nächste Funkeinheit

Tabelle 4.10.: Nachrichtenaufbau Routing data

Tabelle 4.9, 4.10 und 4.11 zeigt den Aufbau des Sende/Empfangplans, die eines Routingeintrages und die der Synchronisierung.

Bytegröße	Bezeichnung	Funktion
4	actualTime	Aktuelle Zeit beim Sendevorgang
4	superframeBegin	Anfang des Superframes

Tabelle 4.11.: Nachrichtenaufbau Synchronisierung

In Abbildung 4.12 ist der Aufbau einer HelloRequest-Nachricht gezeigt. Hier sind die benötigten Informationen zu sehen, die ein Funkeinheit für eine Antwort benötigt. Abbildung 4.13 zeigt die Inhalte einer HelloResponse-Nachricht. Diese Daten werden in dem Graphen der Basisstation gespeichert.

#### 4. Entwurf von Entwicklungsumgebung und Funksystem

---

Bytegröße	Bezeichnung	Funktion
1	bitMask	Vergleichsfeld
1	bitSelection	Bitauswahl für den Vergleich

Tabelle 4.12.: Nachrichtenaufbau Hello Request

Bytegröße	Bezeichnung	Funktion
1	type	Funktion des Funkteilnehmers
1	id	aktuelle id
1	lqi	Empfangsstärke
1	mac	MAC-Adresse
1	battery	Batteriezustand

Tabelle 4.13.: Nachrichtenaufbau Hello Response

Schließlich ist in der Abbildung 4.14 die Nachricht zu den Kindknoteneintrag vorhanden. Diese Nachricht wird in der Schedulingphase verschickt. Eine Funkeinheit weiß dann wie viele Funkeinheiten nach ihr folgen.

Bytegröße	Bezeichnung	Funktion
1	id	ID der Funkeinheit
1	childCount	Anzahl nachfolgender Funkeinheiten

Tabelle 4.14.: Nachrichtenaufbau Kindknoteneintrag

## 5. Realisierung

In diesem Kapitel wird das Design umgesetzt. Zuerst erfolgt die Auswahl der passenden Hardware. Diese wird anschließend miteinander verbunden, so dass die Software entwickelt werden kann. Die Komponenten des Systems werden im Einzelnen erklärt. Dabei erfolgt zuerst die Fertigstellung der Entwicklungsumgebung. Danach wird das Funksystem implementiert. Abschließend wird anhand einer Beispielrechnung die voraussichtliche Betriebszeit einer Funkeinheit berechnet.

### 5.1. Hardwareauswahl

Für die Hardwareauswahl muss ein Mikrocontroller mit einem 802.15.4 Funkchip gewählt werden. Für den Geräteagent sollte am besten ein Mikrocontroller eingesetzt werden, der USB-Schnittstelle aufweist und bei dem die Software möglichst komfortable erstellt werden kann.

Die Firma Dresden Elektronik Ingenieurtechnik GmbH bietet viele Funkchips, die nach dem Standard 802.15.4 funken. Die Produktparte deRFmega128 erfüllt alle nötigen technischen Voraussetzungen. Es ist eine AVR Single-Chip-Lösung, die den Mikrocontroller und Funkchip beinhaltet.



Abbildung 5.1.: deRFmega128-22Ao2 und Board, (vgl. [DresdenElektronik 2012](#))

## 5. Realisierung

---

Der Takt wird von einem 16Mhz-Quarz erzeugt. Verschiedene Energiesparmodi sind möglich und besitzt einen asynchron laufenden 32,768KHz-Quarz. Im Datenblatt ist ein maximaler Stromverbrauch von ca. 18mA angegeben.

Für die Anbindung an Sensorik bietet dieser Chip verschiedene Schnittstellen (UART, I2C, ADC, GPIO, SPI). Zusätzlich lässt sich die Reichweite mittels einer externen Antenne erweitern (vgl. [DresdenElektronik 2012](#)). Das Unternehmen bietet neben den reinen Funkchips auch Zubehör wie passende Platinen und Batteriehalterungen an.

Weitere Recherche ergab, dass viele andere Entwicklungsboards nur für Zigbee ausgelegt sind. Es gibt für diese Boards lediglich eine Software dazu, mit der man die Zigbee-Parameter und die Sensoranbindung einstellen kann. Daher wird als Alternative ein separates Funkmodul genutzt. Als Mikrocontroller sollte ein AVR ATmega64 ausreichen. Seine Eckdaten sind:

- 64 Kbyte Flash Speicher, 4 Kbyte ROM
- kann bis 16 MHz getaktet werden
- SPI, I2C, RTC (Echtzeituhr 32,768KHz), USART

Als Funkmodul wird das XBee DigiMesh 2.4 ausgewählt. Es kann für Zigbee als auch für andere Anwendungen genutzt werden, da es den Standard 802.15.4 benutzt. Die Ansteuerung erfolgt mittels einer seriellen Schnittstelle. Als Verbrauch wird bis zu 50mA angegeben (vgl. [DigiwirelessSolutions 2012](#)).



Abbildung 5.2.: XBee DigiMesh 2.4, (vgl. [DigiwirelessSolutions 2012](#))

Durch den einfacheren Umgang, niedrigeren Stromverbrauch und weniger Arbeit bei der Benutzung sowie Ersteinrichtung ist das Produkt deRFmega128-22A02 von der Firma Dresden Elektronik Ingenieurtechnik GmbH besser geeignet. Dadurch kann mehr Arbeit in die Entwicklung des Funkprotokolls einfließen. Ein Verdrahten der Bauteile wie in der Lösung mit

## 5. Realisierung

---

einem ATmega64 und dem XBee DigiMesh 2.4 ist nicht notwendig.

Als Geräteagent-Hardware wird das „LPC1343 QuickStart Board“ ausgewählt. Es besitzt einen ARM M-3 Prozessor mit bis zu 72MHz und diversen Schnittstellen. Eine USB-Schnittstelle ist direkt auf der Platine vorhanden. Die Softwareaufspielung erfolgt per USB, es wird keine zusätzliche Hardware benötigt. Durch auflötbare Steckleisten lässt sich die Hardware leicht auf einem Breadboard mit anderen Geräten verbinden.

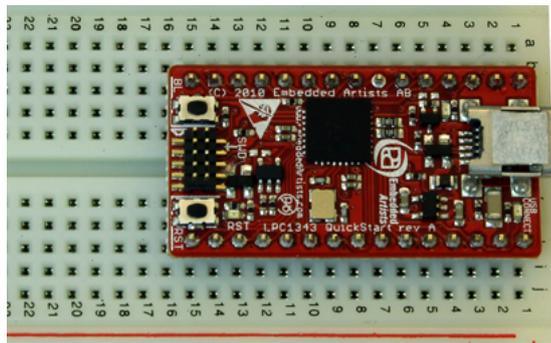


Abbildung 5.3.: LPC1343, (vgl. [embeddedArtists 2012](#))

Es sind noch 2 Computer nötig, die für jeweils eine Basisstation als Hilfsmittel für die Gruppenkommunikation dienen soll. Die Funkeinheiten können auch mit den Computern verbunden werden, um neue Software auf die Geräte zu spielen. Da keine Besonderen Anforderungen neben USB- und Ethernetschnittstelle bestehen, können normale Workstations genutzt werden.

### 5.1.1. Zusammenspiel der Komponenten

Für die Entwicklung des Systems stehen 2 Räume zu Verfügung. Verbunden sind die Räume mit Gigabit-Ethernet, in je einem Raum wird eine Linux-Workstation eingerichtet. Die Anbindung an die den LPC1343 erfolgt per USB-Kabel, worauf später eine virtuelle serielle Verbindung in Betrieb genommen wird. Der LPC1343, der als Geräteagent dient, wird mit dem deRFmega128 verbunden. Der Funkchip erhält Anschluss zur 3,3V Spannungsversorgung, eine Masseverbindung, Sende- und Empfangsleitung der seriellen Verbindung und eine Resetleitung zum RSTN-PIN des AVR-Chips. Das Zusammenspiel für eine Linux-Workstation und zwei Funkchips ist in [Abbildung 5.4](#) zusammengefasst.

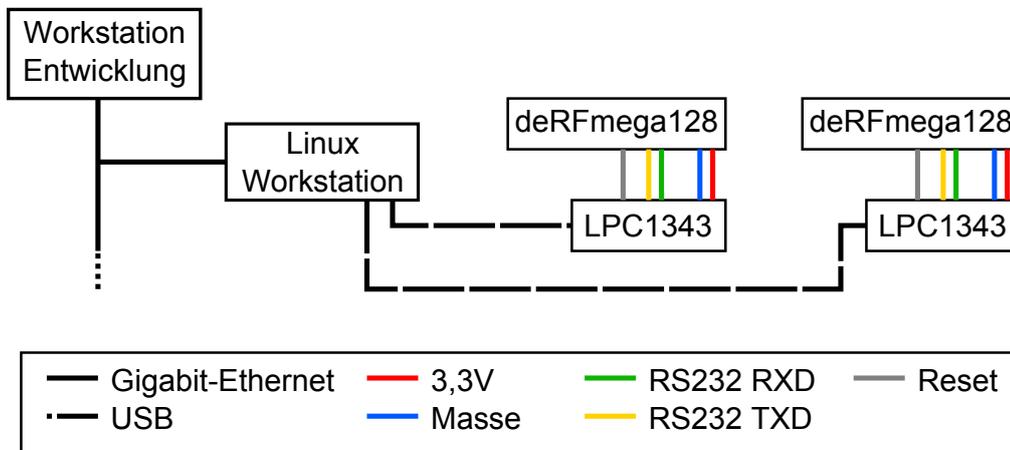


Abbildung 5.4.: Zusammenspiel der Geräte

## 5.2. Entwicklungsumgebung

Die Entwicklungsumgebung dient zum Aufspielen neuer Software auf die Geräte, setzt sie zurück und dient für die Basisstationen als Hilfsmittel für die Gruppenkommunikation.

