

Evaluierung und Erprobung einer Plattform zur Käuferlokalisierung

Diplomarbeit

Marco Rust

s733042

Berlin, 13. Juli 2010



Beuth Hochschule Berlin
Fachbereich VI – Informatik und Medien

Luxemburger Straße 10
13353 Berlin

Betreuer: Prof. Dr. B. Buchholz
Gutachter: Prof. Dr. Teppner

Fraunhofer Institut FOKUS

Kaiserin-Augusta-Allee 31
10589 Berlin

Betreuer: M.Sc. Lasse Thiem

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig angefertigt und mich fremder Hilfe nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichtem oder unveröffentlichtem Schrifttum entnommen sind, habe ich als solche kenntlich gemacht.

Ort, Datum

Unterschrift

Sperrvermerk

Die vorliegende Diplomarbeit enthält vertrauliche Daten des Fraunhofer Institut für Offene Kommunikationssysteme (FOKUS). Veröffentlichung oder Vervielfältigung der Diplomarbeit sind auch auszugsweise - ohne ausdrückliche Genehmigung des Unternehmens bzw. des Autors nicht gestattet. Die Diplomarbeit ist nur den Mitgliedern des Prüfungsausschusses zugänglich zu machen.

Vorwort

Hervorgegangen aus meinem Praktikum am *Fraunhofer Institut für Offene Kommunikationssysteme* (FOKUS) hat sich mir die Möglichkeit geboten, im Kompetenzzentrum *Sensor Applications and Networks* (SANE) folgende wissenschaftliche Arbeit zu verfassen.

Ein großer Dank geht daher an Lasse Thiem und Dr. Thomas Luckenbach, welche diese Arbeit überhaupt ermöglicht haben. Des Weiteren danke ich meinen Kollegen Björn Riemer, Martin Klein und Benjamin Kranz für ihre fachliche und richtungsweisende Hilfe in schwierigen Situationen. Für Ihre langjährige Unterstützung möchte ich mich an dieser Stelle auch bei meinen Eltern bedanken.

Mein besonderer Dank gilt meiner Freundin Pamela Krajczyk, die mich auch in den schwierigen Momenten meiner Arbeit aufzumuntern vermochte und mich stets motiviert hat.

Hinweise zur Formatierung

In dieser Arbeit werden im Folgenden Eigennamen, Titel von Querverweisen sowie wichtige Begriffe die nicht Teil des allgemeinen Wortschatzes sind in *kursiver* Schrift geschrieben. Ausschnitte von Programmcode werden im Text durch die Schriftart *courier* gekennzeichnet.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Anforderungen und Zielsetzungen	1
1.3	Gliederung dieser Arbeit	2
2	Theoretische Grundlagen	3
2.1	Grundlagen zur Lokalisierung	3
2.2	Bestehende Lokalisierungssysteme	6
2.2.1	Infrarot	6
2.2.2	Ultraschall	7
2.2.3	Funk	7
2.2.4	Visuell	8
2.2.5	Koppelnavigation	8
2.2.6	Entwicklungstrends	8
2.3	Grundlagen der Kommunikation	10
2.4	Drahtlose Sensornetzwerke	12
2.4.1	IEEE 802.15.4	13
2.4.2	ZigBee	18
2.4.3	Weitere drahtlose Netzwerke-Technologien	19
2.5	GSM	21
3	Konzept	22
3.1	Technologieentscheidung	22
3.2	Funktionsbeschreibung	22
3.3	Softwarekomponenten	26
4	Pflichtenheft	27
4.1	Zielbestimmungen	27
4.2	Produkteinsatz	28
4.3	Produktübersicht	29
4.4	Produktfunktionen	29

4.5	Produktleistung.....	30
5	Realisierung.....	31
5.1	Hardware und Software.....	31
5.1.1	Motorola C123.....	31
5.1.2	OsmocomBB.....	32
5.1.3	Radiocrafts RC2301AT.....	32
5.1.4	CC-Debugger.....	32
5.1.5	Host-Computer.....	33
5.1.6	TIMAC.....	33
5.1.7	IAR Embedded Workbench für 8051.....	34
5.1.8	Microsoft Visual Studio 2008.....	35
5.1.9	Wireless Protocol Analyser.....	35
5.1.10	HTerm.....	36
5.1.11	SmartRF Flash Programmer.....	36
5.1.12	Weitere Hard- und Software.....	36
5.2	Implementierung.....	38
5.2.1	Verwendung der TIMAC-API.....	38
5.2.2	Das IEEE 802.15.4-Netzwerk.....	39
5.2.3	Rahmenformate für die Kommunikation.....	42
5.2.4	Entwicklung der Basisstation.....	43
5.2.5	Entwicklung der Plattform.....	45
5.2.6	Software des Host-Computer.....	54
5.3	Aufgetretene Probleme.....	57
5.3.1	SDCC Compiler.....	57
5.3.2	Kompilierung des TIMAC-Stacks mit IAR Embedded Workbench.....	58
5.3.3	CC-Debugger mit IAR Embedded Workbench.....	58
5.3.4	Speichern der IEEE-Adresse aus der eigenen Anwendung heraus.....	58
5.3.5	Download des Programms in das Mobiltelefon schlägt fehl.....	58
5.3.6	Fehlerhafte Übertragungen der RSSI-Werte über IEEE 802.15.4.....	58
6	Ergebnis der Evaluierung.....	60

6.1	Funktionsnachweis.....	60
6.1.1	Vorbereitungen.....	60
6.1.2	Kalibrierung der Location Engine.....	60
6.1.3	Installation der Basisstationen und des Host-Computer.....	61
6.1.4	Kommunikation über das IEEE 802.15.4-Netzwerk.....	62
6.1.5	Erstellen der GSM-Radiomap.....	63
6.1.6	Darstellen der Position in der Software.....	63
6.2	Genauigkeit der Lokalisierung.....	64
7	Schlussbetrachtung.....	68
7.1	Zusammenfassung und Fazit.....	68
7.2	Ausblick.....	69
8	Literaturverzeichnis.....	70
9	Abbildungsverzeichnis.....	75
10	Tabellenverzeichnis.....	78
11	Abkürzungen.....	79
A.	Anhang.....	81
B.	Inhalt der CD.....	86

1 Einleitung

In diesem Kapitel werden die Rahmenbedingungen und die Ziele dieser Arbeit erläutert sowie ein Überblick zum Inhalt gegeben.

1.1 Motivation

Das wohl bekannteste Lokalisierungssystem, das *Global Positioning System* (GPS), ist zum Alltag geworden. Zunehmend mehr technische Geräte ermöglichen das Orten ihrer Position über dieses System. Neben dem ursprünglichen militärischen Verwendungszweck befinden sich GPS-Empfänger heutzutage in verschiedenen Anwendungsbereichen, zum Beispiel Kraftfahrzeugen, Seenavigation, Vermessung und sogar Mobiltelefonen im Einsatz. Mit Hilfe von GPS werden *Outdoor*-Genauigkeiten von unter 10 Metern erreicht (vgl. Kapitel 2.2.3), jedoch hat diese Satelliten-gestützte Technologie einen gravierenden Nachteil – der mangelhafte bzw. fehlende Empfang von Funksignalen innerhalb von Gebäuden. Dieser Umstand macht GPS für Anwendungen in Gebäuden praktisch unbrauchbar.

Aus diesem Grund muss im Bereich der *Indoor*-Lokalisierung¹ auf andere Technologien zurückgegriffen werden, denn die Ortung von Gegenständen oder Personen innerhalb von Gebäuden gewinnt immer mehr an Bedeutung. Beispiele hierfür sind die Indoor-Navigation auf Messegeländen oder die Lagerhaltung.

Im Rahmen eines Projektes des Fraunhofer FOKUS zur Analyse des Kaufverhaltens (siehe Kapitel 1.2) hat sich mir daher die Möglichkeit ergeben, eine Plattform zu entwerfen um ausgewählte Lokalisierungsverfahren zu evaluieren und dazu diese Arbeit zu verfassen.

Das *Fraunhofer-Institut für Offene Kommunikationssysteme* (FOKUS)² forscht dazu im Bereich Indoor-Lokalisierung im Kompetenzzentrum *Sensor Applications and Networks* (SANE) unter Verwendung von drahtlosen Sensornetzwerk-Technologien.

1.2 Anforderungen und Zielsetzungen

Die Entwicklung eines präzisen vollautomatischen Systems zur Käuferlokalisierung ist das Ziel eines Projektes beim Fraunhofer FOKUS. Der Ort der Kaufentscheidung ist auch immer häufiger der des Kaufes, dem sogenannten *Point of Sale* (PoS). Beispielsweise fällt die Entscheidung über den Kauf eines neuen Flachbild-Fernsehgeräts oftmals erst vor Ort im jeweiligen Elektronikfachgeschäft.

¹ Innerhalb von Gebäuden

² „FOKUS erforscht und entwickelt bedarfsorientiert Lösungen für Partner aus Industrie, Forschungsförderung und Öffentlicher Verwaltung...“ [FOKUS]

Bewegungsmuster von Personen, wie etwa Aufenthaltsdauer vor einem Artikelsortiment oder Regal, sind daher für die Verhaltensforschung zunehmend relevant. Aus diesem Grund wurden zur Kaufverhaltensanalyse bisher Methoden der verdeckten Beobachtung oder Befragung des Kunden durch Interviewer oder Erfassung mit Hilfe von Videoaufzeichnung eingesetzt. Diese Techniken erfordern jedoch einen hohen Personal- und Zeitaufwand und die Problematik des Datenschutzes darf hierbei auch nicht außer Acht gelassen werden.

Anforderungen

In Folge dessen werden folgende Anforderungen an das System gestellt:

- Die Bewegungen des Käufers sollen innerhalb eines Geschäftes ohne zusätzliches Personal so genau wie möglich erfasst werden.
- Benötigte Technik muss leicht in die Verkaufsumgebung integriert werden können.
- Das System ist leicht zu (de-)installieren und damit mobil.
- Es werden hauptsächlich handelsübliche Hard- und Softwarekomponenten verwendet.

Zielsetzungen

Aus den gegebenen Anforderungen soll im Rahmen dieser Arbeit eine Evaluierungsplattform entwickelt werden, um verschiedene Indoor-Lokalisierungsverfahren zu testen. Dabei wird die Position der Plattform an zentraler Stelle dargestellt. Zum Abschluss sollen die erreichbaren Genauigkeiten der Verfahren mit Hilfe dieser Plattform in einer verkaufsähnlichen Indoor-Umgebung bestimmt werden.

1.3 Gliederung dieser Arbeit

Im Kapitel *Theoretische Grundlagen* werden die theoretischen Vorbedingungen erläutert. Dies umfasst zum Einen die Thematik der Lokalisierung an sich sowie eine Auswahl bereits bestehender Lokalisierungssysteme – und zum Anderen das Thema der Kommunikation.

Das Kapitel *Pflichtenheft* konkretisiert die Anforderungen und Ziele des Unterkapitels *Anforderungen und Zielsetzungen*.

Im darauffolgenden Kapitel *Konzept* wird die Entscheidung über die eingesetzte Technologie sowie die Funktionsweise selbiger für die Lokalisierung erörtert.

Die Implementierung der Plattform wird detailliert im anschließenden Kapitel *Realisierung* beschrieben.

Im Kapitel *Ergebnis der Evaluierung* wird die resultierende Genauigkeit der Lokalisierung betrachtet und beurteilt und der Funktionsnachweis erbracht.

Abschließend resümiert das Kapitel *Schlussbetrachtung* die Ergebnisse und gibt einen Ausblick auf eine mögliche Weiterführung des Projekts.

2 Theoretische Grundlagen

Dieses Kapitel erläutert die für diese Arbeit theoretischen Grundlagen. So werden neben der Problematik Lokalisierung auch theoretische Grundlagen der Kommunikation allgemein und in Sensornetzwerken erläutert. Des Weiteren werden kurz bereits bestehende Lokalisierungssysteme vorgestellt.

2.1 Grundlagen zur Lokalisierung

Für die Ortung gibt es verschiedene Prinzipien, welche in diesem Unterkapitel unter anderem unter Zuhilfenahme des Buches *Protocols and Architectures for Wireless Sensor Networks* [Kar05] S.231-249 vorgestellt werden.

Verfahren zur Positionsbestimmung

Für die Bestimmung der Position einer Person oder eines Objekts, im Folgenden als Knoten bezeichnet, gibt es verschiedene Methoden. Zunächst wird unterschieden zwischen *Tracking* (Verfolgung) und dem *Positioning* (Positionsbestimmung). Das *Tracking* ist ein netzwerkgestütztes Verfahren, in welchem die Ortung des Knotens durch die Infrastruktur durchgeführt wird. Im Vergleich dazu wird beim *Positioning* die eigene Position durch den Knoten selbst bestimmt. Der Begriff Infrastruktur bezeichnet das umgebende Netzwerk, welches zur Kommunikation und Lokalisierung verwendet wird (siehe Abbildung 2.1).

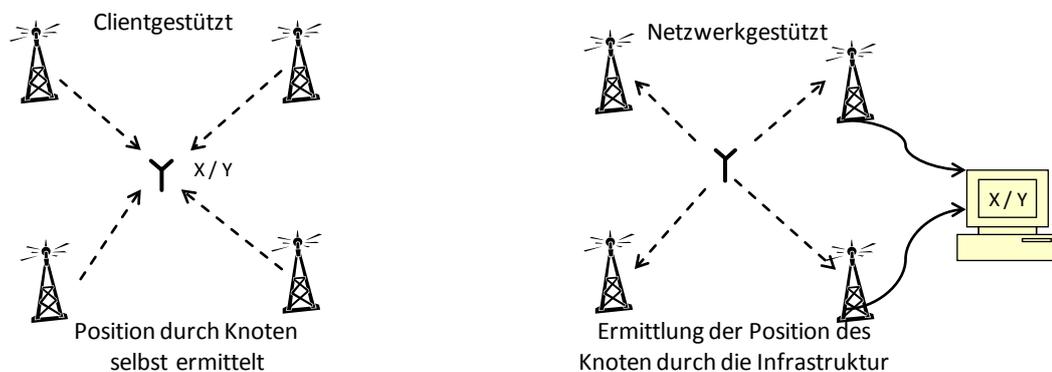


Abbildung 2.1: Tracking und Positioning

Grundlegende mathematische Lokalisierungsverfahren basieren auf einfacher euklidischer Geometrie:

- **Angulation:** Bei der Angulation (AoA – Arrival of Angel) wird mit Hilfe von mindestens zwei oder mehrerer bekannter Winkel eine Position in dem Schnittpunkten der Winkelgeraden ermittelt. Siehe dazu Abbildung 2.2.
- **Lateration:** Die Lateration beschreibt die Positionsbestimmung durch Ermittlung des Schnittpunktes mittels dreier Kreise bzw. Kugelflächen im 3-Dimensionalen (Abbildung

2.2). Die Positionen der Ankerknoten³ sind hierbei bekannt und die Radien werden durch geeignete Entfernungsmessverfahren ermittelt (siehe dazu den folgenden Abschnitt *Technische Messverfahren*).

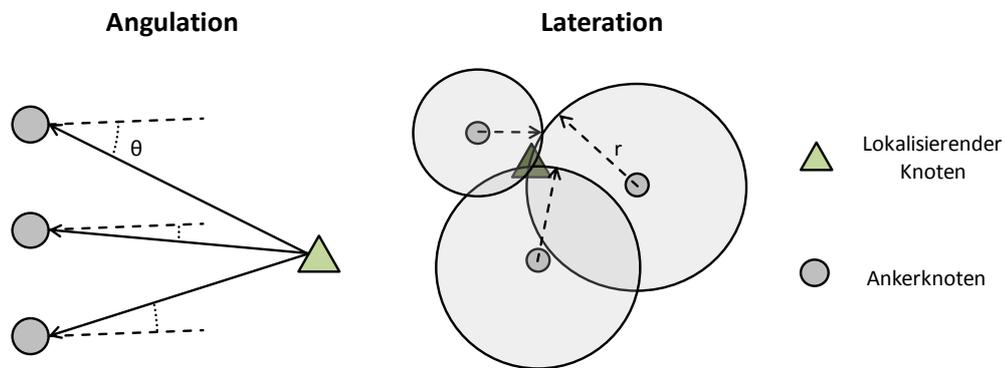


Abbildung 2.2: Angulation und Lateration

Es gibt jedoch auch weitere Methoden die für die Lokalisierung verwendet werden.

- **Zellbasiert:** Zellbasierte Lokalisierung (*Cell of Origin - CoO*) ist eine einfache Form der Ortung. Bei dem Verfahren werden Zellen angenommen, welche eine begrenzte Ausdehnung haben und sich nicht überlappen. Befindet sich ein Knoten innerhalb einer Zelle, so ist seine Position mit der Größe der Zelle genau bestimmt. Je dichter und kleiner die Zellen sind, desto genauer ist die Positionsbestimmung.
- **Radiomap:** Für diese Art der Lokalisierung wird zunächst eine Karte (*Radiomap*) mit bestimmten charakteristischen Eigenschaften, oftmals der Feldstärke, von beliebig vielen verschiedenen Positionen im Raum erstellt. Nach dieser Trainingsphase kann bei erneuter Messung der Signalstärke(n) diese der wahrscheinlichsten Position zugeordnet werden. Neben der Lokalisierung durch eine Feldstärkenkarte könnten auch andere Eigenschaften für die Positionsbestimmung verwendet werden, etwa ermöglicht der Luftdruck eine ungefähre Bestimmung in der Vertikalen. In Abbildung 2.3 wird exemplarisch die *Radiomap* eines Gebäudes graphisch dargestellt.

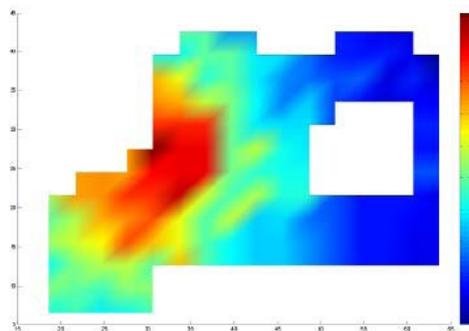


Abbildung 2.3 Graphische Darstellung einer *Radiomap* [Sza10]

³ Ein Knoten dessen Position bekannt ist und zur Berechnung der Position anderer Knoten verwendet wird.