### 5.2.1. Bootloader

Ein Bootloader wird in erster Linie dazu genutzt das Hauptprogramm eines Mikrocontrollers entgegen zunehmen und es in den Flashspeicher zu schreiben. Danach kann es vom Bootloader aus gestartet werden. Daneben kann ein Bootloader mit weiteren Funktionen ausgestattet werden.

Folgende Funktionalitäten soll dieser Softwareteil erfüllen:

- neues Hauptprogramm in den Flashspeicher schreiben
- schreiben und auslesen der Identität: damit sind der Gerätenamen, die MAC-Adresse und die Geräteklasse gemeint
- Rückmeldung zum Kommunikationspartner, dass sich das Gerät im Bootloader befindet

Der Bootloader wird mit der Entwicklungsumgebung AVR-Studio entwickelt. Zum Aufspielen auf das Gerät kommt der AVR-Dragon zum Einsatz. Jede weitere Aufspielung von Software nach

Entwicklung des Bootloaders geschieht mittels der seriellen Schnittstelle. Der AVR-Dragon wird dann nicht mehr benötigt.

### 5.2.2. Geräteagent

Der Geräteagent dient einerseits zum Zurücksetzen der Funkhardware sowie zur Weiterleitung der Daten von der Linux-Workstation zum deRFmega128 und wieder zurück.

Dieses Hilfssystem wird jedoch bei den Funkeinheiten nur in der Entwicklungsphase benötigt. Im Normalbetrieb wird keine weitere Anbindung als die der Funkverbindung benötigt. Nur die Basisstationen sind weiterhin mit dem System gekoppelt, da die Messdaten weitergeleitet werden müssen. Dies kann bei späterer Weiterentwicklung auch wegfallen, wenn die Basisstationen ein eigenes Bussystem zur Kommunikation benutzen.

Die Entwicklung erfolgt in der Entwicklungsumgebung LPCXpresso, verschiedene Bibliotheken für die Ansteuerung von Schnittstellen sind bereits vorhanden. Diese werden für die serielle RS232-Verbindung zum AVR bzw. zur Linux-Workstation genutzt.

Eine Nachricht besteht aus einem Header sowie dem Nachrichteninhalte. Ein Header beinhaltet einen Befehl und die Datenmenge der Nachricht, die nach dem Header zu erwarten sind. Bei dem Befehl kann es sich um eine allgemeine Weiterleitung oder um einen Auftrag für den Geräteagent handeln. Das Verhalten dazu ist in der Abbildung 5.5 zu sehen.

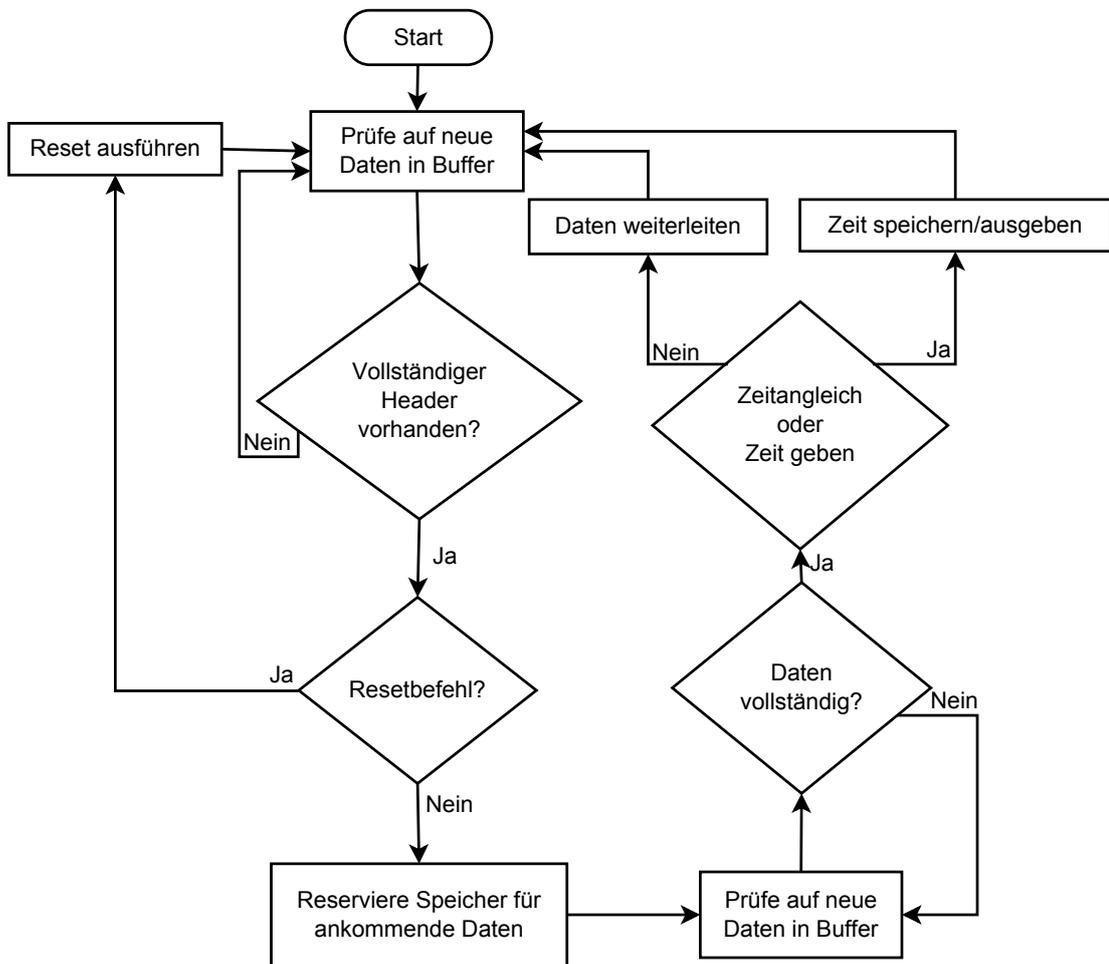


Abbildung 5.5.: Flussplan des Geräteagents

Der Inhalt einer Nachricht ist in Tabelle 5.1 gezeigt:

Byteanzahl	1	4	n
Feldbeschreibung	Typ	Datenlänge	Daten

Tabelle 5.1.: Nachrichtenaufbau

Der Typ einer Nachricht hängt von Verwendungszweck ab. Es gibt synchrone Kommunikation, Lognachrichtenübertragung, Gruppennachrichten und Ausführungsaufträge an den LPC. Diese sind in Tabelle 5.2 zusammengefasst.

## 5. Realisierung

Nachrichtentyp	Umgang mit Nachricht
Synchrone Kommunikation	Weiterleitung
Lognachrichten	Weiterleitung
Gruppennachrichten	Weiterleitung
Reset	Ausführung
Zeitangleich	Ausführung
Zeitabfrage	Ausführung

Tabelle 5.2.: Nachrichtentypen

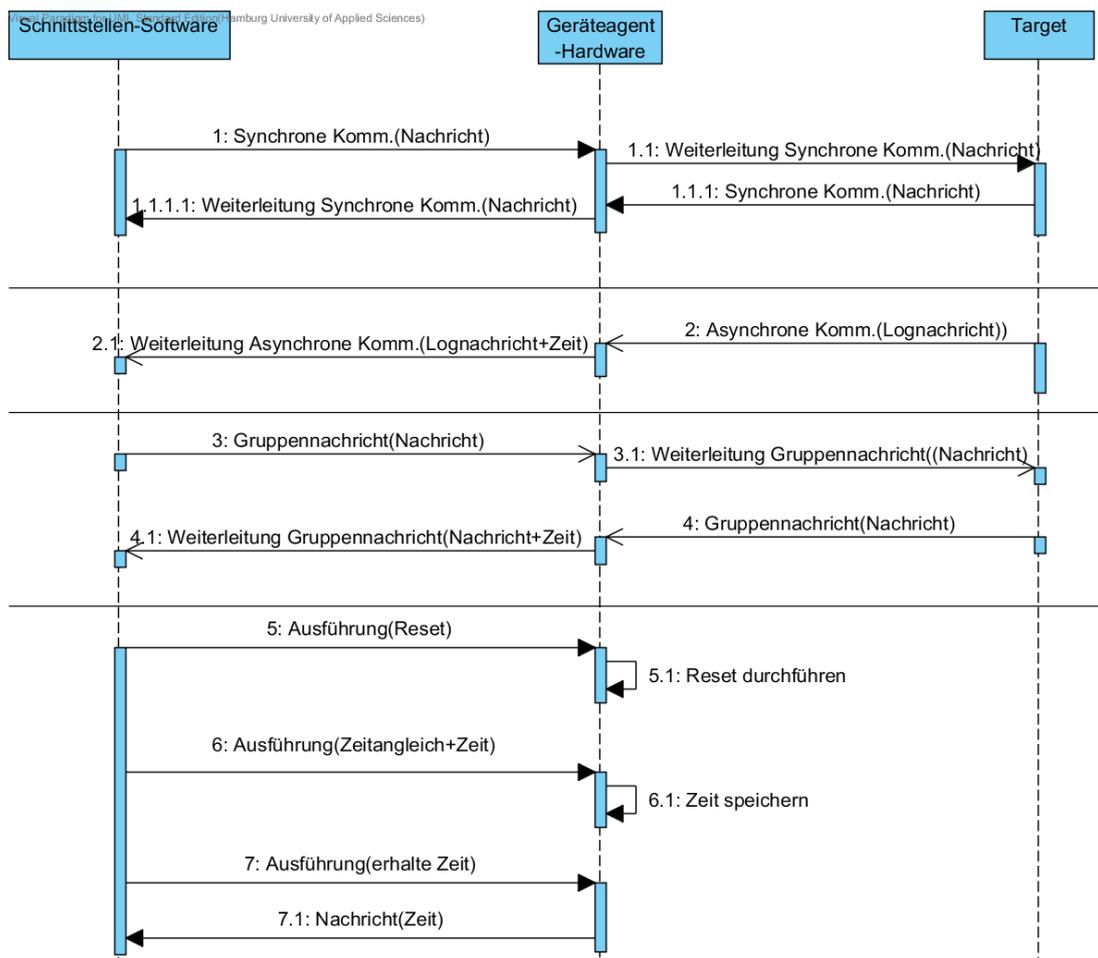


Abbildung 5.6.: Sequenzdiagramm Nachrichtenbehandlung

## 5. Realisierung

Ein Sequenzdiagramm in Abbildung 5.6 zeigt den allgemeinen Verlauf der Nachrichtenverarbeitung. Dabei werden die Nachrichten entweder weitergeleitet oder es wird ein Befehl ausgeführt.

Im Laufe der Entwicklung hatte der LPC1343 bei größeren Datenaufkommen Schwierigkeiten. Zum Teil trafen Daten nicht in der Schnittstellensoftware ein. Abhilfe schaffte eine kurze Sendepause zwischen zwei Datenblöcken. Nach dieser Änderung wurden Daten vollständig weitergeleitet.

### 5.2.3. Zentrale Benutzeroberfläche

Die Benutzeroberfläche erlaubt es mit der Schnittstellensoftware zu kommunizieren. Damit können Befehle an die Schnittstellensoftware gesendet werden, um z.B. eine Reset am Gerät durchzuführen. Von der GUI aus werden Verbindungen aufgebaut und die Funkchips lassen sich hierüber steuern. Für die Entwicklung wird Java verwendet, die Oberfläche wird mit Swing dargestellt. Einen Ausschnitt der Oberfläche ist in der Abbildung 5.7 zu sehen, ein Beispiel für die Logausgabe beinhaltet Abbildung 5.8.

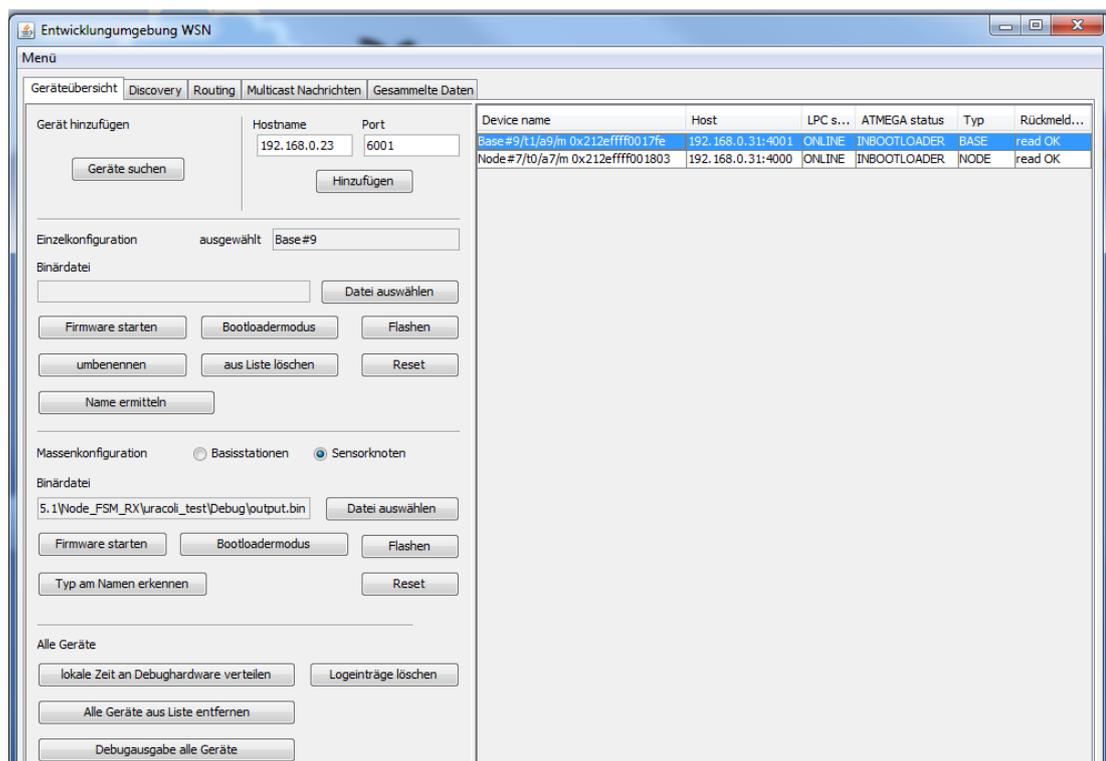


Abbildung 5.7.: Zentrale Benutzeroberfläche

## 5. Realisierung

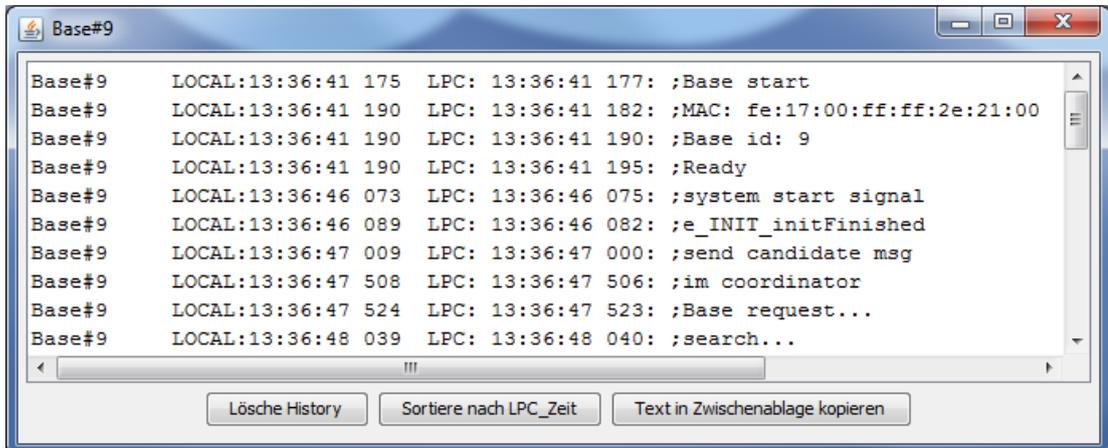


Abbildung 5.8.: Logfenster

### 5.2.4. Schnittstellensoftware

Die Schnittstellensoftware wurde auf den Linux-Workstations entwickelt. Dafür kommt Eclipse mit dem C/C++ Plugin zum Einsatz. Die Aufgaben werden auf Threads verteilt. Ein Thread wartet auf ankommende Daten von der Anwendungsoberfläche (TCP), der zweite lauscht auf der seriellen Schnittstelle zum LPC1343. Der dritte Thread warte auf Gruppennachrichten bzw. erstellt Gruppennachrichten (UDP-Multicast). Da mehrere Threads an der Kommunikation beteiligt sind, müssen diese synchronisiert werden. In der Linux-Umgebung steht für diesen Zweck die POSIX-Bibliothek bereit.

In der Abbildung 5.9 ist die Threadkoordinierung allgemein dargestellt. Zu jeder Verbindung gibt es ein Thread. Eine gleichzeitige Ausführung von Sendebefehlen wird von Mutexen verhindert. Da zwischen den Threads der TCP-Verbindung und der seriellen Verbindung Daten ausgetauscht werden müssen, erfolgt die Verarbeitung mittels Semaphoren und gemeinsame Datenpuffer.