Technische Messverfahren

Zur Bestimmung der im vorherigen Kapitel erläuterten Methoden gibt es verschiedene technische Messverfahren:

- **Winkelmessung:** Für die Bestimmung der Winkel werden gerichtete Sender- und Empfänger verwendet. Mit Hilfe der gegebenen Winkel kann eine Triangulation durchgeführt werden.
- **Signalstärkemessung:** Die Signalstärke wird durch den *Received Signal Strength Indication* (RSSI) im Empfänger des Signals dargestellt. Da die im Empfänger gemessene Signalstärke radial mit der Distanz zum Sender quadratisch abnimmt ([Rot02] S.272), kann somit die Entfernung unter idealen Bedingungen⁴ ermittelt werden.
- **Laufzeitmessung:** Da sich Funksignale mit annähernd Lichtgeschwindigkeit ausbreiten, kann nach Messung der Ausbreitungszeit unter idealen Bedingungen über die Gleichung $c * t = s$ die Distanz ermittelt werden. Im Detail gibt es hierbei jedoch verschieden genaue Ansätze.
 - **ToA:** (Time of Arrival) Bei dem einfachsten Laufzeitmessungsverfahren wird die Zeit gemessen, die das Signal bis zum Eintreffen beim Empfänger benötigt hat. Ein gravierender Nachteil ist jedoch, dass eine hochgenaue Synchronisation der Sender- und Empfängeruhren im pico-Sekunden-Bereich stattfinden muss um eine hochgenaue Ortung zu ermöglichen. Weniger genaue Uhren lassen für Funksignale nur eine Distanzmessung im Zehn-Meterbereich zu.
 - **TDoA:** (Time Difference of Arrival). Das Verfahren arbeitet mit verschiedenen Technologien deren Laufzeiten im gleichen Medium unterschiedlich sind. Als Beispiel kann das gleichzeitige Senden von Funk- und Ultraschallsignalen betrachtet werden ($v_{\text{Funk}} \approx 3 \times 10^8 \text{ m/s}$; $v_{\text{Ultraschall}} \approx 333 \text{ m/s}$; $\text{Medium} = \text{Luft}$). Hierbei erreicht zunächst das Funksignal den Empfänger, die Laufzeitmessung wird gestartet. Sobald das Ultraschallsignal den Empfänger erreicht, wird anschließend die Laufzeitdifferenz gemessen. Da die Laufzeit des Funksignals gegenüber dem Ultraschallsignal um circa sechs Größenordnungen kleiner ist, kann diese vernachlässigt werden. Ein weiterer Vorteil ist, dass bei diesem Verfahren keine Synchronisierung der Uhren mehr benötigt wird, jedoch stehen dem höhere Kosten und eine begrenzte Reichweite gegenüber.

⁴ Dies gilt im freien Raum, ohne Reflexionen, Interferenzen und Dämpfungen bei der Signalausbreitung.

- **RToF:** In der Tierwelt senden beispielsweise Fledermäuse ein Ultraschallsignal aus, welches zum Sender zurückreflektiert und zur Entfernungsmessung verwendet wird. Dieses *Reflection Time of Flight* genannte Verfahren wurde in der Technik als *Roundtrip Time of Flight* (RToF) umgesetzt. Dieser Ansatz ist jedoch nicht nur auf Ultraschallsignale begrenzt. Im Detail wird nicht das Signal reflektiert, sondern vom Empfängerknoten zurück an den Sendeknoten gesendet. Da die benötigte Empfangs-, Bearbeitungs- und Rücksendezeit des Empfängerknotens sehr genau bestimmbar ist, kann die Entfernung beider Knoten zueinander als halbe Laufzeit des „RoundTrip“-Signals ermittelt werden. Bei diesem Verfahren ist keine Uhrensynchronisation der Knoten notwendig, jedoch werden aufgrund der hohen Ausbreitungsgeschwindigkeit von Funksignalen hohe Anforderungen an die taktgebenden Bauteile gestellt. Die Ganggenauigkeit muss sehr groß sein, was jedoch mit hohen Kosten verbunden ist.
- **SDS-TWR:** Um den Gangfehler (*clock drift*) weiter zu reduzieren, wurde ein weiteres ToF-Verfahren entwickelt [Nan10]. Symmetrical Double-Sided Two Way Ranging führt zweimalig eine Laufzeitmessung durch und mittelt die gemessene Laufzeit um eine höhere Genauigkeit zu erreichen.
- **Hop-Basierte Distanzmessung:** Es wird ein räumlich gleichverteiltes Netz aus beliebig vielen Knoten angenommen. Die Distanzen zwischen allen Knoten werden gemittelt als Knotendistanz k . Im Folgenden wird die Anzahl der Sprünge (Hops) vom Sender zum Empfänger ermittelt und mit der gemittelten Knotendistanz k multipliziert, dies ergibt die absolute Distanz. Eine höhere Dichte des Netzes erhöht hierbei die Genauigkeit, jedoch auch die notwendigen Kosten.

Die hier dargestellten Verfahren erzielen die höchste Genauigkeit nur unter idealen Bedingungen, welche jedoch in der Realität nicht gegeben sind. Durch Reflexionen und Interferenzen der Signale kommt es zu Überlagerungen und Auslöschungen, was eine genaue Distanzmessung erschwert.

2.2 Bestehende Lokalisierungssysteme

Inzwischen gibt es zahlreiche Lokalisierungssysteme. Einige davon sind bereits im Alltag im Einsatz, während andere eher Forschungsgegenstand sind. An dieser Stelle gibt es eine Auswahl solcher Systeme.

2.2.1 Infrarot

Active Badge [WHFG92]: ist ein reines Indoor-Ortungssystem. Es besteht aus tragbaren Infrarot-Sender (*Badge*) und einen Infrarotempfänger-Netzwerk. Konkret wird in jedem Raum ein Infrarotempfänger angebracht, welcher eine einzigartige Kennung des Badge empfängt und an dem am Sensornetzwerk angeschlossen Server zentral auswertet.

Im Gegensatz dazu ist *WIPS* (*Wireless Indoor Positioning System*) [WIPS00] infrastrukturbasiert, da hier die ortsfesten Infrarotbaken eine Kennung senden, welche vom Badge empfangen und über WLAN (siehe Kapitel 2.4.2) an die zentralen Server zur Auswertung übermittelt werden. Eine Vernetzung der IR-Baken ist dabei nicht mehr notwendig. Nachteil dieser Systeme ist, dass die Position aufgrund der Begrenzung auf den Raum lediglich auf die Größe des Raumes genau bestimmt werden kann. Da jedoch eine räumliche Trennung zur Unterscheidung der Infrarotsignale benötigt wird, ist das System z.B. gänzlich ungeeignet für große Lagerhallen oder Verkaufsflächen.

2.2.2 Ultraschall

Eine weitaus höhere Genauigkeit lassen sich durch Ultraschallverfahren erreichen. Bei dem Indoor-System *ActiveBat*[WJH97] wird ein 1,2 m dichtes Netz von Ultraschallempfängern an der Decke des Raumes angebracht. Ein zentraler Server gibt per Funk an einen *Bat*⁵ die Anforderung einen Ultraschallimpuls zu erzeugen, welcher von dem Empfängernetz an der Decke detektiert wird. Der zentrale Server ermittelt mit Hilfe dieses Netzes die Position des Bat auf bis zu 10 cm genau.

Darüber hinaus gibt es das jüngere System *Cricket* [PCB00], wobei die festinstallierten Baken an der Decke Ultraschallimpulse und zusätzlich ein Funksignal aussenden, welche vom mobilen Gerät empfangen werden. Durch TDOA (vgl. Kapitel 2.1) wird dadurch die Laufzeit bestimmt. Das „serverlose“ System erreicht hierbei eine Genauigkeit von 1 - 3 Zentimeter.

2.2.3 Funk

Auch im Bereich der Funkortung gibt es verschiedene Lösungen. Das bekannteste Lokalisierungsverfahren *Global Positioning System* (GPS) ist seit 1993 in Betrieb und erzielt heute nach zahlreichen Verbesserungen weltweit Genauigkeiten von bis 10 m Metern [Ele10]. In Kombination mit Korrektursignalen (D-GPS), die von Basisstationen gesendet werden, kann die Genauigkeit auf unter 1 Zentimeter gesteigert werden ([DH10] S. 226). Jedoch hat GPS einen gravierenden Nachteil, so ist der Empfang der Signale innerhalb von Gebäuden aufgrund der hohen Dämpfung schwer - in den meisten Fällen sogar unmöglich.

SpotON [HBW00] basiert auf RFID-Technologie und verwendet die Lateration mit Hilfe von Signalstärkemessungen. Die Position wird durch einen an die Infrastruktur angeschlossenen Server berechnet und dabei wird Indoor eine Genauigkeit von 3m erreicht.

Ein weiteres Funk-Lokalisierungsverfahren von *Microsoft Research* ist das System *RADAR* [Bahl00]. Es verwendet die Signalstärken von verschiedenen Wireless LAN Accesspoints, um diese in einem zweistufigen Verfahren zunächst in einer Datenbank zu speichern und in der zweiten Phase mit Hilfe dieser Datenbank die wahrscheinlichste Position zu ermitteln.

⁵ Gerät zur Positionsbestimmung

Ebenfalls auf einer *Radiomap* basiert das Prinzip der GSM-Lokalisierung [VLHLO07], jedoch können hier wegen des größeren Frequenzbandes eine größere Zahl von Kanälen genutzt werden.

2.2.4 Visuell

Visuelle Ortung ist unter Verwendung von festinstallierten (Netzwerk-gestützt) und „mobilen“ Kameras (Client-gestützt) möglich. Für beide Varianten ist aufwändige Bildverarbeitung notwendig, um die Position anhand der Umgebung zu bestimmen. Verfahren zur clientgestützten Selbstlokalisierung werden hauptsächlich in der Robotik eingesetzt, z.B. [Lan00], aber auch infrastrukturbasierende Ansätze [TUD10] sind vorhanden. Da die Objekterkennung bei der Bildverarbeitung sehr rechenintensiv ist, sind solche Verfahren für drahtlose Netzwerke eher ungeeignet.

2.2.5 Koppelnavigation

Ohne Infrastruktur, nur durch die Kombination verschiedener Sensoren kann die Position aus dem zurückgelegten Weg berechnet werden. In [Wan07] wird mit Hilfe von Beschleunigungssensoren der Bewegungsablauf von Personen analysiert und Schritte detektiert, woraus schließlich die Geschwindigkeit ermittelt wird. In Verbindung mit einem Kompasssensor kann der begangene Weg und letztendlich die relative Endposition berechnet werden. Nachteilig ist jedoch, dass sich Positionsfehler bei der Koppelnavigation aufsummieren, so dass mit steigender Weglänge die Abweichung von der realen Position zu der Berechneten zunimmt. Des Weiteren muss die absolute Startposition bekannt sein, was ggf. mit weiteren Sensoren, beispielsweise GPS, realisiert werden kann.

2.2.6 Entwicklungstrends

Da im Bereich der Indoor-Lokalisierung besondere Probleme wie der Dämpfung, Reflexion und Interferenzen gegeben sind, gibt es kein optimales System für die Ortung innerhalb von Gebäuden. Künftige Entwicklungen gehen daher in Richtung Fusion der Sensordaten bzw. Verfahren, um die Genauigkeit der Positionsbestimmung zu erhöhen. Abbildung 2.4 zeigt eine Plattform mit kombinierten Sensoren.

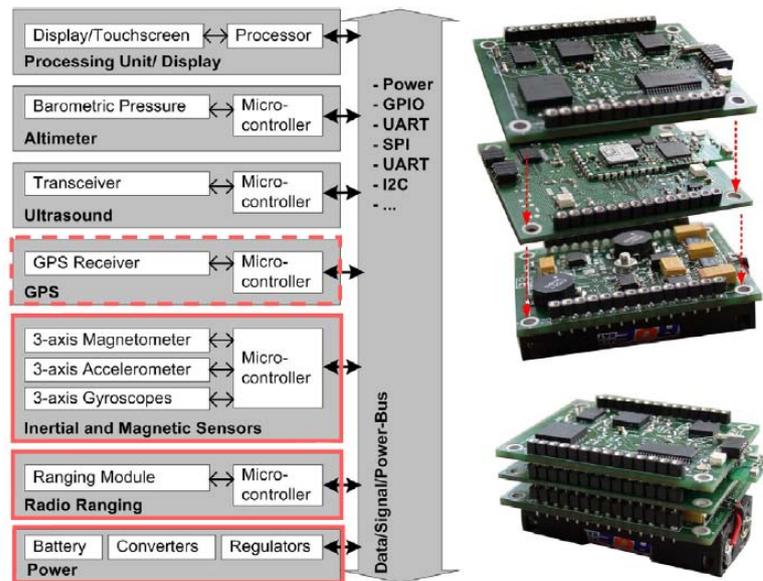


Abbildung 2.4: Kombination verschiedener Sensordaten [Kli10]

Weiterhin entwickelt sich der Trend in Richtung der opportunistischen Lokalisierung - bereits bestehende Netzwerkinfrastruktur wird, wenn vorhanden, genutzt und erhöht damit die Genauigkeit. Das Projekt *MagicMap* [HUB10] ist eine reine Softwarelösung zur Indoor-Lokalisierung mit Wireless LAN. Durch Schnittstellen kann es jedoch um weitere Technologien, z.B. (GSM, GPS usw.) erweitert werden. Ein kommerzielles System, welches Wireless LAN, GPS und die GSM-Trilateration vereint, bietet *Skyhook Wireless* [Sky10]. Der Anbieter verspricht In- und Outdoor aufgrund der flächendeckenden Verbreitung von Wireless LAN eine Genauigkeit von bis zu 10m.

Tabelle 2.1 zeigt eine Übersicht über die vorgestellten Verfahren mit Genauigkeiten.

	Infrarot	Ultraschall	Funk	Koppelnavigation
Client-basiert	<p>WIPS Genauigkeit: Raum Kosten: mittel</p>	<p>Cricket Genauigkeit: 1-3 cm Kosten: hoch</p>	<p>GPS Genauigkeit: <10 m Kosten: gering</p> <p>GSM (Radiomap) Genauigkeit: <10 m Kosten: mittel</p> <p>Laufzeitbasierte Lateration [Röh09] Genauigkeit: ~50 cm Kosten: mittel</p>	<p>Lokalisierung und Detektion von Personenbewegung [Wan07] Genauigkeit: 1-3 cm Kosten: gering</p>
Infrastruktur-basiert	<p>Active Badge Genauigkeit: Raum Kosten: mittel</p>	<p>ActiveBat Genauigkeit: ~10 cm Kosten: hoch</p>	<p>SpotON Genauigkeit: 3 m Kosten: hoch</p>	<p>? Unabhängig von Infrastruktur</p>

Tabelle 2.1: Einige wichtige Indoor-Lokalisierungsverfahren⁶

⁶ Die angegebenen Kosten beziehen sich nicht auf die Beschaffung der Hardware, sondern den Aufwand der Installation und Wartung des Systems.

2.3 Grundlagen der Kommunikation

Neben der Lokalisierung ist auch die Verarbeitung der Daten von Bedeutung. Dies geschieht nicht zwangsläufig in dem System, dessen Position ermittelt wurde, sondern einem weiteren zentralen System zur Weiterverarbeitung oder Darstellung. Dieses Unterkapitel beschreibt eine Auswahl verschiedener Verfahren bzw. Systeme der Kommunikation und wurde unter Verwendung der Bücher *Computernetzwerke* [Tan00], *Mobile Computing* [Rot02] und *ZigBee* [Kup07] verfasst.

OSI-Referenzmodell

In drahtgebundenen sowie drahtlosen Netzen wird die Kommunikation über eine von „übereinandergestapelte“ Ebenen oder Schichten dargestellt. Einzelne Schichten bieten ihren jeweilig höheren Schichten bestimmte Dienste an, ohne dass diese Schichten über die Implementierung der anbietenden Schicht Kenntnis haben. Beide Schichten der Kommunikationspartner vermitteln die Information nicht direkt über ihr spezifisches Protokoll, sondern nutzen die angebotenen Dienste über Schnittstellen (*Service Access Point* – SAP) und leiten damit die Daten an die darunterliegende Schicht weiter. Das physische Medium liegt unter der untersten Schicht, wo die eigentliche Kommunikation stattfindet.

Es gibt für die Kommunikation zwischen den Schichten vier verschiedene Dienstprimitive. In Senderichtung ist es die nächsthöhere Schicht die der darunterliegenden Schicht durch **.request** einen Auftrag signalisiert, welcher von dieser durch **.confirm** bestätigt wird. In Empfangsrichtung zeigt die niedrigere Schicht der darüber liegenden Schicht durch **.indicate** ein Eintreffen einer Nachricht an, was von der höheren Schicht durch **.response** bestätigt wird.

Das OSI-Referenzmodell baut auf dieser Abstraktion auf. Wenn auch dieses Modell als Referenz vorliegt, so findet es in der Praxis in der vollen Komplexität selten Anwendung - der hohe Abstraktionsgrad erhöht den Implementierungsaufwand. Abbildung 2.5 zeigt das siebenschichtige Modell und wird im Folgenden kurz erläutert.

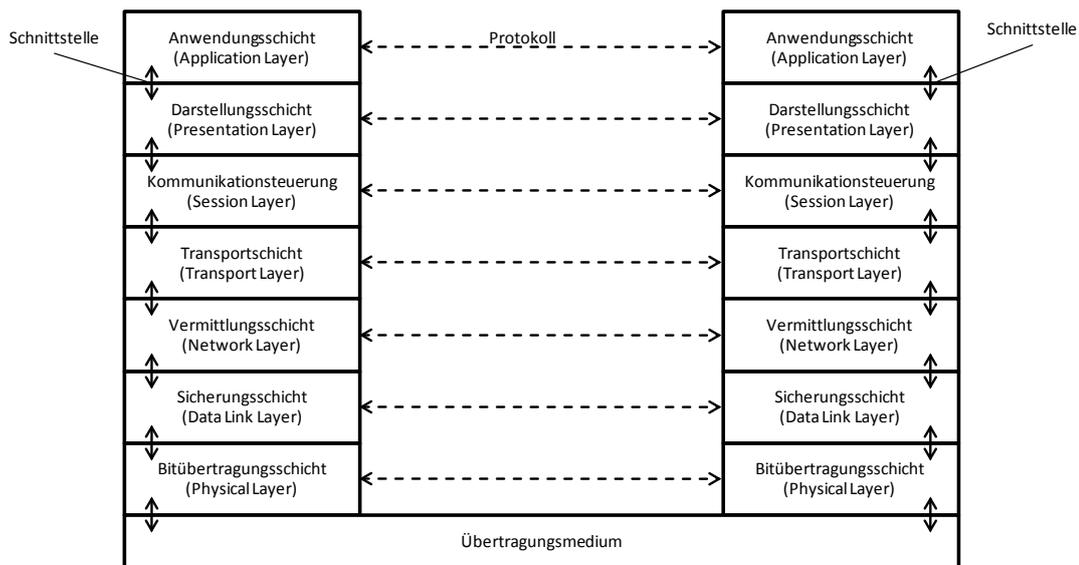


Abbildung 2.5: OSI-Referenzmodell

Die Bitübertragungsschicht

Der *Physical Layer* auf der ersten Schicht beschreibt die Übertragung einzelner Bits auf dem Medium (Kabel, Äther). Es wird beschrieben, welche mechanischen, elektrischen oder elektromagnetischen Eigenschaften die Bitinformation darstellen. Bei der drahtlosen Übertragung beispielsweise kann hier festgelegt sein, mit welcher Leistung und auf welcher Frequenz gesendet und empfangen wird. Im Allgemeinen wird diese Schicht u.a. als PHY-Layer bezeichnet.

Die Sicherungsschicht

Diese Ebene, im Englischen *Data Link Layer* genannt, repräsentiert die Zusammenfassung einzelner Bits im Datenstrom des PHY-Layer in Rahmen (Frames). Im Detail sollte die Sicherungsschicht unter anderem Rahmengrenzen anhand festgelegter Bitmuster erkennen, durch Störsignale zerstörte Frames feststellen (CRC⁷) und ggf. erneut übertragen (ARQ⁸) oder korrigieren können (FEC⁹).