## 5. Realisierung

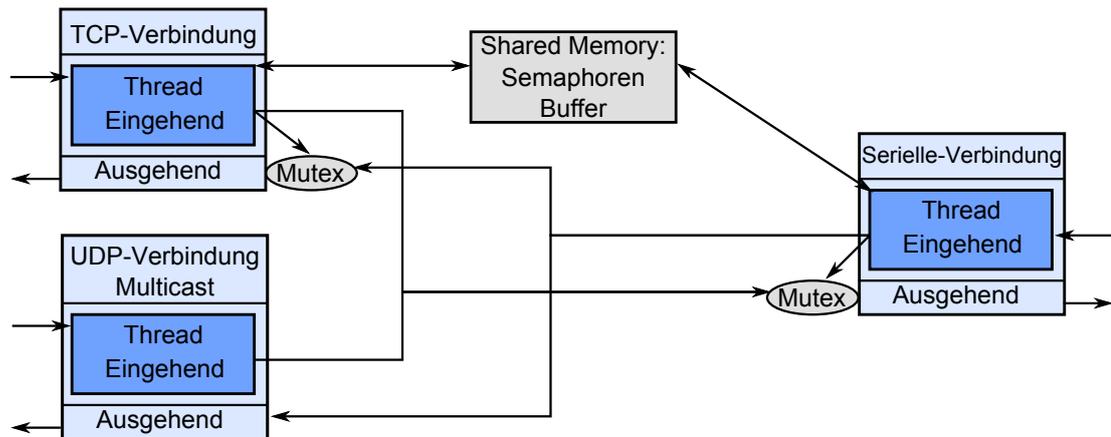


Abbildung 5.9.: Threadkoordinierung

Für zeitkritische Situationen muss die Verzögerung der Gruppennachrichten bekannt sein. Für die Messung dieser Zeit wurde eine Anwendung für 2 Funkchips geschrieben. Beide sind mit einer Leitung verbunden, die einen Interrupt auslösen soll. Gerät A speichert den Startzeitpunkt und verschickt eine Gruppennachricht. Gerät B erhält die Nachricht und verändert den Leitungspiegel zu Gerät A. In Gerät A wird ein Interrupt ausgelöst und es wird der Endzeitpunkt gespeichert.

Die Zeitdifferenz zwischen Start- und Endzeitpunkt wird anschließend in das Log gespeichert. Dieser Vorgang wird 10000-Mal wiederholt. Die Daten aus dem Log werden mittels Tabellenkalkulation ausgewertet und es wird eine Häufigkeitstabelle erstellt. Diese ist in [Abbildung 5.10](#) einzusehen.

Zeit in $\mu\text{s}$	4636	4880	5124	5368	5612	5856	6100	6832	8540	10004
Häufigkeit	76	1730	2445	2425	2426	893	2	1	1	1

Tabelle 5.3.: Tabelle Nachrichtenverzögerung

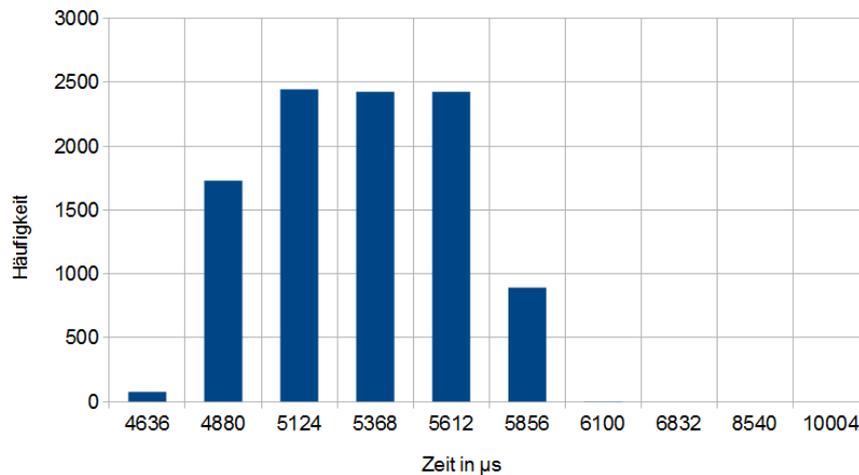


Abbildung 5.10.: Diagramm Nachrichtenverzögerung

Die Verzögerung schwankt im Mittelfeld bei den häufigsten Zeiten um  $976 \mu\text{s}$ . Im Design wurde für diese Verzögerung  $1\text{ms}$  eingeplant. Mit den Verzögerungen kann das Funksystem demnach noch umgehen.

### 5.3. Funksystem

Das Funksystem wird aus 2 Softwarebestandteilen aufgebaut: Die Software für die Basisstation und die der Funkeinheit. Softwaremodule für Sensoren lassen sich einfach hinzufügen.

#### 5.3.1. Softwareentwicklung

Das Hauptprogramm für die Basisstationen und die Funkeinheiten können unterschiedlich implementiert werden. Zur Auswahl stehen zwei Möglichkeiten

- TinyOS: Ist ein kleines Betriebssystem, das nur wenig Ressourcen verbraucht. Es ist speziell für den Bereich der drahtlosen Sensornetze entwickelt worden. TinyOS ist in einem C-Dialekt, nesC, geschrieben. Es wird also eine Entwicklungsumgebung mit einem nesC-Compiler benötigt. Zur Reaktion auf Einflüsse werden Events benutzt. Diese Events steuern dann das Verhalten des Programms
- $\mu\text{racoli}$ : Ist eine Bibliothek, die direkt in einem C-Projekt eingebunden werden kann. Es stehen verschiedene High- und Low-Level-Befehle zu Verfügung, die die Hardware

## 5. Realisierung

---

z.B. initialisieren und steuern. Frames können mit Befehlen in den Buffer geschrieben werden, die anschließend versendet werden

Für die Softwareentwicklung wird `µracoli` verwendet. Die herstellereigene Entwicklungsumgebung AVR-Studio kann benutzt werden, die Einbindung der `µracoli`-Bibliothek ist unkompliziert.

Die Handhabung als Bibliothek gewährleistet weiterhin vollständige Kontrolle beim Programmablauf. TinyOS nutzt bereits ein Grundgerüst zum Programmaufbau und es muss die Programmiersprache nesC verwendet werden.

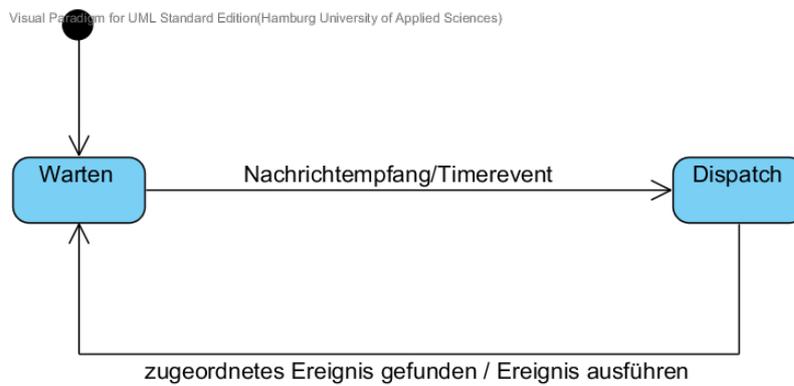


Abbildung 5.11.: Allgemeines Verhalten Zustandsautomat

Für die Funkchips wird jeweils ein Zustandsautomaten implementiert. Dieser Automat soll auf Ereignisse wie den Empfang einer neuen Nachricht oder einen Timer-Callback ausgelöst werden. In der Programmiersprache C wird dafür eine zweidimensionales Array aufgebaut, in der Funktionen aller Zustandsereignisse gespeichert sind. Die Ereignisse werden aufgefangen und je nach Zustand zugeordnet. Der allgemeine dazugehörige Ablauf ist in [Abbildung 5.11](#) gezeigt.

Es gab es beim Synchronisieren der Funkeinheiten kleinere Probleme. Ab und an hat eine Funkeinheit den Sende/Empfangszeitpunkt verpasst. Nach einer Auswertung am Oszilloskop stellte sich heraus, dass die berechneten Zeiten für die Nachrichtenverzögerung nicht optimal waren. Ein schrittweiser Abgleich brachte eine maximale Abweichung der Zeiten von wenigen Mikrosekunden.

### 5.3.2. Sensorik

Als Anwendungsfall soll das System Temperaturmessungen durchführen. Da in dieser Arbeit der Schwerpunkt auf das Funkprotokoll liegt, wird nur eine einfache Sensorverbindung genutzt. OneWire-Sensoren können einfach angesprochen werden und sind kostengünstig. Auch ist die Verdrahtung eines Sensors an den Funkchip einfach.

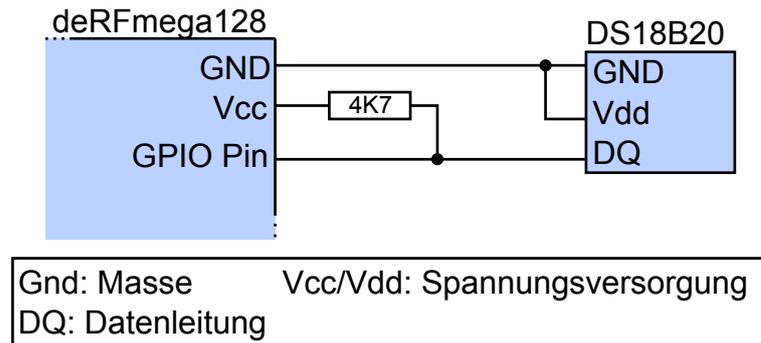


Abbildung 5.12.: Schaltplan One Wire - Parasitär

Die Abbildung 5.12 zeigt die Verdrahtung eines DS18B20-OneWire-Tempersensors. Der Baustein kann parasitär versorgt werden. Das heißt, dass die Energie direkt über die Datenleitung bezogen wird. Deshalb ist der Pin für die Spannungsversorgung an dem Temperatursensor auf Masse gelegt. Dafür ist ein Pull-Up-Widerstand nötig, der zwischen Spannungsversorgung und Datenleitung angeschlossen wird. In einer Messung erhält er seine Energie von der Versorgungsspannung des Funkchips über den Pull-Up-Widerstand.

Der Temperatursensor verbraucht im Standby 750-1000 nA. Eine Messung im parasitären Modus benötigt ca. 1-1,5 mA für 750ms (vgl. Maxim 2012). Dieser Verbrauch ist im Vergleich zum Funkchip recht klein und ist damit als Sensor für den Batteriebetrieb geeignet.

### 5.3.3. Maximale Anzahl der Knoten

Um die Anzahl der Knoten festzustellen, die eine Basisstation verwalten kann, müssen zuerst einige Informationen vorhanden sein. Der Hauptspeicher des AVR beträgt 16 Kilobyte. Von diesen 16 Kilobyte benötigt das Programm fest zugewiesene 4240 Bytes. Einige Datenstrukturen wachsen mit der Anzahl Funkeinheiten, sie allozieren nach Bedarf Hauptspeicher.

Datenstrukturname	Benötigte Bytes für ein Eintrag
Graph	$21+3 \cdot (\text{Anzahl Nachbarn})$
DiscoveredNodes	13
SensorData	10
Summe	$44+3 \cdot (\text{Anzahl Nachbarn})$

Tabelle 5.4.: Dynamisch wachsende Datenstrukturen

Neben diesen Datenstrukturen gibt es weitere Allokationen, die für kurze Zeit benötigt werden. Darunter fallen z.B. das Erstellen von Routeninformationen und einige Algorithmen im Graphen. Dafür werden pauschal 1024 Bytes reserviert. Damit ergeben sich 5264 Bytes für den allgemeinen Programmablauf. Es verbleiben 11.120 Bytes für die Funkeinheiten. Als Beispiel haben die Funkeinheiten im Durchschnitt 5 Nachbarn.

$$\text{Anzahl}_{\text{Funkeinheiten}} = \frac{\text{Bytes}_{\text{zurVerfuegung}}}{\text{Bytes}_{\text{proFunkeinheit}}} = \frac{11120 \text{ Bytes}}{44 \text{ Bytes} + 3 \text{ Bytes} \cdot 5} = 188,47$$

Es sind in dieser Beispielrechnung bis zu 188 Funkeinheiten möglich. Jedoch kann eine Basisstation bei einem höheren Vermaschungsgrad weniger Funkeinheiten aufnehmen, da zusätzliche Nachbarschaftseinträge hinzukommen.

#### 5.3.4. Energiebedarf

Der AVR besitzt einige Energiesparmodi im Detail sind dies:

- Idle: Die CPU wird gestoppt, so dass kein weiteren Befehle ausgeführt werden. Weiterhin sind aber alle Hardwarebausteine wie ADC/USART etc. aktiv. Geweckt werden kann die CPU durch alle aktiven Interrupts
- ADC noise reduction mode: Dieser Energiesparmodus dient zur leichten Verbesserung der Analogmessung, da es weniger Störungen bei der Messung gibt. Ähnlich wie bei dem Idle-Mode ist die CPU gestoppt. Die CPU wacht auf, wenn die Messung fertig ist
- Power-Down-Mode: Es ist der energiesparendste Modus. Neben der CPU sind fast alle Hardwarebaugruppen ausgeschaltet und nur ein externen Interrupt oder ein I2C-Adress-Match kann die CPU wieder aufwecken
- Power-Save-Mode: Dieser Energiesparmodus fast mit dem Power-Down identisch. Jedoch kann noch der Timer2 asynchron betrieben werden, der die CPU aufwecken kann. So lässt sich der Tiefschlaf zeitlich steuern

## 5. Realisierung

- Standby-Mode: Ähnelt dem Power-Down . Der Unterschied zum Power-Down ist, dass die CPU weiterhin aktiv ist. Das Aufwecken kann so in wenigen Takten erfolgen
- Extended-Standby-Mode: Ähnelt dem Power-Save. Der Unterschied zum Power-Save ist, dass die CPU weiterhin aktiv ist. Das Aufwecken erfolgt in wenigen Takten

Zusätzlich lassen sich viele Hardwarebausteine extra abschalten. Als Beispiel werden nicht alle Timer benötigt, sie können abgeschaltet werden. Auch wird der Analog-Digital-Converter nicht benutzt und wird daher deaktiviert.

Die Abbildung 5.13 zeigt welcher Energiesparmodus zu welcher Phase im Funksystem aktiv ist.

Setup	Superframe				
Senden/Empfangen	Inaktiv	Empfang	Inaktiv	Senden	Inaktiv
Idle/Aktiv	Power-save	Idle	Power-save	Aktiv	Power-save

### Energiebedarf

Sehr gering ■ Hoch ■ Sehr hoch ■

Abbildung 5.13.: verschiedene Energiesparmodi während des Betriebs

Für die Berechnung der voraussichtlichen Betriebsdauer einer Funkeinheit können die Energiesparmodi mit den dazugehörigen Zeiten kombiniert werden, um den ungefähren Stromverbrauch festzustellen. Gerechnet wird mit einer Basisstation und 8 Funkeinheiten, jede Funkeinheit wird einem Sende und Empfangsslot zugeordnet.

Beschreibung	Zeit in ms	Strombedarf
Erste Erkundung	$8 \cdot 240ms$	$(8 \cdot 240ms) \cdot 16mA = 30,73mA \cdot s$
Scheduling+Synchronisieren	$200ms$	$200ms \cdot 16mA = 3,2mA \cdot s$
Superframe: Tiefschlaf	$300s - 2 \cdot 15,6ms$	$299,9688s \cdot 50\mu A = 15mA \cdot s$
Superframe: Empfang	$15,6ms$	$15,6ms \cdot 16mA = 250\mu A \cdot s$
Superframe: Senden	$15,6ms$	$15,6ms \cdot 18mA = 280\mu A \cdot s$
Messvorgang	$750ms$	$750ms \cdot (50\mu A + 1mA) = 788\mu A \cdot s$
Synchronisieren	$150ms$	$150ms \cdot 16mA = 2,4mA \cdot s$
Zweite Erkundung	$8 \cdot 100ms$	$(8 \cdot 100ms) \cdot 16mA = 12,8mA \cdot s$

Tabelle 5.5.: Stromverbrauch Funkeinheit

## 5. Realisierung

---

In der Beispielkonfiguration gibt es drei Superframes. Im ersten Superframe erfolgt eine komplette Erkundung, für die anderen Superframes ist nur ein Synchronisieren nötig. In der Berechnung fließt eine erste Erkundung und zwei mal die zweite Erkundung ein. Dazu kommt zur ersten Erkundung Scheduling sowie die Synchronisierung. Für die nachfolgenden Superframes ist jeweils nur das Synchronisieren nötig. Anschließend kommen noch die Superframe-Phasen Tiefschlaf, der Empfang, das Senden und die Temperaturmessung hinzu.

Summe Stromaufnahme:

$$Q_{Sum} = 30,73mA \cdot s + 2 \cdot 12,8mA \cdot s + 3,2mA \cdot s + 2 \cdot 2,4mA \cdot s + 3 \cdot (15mA \cdot s + 250\mu A \cdot s + 80\mu A \cdot s + 788\mu A \cdot s) = 0,112684A \cdot s$$

Summe Zeit:

$$t_{Sum} = 1,92s + 2 \cdot 0,8s + 0,2s + 2 \cdot 0,15s + 3 \cdot 300s = 904,02s$$

Durchschnittliche Stromaufnahme:

$$I = \frac{Q}{t} = \frac{Q_{Sum}}{t_{Sum}} = \frac{0,112684A \cdot s}{904,02s} = 124,65\mu A$$

Mit den für das Projekt zu Verfügung stehenden Akkus mit einer Kapazität von 1500mAh kann man mit dieser Stromaufnahme 12.033,7 Stunden bzw. 501,4 Tage eine Funkeinheit betreiben, vorausgesetzt der Akku wird keinen extremen Temperaturen ausgesetzt.

## 6. Tests

In Diesem Kapitel werden einige Tests für verschiedene Zwecke durchgeführt. Alle Tests fanden in den zwei zur Verfügung stehenden Räumen statt.

### 6.1. Energieverbrauch

Dieser Test soll den tatsächlichen Energieverbrauch über einen Zeitraum bestimmen. Der Stromverbrauch wird mit Hilfe eines Kondensators errechnet. Der Mikrocontroller arbeitet in einem Bereich von 1,8V-3,6V, der ungefähre Stromverbrauch wurde aus dem Abschnitt **Energiebedarf** entnommen. Folgende Kapazität wird benötigt:

$$Q = C \cdot U; C = \frac{Q}{U} = \frac{0,112684A \cdot s}{1,8V} = 0,626F$$

Für den Test wird ein Kondensator mit mindestens 0,626F benötigt. In dieser Größenordnung ist die nächste Kapazität 1F. Ausgewählt wurde der Kondensator „Gold Cap 5.5 V“. Laut Datenblatt ist hier ein Leckstrom von 315µA zu erwarten und kann im Test mit eingerechnet werden.