Des Weiteren ist der *Data Link Layer* z.B. für die Unterbindung einer möglichen Datenduplizierung, beispielsweise durch zerstörtes Bestätigungsframe und der darauffolgenden erneuten Übertragung zuständig. Das „Wetteifern um die freie Leitung“ wird auch durch einen Teil der Sicherungsschicht, der MAC-Teilschicht, realisiert. Genauer beschreibt diese den koordinierten Zugriff mehrerer Sender auf das Übertragungsmedium, so dass Kollisionen und damit Datenverluste vermieden werden.

⁷ Cyclic Redundancy Check – Bildung einer Prüfsumme

⁸ Automatic Repeat reQuest – Wiederholtes Senden im Fehlerfall

⁹ Forward Error Correction – Fehlerkorrektur durch redundante Daten

Die Vermittlungsschicht

Der *Network Layer* ist für das Routing der Frames zuständig. Die Routen können statisch definiert sein oder dynamisch für jedes Paket¹⁰ einzeln gelten und gehen über die Punkt-zu-Punkt-Verbindungen der Sicherungsschicht hinaus. Eine weitere Aufgabe der Vermittlungsschicht ist die Steuerung der Pakete, um eine gute Auslastung des Netzes zu garantieren und Überlastung desselben zu verhindern.

Die Transportschicht

Der *Transport Layer* bietet höheren Schichten eine transparente Mehrfachbenutzung durch Multiplexing an. Es stellt eine echte *Ende-zu-Ende-Schicht* dar – darunterliegende Schichten bestehen im Gegensatz dazu aus verketteten Punkt-zu-Punkt-Verbindungen.

Die Sitzungsschicht

Zur Herstellung von Transaktionen wurde der *Session Layer* erdacht, beispielsweise werden damit Sitzungen unterbrochener Dateitransfers dargestellt. Diese zusätzliche Abstraktionsebene hat sich in den meisten etablierten Modellen nicht durchsetzen können und wird in der Regel in der Applikation selbst realisiert.

Die Darstellungsschicht

Auch die Darstellungsschicht, englisch *Presenting Layer* genannt, findet in der Regel in heutigen Modellen keine Anwendung. Hier wird die Syntax und Semantik der übertragenen Daten, z.B. verschiedene Kodierungen, auf eine standardisierte Art und Weise übertragen.

Die Verarbeitungsschicht

Auf dieser Schicht wird das Protokoll der eigentlichen Anwendung vereinbart. Wie bereits erwähnt hat es sich durchgesetzt, dass z.B. im Internetprotokoll (IP) keine Sitzungs- und Darstellungsschicht vorhanden sind, diese werden dann oftmals in der Ebene der Anwendung realisiert.

2.4 Drahtlose Sensornetzwerke

Es gibt eine Vielzahl von drahtgebundenen und drahtlosen Standards, jedoch gibt es im Bereich der drahtlosen Sensornetzwerke, im Englischen *Wireless Sensor Networks* (WSN) genannt, spezielle Anforderungen:

- geringer Energieverbrauch
- hohe Betriebsdauer
- geringe Kosten
- hohe Reichweite
- hohe Datenrate

¹⁰ Die Frames der Sicherungsschicht werden in der Vermittlungsschicht *Pakete* genannt.

Alle Anforderungen umzusetzen ist nur schwer möglich, da die einzelnen Teilprobleme zueinander im Konflikt stehen.

Die Standards des amerikanischen Ingenieurverbands Institut of Electrical and Electronics Engineers (IEEE) [IEEE] spielen für zahlreiche im Alltag vorkommende Technologien eine große Rolle, WLAN oder Bluetooth beispielsweise, wurden von IEEE erarbeitet und standardisiert.

Für drahtlose Sensornetzwerke ist besonders die Arbeit der Arbeitsgruppe 802.15 relevant – diese beschäftigt sich hauptsächlich mit der Definition der Bitübertragung- und Sicherungsschicht in drahtlosen Nahbereichsnetzen, sogenannten *Wireless Personal Area Networks* (WPANs). Daraus hervorgehend hat die untergeordnete *Task Group* den Standard IEEE 802.15.4 erarbeitet, welcher die zuvor genannten Anforderungen an drahtlose Sensornetzwerke (WSN) erfüllt und im folgenden Unterkapitel erläutert wird.

2.4.1 IEEE 802.15.4

Ein Standard für einen extrem geringen Energieverbrauch sowie einer geringen Komplexität und günstige Hardware- und Infrastrukturkosten drahtloser Sensornetzwerke war das Ziel, welches durch die im Jahr 2000 gegründete *IEEE 802.15 WPAN Task Group 4* spezifiziert werden sollte. Bereits 2003 wurde der Standard verabschiedet, welcher in den lizenzfreien Bändern 868/915 MHz sowie 2,4 GHz Datenraten von bis zu 250 Kbit/s erreicht.

802.15.4 steht synonym für ZigBee, obwohl dieser nur den PHY- und den MAC-Layer definiert.[ITW10a]

Nur die Bitübertragungsschicht und Sicherungssicht werden durch den IEEE-Standard, andere Ebenen des OSI-Referenzmodells werden jedoch beispielsweise durch *ZigBee* (siehe Kapitel 2.4.2) beschrieben.

Diverse Untergruppen (802.15.4a/b/c) haben den Standard um weitere PHY Schichten erweitert, bis er in der Version IEEE 802.15.4-2006 eine größere Überarbeitung erfuhr.[IEEE06]

Der PHY-Layer von IEEE 802.15.4

Wie bereits erwähnt sind in dem Standard die lizenzfreien Bänder 868/915 MHz sowie 2,4 GHz spezifiziert, jedoch hat sich der zweite Frequenzbereich bei den Halbleiterherstellern durchgesetzt. In diesem 83,5 MHz breiten Frequenzbereich steht eine höhere Bandbreite zur Verfügung, welche durch FDMA¹¹ in 16 Kanäle zu je 4 MHz und 1 MHz Kanalabstand aufgeteilt werden und weltweit einsetzbar sind (siehe Tabelle 2.2).

¹¹ FDMA: Frequency Division Multiple Access – Aufteilung eines Frequenzbandes in mehrere schmalbandige Frequenzkanäle

Kanal	11	12	13	14	15	16	17	18
Mittelfrequenz (MHz)	2405	2410	2415	2420	2425	2430	2435	2440
Kanal	19	20	21	22	23	24	25	26
Mittelfrequenz (MHz)	2445	2450	2455	2460	2465	2470	2475	2480

Tabelle 2.2: Kanäle im 2,4 GHz-Band

Wie bereits in Kapitel 2.4 erwähnt, wird zwischen den Schichten mit Hilfe der Dienstprimitive kommuniziert. Der PHY-Layer stellt dem darüber liegenden MAC-Layer drei Dienstprimitive zur Verfügung (`.request`, `.confirm`, `.indication`). Dabei unterscheidet man zwischen Dienstprimitive für den Datenverkehr und das Management der Übertragung.

Der Datenverkehr besteht aus `PD-DATA.request` für die Übertragung von Datenrahmen sowie `PD-DATA.confirm` für deren Bestätigung und einem `PD-DATA.indication` Dienstprimitive womit die MAC-Schicht über den erfolgreichen Empfang eines Datenrahmens informiert werden kann.

Für das Übertragungsmanagement kann durch die Dienstprimitive `PLME-CCA.request/.confirm` abgefragt werden, ob der Kanal frei ist. Zum Prüfen auf Störquellen auf dem Kanal kann mittels `PLME-ED.request/.confirm` ein *Energy Detect Scan*¹², durchgeführt werden. Des Weiteren können zentrale Einstellungen, wie Kanal, Sendeleistung und Kanalüberwachungsmodi, für das Netzwerk durch `PLME-SET` und `PLME-GET` abgefragt und gesetzt werden.

Der MAC-Layer von IEEE 802.15.4

In der Sicherungsschicht wird der „Bitstrom“ in Rahmen (Frames) aufgeteilt. Diese Frames bestehen aus einer Nutzlast (Payload), einem Header, welcher Steuerinformationen enthält und in der Regel auch aus einem Footer für die Fehlererkennung.

Der PHY-Header ist sechs Byte lang, er besteht aus einer vier Byte langen Präambel für die Synchronisation des Empfängers, einem *Start of Frame Delimiter* (SFD)¹³ und einem weiteren Byte, in welchem nur 7 Bit genutzt werden, welches die Länge des darauf folgenden MAC-Frames (MPDU) enthält (max. 127 Byte).

Der anschließende MAC-Datenrahmen enthält seinerseits einen Header, einen Footer und die Payload. Im Header des Frames bestimmen die ersten beiden Bytes den Rahmentyp. Es folgt eine 1-Byte lange Sequenznummer, um den Rahmen eindeutig identifizieren zu können und abschließend die Quell- und Zieladresse des Rahmens, welche zusammen 4 bis 20 Bytes belegen. Nach der Nutzlast des Datenrahmens schließt sich der Footer mit einer *Frame Check Sequence* (FCS) an. Diese stellt einen 16 Bit langen CRC-Wert des gesamten MAC-Datenrahmens für die Fehlererkennung dar.

¹² Messung der Signalstärke auf dem Kanal

¹³ SFD: Start of Frame Delimiter – Zeigt Beginn der Nutzlast an

Davon ausgehend, dass der Overhead von PHY- und MAC-Layer bis zu 31 Bytes groß sein kann, beträgt der maximale Overhead durch Header und Footer bei einer Gesamtrahmenlänge von 133 Byte circa 23 Prozent.

Das folgende Zitat bezieht sich auf die Darstellung von Frames (siehe Abbildung 2.6).

Auch wenn es bei Funksystemen allgemein verbreitet ist, diese Präambel der Bitübertragungsschicht zuzuordnen, so gehört sie streng genommen nach dem OSI-Referenzmodell zur Sicherungsschicht. [Kup07] S. 66

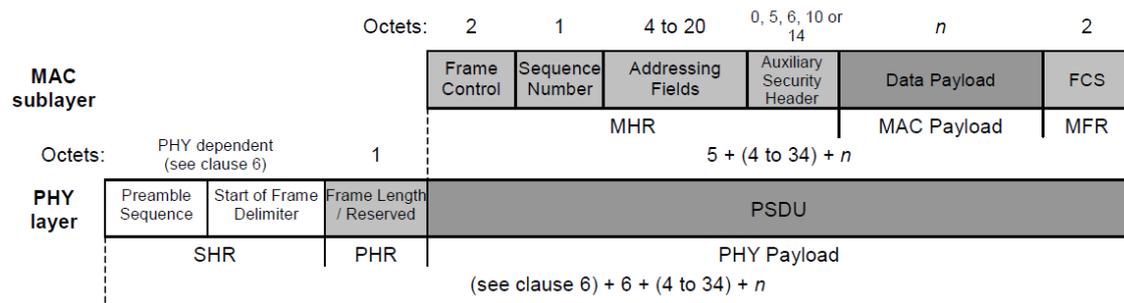


Abbildung 2.6: PHY- und MAC-Datenrahmen [IEEE06]

Es gibt vier verschiedene Datenrahmen, welche in IEEE 802.15.4 definiert sind. Mit Hilfe von Beacon-Rahmen können Koordinatoren andere Knoten periodisch informieren bzw. synchronisieren. Datenrahmen (siehe Abbildung 2.6) enthalten die eigentliche Payload, wobei Bestätigungsrahmen (Acknowledgment) den fehlerfreien Empfang von vorausgegangen Datenrahmen bestätigen. Steuerrahmen letztlich sind für die Netzwerkverwaltung, beispielsweise das An- und Abmelden von Stationen, zuständig.

In IEEE 802.15.4 sind zwei verschiedene Möglichkeiten der Adressierung von Knoten vorgesehen. Zum Einen gibt es den 8 Bytes langen *Extension Unified Identifier* (EUI-64 oder *IEEE-Adresse*) und Kurzadressen mit einer Länge von 2 Bytes. Da die Verwendung langer Adressen einen großen Overhead bei der Daten-Kommunikation verursacht, kann während der erstmaligen Anmeldung eines Knotens durch einen Koordinator solch eine Kurzadresse zugewiesen werden. Jedoch können in diesem Netz nun „nur noch“ bis zu 65536 Knoten teilnehmen.

Um das Netz, in dem sie ihren Dienst verrichten, eindeutig zu identifizieren, haben die Knoten zusätzlich zur Ihrer Adresse noch eine *PAN ID*, daher ist es möglich, dass die gleichen Kurzadressen mehrmals vergeben werden können.

Jedes *Personal Area Network* hat mindestens einen *PAN Coordinator*, welcher für die Verwaltung des Netzwerks verantwortlich ist. Ferner gibt es in dem Standard zwei verschiedene Gerätetypen. Das *Full Function Device* (FFD) kann in beliebigen Topologien eingesetzt werden, da es mit beliebigen Geräten kommunizieren und selbst einen *PAN Coordinator* (PC) darstellen kann. Im Gegensatz dazu kann ein *Reduced Function Device* (RFD)

kein *PAN Coordinator* werden und hat nur die Möglichkeit mit einem FFD zu kommunizieren. Das RFD ermöglicht jedoch eine preisgünstigere Implementierung.

Mit den erläuterten Gerätetypen sind nun die folgenden zwei Netztopologien möglich (siehe Abbildung 2.7).

In einer Stern-Topologie kommunizieren alle Geräte als Endknoten mit dem *PAN Koordinator* und können ein FFD, aber auch ein RFD sein. Die Peer-to-Peer-Topologie ermöglicht ein voll vermaschtes Netz der in Reichweite befindlichen FFD-Knoten, wobei ein FFD die Rolle des *PAN Coordinator* übernimmt. RFDs können in einer Mischform wiederum nur mit dem PC direkt kommunizieren.

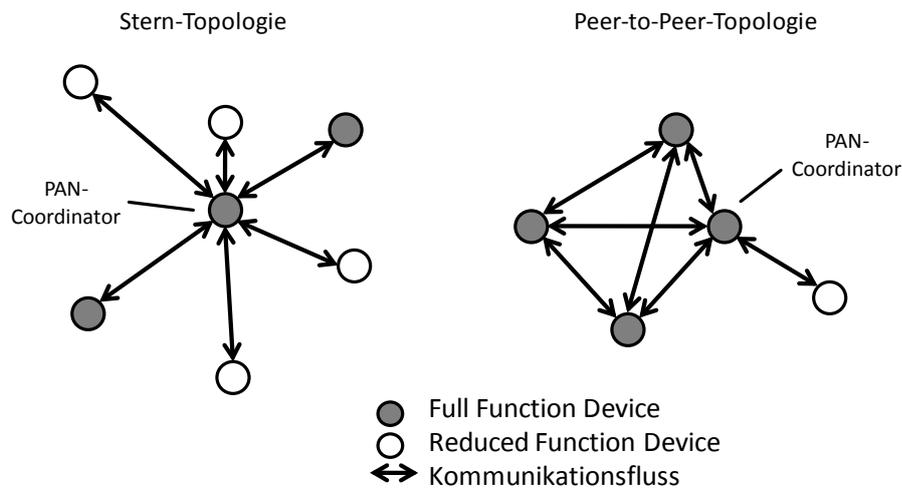


Abbildung 2.7: Topologien nach IEEE 802.15.4 - vgl. [IEEE06]

Gemäß dem Standard können Daten zwischen den Knoten erst nach dem Anmeldevorgang ausgetauscht werden. Der Ablauf des Anmeldens wird daher exemplarisch kurz verbal erläutert. Zwischen dem PHY- und dem MAC-Layer wird dabei nicht differenziert, sondern beide Schichten zusammen betrachtet.

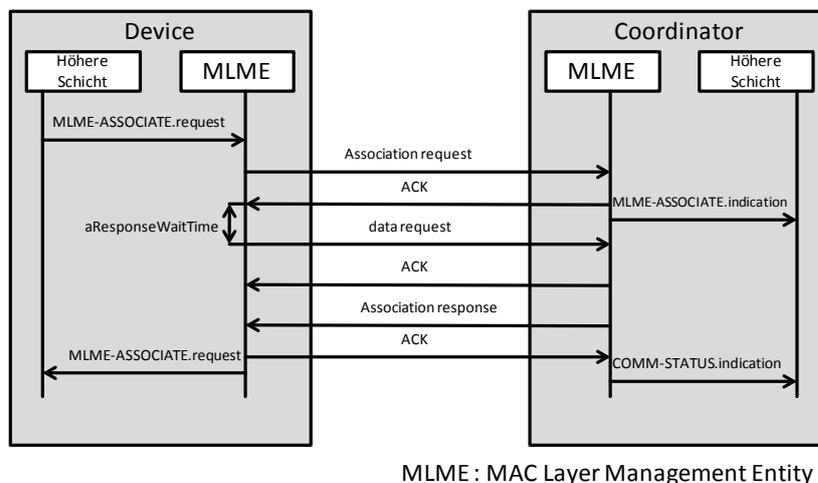


Abbildung 2.8: Anmeldung eines Knotens im Netzwerk - vgl. [Kup07] S. 71

Aufgrund der Aufforderung einer höheren Schicht sendet zu Beginn das Device ein *Association Request* an den *PAN Coordinator*, nach einer gewissen „Überdenkzeit“ des PCs sendet das Device erneut ein *Data Request* an den PC. Ist der anfragende Knoten ein zulässiges Device sendet nun der PC eine *Association Response*, damit ist das Anmelden abgeschlossen. Je nach Einstellungen des PAN können sämtliche Frames durch ein Bestätigungsframe bestätigt werden, dies wird für alle Teilnehmer vom PC vorgegeben. Der Vorgang wird in Abbildung 2.8 dargestellt.

Abschließend wird auch der Kanalzugriff auf das Medium durch den MAC-Layer beschrieben. Dazu gibt es das Verfahren *Carrier Sense Multiple Access Collision Avoidance (CSMA/CA)* sowie das *Beacon-Enabled-Verfahren*.

CSMA/CA prüft vor dem Senden, ob der Kanal frei ist und wartet im negativen Fall eine bestimmte Zeit. Das Warten wird bis zu fünf Mal wiederholt (*Number of Backoffs*), wobei jedes Mal die Wartezeit erhöht wird, bis schließlich der Zugriffsversuch abgebrochen wird. Durch dieses Verhalten wird die Kollisionswahrscheinlichkeit verringert, wenn der Kanal durch eine Übertragung belegt wurde, und im Anschluss viele Knoten einen Übertragungswunsch haben.

Verschiedene Zeitschlitze (*Timeslots*) werden im Beacon-Enabled-Verfahren für jeden angemeldeten Knoten reserviert. Zu Beginn sendet der Coordinator einen Beacon Frame, welcher Steuerinformation wie Beacon-Intervall und *Superframe-Dauer* enthält und der Synchronisation teilnehmender Knoten dient. Anschließend kann jeder Knoten in seinem zugeteilten Zeitschlitz Daten senden und empfangen, worauf bis zum nächsten Beacon eine Zeitspanne folgt, in welcher das Netzwerk inaktiv ist. Nun können die Knoten einschließlich des Koordinators in einen Stromsparmodus gehen. Da das Superframe, welches die Timeslots enthält, nur einen Bruchteil des Beacons lang sein kann und das Gerät damit aktiv sein muss, können Knoten über einen längeren Zeitraum durch eine Batterie betrieben werden. Abbildung 2.9 zeigt ein Netzwerk im Beacon-Enabled-Modus.