Die Genauigkeit ist für einen 1F Kondensator mit 0,8. bis 1,8 F angegeben (vgl. **Conrad 2012**, S. 3). Es muss also zunächst die Kapazität näher bestimmt werden. Das vorhandene Messgerät „Fluke 115“ kann nur Kapazitäten bis 9999 µF messen. Es wird daher eine Schaltung aufgebaut, um mittels der Formel  $\tau = R \cdot C$  die gesuchte Größe zu finden. Ein Widerstand mit 269,5 Ω wird mit dem Kondensator in Reihe geschaltet und das Messgerät wird parallel zum Kondensator angeschlossen. Eine Spannung von 3,6V wird angelegt und es wird die Zeit gemessen. Liegt an dem Kondensator eine Spannung von 2,275V an, so entspricht dies 63,2% der Spannungsquelle bzw. ein  $\tau$ . Es wurde eine Zeit von 244 Sekunden gemessen. Nun kann die Kapazität bestimmt werden.

$$C = \frac{\tau}{R} = \frac{244s}{269,5\Omega} = 0,90538F$$

## 6. Tests

---

Für den Test wird der Kondensator auf 3,6V mit einem Labornetzteil aufgeladen. Anschließend erfolgt in etwa zur gleichen Zeit das Abklemmen der Spannungsversorgung und das Verbinden der Funkeinheit an den Kondensator. Kurz darauf wird das System gestartet. Für eine Stunde wurde das System beobachtet und den Spannungsverlauf der Funkeinheit mitverfolgt. Diese Daten sind in Tabelle 6.1 zu entnehmen.

Zeit nach Systemstart in Minuten	Spannung an Kondensator
0	3,602V
15	3,501V
30	3,435V
45	3,370V
60	3,308V

Tabelle 6.1.: Spannungsverlauf während des Tests

$$Q_{\text{mitLeckstrom}} = C \cdot U = 0,90538F \cdot (3,602V - 3,308V) = 0,26618172A \cdot s$$

Der Leckstrom wurde mit 315  $\mu\text{A}$  je 30 Minuten angegeben, dies entspricht 175 nA in einer Sekunde.

$$Q_{\text{ohneLeckstrom}} = Q_{\text{mitLeckstrom}} - 175nA \cdot s = 0,26618154A \cdot s$$

$$I_{\text{Durchschnitt}} = \frac{Q}{t} = \frac{Q_{\text{ohneLeckstrom}}}{3600s} = \frac{0,26618154A \cdot s}{3600s} = 73,94\mu A$$

Durch Ungenauigkeiten wird der Verbrauch auf 75  $\mu\text{A}$  geschätzt. Mit dem Durchschnittsverbrauch kann nun die ungefähre Betriebszeit einer Funkeinheit ermittelt werden, es stehen 1500mAh zur Verfügung.

$$t_{\text{Betriebszeit}} = \frac{Q}{i} = \frac{1,5Ah}{75\mu A} = 20000h = 833,33Tage$$

Es sind demnach 332 mehr Tage als im Kapitel 5 berechnet. Allerdings fand dieser Test mit den Geräten innerhalb eines Raumes statt. In einer richtigen Anwendung würde die Funkeinheit je nach Standort mehr Daten weiterleiten und damit einen höheren Verbrauch haben.

Im Testzeitraum wurde noch zusätzlich die Stromaufnahme im Tiefschlaf gemessen. In einer Zeit von 32 Sekunden sank die Spannung um 1mV.

$$I_{Tiefschlaf} = \frac{Q}{t} = \frac{U \cdot C}{t} = \frac{1mV \cdot 0,90538F}{32s} = 28,29\mu A$$

Der Stromverbrauch im Tiefschlaf beträgt ca. 30µA. Damit ist der reale Stromverbrauch im Power-Save-Mode ca. 20 µA weniger als in der Theorie angenommen.

### 6.2. Zuverlässigkeit

Diese Tests gelten der Zuverlässigkeit des Funkprotokolls für die Datenübertragung. Im ersten Testaufbau sind 8 Funkeinheiten und eine Basisstation in einem Raum verteilt. Die Basisstation baut das Netzwerk auf. Anschließend sendet die Basisstation rundenbasiert jeder Funkeinheit eine Befehl, der eine Rückantwort erfordert. So kann die Basisstation feststellen, ob die Datenübertragung erfolgreich war. Diese Runde wurde 10.000 Mal wiederholt. Bei 8 Funkeinheiten müssten demnach 80.000 Rückantworten eintreffen. Für diesen Test wurden die WLAN-Access-Points umkonfiguriert, so dass es keine Datenübertragungen im gleichen und den umliegenden Kanälen geben kann.

Für diesen Zweck wurde das Hauptprogramm der Basisstation umgeschrieben, die Software der Funkeinheiten musste nicht geändert werden. Das Ergebnis wurde im Log ausgegeben. Testergebnis: 79.992 von 80.000 Rückmeldungen. Dies entspricht einer Fehlerrate von 0,0001%.

### 6.3. Störeinwirkung

Als Grundlage für diesen Test wird das umgeschriebene Programm des Zuverlässigkeitstests übernommen. Die Rundenanzahl werden verkleinert und die Störeinwirkung je Test schrittweise erhöht. Die Störung wird dabei von einem WLAN verursacht. Dafür wird ein Notebook in dem gleichen Raum gestellt und der WLAN-Access-Point umkonfiguriert. Der Access-Point erhält den Kanal 13 bzw. die Frequenz 2472Mhz. Die Funkeinheiten und Basisstation werden die Frequenz 2475MHz nutzen.

Eine Datenübertragung wird am Notebook gestartet und schrittweise wird nach jeder Übertragung die Bandbreite erhöht. Angefangen wird der Test ohne eine aktive Datenübertragung, lediglich die WLAN-Verbindung wird gehalten. Das Ergebnis ist in Abbildung 6.1 zu sehen.

## 6. Tests

---

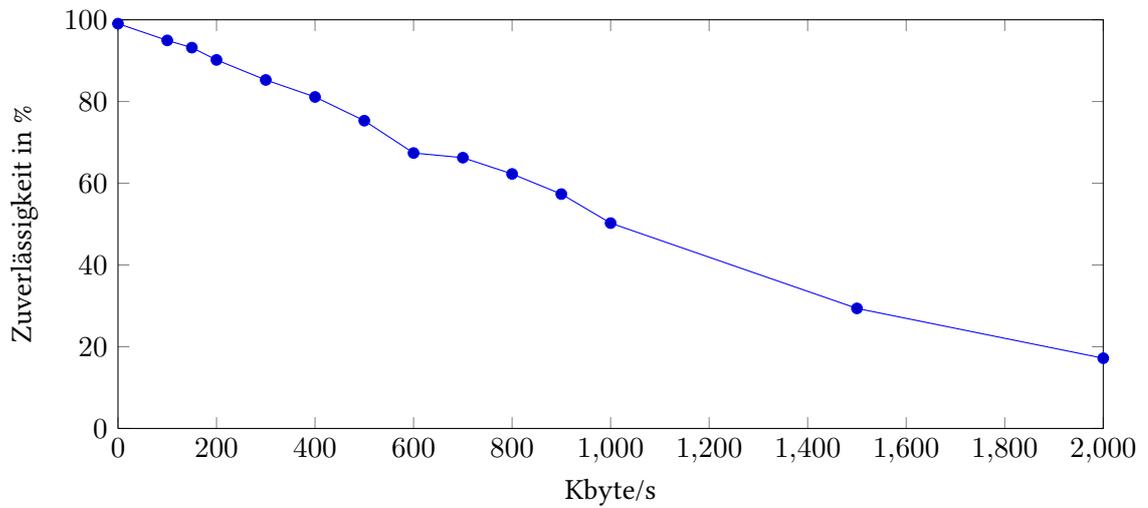


Abbildung 6.1.: Testergebnis Störeinwirkung

Es ist also auszugehen, dass mit einem WLAN im gleichen Kanal je nach Auslastung der Datenaustausch oft nur noch mit Sendewiederholung möglich ist. Ab einer bestimmten Auslastung des Kanals durch das andere Funknetz brach das Sensornetzwerk fast zum Erliegen. Eine Zuverlässigkeit von nur noch 17,19% wurde in dem Test gemessen.

## 7. Weiterentwicklungsmöglichkeiten

Das Funkprotokoll unterstützt bis zu 172 Teilnehmer, die Funkchips können aktuell bis zu 188 Funkeinheiten mit durchschnittlich 5 Nachbarn aufnehmen. Dabei spielt die Anordnung der Geräte keine Rolle, auch ist die lineare Verkettung möglich. Die Übertragungsintervalle können angepasst werden, im Testbetrieb sendeten die Funkeinheiten alle 5 Minuten ihre Messwerte, alle 15 Minuten wurde das Netzwerk neu erkundet.

Es gibt aber auch Verbesserungspotenzial. Das Suchverfahren ist mit 240ms pro Gerät noch relativ zeitintensiv. Ein kürzeres Verfahren kann die Betriebsdauer deutlich steigern. Da die aktuelle Entwicklungsumgebung auf Komfort ausgelegt ist, gibt es in der Gruppenkommunikation zwischen den Basisstationen relativ große Verzögerungen. Wenn die Geräte direkt an einem Bussystem gekoppelt sind, können die Synchronisierung und die Zeiten in Zeitslots verkürzt werden.

Im Funkprotokoll wird momentan von 100 ppm Uhrenungenauigkeit ausgegangen. Ein System, das die Gangart einer Uhr bestimmt, kann hier wesentlich genauer arbeiten. Es wird festgestellt, wie schnell oder langsam eine Uhr im Vergleich zu einer anderen Uhr ist. Die Ungenauigkeiten können auf wenige ppm reduziert werden.