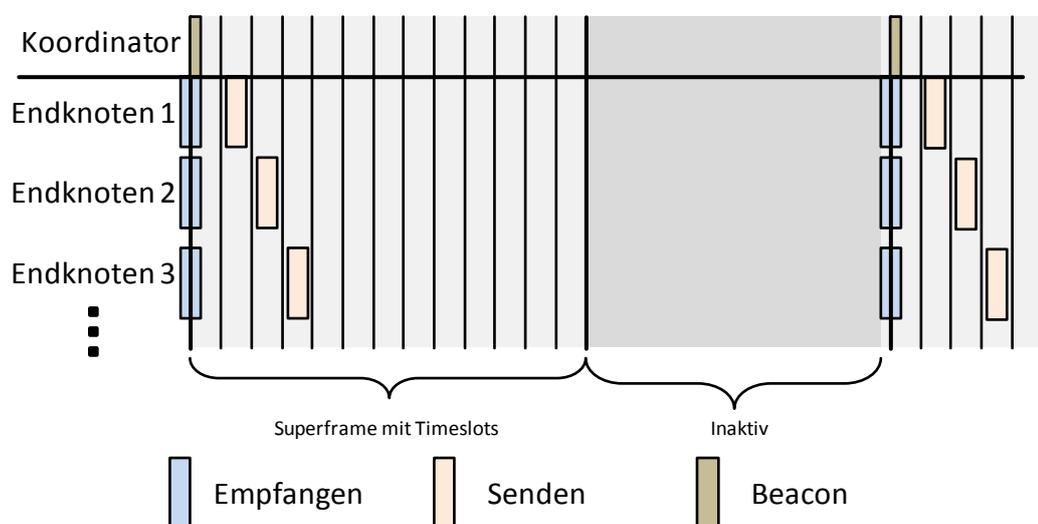


Abbildung 2.9: Beacon-Enabled Modus

2.4.2 ZigBee

IEEE 802.15.4 beschreibt die Bitübertragungs- und Sicherungsschicht, während die Protokolle der höheren Schichten unter dem Namen *ZigBee* veröffentlicht wurden.

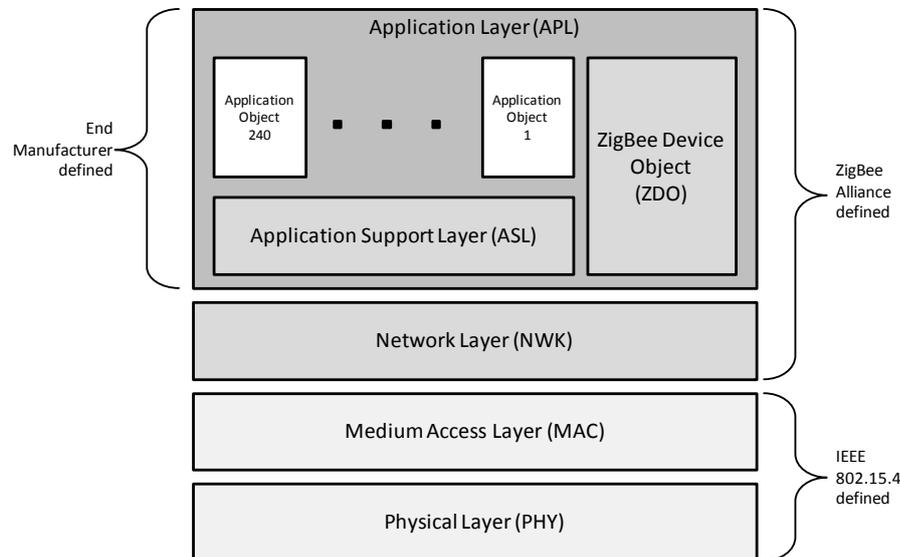


Abbildung 2.10: IEEE 802.15.4 und ZigBee im Überblick - vgl. [Kup07] S. 21

Aufbauend auf dem MAC-Layer folgt der *Network Layer* (NWK), diese Schicht ist für das Eintreten und Verlassen des Knotens in ein Netzwerk, das Weiterleiten von Datenpaketen zu ihren Zielknoten sowie Feststellen und Instandhalten von Routen zuständig. Dem Network Layer sind darüber hinaus Informationen über die direkten Nachbarknoten bekannt. In der nächst höheren Schicht, dem *Application Layer* bietet der *Application Support Sublayer* (ASL) die Anbindung von bis zu 255 *Application Objects* an den Network Layer. Darüber hinaus steuert in dieser Schicht das *ZigBee Device Object* (ZDO) Funktionen des Network Layer und bietet den *Application Objects* dafür eine Schnittstelle an. Abbildung 2.10 zeigt den kompletten Protokollstapel *ZigBee*.

Des Weiteren wird in *ZigBee*-Netzen die Netztopologie des IEEE 802.15.4-Standards erweitert, es gibt nun genau einen *ZigBee*-Koordinator, welcher hauptsächlich für den Aufbau des Netzes verantwortlich ist - ein FFD im Sinne des IEEE 802.15.4-Standards. Zusätzlich gibt es *ZigBee*-Router (FFD), deren wichtigste Aufgabe es ist, Daten im Netzwerk weiterzuleiten und schließlich *ZigBee*-Endknoten, welche als RFD die eigentliche Anwendungsfunktionalität bereitstellen.

Vertiefende Informationen zum ZigBee-Standard werden in [Kup07] erläutert.

2.4.3 Weitere drahtlose Netzwerke-Technologien

Bluetooth

Das nach einem Wikingerkönig benannte Bluetooth (Blauzahn) ([Mul01] S. 33) repräsentiert den Standard IEEE 802.15.1 und sendet im 2,4 GHz lizenzfreien ISM-Band¹⁴. Um zu anderen Technologien im gleichen Frequenzbereich weites gehend störungsfrei zu bleiben, verwendet Bluetooth verschiedene Methoden.

Zum Beispiel verwendet das Frequenzsprungverfahren *Frequency Hopping Spread Spectrum* (FHSS) eine höhere Bandbreite bei gleichzeitig geringerer benötigter Sendeleistung. Hierbei verwendet FHSS 1600 Sprünge pro Sekunde auf bis zu 79 Frequenzen. Für das Timing der „Hops“ wird eine Synchronisierung der Frequenzen sämtlicher Teilnehmer notwendig. Dies wird durch den Takt eines Masters realisiert (siehe Abbildung 2.11).

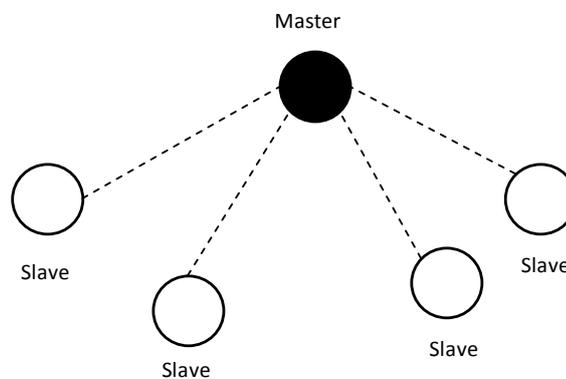


Abbildung 2.11: Bluetooth-Pico-Netz

Aufgrund der hohen Komplexität des Protokolls werden hohe Anforderungen an die Hardware gestellt – durch die langen Synchronisationszeiten erfordert das FHSS einen stetig hohen Strom, so dass es für drahtlose Sensornetzwerke weniger geeignet ist. Ein weiterer Nachteil von Bluetooth ist, dass Verbindungen nur Punkt-zu-Punkt oder als Stern-Topologie im Rahmen eines Pico-Netzes möglich sind [Mul01].

Wireless LAN

Wireless LAN (WLAN) basiert auf dem Standard IEEE 802.11 und stellt im Gegensatz zu Bluetooth oder 802.15.4 kein WPAN dar - es ist für hohe Datenraten über größere Distanzen (<100 m) ausgelegt. Der hohe Energieverbrauch von Wireless LAN spricht gegen den Einsatz in drahtlosen Sensornetzwerken. Wireless LAN sendet und empfängt wie Bluetooth im 2,4 GHz-Band¹⁵ und verwendet auf der Bitübertragungsschicht das *Direct Sequence Spread Spectrum*-Verfahren (DSSS) mit in Verbindung mit *Code Division Multiple Access* (CDMA) auf 14 Kanälen. Das Spreizen des Datensignals erhöht die Störsicherheit, erfordert jedoch auch

¹⁴ Industrial, Scientific, Medical

¹⁵ IEEE 802.11a/n sehen zusätzlich das ebenfalls lizenzfreie 5 GHz Band vor.

eine höhere Bandbreite, womit es andere Netzwerke, beispielsweise Bluetooth oder 802.15.4 erheblich beeinflussen kann. Für den Einsatz von WLAN sind von dem Standard zwei Topologien vorgesehen. Es kann im Infrastruktur-Modus betrieben werden, wobei die Kommunikation über *Access Points* stattfindet, welcher seinerseits in Zugang zu einem stationären Netzwerk bietet. Der sogenannte Ad-hoc-Modus bietet lediglich die Möglichkeit eine direkte Verbindung zwischen zwei Stationen herzustellen, nicht jedoch über ein höheres Protokoll, wenn sich diese zum Beispiel außerhalb der Kommunikationsreichweite befinden (siehe Abbildung 2.12) [Rot02].

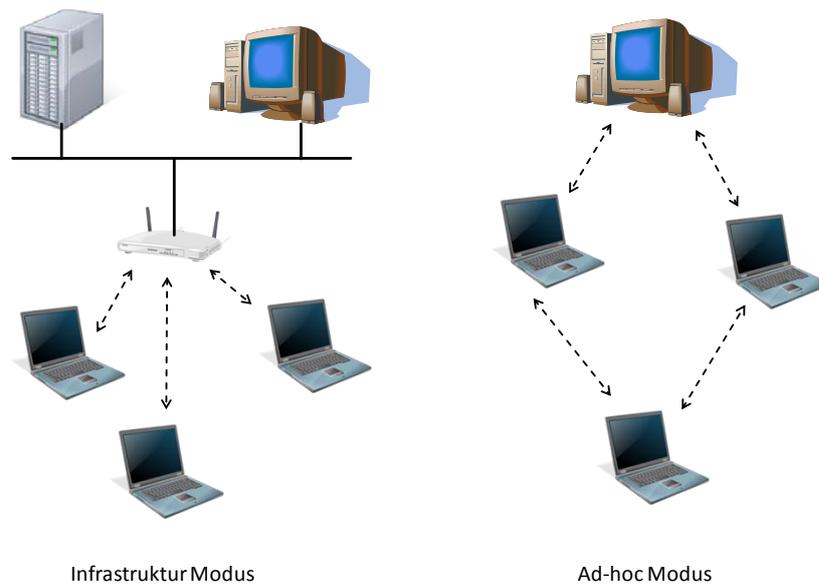


Abbildung 2.12: Wireless LAN Modi

2.5 GSM

Global System for Mobile Communications, kurz GSM, dessen Entwicklung 1982 begann, stellt ein zellbasiertes Funkübertragungssystem dar. Ursprünglich wurde es für die Sprachkommunikation entwickelt, wobei der Standard zunehmend auch für die Datenübertragung genutzt wird. Es werden derzeit in Deutschland zwei Frequenzbänder genutzt (900MHz / 1800MHz in Europa und 850MHz / 1900Mhz in Amerika). Diese Bänder werden in Europa durch FDMA in 548 Kanäle[Neu10] zu je 200 kHz unterteilt (ARFCN¹⁶). Die Hälfte dieser Kanäle sind jeweils für den Download bzw. Upload der (Sprach-)Daten vorgesehen sind. Durch den zellulären Aufbau des Netzes und den speziell ausgerichteten Antennen der Basisstationen (BTS¹⁷) wird durch den Netzbetreiber eine optimale Abdeckung ohne Überlappung gleicher Kanäle erreicht.

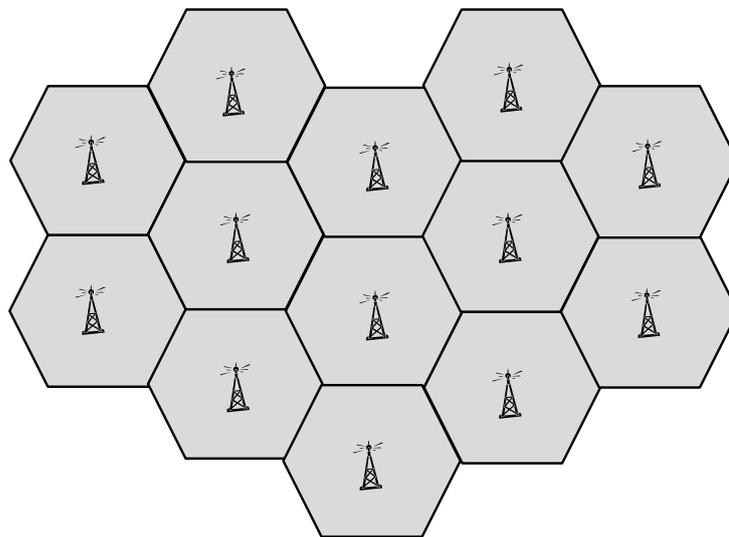


Abbildung 2.13: Einfache Darstellung der Zellstruktur von GSM

Die einzelnen Zellen werden durch die *Cell Identity* (CI) eindeutig identifiziert, was eine grobe Lokalisierung von Mobilfunkteilnehmern ermöglicht. Weiterführende Einzelheiten, beispielsweise über die Einteilung der Kanäle in weitere logische Kanäle durch TDMA¹⁸, der exakten Kalibrierung der Frequenz auf einen Kanal (FCCH¹⁹), der Rahmenstruktur dieser Kanäle sowie die Topologie des GSM-Netzwerks allgemein werden in *An Introduction To GSM* [Red95] detailliert erläutert.

¹⁶ *Absolut Radio Frequency Channel Number*

¹⁷ *Base Transceiver Station*

¹⁸ *Time Division Multiple Access* – Aufteilung eines Kanals in mehrere Zeitschlitz

¹⁹ *Frequency Correction Channel* – Kanal zur Anpassung der Frequenz

3 Konzept

In diesem Kapitel werden die Entscheidungen für die verwendeten Technologien sowie der Hardware begründet und die Funktionen der Plattform beschrieben.

3.1 Technologieentscheidung

Von Seiten des Kompetenzzentrums *Sensor Applications and Networks* (SANE) wurde im Rahmen dieser Arbeit eine Evaluierung der Übertragungstechnologie IEEE 802.15.4 für die Lokalisierung gewünscht. Neben den bekannten Vorteilen, wie z.B. geringer Energieverbrauch, lange Betriebsdauer im Batteriebetrieb (vgl. Kapitel 2.4.1), hat die Technologie den weiteren Nutzen, dass die Kommunikation darüber durchgeführt werden kann. Als Bauteil für diese Technologie wurde das Funkmodul RC2301AT der Firma *Radiocrafts* ausgesucht. Das Modul beinhaltet den Chip CC2431 von *Texas Instruments*. Die Besonderheit dieses Bausteins ist neben dem integrierten Mikrocontroller eine sogenannte *Location Engine*, eine Hardwarekomponente, welche Positionen anhand von Signalstärken von Basisstationen berechnen kann.

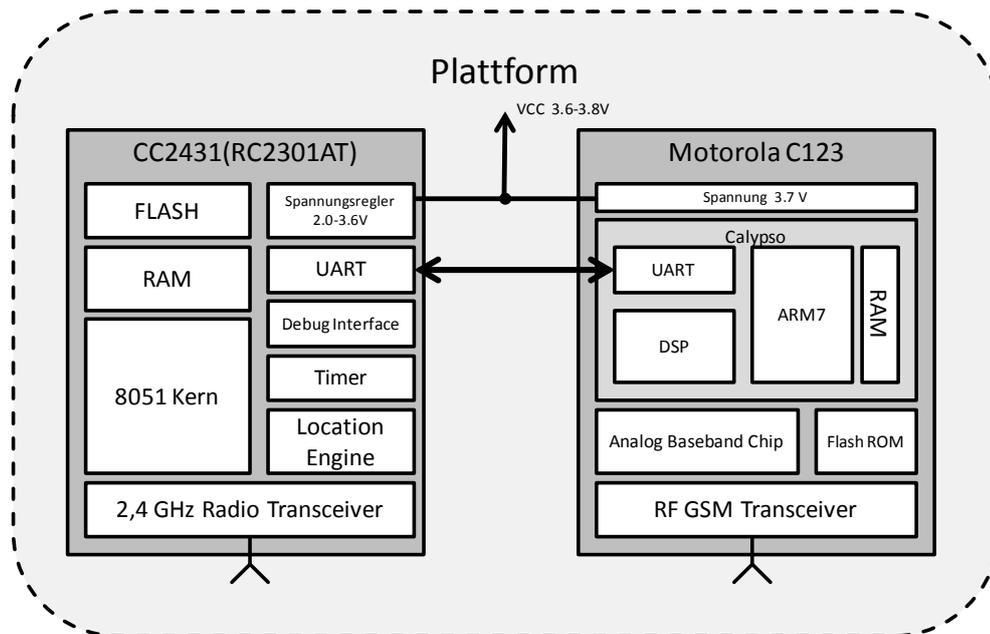
Des Weiteren wird konträr dazu ein *Radiomap*-Verfahren evaluiert, welches auf dem GSM-Standard beruht. Aufgrund der flächendeckenden Verfügbarkeit der Mobilfunksignale, auch innerhalb von Gebäuden, sowie der großen Verbreitung der Endgeräte ist es sinnvoll auch dies zu evaluieren. Da das *Radiomap*-Verfahren mit zunehmender Zahl von Signalstärke-Werten bessere Ergebnisse erzielt, wurde als Hardware hierfür das Mobiltelefon *Motorola C123* gewählt. Im Gegensatz zu anderen Telefonen bzw. GSM-Modulen ermöglicht das C123 eine größere Zahl von Signalstärken (fast) gleichzeitig zu ermitteln.

Weitere Details zu der Hardware sind im Kapitel 5.1 zu finden, während im Folgenden Unterkapitel *Funktionsbeschreibung* die beiden Lokalisierungsverfahren auf Basis der beiden Technologien erläutert werden.

3.2 Funktionsbeschreibung

Die Evaluierungsplattform besteht, wie bereits in der vorherigen Technologieentscheidung beschrieben, aus dem IEEE 802.15.4-Funkmodul *Radiocrafts* RC2301AT und dem Mobiltelefon *Motorola C123*. Das Funkmodul und das Mobiltelefon sind über eine serielle Schnittstelle verbunden. Über diese empfängt das RC2301AT-Modul die ermittelten RSSI-Werte des C123 Mobiltelefons. Abbildung 3.1 zeigt ein Blockschaltbild der verwendeten Hardware.

Sämtliche ermittelten Positionsdaten (RSSI und Koordinaten der *Location Engine*) werden zur Auswertung bzw. Darstellung über die IEEE 802.15.4-Funkschnittstelle des Funkmoduls an ein Gateway übertragen.

Abbildung 3.1: Blockschaltbild der Plattform²⁰

Lokalisierung durch Signalstärke-Lateration

Für die Lokalisierung durch das *Radiocrafts* Funkmodul ist eine zusätzliche Infrastruktur in Form von Basisstationen, welche ebenfalls über Module des IEEE 802.15.4-Standards verfügen, erforderlich. Da Geräte des IEEE 802.15.4-Standards einen geringen Energieverbrauch und Herstellungskosten haben, bleiben dabei die Gesamtkosten dieses Netzes gering. Hierbei werden diese an ausgewählten Positionen im Raum verteilt, um eine optimale Abdeckung aller Signalausbreitungswege zu erreichen. Sämtliche Basisstationen senden fortan ihre eigene Position kontinuierlich an die zu lokalisierende Evaluierungsplattform, diese wird im Folgenden daher als *Blind Node* bezeichnet. Nach dem Empfangen dieser Positionsangaben in dem Funkmodul kann diese mit Hilfe der Entfernungen zu den Basisstationen, welche aus den Signalstärkemessungen ermittelt wurden, berechnen. Umso mehr Basisstationen in Empfangsreichweite des Funkempfängers sind, desto genauer kann die Position durch *Multi-Lateration* berechnet werden – ein mathematischer Berechnungsansatz ist unter [Kar05] S. 237-239 zu finden. Das vorliegende Funkmodul enthält jedoch eine Hardware-Komponente zur Berechnung der Position aus diesen Werten.

Schließlich wird die berechnete Position an das Gateway zur Darstellung übertragen. Eine schematische Darstellung hierfür ist in Abbildung 3.2 zu finden.

²⁰ Die internen Komponenten des Funkmodul und des Mobiltelefons in dem Blockschaltbild sind stark vereinfacht dargestellt. Detailliertere Informationen zum internen Aufbau sind im Anhang in Abbildung A.3, Abbildung A.4 und Abbildung A.5 sowie auf der Webseite des *OsmocomBB*-Projektes [OSMOCOM] zu finden.

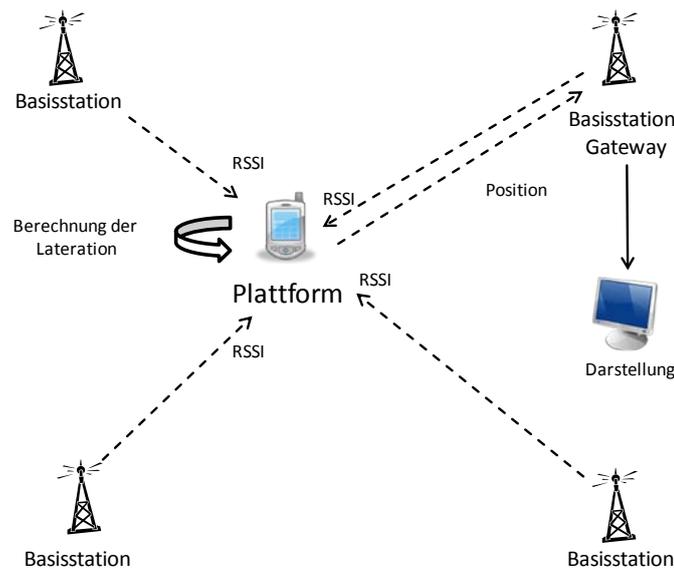


Abbildung 3.2: Positionsermittlung der Plattform durch Lateration mit IEEE 802.15.4

Lokalisierung mit Hilfe einer GSM-Radiomap

Bei der Positionsbestimmung im Indoor-Bereich durch GSM ist die Nutzung von Verfahren, welche auf Entfernungsmessung beruhen, aufgrund der hohen Dämpfung durch Hauswände hindurch ungeeignet. Daher wird bei der GSM-Lokalisierung der Plattform das *Radiomap*-Verfahren eingesetzt. Wie bereits im Kapitel 2.5 erwähnt, wird hierbei die Tatsache zunutze gemacht, dass benachbarte Basisstationen (BTS) auf unterschiedlichen Kanälen senden. An jeder beliebigen Position können theoretisch bis zu 548 Signalstärken verschiedener Kanäle gemessen werden (vgl. Kapitel 2.5). Dies ist jedoch aufgrund der geringen Dichte der GSM-Zellen bzw. der GSM-Basisstation und damit der großen Fläche auszuschließen, da die Empfindlichkeit des GSM-Moduls bzw. des C123 entsprechend hoch sein müsste. Allerdings kann von einer größeren Anzahl nutzbarer Kanäle ausgegangen werden, weil gerade im städtischen Raum die Dichte der BTS aufgrund Signaldämpfung und größerer Teilnehmerkapazitäten höher ausfallen müsste.

In der Abbildung 3.3 wird die Lokalisierung durch GSM schematisch dargestellt. Das zweistufige Verfahren ist unterteilt in die Trainingsphase und die eigentliche Lokalisierungsphase.

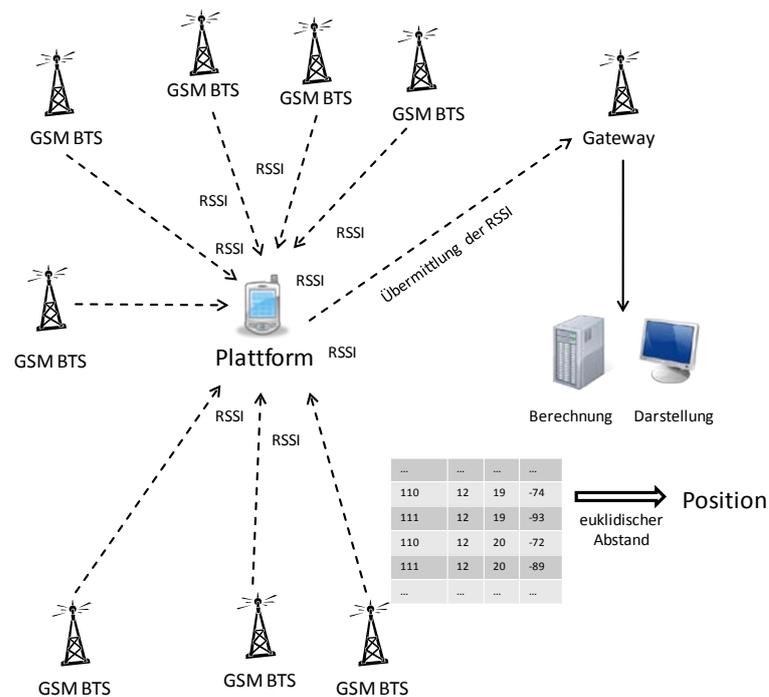


Abbildung 3.3: Positionsermittlung der Plattform durch GSM-Radiomap

Trainingsphase

In der Trainingsphase wird zunächst das gesamte Indoor-Areal eingemessen - dazu wird zunächst über die gesamte Fläche ein Gitter gelegt. Im Anschluss werden durch das *Motorola C123* für jede Teilfläche des Gitters die Signalstärke-Werte der verschiedenen Kanäle eingelesen. Diese werden daraufhin an das Gateway und einer dahinterstehenden Datenbank übertragen und zu der entsprechenden Position zugeordnet abgespeichert. Wie bereits erwähnt, ist es nicht sinnvoll alle Kanäle abzuspeichern, da die Empfindlichkeit des Telefons nicht ausreicht. Die richtige Wahl der n -besten Kanäle ist Teil empirischer Versuche. Des Weiteren muss beachtet werden, dass eine Verfeinerung des Gitters zwar die Genauigkeit erhöht, jedoch auch den Einmessaufwand beachtlich erhöht. Im Rahmen dieser Arbeit wird aus Zeitgründen eine Gitterteilfläche von 1 m^2 verwendet.

Lokalisierungsphase

Ist die Trainingsphase abgeschlossen und das gesamte Areal eingemessen kann die Plattform nun mit Hilfe der Datenbank die Position bestimmen. Dazu misst das C123 zunächst die Signalstärken aller n -besten Kanäle, sendet selbige an das Gateway, um diese mit den verschiedenen gespeicherten Positionen in der Datenbank zu vergleichen. In diesem Schritt wird daher die Position in der Datenbank als die wahrscheinlichste angenommen, welche den kleinsten *Euklidischen Abstand* [EuAb] hat. Der euklidische Abstand berechnet sich wie folgt

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Formel 3.1: Euklidischer Abstand

wobei x die Menge der n -besten zugeordneten Kanäle einer Position aus der Datenbank darstellt und y die Menge der aktuell gemessenen n -besten Kanäle.

Da Messungen auf Basis der Signalstärke großen Schwankungen durch Interferenzen, Reflexion und Dämpfung unterliegen, und leichte Änderungen in der Umgebung die Signalausbreitungscharakteristik stark beeinflussen können, wird zusätzlich der Algorithmus der *K-Nächste-Nachbarn* [KNN] angewendet. Die Mittelung der Koordinaten von den k nächsten Nachbarn (euklidische Abstände) ergibt in der Regel eine Position, welche näher an der wahren Position ist (siehe Abbildung 3.4). Für zu große k besteht jedoch die Gefahr, dass Koordinaten, welche von der korrekten Position weit entfernt sind, die Berechnung der ermittelten Position negativ beeinflussen.

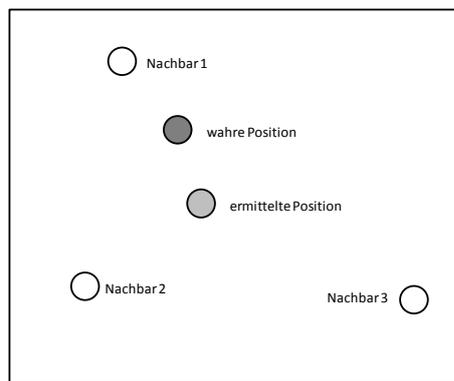


Abbildung 3.4: Verbesserung der Ortung durch Mittelung der Koordinaten

3.3 Softwarekomponenten

Die Abbildung 3.5 zeigt die eingesetzten Softwarekomponenten auf den verschiedenen Hardwarekomponenten und deren Kommunikationswege.

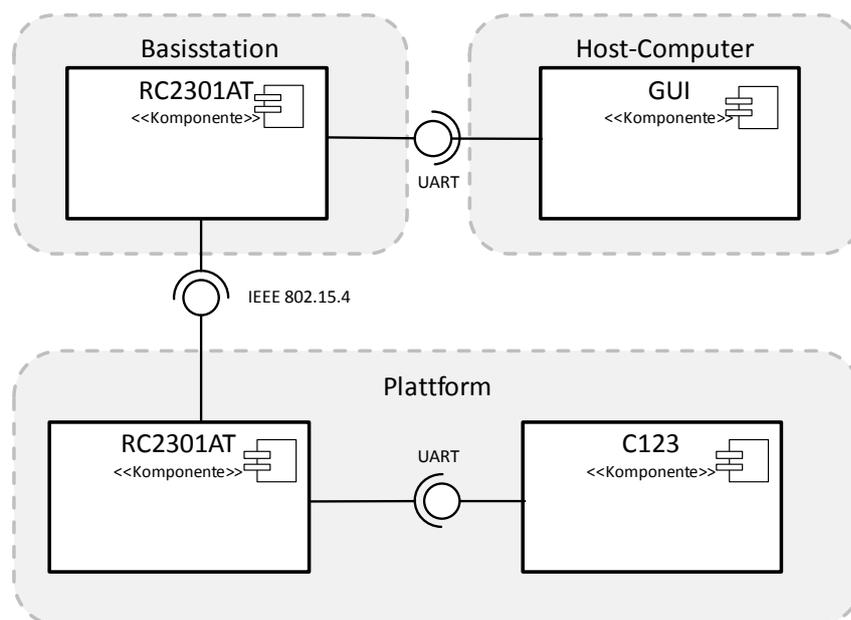


Abbildung 3.5: Softwarestruktur des Systems als vereinfachtes Komponentendiagramm

4 Pflichtenheft

In diesem Kapitel wird ein wichtiger Teil eines Hardware- oder Software-Projekts, das Pflichtenheft, erläutert. Als Leitfaden für den Aufbau des Pflichtenheftes wurde das *Lehrbuch der Software-Technik – Software Entwicklung* von Helmut Balzert [Bal00] herangezogen. Zur Aufgabe des Pflichtenheftes schreibt der Autor: „Das Pflichtenheft enthält eine Zusammenfassung aller fachlichen Anforderungen, die das zu entwickelnde Software-Produkt aus der Sicht des Auftraggebers erfüllen muss.“ [Bal00] S. 113.

Im Folgenden werden die aus Kapitel 1.2 dargelegten Anforderungen konkretisiert und das Pflichtenheft, entsprechend dem oben genannten an diese Arbeit angepassten Leitfaden, angefertigt.

4.1 Zielbestimmungen

Die Zielbestimmungen legen fest, welche Intention die Entwicklung eines Systems hat. Im Detail dient die zu entwickelnde Plattform der Evaluierung neuer Indoor-Lokalisationsverfahren, um so Ergebnisse für weitere Forschung oder Entwicklung am Fraunhofer Institut für Offene Kommunikationssysteme bereit zu stellen.

Musskriterien

Musskriterien umfassen Leistungen, welche für ein Produkt oder System „unabdingbar“ sind, und damit realisiert werden müssen:

- Die Lokalisierungsplattform verwendet den Funkstandard IEEE 802.15.4.
- Des Weiteren verwendet die Plattform mindestens eine weitere Funktechnologie zur Positionsbestimmung.
- Positionsdaten werden über eine geeignete Funkinfrastruktur, beispielsweise IEEE 802.15.4 PAN (vgl. Kapitel 2.4.1), an ein zentrales Gateway zur Berechnung bzw. Darstellung übertragen.
- An zentraler Stelle wird die ermittelte Position durch ein graphisches Benutzerinterface (GUI) ausgegeben.

Wunschkriterien

Die Realisierung von Wunschkriterien sollte soweit wie möglich angestrebt werden, jedoch sind diese für die Funktionalität des Systems nicht unabdingbar.

- Für die Übertragung der Daten stehen mehrere Funktechnologien parallel zur Verfügung.
- Die Plattform verwendet anhand bestimmter Genauigkeitskriterien die geeignetste Lokalisierungstechnologie.

- Durch Fusion der ermittelten Positionen der eingesetzten Technologien wird eine höhere Genauigkeit erzielt.

Abgrenzungskriterien

Abgrenzungskriterien sollen darstellen, welche Funktionen im System ausdrücklich nicht realisiert werden sollen. Sie werden in dieser Arbeit auch nicht weiter betrachtet.

- Die aufzubauende Infrastruktur ist einfach gehalten, so dass kein aufwendiges Routing implementiert werden muss.
- Bei der Implementierung wird auf eine bestehende Bibliothek des IEEE 802.15.4 Standards zurückgegriffen, damit muss keine umfangreiche Implementierung selbiger stattfinden.
- Da es sich lediglich um eine Evaluierung der Genauigkeit der Ortung handelt, erfolgt keine Betrachtung bezüglich der Sicherheit der Funkübertragung.
- Die Problematik des Datenschutzes beim Einsatz des Produkts wird in dieser Arbeit nicht genauer beleuchtet.

4.2 Produkteinsatz

Bei der Entwicklung ist auch der spätere Einsatz des Systems relevant, daher werden hierbei die wichtigen Aspekte Anwendungsbereiche, Zielgruppen und Betriebsbedingungen erörtert.

Anwendungsbereiche

Der Hauptanwendungsbereich liegt im Bereich der Indoor-Lokalisierung, speziell in Märkten und Kaufhäusern mit den dortigen gegebenen Umgebungsbedingungen sollen die Verfahren evaluiert werden. Negative Umgebungseigenschaften für die Funkausbreitung können zum Beispiel hohe reflektierende metallische Regale, signaldämpfende Wände oder gar Flüssigkeiten sein.

Das Evaluieren durch das Produkt ist jedoch nicht darauf beschränkt, und kann somit auch in anderen Gebäuden, wo eine Positionsbestimmung gewünscht ist, eingesetzt werden.

Zielgruppe

Die wichtigste Zielgruppe dieser Arbeit sind Entwickler, denen die gewonnenen Lokalisierungsergebnisse für die weitere Entwicklung im Rahmen der Indoor-Lokalisierung zur Verfügung stehen.

Betriebsbedingungen

Die Plattform wird unter Laborbedingungen entwickelt und evaluiert um eventuelle Probleme und die darauffolgende Fehlersuche zu minimieren. Eine Evaluierung unter realen Bedingungen in einem realen Geschäft ist aus Zeitgründen im Rahmen dieser Arbeit nicht möglich.

4.3 Produktübersicht

Die Produktübersicht gibt eine Übersicht des Systems in Form eines Usecase-Diagrammes. Ein mögliches Anwendungsszenario zeigt Abbildung 4.1.

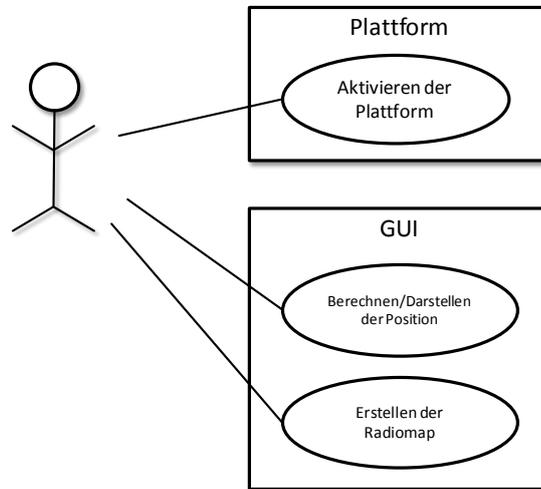


Abbildung 4.1: Übersichtsdiagramm für die Evaluierungsplattform

4.4 Produktfunktionen

Die für den Anwender wichtigsten Produktfunktionen werden stichpunktartig beschrieben.