Störungen, die durch andere Geräte im gleichen Frequenzband verursacht werden, können nicht vermieden werden. Ab einer bestimmten Auslastung im Kanal ist das Funksystem nicht mehr arbeitsfähig. Eine Weiterentwicklung in diesem Bereich macht das System robuster. So könnten z.B. die Kanäle gewechselt werden, wenn die Störungen das System beeinträchtigen.

Interessant ist auch das Verhalten zwischen dem Sensornetzwerk und einem anderen funkbasierten Netzwerk im gleichen bzw. im benachbarten Kanal. So würde Bluetooth andere Funkteilnehmer registrieren, während WLAN-Access-Points nur selten nach anderen Stationen suchen und ihren Kanal anpassen. Dazugehörige Tests in Abhängigkeit von der Entfernung zu anderen Geräten oder vom Abstand im Funkkanal wären sehr aufschlussreich.

## 7. Weiterentwicklungsmöglichkeiten

---

Eine zusätzliche Simulation, in der Anordnung und Kostenfunktion geändert werden können, optimiert die Wegeauswahl. Die Testauswertungen können dann in einem Realsystem getestet werden.

## 8. Zusammenfassung

Das Ziel dieser Arbeit war es ein Funksystem zu entwickeln, in der mehrere Basisstationen beteiligt sind. Wege werden durch eine Kostenfunktionen festgelegt und Funkeinheiten werden Basisstationen zugeordnet. Die Kostenfunktion soll helfen unnötige Hops über Funkeinheiten zu verhindern und eine zuverlässige Datenübertragung zu ermöglichen sowie Funkeinheiten mit wenig Energiereserven zu schonen.

Die im Kapitel [Analyse Nahfunktechnologien](#) angegebenen Anforderungen wurden vollständig umgesetzt: Jede Funkeinheit kann 10 Bytes Messdaten je Superframe verschicken, der Weg der Daten von der Funkeinheit zur Basisstation wird anhand einer Kostenfunktion errechnet. Das System unterstützt mehrere Basisstationen, der Betrieb wurde mit zwei Basisstationen und 8 Funkeinheiten erfolgreich getestet.

Der Test fand innerhalb eines Raumes statt. Dabei gab es keine Störungen durch andere Funk-systeme. 7 Funkeinheiten wurden per USB mit Energie versorgt, eine Funkeinheit wurde während der Testphase mit einem Kondensator gespeist. Anhand der fallenden Spannung an dem Kondensator wurde der Stromverbrauch ermittelt. Die Betriebsdauer einer Funkeinheit wurde mit 501 Tage errechnet, im Realtest wurde in einem Zeitraum von einer Stunde ein Durchschnittsverbrauch von ca.  $75\mu\text{A}$  gemessen, dies entspricht bei 1500mAh-Energievorrat eine Betriebsdauer von ca. 833 Tagen. Das Funksystem erkennt bei jedem Erkundungsvorgang neue bzw. nicht mehr vorhandene Funkeinheiten. Störungen und Verbindungsabbrüche werden durch Sendewiederholungen ausgeglichen.

Insgesamt fielen die Realtests im Kapitel 6 positiv aus. Die Nachrichtenzustellung in auf-gebauten Netzwerken ohne Störungen besitzt eine Fehlerrate von 0,0001%. Jedoch ist das Sensornetzwerk gegenüber anderen Netzerken im gleichen Kanal recht anfällig. Eine geringe Kanalausnutzung kann noch mit Sendewiederholungen ausgeglichen werden. Bei einer aktiven WLAN-Verbindung mit Datenübertragung wurde aber nur noch jeder 5 Frame vollständig übertragen. Ausbaufähig ist daher vor allem die Anpassung des Kanals, wenn Störungen auf-

## 8. Zusammenfassung

---

treten. Diese Funktion konnte leider nicht mehr hinzukommen, da es zeitlich nicht mehr in das Design passte.

Zusammengefasst lässt sich sagen, dass alle Ziele erreicht wurden und dass die Kurzzeittests erfolgreich waren. Weiterentwicklungen in verschiedenen Bereichen würden das Funksystem mit der Zeit marktreif machen.