Evaluierungsplattform

- Aktivieren der Plattform
 - **Ziel:** Position berechnen und GSM-Signalstärken ermitteln
 - **Vorbedingung:** Mindestens drei Signalstärke-Werte von Basisstationen in Reichweite liegen vor
 - **Nachbedingungen Erfolg:** Position berechnet bzw. ausreichend GSM-Signalstärken ermittelt und Daten an das Gateway gesendet
 - **Nachbedingungen Fehlschlag:** Keine oder zu wenig Signalstärken ermittelt
 - **Akteure:** Benutzer
 - **Auslösendes Ereignis:** Benutzer aktiviert die Plattform
 - **Beschreibung:** Die *Location Engine* des Funkmoduls berechnet mit Hilfe von vorliegenden Signalstärkewerten von Basistationen eine Position und das Mobiltelefon *Motorola C123* bestimmt auf sämtlichen verfügbaren GSM-Kanälen die Signalstärke. Die Plattform überträgt die Daten an das Gateway.
 - **Erweiterung:** -
 - **Alternative:** -

Graphische Benutzerschnittstelle

- **Berechnen/Darstellen der Position**
 - **Ziel:** Darstellung der aktuellen Position der Plattform in GUI
 - **Vorbedingung:** Gateway ist aktiv und es wurden Daten darüber erhalten
 - **Nachbedingungen Erfolg:** Position dargestellt
 - **Nachbedingungen Fehlschlag:** Position konnte nicht berechnet oder dargestellt werden.
 - **Akteure:** Benutzer
 - **Auslösendes Ereignis:** Positionsdaten wurden erfolgreich an das Gateway übermittelt
 - **Beschreibung:** Das Gateway berechnet anhand der empfangenen Daten die Position und stellt diese in der GUI dar.
 - **Erweiterung:** -
 - **Alternative:** -

- **Erstellen der GSM-Radiomap**
 - **Ziel:** Hinzufügen von Signalstärkewerten zur Datenbank
 - **Vorbedingung:** Gateway ist aktiv und es wurden Signalstärkedaten darüber erhalten
 - **Nachbedingungen Erfolg:** Daten hinzugefügt
 - **Nachbedingungen Fehlschlag:** Daten konnten nicht hinzugefügt werden.
 - **Akteure:** Benutzer
 - **Auslösendes Ereignis:** Positionsdaten wurden erfolgreich an das Gateway übermittelt
 - **Beschreibung:** Signalstärkewerte werden in einer *Radiomap*-Datenbank eingetragen.
 - **Erweiterung:** -
 - **Alternative:** -

4.5 Produktleistung

Es werden Anforderungen an die Leistungsfähigkeit eines Systems gestellt. Für die zu entwickelnde Plattform ist die Hauptanforderung, dass das System umgehend auf die Eingaben reagieren soll. Die Verarbeitungsdauer der Positionsbestimmung unterliegt jedoch wegen des Evaluierungshintergrundes keiner Vorgabe. An die Genauigkeit der Lokalisierung kann ebenfalls keine Anforderung gestellt werden, da es Gegenstand dieser Arbeit ist, diese zu evaluieren.

5 Realisierung

In diesem Kapitel wird die Umsetzung des in dem Kapitel *Konzept* entwickelten Systems dokumentiert. Zunächst wird die eingesetzte Hardware und Software beschrieben und im Anschluss der Aufbau der einzelnen Hardwarekomponenten vorgestellt. Daraufhin werden wichtige Punkte der Software-Implementierung erläutert. Abschließend werden eventuell aufgetretene Probleme dargelegt und eventuell durchgeführte Lösungsvorschläge erörtert.

5.1 Hardware und Software

Falls nötig wird im Folgenden die Verwendung der vorgestellten Hard- und Software begründet. Sämtliche Datenblätter der verwendeten folgenden beschriebenen Hardware sind auch auf der beigelegten CD vorhanden. Wenn nicht anders erwähnt, wird im Anschluss das Windows-Betriebssystem verwendet.

5.1.1 Motorola C123

Im Rahmen dieser Arbeit wurde das GSM-Modul *SIMCom* SIM508 und *Sony Ericsson* GM29 evaluiert. Es hat sich gezeigt, dass diese lediglich bis zu 6 bzw. 35 Kanäle der aktuell verwendeten bzw. benachbarten Zellen ermitteln können.

Diese Einschränkungen der jeweiligen Originalfirmware können durch das Projekt *OsmocomBB* (siehe folgendes Kapitel) in Verbindung mit dem Mobiltelefon C123 der Firma *Motorola* (Abbildung 5.1) behoben werden. Das Gerät unterstützt das europäische GSM 900 und GSM 1800 Band und damit können die Signalstärke-Werte von sämtlichen in Europa verfügbaren GSM-Kanälen ausgegeben werden. Die Kernkomponente des Telefons ist der Chip *Calypso* [CALYPSO] von Texas Instruments. Dieser besteht aus einem 32 Bit ARM7-Kern und einem integrierten digitalen Signalprozessor für die Verarbeitung der hochfrequenten Datensignale. Des Weiteren stehen dem Telefon 2 MB Flash-ROM und 256 kB SRAM für größere Programme zur Verfügung.



Abbildung 5.1: Motorola C123 [C123]

5.1.2 OsmocomBB

Das Open-Source-Projekt hat das Ziel einen kompletten freien Ersatz der Firmware für verschiedene Mobiltelefone der Firma *Motorola* zur Verfügung zu stellen. Dabei wurden bisher größtenteils die Schichten 1 und 2 des OSI-Modells im Protokoll-Stack implementiert. Da für die Lokalisierung Signalstärkeinformationen der Kanäle benötigt werden, kann die Bitübertragungsschicht von einem Teil des Projekts modifiziert für die Plattform verwendet werden [OSMOCOM].

Für Übersetzen des Programms wird der C-Compiler *GCC* für die ARM-Prozessorarchitektur eingesetzt. Das gesamte Projekt wurde in einer virtuellen Maschine unter Linux modifiziert und kompiliert [GCCARM].

5.1.3 Radiocrafts RC2301AT

Das *Radiocrafts*-Modul [RC2301AT] beinhaltet einen Mikrocontroller mit integriertem Funkmodul und Antenne. Die Hauptkomponente stellt der Funk-SoC²¹ CC2431 [CC2431] von Texas Instruments dar. Der Chip hat einen erweiterten *Intel* 8051-Kern mit 8 kB RAM und 128 kB Flash-Speicher, welcher mittels Bank-Switching voll adressierbar ist. Neben Standardkomponenten, z.B. seriellen Schnittstellen oder Analog-Digital-Wandlern, ist der im 2,4-GHz-Band funkende Transceiver vollständig IEEE 802.15.4-kompatibel und unterstützt CSMA-CA seitens der Hardware (vgl. Kapitel 2.4.1). Weiterhin eignet sich der CC2431 mit verschiedenen Power-Modi für *Ultra-Low-Power*-Systeme, wobei die zulässige Betriebsspannung des Moduls im Bereich von 2 – 3,6 Volt liegt.

Als Besonderheit bietet der Baustein eine sogenannte *Location Engine*, welche die Position des Knotens mit Hilfe bekannter Positionen anderer Knoten im Netzwerk bestimmen kann. Ein Prinzipschaltbild des CC2431 befindet sich im Anhang Abbildung A.3.



Abbildung 5.2: Radiocrafts-Modul RC2301AT

5.1.4 CC-Debugger

Das Funkmodul CC2431 von *Texas Instruments* wird über ein serielles proprietäres Debug-Protokoll²² programmiert und besitzt auch Debugging-Fähigkeiten. Der CC-Debugger [CCDEBUG] von *Texas Instruments* kann zum Flashen und Debuggen von fast allen Funk-SoC von *Texas Instruments*, einschließlich des CC2431, verwendet werden. Nach der Installation

²¹ SoC: System on Chip

²² Weitere Informationen unter [Goo10]

der notwendigen Treibersoftware ist die über die USB-Schnittstelle angeschlossene Hardware in der Entwicklungsumgebung *IAR Embedded Workbench* verfügbar und kann dort direkt für die Fehlersuche genutzt werden. Abbildung 5.3 zeigt neben dem *CC-Debugger* die Minimalbeschaltung für das Debugging eines Funkchips ohne Peripherie. Für den *CC-Debugger* ist auch die Software *SmartRF Flash Programmer* [TIFLASH] von *Texas Instruments* verfügbar, diese ermöglicht es Programme im Hex-Format in die Hardware zu übertragen sowie das exklusive Setzen der IEEE-Adresse in dem Baustein.

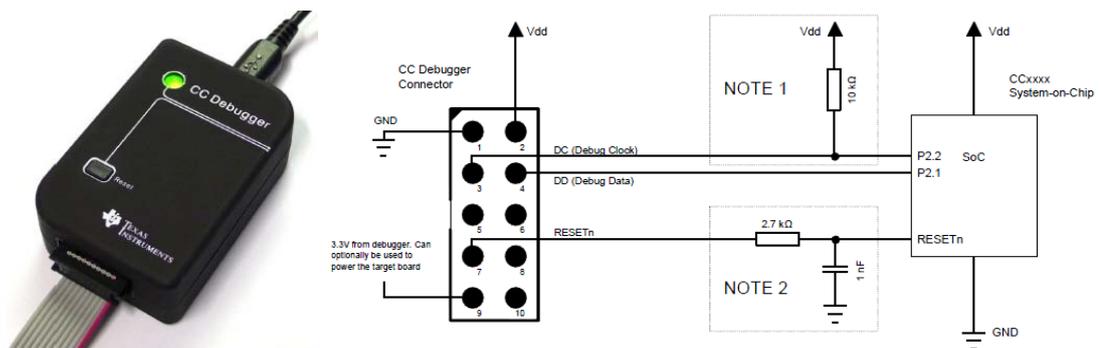


Abbildung 5.3: CC-Debugger von Texas Instruments und Minimalbeschaltung [CCDEBUG]

5.1.5 Host-Computer

Bei dem Host-Computer handelt es sich um ein handelsübliches Notebook mit dem Windows 7 Betriebssystem. Das Gerät wird durch ein USB-zu-Seriell-Adapter²³ um eine serielle Schnittstelle ergänzt.

5.1.6 TIMAC

Für die Kommunikation zwischen den Basisstationen und dem *Blind Node* wird ein drahtloses Netzwerk eingesetzt. Die Implementierung des dafür geeigneten IEEE 802.15.4-Standards würde jedoch den Rahmen dieser Arbeit sprengen, so dass eine bereits bestehende Lösung verwendet wird.

*TIMAC*²⁴ von Texas Instruments ist ein kostenloser IEEE 802.15.4-konformer Software Stack, welcher für drahtlose Sensornetze entwickelt wurde und dabei die Anforderung von WSNs erfüllt (vgl. Kapitel 2.4).

Die Software von TI ist modular und besteht aus drei Teilkomponenten (siehe Abbildung 5.4). Dieses Konzept ermöglicht eine vereinfachte Anwendungsentwicklung, unabhängig von der eingesetzten Hardware von *Texas Instruments* (vgl. [OSALHAL]).

²³ Adapter des c't-Magazin: [USB2BOT]

²⁴ IEEE802.15.4 Medium Access control (MAC) software stack [TIMAC]

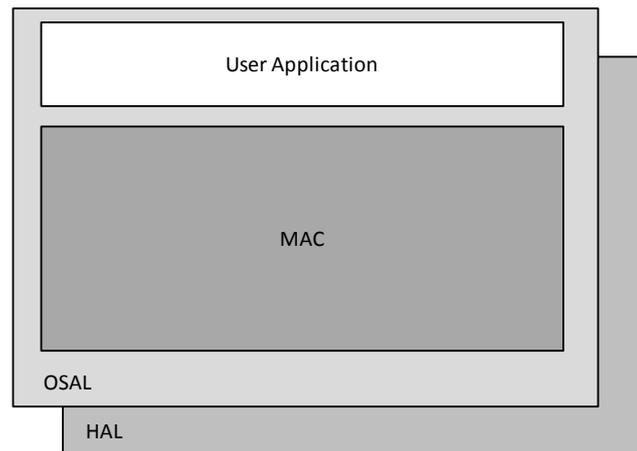


Abbildung 5.4: Aufbau des TIMAC Stack

OSAL

Der *Operation System Abstraction Layer* (OSAL) stellt zur Entwicklung eine API zur Verfügung, welche die Steuerung von Tasks, Messages, Interrupts, Timer, der Speicherverwaltung sowie Power Management anbietet.

HAL

Die API des *Hardware Abstraction Layer* (HAL) ermöglicht den vereinfachten Zugriff von der Software auf die Hardware. Der Softwareentwickler benötigt jedoch keine genauen Kenntnisse zur Ansteuerung der Hardware.

MAC

Die dritte Komponente *Medium Access Control* (MAC) beinhaltet ebenfalls eine API, diese stellt Funktionen für die Kommunikation nach IEEE 802.15.4 bereit (vgl. Kapitel 2.4.1).

TIMAC bietet eine Beispiel-Funkanwendung, welche für diese Arbeit verwendet und angepasst wurde.

5.1.7 IAR Embedded Workbench für 8051

Die IDE *IAR Embedded Workbench* (siehe Abbildung 5.5) bietet einen auf den 8051 angepassten C-Compiler sowie für verschiedene Mikroprozessoren zahlreicher Hersteller modifizierte Konfigurationsdateien. Darüber hinaus unterstützt IAR die verschiedenen Speicherbereiche des 8051, unter anderem auch das Bit-weise Adressieren von *Special Function Register* (SFR). Die Software bringt neben einem Simulator auch zahlreiche Plugins für Hardware-Debugger mit – unterstützt wird hierbei auch der *CC-Debugger*, was die Implementierung der Plattformsoftware erleichtert [IAREW].

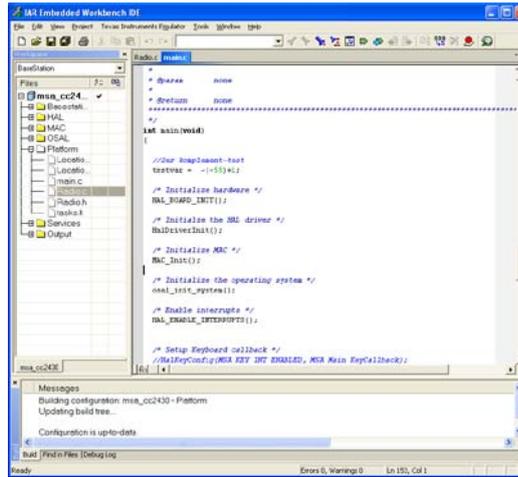


Abbildung 5.5: Die integrierte Entwicklungsumgebung IAR Embedded Workbench

5.1.8 Microsoft Visual Studio 2008

Visual Studio ist eine integrierte Entwicklungsumgebung und unterstützt zahlreiche Programmiersprachen. Es wird vornehmlich bei Entwicklung von Windows-Anwendungen eingesetzt, kann jedoch auch für Webservices oder dynamische Webseiten eingesetzt werden. Das Hauptaugenmerk lag in dieser Arbeit bei der Erstellung einer graphischen Benutzeroberfläche mit Hilfe der Programmiersprache C# für die Darstellung der ermittelten Position [VS2008].

5.1.9 Wireless Protocol Analyser

Während der Entwicklung der Plattform wurde die Software *Wireless Protocol Analyser (WPA)* (Abbildung 5.6) von der früheren Firma Integration genutzt. In Verbindung mit dem *CompXs IEEE 802.15.4 USB Dongle* (Abbildung 5.7) kann der Datenverkehr über die Luftschnittstelle analysiert werden, da die Software alle Datenrahmen aller möglichen IEEE802.15.4 Kanäle zeitlich darstellt.

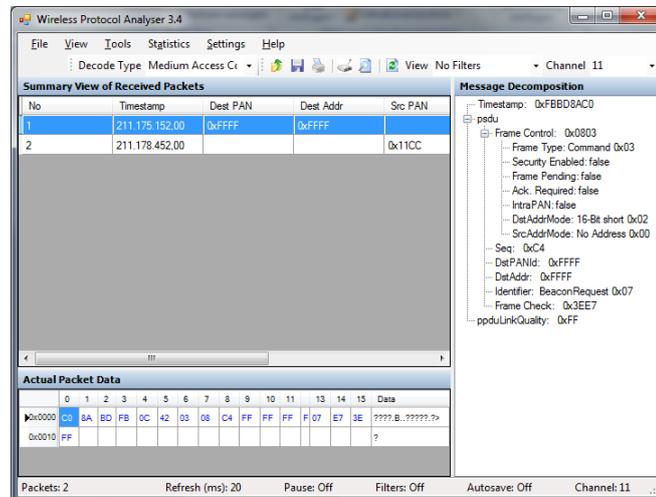


Abbildung 5.6: Wireless Protocol Analyser

Obwohl mit Hilfe des TIMAC Stacks standardkonforme Datenrahmen erzeugt und damit auch korrekte Kanalzugriffe sichergestellt werden, ist die Realisierung unter Verwendung des Sniffers²⁵ unabdingbar. Das Werkzeug ermöglicht es, die sequentielle Kommunikation verschiedener Knoten zu untersuchen, was die Entwicklung und die Fehlersuche sehr vereinfacht.



Abbildung 5.7: ZigBee USB Dongle von Integration [DONGLE]

5.1.10 HTerm

Analog zu WPA dient das Programm *HTerm* dazu, die übertragenen Daten der seriellen Schnittstelle protokollieren und im Anschluss analysieren zu können.

5.1.11 SmartRF Flash Programmer

Das Tool dient normalerweise zum Speichern von Programmen auf TI-Funk-SoC's, im Rahmen dieser Arbeit wurde es jedoch ausschließlich für das Speichern der IEEE-Adresse am Ende des Flash-Speichers verwendet (Abbildung 5.8).

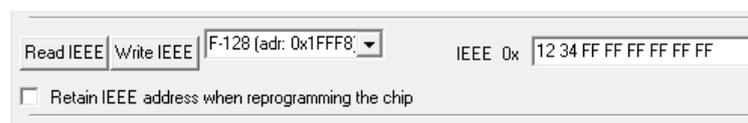


Abbildung 5.8: SmartRF Flash Programmer - IEEE Adresse

5.1.12 Weitere Hard- und Software

Die folgenden Hard- und Software-Komponenten wurden zu Beginn der Arbeit eingesetzt bzw. evaluiert, konnten jedoch im weiteren Verlauf aus verschiedenen Gründen nicht weiter verwendet werden.

²⁵ Werkzeug für die Analyse des Datenverkehrs auf einem Medium

m-Stack

m-Stack ist ein *Low-Power-Wireless-Software* Protokoll-Stack, welcher 6loWPAN²⁶ für die Kommunikation zwischen den Knoten verwendet. In dem freien Softwarepaket sind die Software sowie der Compiler und eine Entwicklungsumgebung enthalten [MSTACK].

Die vielversprechenden ersten Implementierungen mit der Software mussten jedoch aufgrund von Problemen mit dem SDCC-Compiler verworfen werden – Details dazu sind in Kapitel 5.3 zu finden.

TinyOS

Für das komponentenbasierte Betriebssystem *TinyOS* gibt es eine Portierung für 8051-Mikroprozessoren, speziell auch für den Funkchip CC2430/1 von *Texas Instruments* [TINYOS51].

Das Projekt bietet dem Entwickler jedoch nur eine reduzierte *SimpleMAC*-Bibliothek, wobei zusätzlich das Senden solcher „abgespeckter“ Frames durch die Komponente nicht gelang. Des Weiteren verwendet *TinyOS* auch den SDCC, was womöglich bei der weiteren Entwicklung ebenfalls Probleme mit sich gebracht hätte (siehe Kapitel 5.3).

CC Flasher

Da außer den Development-Kits von *Texas Instruments* zunächst keine Hardware zum Flashen der Programme auf die Funkmodule bekannt war, wurde hierzu das Projekt *CC Flasher* [CCFLASHER] vom gleichen Autor des *m-Stack*-Entwicklungspakets genutzt. Das Open-Source-Projekt bietet einen Schaltplan für die Herstellung einer Platine sowie die quelloffene Firmware für den Atmel AVR ATtiny2313, womit die Platine bestückt wurde. Mit Hilfe von *AVRStudio*²⁷ wurde die Firmware auf den Mikrocontroller übertragen. Nun konnte unter Verwendung der gleichnamigen Software *cc_flasher* auf dem Host-Computer das zu „flashende“ Programm über die USB Schnittstelle und der „Flasher“-Platine auf den Funkchip übertragen werden. Abbildung 5.9 zeigt die um einen ISP-Anschluss erweiterte Platine, wobei der Schaltplan im Anhang Abbildung A.2 zu finden ist.

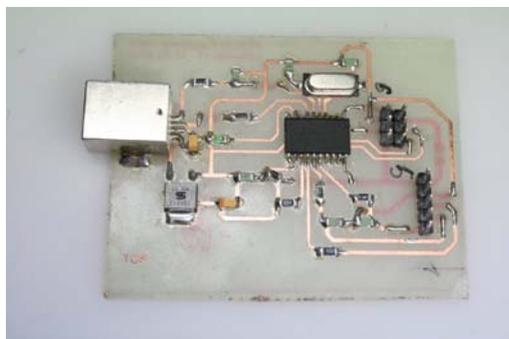


Abbildung 5.9: CC Flasher - Modifizierte Platine

²⁶ IPv6-Implementierung für IEEE 802.15.4-Netzwerke - detaillierte Erläuterung des Protokolls unter [6loWPAN]

²⁷ Integrierte Entwicklungsumgebung für 8 Bit AVR Mikrocontroller der Firma Atmel [Atm10]

Als Negativ war jedoch die fehlende Unterstützung eines leicht zu nutzenden Debuggers zu betrachten, zwar offeriert `cc_flasher` als Server eine Möglichkeit des Debuggens über das IP-Protokoll²⁸, jedoch konnte dies in keiner Entwicklungsumgebung sinnvoll eingebunden werden. Das Problem ließ sich unter Verwendung des Tools *CC-Debugger* lösen.

5.2 Implementierung

In diesem Unterkapitel werden Rahmenformate der Funkkommunikation, wichtige Kernpunkte für die Verwendung der *TIMAC*-Stacks, die Entwicklung der Plattform sowie der Basisstationen und der Software zur Darstellung und Berechnung auf dem Host-Computer beschrieben.

5.2.1 Verwendung der TIMAC-API

In der Implementierung wurde die Struktur der Beispielanwendung von *TIMAC* übernommen. Diese verwendet das Task-, Message- und Event-System der OSAL-API.

Zunächst werden mit Hilfe der OSAL-API für die MAC, die HAL und das Anwendungsprogramm eigene nebenläufige Tasks erzeugt und registriert. Die Reihenfolge der Registrierung gibt dabei die Priorität vor, wobei der Anwendungstask die geringste Priorität hat.

Die MAC-API, welche nur als Binär-Bibliothek vorliegt, repräsentiert größtenteils die Dienstprimitive des IEEE 802.15.4-Standards. Die Callback-Funktion der API `MAC_CbackEvent()` gibt verschiedene Nachrichten an das Anwendungsprogramm weiter und muss im Anwendungsprogramm implementiert werden. In dieser Funktion dürfen jedoch keine Prozessor-lastigen Operationen durchgeführt werden ([MAC] S. 2). Daher wurde in dieser Funktion die empfangene Nachricht des MAC-Tasks an den Anwendungstask gesendet, worauf dadurch ein Systemevent ausgelöst wird und die Nachricht in der OSAL-Funktion `ProcessEvents()` behandelt werden kann. Nachrichten werden durch die Funktionen `osal_msg_send()` und `osal_msg_receive()` (siehe [OSAL]) gesendet und empfangen. Des Weiteren sieht OSAL neben dem System-Event vor bis zu 15 weitere Ereignisse zu definieren.

Mit Hilfe der HAL-API kann einfach auf die Hardware zugegriffen werden. Es können LEDs geschaltet und Taster an Ports durch Callback-Funktionen abgefragt werden. Ferner stellt die API für die serielle Kommunikation Funktionen zum Lesen und Schreiben auf der Schnittstelle bereit. Die Interrupt-gesteuerte Callback-Funktion `HalUARTCback()` gibt Informationen über den aktuellen Zustand des UART-Interfaces zurück.

In der Abbildung 5.10 wird der Aufbau und Verwendung von Tasks, Events und Nachrichten dargestellt.

²⁸ gdb Remote Serial Protocol – weitere Informationen in [Gat99]

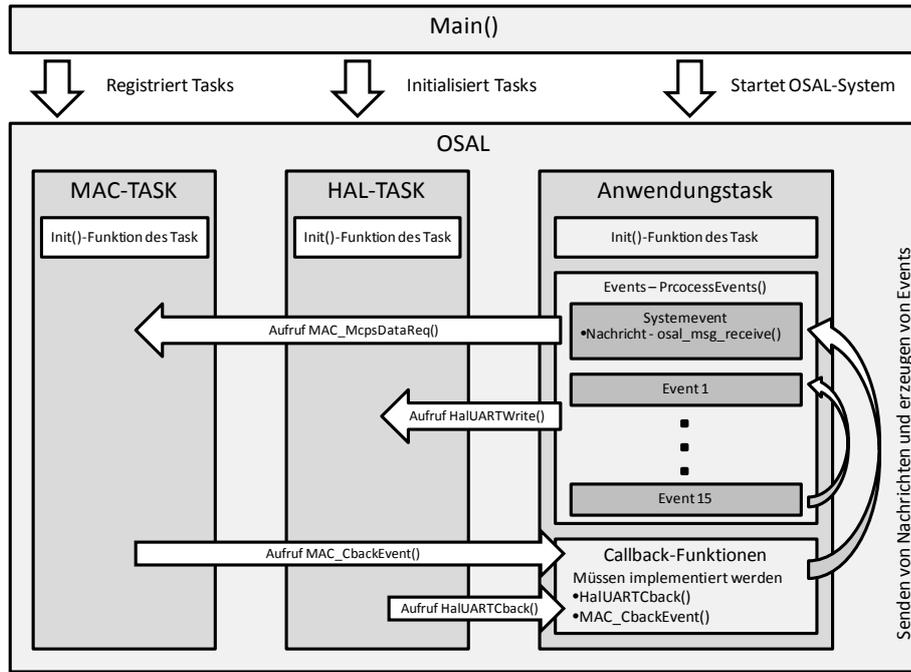


Abbildung 5.10: Struktur einer OSAL-Anwendung

5.2.2 Das IEEE 802.15.4-Netzwerk

Die IEEE-Adresse des CC2431

Da die IEEE-Adresse in einem Speicherbereich liegt, welcher bei richtiger Einstellung durch das Flashen des Programms nicht automatisch überschrieben wird, können Programm und IEEE-Adresse unabhängig voneinander durch die Software *SmartRF Flash Programmer* geschrieben werden. Das hat den Vorteil, dass man weitere Parameter in dem Flash-Speicher hinterlegen kann ohne das komplette Programm mit den veränderlichen Informationen erneut kompiliert werden muss. Mögliche Parameter sind zum Beispiel die Position einer Basisstation (Kapitel 5.2.4) oder die Kalibrierungsparameter *A* und *n* (Kapitel 5.2.5.2).

Im Detail wird die acht Byte lange IEEE-Adresse an das Ende des physikalischen Flash-Speichers geschrieben (**0x1FFF8 - 0x1FFFF**). Der Zugriff auf diesen Speicherbereich ist aber aufgrund der Begrenzung der 8051-Mikroprozessoren von 64 kB nicht direkt adressierbar (siehe Abbildung 5.11).

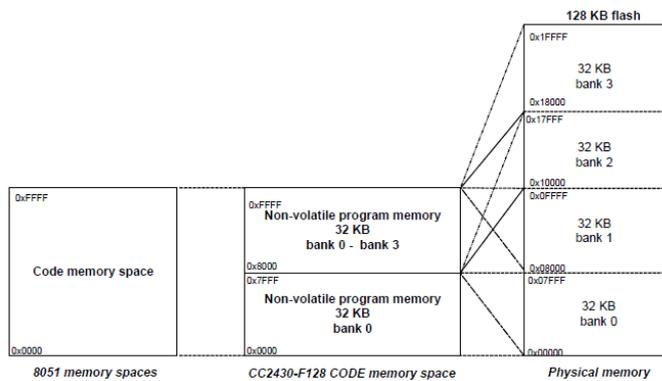


Abbildung 5.11: Das Speicherbank-Mapping

Um die Adresse, unabhängig von dem automatisierten Bank-Switching, durch den Compiler zur Laufzeit auslesen zu können, muss die letzte Bank in den Teil des Speicherbereichs des CODE-Segments gemappt werden. Dies geschieht durch Setzen der Bits `FMAP[1:0]` im SFR-Byte `MEMCTR` (siehe Abbildung 5.12). Nun kann die IEEE-Adresse im adressierbaren Bereich an `0xFFFF8` - `0xFFFFF` ausgelesen werden. Daraufhin wird der Speicherbereich wieder auf die vorherige Bank zurückgesetzt. Hierbei muss jedoch beachtet werden, dass Interrupts während des manuellen Umschaltens deaktiviert sein müssen und der ausführende Programmteil sich im ungemappten Codesegment `NEAR` befindet. Dies wird in *Embedded Workbench* durch das Schlüsselwort `__near_func` erreicht.

```

/* wird im codesegment NEAR der Bank 0 ausgeführt */
__near_func void getIEEEAddress(unsigned char *a)
{
    unsigned char bank;
    bank = MEMCTR; //Mapping-Register zwischenspeichern
    MEMCTR |= 0x30; //letzte Bank auf 0x8000-0xFFFF mappen
    /* Speichern der IEEE adresse in a */
    a[0] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 0);
    a[1] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 1);
    a[2] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 2);
    a[3] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 3);
    a[4] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 4);
    a[5] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 5);
    a[6] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 6);
    a[7] = *(unsigned char __code *)(IEEE_ADDRESS_ARRAY + 7);
    MEMCTR = bank; //vorherige Bank zurücksichern
}

```

Abbildung 5.12: Auslesen der IEEE Adresse

Bildung des Netzwerks

Im Kapitel 3 *Konzept* wurde der Aufbau eines IEEE 802.15.4 Netzwerks erläutert. In diesem Konzept spielt das Gateway eine große Rolle, es ist die Schnittstelle zwischen dem Netzwerk und dem Host-Computer für die Berechnung bzw. Darstellung der Position. Daher stellt das Gateway im Rahmen dieser Arbeit zwar lediglich eine Basisstation dar, jedoch sendet diese aber zusätzlich sämtlich Datenframes über die serielle Schnittstelle an den nachfolgenden verbundenen Host-Computer für Darstellung bzw. Auswertung der Position.

Das Gateway entspricht hierbei dem *PAN Coordinator* eines IEEE 802.15.4-Netzes. Welcher Knoten die Funktion des PC übernimmt wird jedoch nicht in dem Programm festkodiert, sondern beim Beginn des Netzwerkaufbaus festgelegt. Beim Start einer Basisstation wird diese zunächst durch einen Scan prüfen, ob bereits ein Koordinator mit der vorher vereinbarten *PAN ID* den Kanal belegt. Wenn das der Fall ist, wird die Basisstation sich mit dem Koordinator verbinden (Assoziieren), im anderen Fall wird der Knoten selbst ein Netzwerk mit der *PAN ID* starten und fortan als *PAN Coordinator* fungieren. Zusätzlich wird er, wie bereits erwähnt, sämtliche an ihn adressierten Datenrahmen weiterleiten.

Das Assoziieren der Evaluierungs-Plattform geschieht analog, nur dass diese bei einem negativen Scan-Vorgang nicht die Funktion eines *PAN Coordinator* übernehmen kann.

Da sämtliche Basisstationen sowie die Plattform mit mehr als einem weiteren Knoten kommunizieren können müssen, stellen sie im Sinne von IEEE 802.15.4 *Full Function Devices* (FFD) dar.

Sämtliche eben erläuterten Vorgänge werden durch die Funktionen `MAC_MlmeScanReq()`, `MAC_MlmeStartReq()` und `MAC_MlmeAssociateReq()` für das `.request`-Primitiv des TIMAC-Stacks realisiert. Analog gibt es für `.confirm` und `.indication` `MAC_MLME_ASSOCIATE_IND`, `MAC_MLME_ASSOCIATE_CNF` sowie `MAC_MLME_SCAN_CNF` und `MAC_MLME_START_CNF`. Diese Events werden als Callback-Funktionen umgesetzt. Das Sequenzdiagramm in Abbildung 5.13 zeigt den zeitlichen Verlauf des Nachrichtenaustauschs beim Assoziieren einer Basisstation, vgl. dazu auch Kapitel 2.4.1.

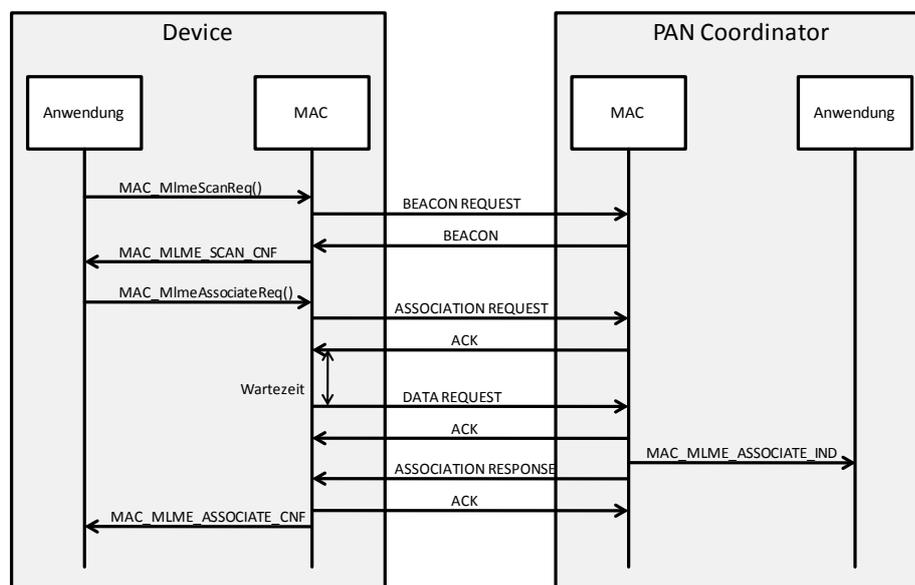


Abbildung 5.13: Sequenzdiagramm beim Assoziieren einer Basisstation oder Plattform

Senden und Empfangen von Datenrahmen

Nach dem Bilden des Netzwerks können Datenframes zwischen allen Teilnehmern verschickt werden. Dies geschieht auf der Sender-Seite durch die Funktion `MAC_McpsDataReq()`. Dafür muss zunächst durch die Methode `MAC_McpsDataAlloc()` Speicher mit der Größe des gewünschten Datenrahmens allokiert werden, zusätzlich wird die darin enthaltene Struktur `macDataReq_t` mit den notwendigen Adressparametern beschrieben. Danach wird die gewünschte Nutzlast in den reservierten Bereich kopiert, bevor abschließend die Funktion `MAC_McpsDataReq()` aufgerufen wird.

Auf der Seite des Empfängers wird beim Empfang eines Datenrahmens das Event `MAC_MCPS_DATA_IND` ausgelöst, wobei die Struktur `macDataInd_t` Parameter wie die Adresse, Signalstärke und Sequenznummer enthält. In der Struktur `sData_t` ist die eigentliche Nutzlast des Frames enthalten.

Abbildung 5.14 zeigt einen Programmausschnitt für das Senden und Empfangen von Datenframes.

```

//EMPfangEN VON DATENRAHMEN
//Systemevent
case MAC_MCPS_DATA_IND:
    pData = (macCbackEvent_t*)pMsg;
    //BSP: acht es Byte der Nutzlast
    uint8 byte7 = pData->dataInd.msdu.p[7];
    //Kurzadresse des Absenders
    uint16 short_addr = pData->dataInd.mac.srcAddr.addr.shortAddr;
    //Sequenznummer
    uint8 seq_nr = pData->dataInd.mac.dsn;

//SENDEn VON DATENRAHMEN
//Datenrahmen vorbereiten
pData = MAC_McpsDataAlloc(RADIO_RSSI_PACKET_LENGTH, MAC_SEC_LEVEL_NONE,
MAC_KEY_ID_MODE_NONE);
//Adressparameter definieren
pData->mac.srcAddrMode = SADDR_MODE_SHORT;
pData->mac.dstPanId = RADIO_PAN_ID;
pData->mac.dstAddr.addr.shortAddr = sh_addr;
//Nutzlast
Radio_Frame_Data[0] = RSSI_REQUEST;
Radio_Frame_Data[1] = actual_seq;
...
//Zusammengesetzte Nutzlast kopieren
osal_memcpy (pData->msdu.p, Radio_Frame_Data, RADIO_RSSI_PACKET_LENGTH);
// Datenrahmen in Sendepuffer
MAC_McpsDataReq(pData);

```

Abbildung 5.14: Programmausschnitt zum Senden und Empfangen von Datenrahmen

5.2.3 Rahmenformate für die Kommunikation

Für die Kommunikation in dem Netzwerk werden für die Lokalisierung bestimmte Rahmenformate festgelegt. Die Nutzlast der IEEE 802.15.4 Daten-Frames besteht aus einem Byte **frametype**, gefolgt von einer ein Byte-langen Sequenznummer und darauffolgend der eigentlichen Payload (siehe Abbildung 5.15).

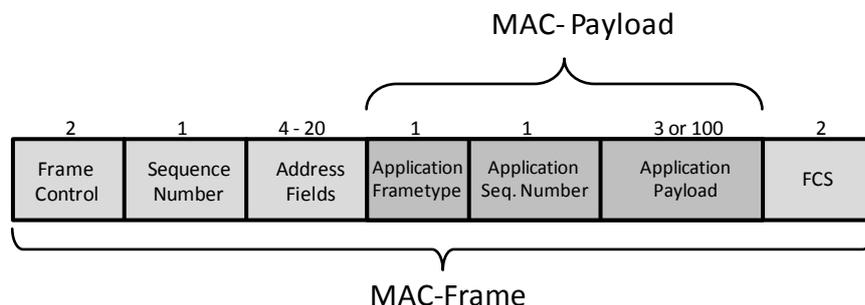


Abbildung 5.15: Rahmenstruktur der MAC-Payload

Es wurden vier verschiedene Rahmentypen für die Kommunikation festgelegt, deren Verwendungszweck in Tabelle 5.1 erläutert werden.

Frameotyp	Wert	Länge	Verwendungszweck
<i>RSSI_REQUEST</i>	<i>0x11</i>	5 Byte	Wird von Blind Nodes per Broadcast für als Aufforderung zur RSSI-Ermittlung versendet
<i>RSSI_REQUEST_RESPONSE</i>	<i>0x22</i>	5 Byte	Wird von der Basisstation zurück an den Blind Node gesendet und enthält den ermittelten RSSI-Wert.
<i>LOCATION_DATA</i>	<i>0x33</i>	55 Byte	Enthält von der Location Engine ermittelte Koordinaten und wird zur Darstellung an das Gateway gesendet
<i>GSM_DATA</i>	<i>0x44</i>	102 Byte	Enthält RSSI-Werte von 100 GSM-Basisstationen und wird zur Auswertung an das Gateway gesendet.