## A. Anhang

## A.1. Zustandsautomaten

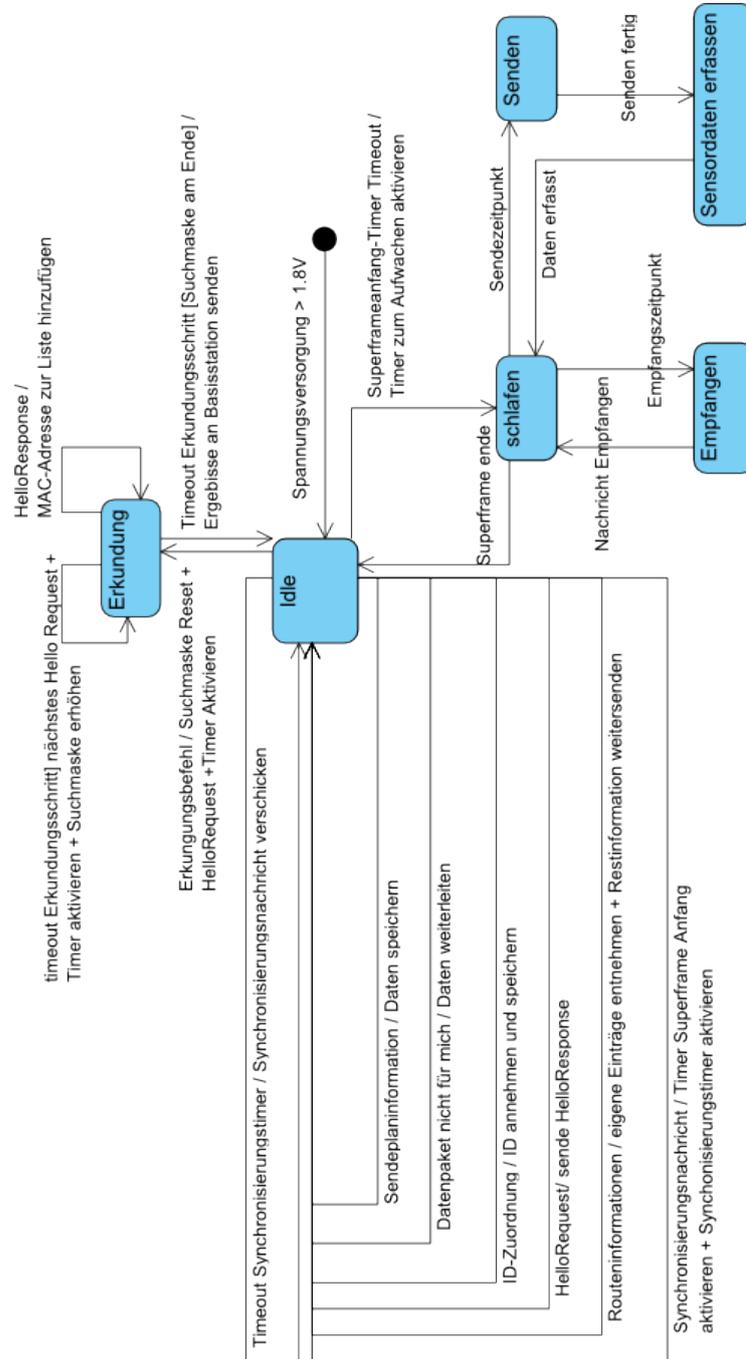


Abbildung A.1.: Zustandsautomat der Funkeinheit

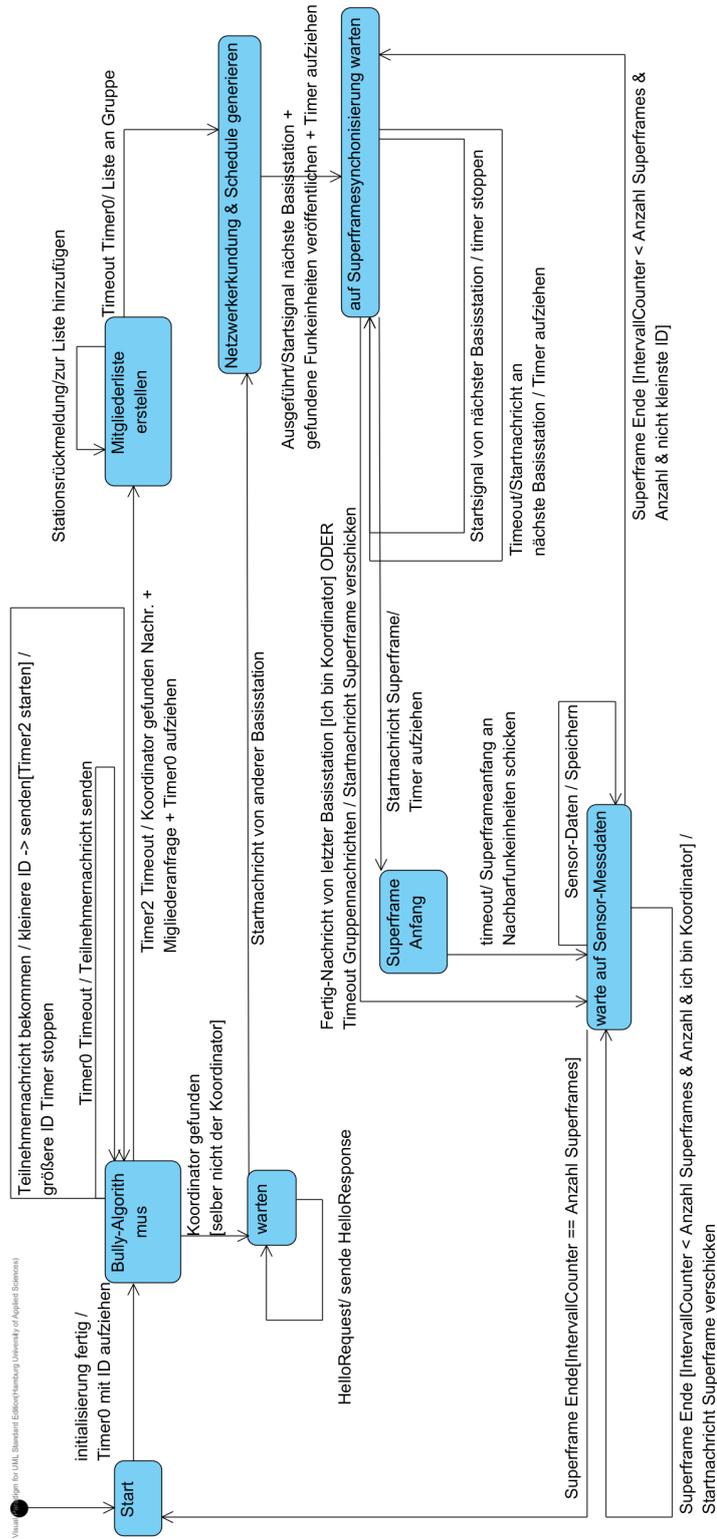


Abbildung A.2.: Zustandsautomat der Basisstation

## A.2. Inhalt DVD

Auf der DVD befinden sich mehrere Verzeichnisse. Es wurde nach benötigten Programmen (Entwicklungsumgebungen), Projektdateien und Diagrammen kategorisiert. In dem Verzeichnis Entwicklungsumgebung befinden sich Installationsdateien für das AVR-Studio, LPCXpresso und Eclipse(mit Java Runtime Environment).

Im Projektdateien-Ordner sind weitere Verzeichnisse enthalten, die zu den einzelnen Projekten der Entwicklungsumgebungen führen. Im Ordner Diagramme sind einige Abbildungen enthalten, die in der gedruckten Version hätten größer erscheinen können. Darunter fallen z.B. Zustandsautomaten und Sequenzdiagramme. Diese Verzeichnisstruktur ist in [A.3](#) abgebildet.

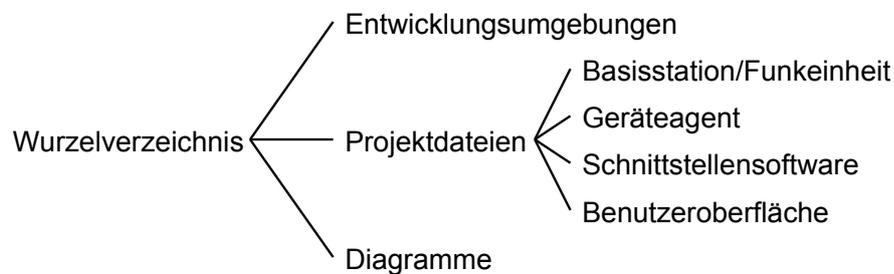


Abbildung A.3.: Verzeichnisstruktur DVD

Dabei gibt es folgende Zuordnungen zu den Projektdateien:

- Basisstation+Funkeinheit: AVR-Studio
- Geräteagent: LPCXpresso
- Schnittstellensoftware: Eclipse mit C/C++ Plugin
- Benutzeroberfläche: Eclipse

# Literaturverzeichnis

- [AtmelCorporation 2012] ATMELCORPORATION: *ATmega128RFA1 PRELIMINARY*. 2012. – URL <http://www.atmel.com/Images/doc8266.pdf>. – Zugriffsdatum: 09.11.2012
- [BluetoothSIG 2010] BLUETOOTH SIG: *BLUETOOTH SPECIFICATION Version 4.0*. 2010. – URL [http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc\\_id=229737](http://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737). – Zugriffsdatum: 08.10.2012
- [Bogue 2009] BOGUE, Robert: Energy harvesting and wireless sensors: a review of recent developments. In: *Sensor Review* 28 (2009), Nr. 3, S. 194 – 199
- [BSI 2009] BSI: *Auswahl geeigneter Kryptoverfahren für WLAN*. 2009. – URL <https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m02/m02384.html>. – Zugriffsdatum: 08.10.2012
- [Conrad 2012] CONRAD: *Datenblatt GoldCap*. Conrad Electronic SE. 2012. – URL [http://www.produktinfo.conrad.com/datenblaetter/450000-474999/473120-da-01-en-Gold\\_Cap.pdf](http://www.produktinfo.conrad.com/datenblaetter/450000-474999/473120-da-01-en-Gold_Cap.pdf). – Zugriffsdatum: 25.10.2012
- [DigiwirelessSolutions 2012] DIGIWIRELESS SOLUTIONS: *XBee DigiMesh 2.4 low power module w/ RPSMA connector*. 2012. – URL [http://www.digiwireless-solutions.com/Bizit/store/product.php?id\\_product=218](http://www.digiwireless-solutions.com/Bizit/store/product.php?id_product=218). – Zugriffsdatum: 09.11.2012
- [DresdenElektronik 2012] DRESDEN ELEKTRONIK: *Technische Daten Evaluations-Module deRF-mega128*. 2012. – URL <http://www.dresden-elektronik.de/funktechnik/products/radio-modules/eval-derfmega/technical-data/>. – Zugriffsdatum: 16.10.2012
- [embeddedArtists 2012] EMBEDDED ARTISTS: *LPC1343 QuickStart Board*. 2012. – URL [http://www.embeddedartists.com/products/boards/lpc1343\\_qsb.php](http://www.embeddedartists.com/products/boards/lpc1343_qsb.php). – Zugriffsdatum: 16.10.2012
- [Holtkamp 2003] HOLT KAMP, Heiko: *Einführung in Bluetooth*. 2003. – URL [www.sorex-austria.com/tl\\_files/news/News\\_Pdf/bluetooth.pdf](http://www.sorex-austria.com/tl_files/news/News_Pdf/bluetooth.pdf). – Zugriffsdatum: 05.11.2012

- [IEEE 2011] IEEE: *IEEE Standard for Local and metropolitan area networks Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*. 2011. – URL <http://standards.ieee.org/about/get/802/802.15.html>. – Zugriffsdatum: 09.10.2012
- [ITU 2007] ITU: *UHF Band Table*. 2007. – URL <http://www.itu.int/ITU-R/terrestrial/faq/index.html#g013>. – Zugriffsdatum: 08.10.2012
- [Kupris und Sikora 2007] KUPRIS, Dr.Ing. G. ; SIKORA, Prof.Dr.-Ing. A.: *ZigBee: Datenfunk mit IEEE 802.15.4 und Zigbee*. Franzis, 2007. – ISBN 978-3772341595
- [Lan u. a. 2008] LAN, Shi ; QILONG, Miao ; DU, Jinglin: *Architecture of Wireless Sensor Networks for Environmental Monitoring*. In: *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on* Bd. 1, dec 2008, S. 579 –582. – ISBN 978-0-7695-3563-0
- [Maxim 2012] MAXIM: *Data Sheet: DS18B20*. 2012. – URL <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. – Zugriffsdatum: 17.10.2012
- [NationalInstruments 2012] NATIONALINSTRUMENTS: *Five Factors to Consider When Implementing a Wireless Sensor Network (WSN)*. 2012. – URL <http://www.ni.com/white-paper/10789/en>. – Zugriffsdatum: 12.11.2012
- [Osterhage 2010] OSTERHAGE, Wolfgang W.: *Bluetooth*. In: *sicher & mobil*. Springer Berlin Heidelberg, 2010 (Xpert.press), S. 133–148. – URL [http://dx.doi.org/10.1007/978-3-642-03083-3\\_8](http://dx.doi.org/10.1007/978-3-642-03083-3_8). – ISBN 978-3-642-03083-3
- [Schmidt 2011] SCHMIDT, Prof. Dr. T.: *Netzwerktechnologien Grundlagen*. 2011. – URL [http://inet.cpt.haw-hamburg.de/teaching/ss-2011/rechnernetze/03\\_LAN.pdf](http://inet.cpt.haw-hamburg.de/teaching/ss-2011/rechnernetze/03_LAN.pdf). – Zugriffsdatum: 18.10.2012
- [Sikora 2004] SIKORA, Prof. Dr. A.: *ZigBee: Grundlagen und Applikation*. In: *Elektronik Wireless 1* (2004), S. 18–24
- [Sonavane u. a. 2008] SONAVANE, S.S ; KUMAR, V. ; PATIL, B.P.: *Designing wireless sensor network with low cost and low power*. In: *Networks, 2008. ICON 2008. 16th IEEE International Conference on, IEEE*, 2008, S. 1–5. – URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4772585>. – ISBN 978-1-4244-3805-1

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 16. November 2012

---

Rico Flaegel