Tabelle 5.1: Verschiedene Rahmentypen

5.2.4 Entwicklung der Basisstation

Für die Basisstation wurden mit Hilfe der Software *EAGLE*²⁹ Schaltplan und Layout für eine kleine Platine entworfen und mit den *Radiocrfts*-Modulen RC2301AT bestückt. Nach dem Anschalten durch einen kleinen DIP-Schalter, versorgen zwei AAA-Akkus die Schaltung mit der benötigten Spannung von 2,4 V. Über eine fünf-polige Pinleiste für den Anschluss des *CC-Debugger* kann die Schaltung ebenfalls mit Strom versorgt werden. Eine weitere Pinleiste dient dem Anschluss einer seriellen Datenverbindung. Damit kann jede einzelne Basisstation auch als Gateway verwendet werden (siehe Kapitel 3). Zwei LEDs können zusätzliche Informationen über den Zustand des Knotens signalisieren. In der Abbildung 5.16 ist eine der Basisstationen dargestellt, der zugehörige Schaltplan befindet sich im Anhang Abbildung A.1.

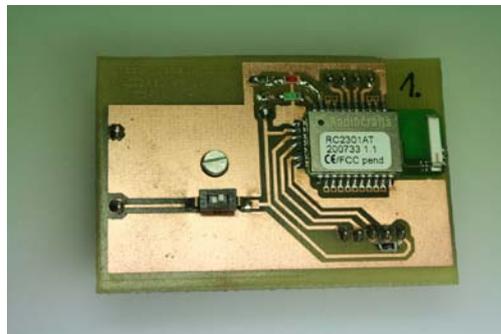


Abbildung 5.16: Basisstation

Position einer Basisstation

Die Position einer Basisstation ist als X-Y-Koordinate in den letzten beiden Byte der IEEE-Adresse kodiert. Siehe dazu Abbildung 5.17. Bei der Anmeldung einer Basisstation im Netzwerk (Assoziieren) vergibt der *PAN Coordinator* diese zwei Byte als Kurzadresse an den Knoten und verwendet Selbige auch für die eigene Kurzadresse. Somit ist die Position einer Basisstation direkt mit ihrer Kurzadresse verknüpft.

A1	B2	C3	D4	E5	F6	25	21
----	----	----	----	----	----	----	----

Abbildung 5.17: Kodierte Position in der IEEE-Adresse

²⁹ „Der EAGLE Layout Editor ist ein einfach zu benutzendes, aber dennoch leistungsfähiges Werkzeug für die Entwicklung von Leiterplatten.“ [Cad10]

Senden der Position

Entgegen der Beschreibung in Kapitel 3.2 senden Basisstationen nicht kontinuierlich ihre Positionen. Stattdessen senden die *Blind Nodes* bzw. die Plattform eine Anfrage (*RSSI_REQUEST*) per Broadcast in das Netz, daraufhin ermittelt jede Basisstation in Reichweite die Signalstärke und sendet diesen RSSI-Wert mit dem **frametype** *RSSI_REQUEST_RESPONSE* zurück an den *Blind Node* (siehe Abbildung 5.18). Mit diesen Werten kann nun die Position errechnet werden. Diese Vorgehensweise ist sinnvoll, da hiermit eine höhere Lokalisierungs-Genauigkeit erzielt werden kann, da alle Basisstationen die Signalstärke zum gleichen Zeitpunkt ermitteln.

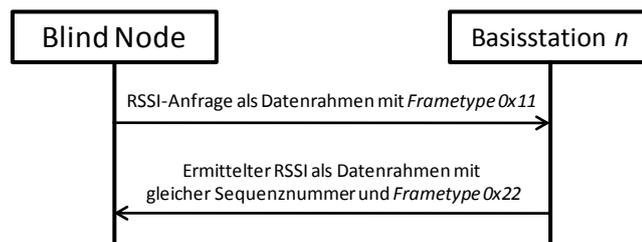


Abbildung 5.18: Ablauf der Signalstärkeermittlung

Bestimmung der Signalstärke eines Datenrahmens im CC2431

Die Signalstärke eines empfangenen Datenrahmens, beispielsweise ein *RSSI_REQUEST*-Frame, wird automatisch durch den CC2431-Chip ermittelt (vgl.[CC2430] S.163) und an die Payload der MAC-Schicht angehängen. Der *TIMAC*-Stack löst beim Erhalt eines Frames das Event **MAC_MCPS_DATA_IND** aus, wobei nun in der Callback-Funktion die Struktur **macDataInd_t** Zugriff auf die Signalstärke des empfangenen Datenrahmens und weiteren Informationen wie den Adressen, Nutzlastlänge oder Sequenznummer bietet (Abbildung 5.19).

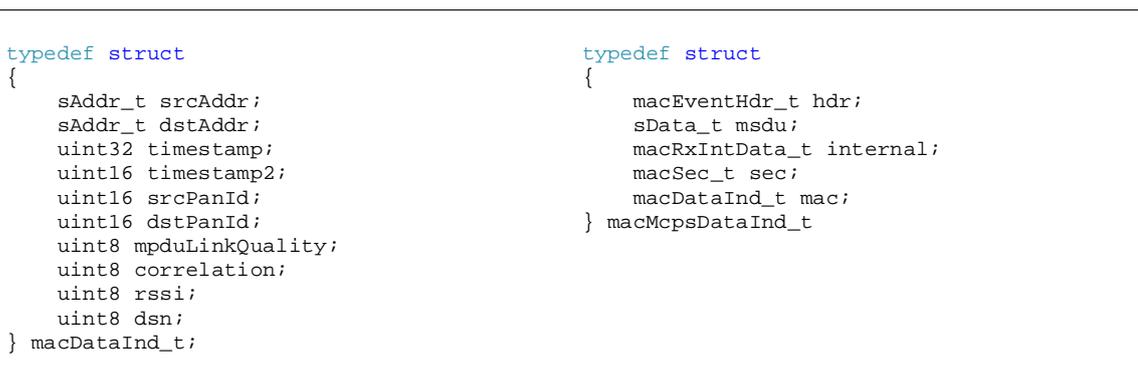


Abbildung 5.19: Datenstruktur beim Erhalt eines Datenrahmens

Übertragung von empfangenen Daten über die serielle Schnittstelle des Gateway

Datenrahmen des **frametype** *LOCATION_DATA* und *GSM_DATA* werden über die serielle Schnittstelle des Gateway an den Host-Computer übertragen. Das Gateway ist die Basisstation, welche als erstes aktiviert wurde. Zunächst muss diese vor dem Aufruf der Funktion **Ha1UARTOpen()** mit 115,2 kBits, 8 Daten- sowie 1 Stop-Bit ohne Parität parametrisiert werden.

Danach wird bei Erhalt eines durch `MAC_MCPS_DATA_IND` ausgelösten Events das Datenframe auf den Typ (`GSM_DATA` oder `LOCATION_DATA`) überprüft und die „GSM-Datenrahmen“ anhand der Sequenznummer in die richtige Reihenfolge gebracht, da diese durch die MAC-API in verschiedener Reihenfolgen empfangen werden können (siehe Kapitel 5.2.5.2).

Abschließend werden die Datenrahmen an den Host-Computer übertragen. Dazu wird zusätzlich zur leichteren Dekodierung der Datenrahmen am Host-Computer vor jedem Rahmen die Zeichenkette `FOKUS` geschrieben.

Programmablauf

Der eben beschriebene Ablauf wird im Anwendertask *Radio* (`radio.c`) realisiert. In der folgenden Abbildung 5.20 wird der Programmablaufplan dargestellt.

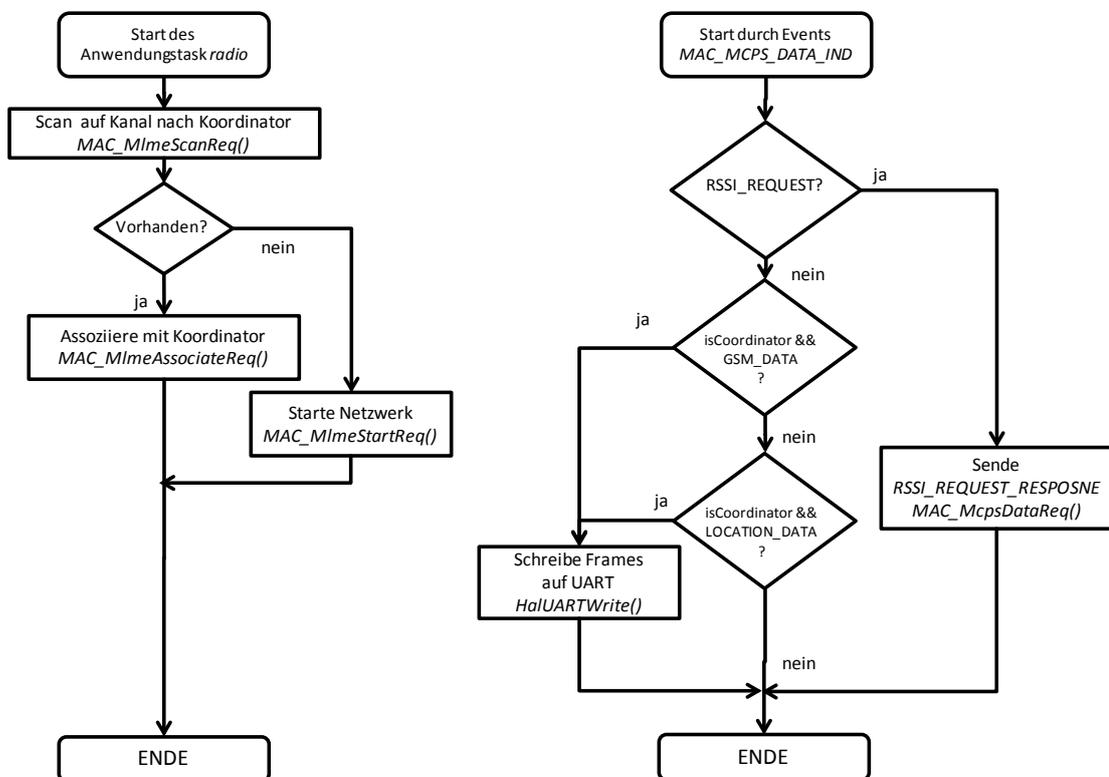


Abbildung 5.20: Programmablaufplan einer Basisstation beim Start

5.2.5 Entwicklung der Plattform

Die Entwicklung der Evaluierungsplattform ist zweigeteilt. Die Hauptkomponenten der Plattform sind, wie bereits erwähnt, das Funkmodul RC2301AT mit integriertem Mikrocontroller von *Radiocrafts* und das Dualband-Handy C123 von *Motorola*. Als Trägerplatine wurde eine Lochrasterplatine verwendet.

Das Mobiltelefon wird über den Kopfhöreranschluss mit dem UART-Port des Moduls verbunden, dies ermöglicht eine serielle Datenkommunikation. Da alle Komponenten mit einer Versorgungsspannung von 3,6 V betrieben werden können, wird die Stromversorgung mit drei

1,2 Volt Akkus in einem Batteriehalter sichergestellt. Die entwickelte Plattform ist in Abbildung 5.21 dargestellt.



Abbildung 5.21: Evaluierungsplattform

5.2.5.1 Software des Motorola C123

Die Anwendung *11test* des Open Source Projekts *OsmocomBB* ist ein Programm, welches verschiedene Parameter der Bitübertragungsschicht des GSM-Standards bestimmt. Es ermittelt zunächst die zehn stärksten Kanäle und versucht eine Frequenz-Kalibrierung auf den stärksten Kanal durchzuführen, um im Anschluss erste rudimentäre Informationen über die Zelle zu ermitteln und diese über die serielle Schnittstelle auszugeben (siehe [Red95]). Das Programm wird für die Plattform dahingehend modifiziert und vereinfacht, dass nun kontinuierlich die Signalstärkeinformationen der 100 stärksten Kanäle ausgegeben werden. Des Weiteren bietet das Programm eine simple Möglichkeit Display und Tastatur des Handys anzusteuern. Der Ansatz, die Plattform darüber zu steuern, wurde aufgrund zusätzlich benötigter Kommunikation zwischen dem C123 und dem *Radiocrafts*-Modul aus Zeitmangel verworfen, ist aber für eine mögliche Erweiterung vorgesehen.

Download der Software ins Telefon

Trotz der Tatsache, dass das Telefon über 2 MB Flash-ROM verfügt, können Programme in dem Projekt derzeit nicht im ROM abgespeichert werden. Stattdessen werden Anwendungen beim Start des C123 über einen RAM-Bootloader in den Arbeitsspeicher des Telefons geladen und ausgeführt. Der für den Download notwendige Handshake³⁰ wird in Abbildung 5.22 dargestellt.

Für die Plattform wurde anfangs das Handshake-Protokoll des originalen RAM-Bootloaders im *Radiocrafts*-Modul nachgebildet, so dass es beim Start über die serielle Schnittstelle in das Telefon geladen werden kann. Als Problem hat sich jedoch die Größe des kompilierten Programms ergeben (~38 kB), welches in den RAM des Telefons geladen werden muss. Da

³⁰ Detaillierte Informationen unter [OSMOLOAD]

die Größe des 128 kB großen Flash-Speicher des CC2431 zusammen mit dem *TIMAC*-Stack nicht ausreicht, wurde auch dieser Ansatz fallengelassen. Denkbar wäre es jedoch einen externen EEPROM dafür zu nutzen.

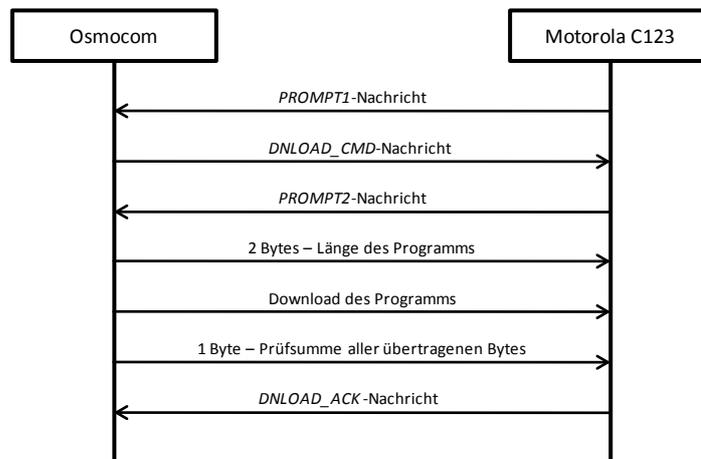


Abbildung 5.22: Handshake des RAM-Bootloaders

Nach dem Einschalten der Plattform bzw. des Mobiltelefons muss daher das hochzuladende Programm durch eine serielle Datenverbindung mit einem PC in den Arbeitsspeicher des Handys transferiert werden. Das Open-Source-Projekt stellt dazu das Kommandozeilenwerkzeug *osmocom* zur Verfügung (siehe Abbildung 5.23). Nach dem Download-Vorgang dient es normalerweise der Ausgabe des zuvor übertragenen Programms. Hier wird jedoch durch einen mechanischen Schalter die Kommunikation auf den seriellen Port (UART) des RC2301AT-Modul umgeschaltet.

```

$ osmocom -mc123xor -p /dev/ttyUSB0 target/firmware/board/compal_e88/lltest.bin
got 2 bytes from modem, data looks like: 04 81
...
Received PROMPT1 from phone, responding with CMD
read_file(lltest.bin): file_size=38024, hdr_len=4, dload_len=38031
got 1 bytes from modem, data looks like: 1b
...
got 1 bytes from modem, data looks like: 43
Received PROMPT2 from phone, starting download
handle_write(): 4096 bytes (4096/38031)
...
handle_write(): 1167 bytes (38031/38031)
handle_write(): finished
got 1 bytes from modem, data looks like: 1b
...
Received DOWNLOAD ACK from phone, your code is running now!
  
```

Abbildung 5.23: Aufruf und verkürzte Ausgabe des Tools *osmocom*

Das Funkmodul kann die Signalstärkeinformationen nun zur Auswertung an das Gateway übertragen.

Da die Prioritäten des OsmocomBB-Projekts bisher bei der Implementierung des GSM-Protokolls lagen, ist eine Anwendung, welche zur Laufzeit den Flash-Speicher beschreibt noch nicht realisiert worden. Sollte dies in nächster Zeit umgesetzt werden, so könnte der RAM-Loader durch einen Bootloader ersetzt werden, welcher das Programm *lltest* aus dem Flash-Speicher des Telefons lädt.

Modifizierung des Programms `l1test`

Das Programm `l1test` besteht aus der `main()`-Funktion, in welcher die einzelnen Komponenten (*Calypso*-Chip, Display, Layer1) initialisiert werden. Konkret wird im Hauptprogramm die Callback-Funktion `l1s_signal_cb()` deklariert. Diese wird durch den Layer1 des *OsmocomBB*-Stack aufgerufen, wenn ein Ergebnis der Signalstärkemessung oder der Frequenzkorrektur vorliegt.

Das Programm wurde dahingehend modifiziert, dass in der Callback-Funktion immer eine neue Signalstärkemessung durch `l1s_pm_test()` mit dem darauf folgenden Kanal gestartet wird, während hingegen die Frequenzkorrektur komplett deaktiviert wurde.

Die übergebene ermittelte Signalstärke in der Callback-Funktion wird zusammen mit dem Kanal in einem Array abgespeichert. Dieses Array enthält eine Struktur, welche wiederum ein Array `rxlev` und eine Indexvariable `buf_index` beinhaltet (siehe Abbildung 5.24). Das Abspeichern geschieht durch die Funktion `arfcn_rxlev_add()`, welche bei jedem Aufruf den Signalstärkewert für den entsprechenden Kanal in dem Unter-Array `rxlev` mit zyklischem Index abspeichert. Diese Implementierung stellt somit einen einfachen Ringpuffer dar. Beim Aufruf der Funktion `get_avg_rxlev()` wird der Mittelwert des Kanals durch die Werte des Ringpuffers zurückgegeben. Die Funktion `top_arfcn_add()` speichert die 100 stärksten RSSI-Werte in dem Array `top_arfcn_list` ab.

```
typedef struct
{
    uint8_t rxlev[ARFCN_RXLEV_BUF_SIZE];
    uint8_t buf_index;
} arfcn_rx;
arfcn_rx arfcn_arraylist[ARFCN_ARRAY_SIZE];
```

Abbildung 5.24: Array mit Ringpuffer für RSSI-Werte

Nach Messung aller Kanäle des in Europa verfügbaren GSM-Bandes, wird die Kanalliste der besten Signalstärken `top_arfcn_list` über die serielle Schnittstelle versendet. Vorher muss jedoch zunächst das HLDC-Protokoll³¹, welches im Projekt für die serielle Schnittstelle vorgesehen ist, abgeschaltet werden. Dies wird durch Setzen der Konstanten `CONS_UART_NR` 1 und `SERCOMM_UART_NR` 0 erreicht. Standardmäßig sind für die Schnittstelle 115,2 kBits, 8 Daten- und 1 Stop-Bit ohne Parität eingestellt.

Mit der Funktion `cons_putchar()` des *OsmocomBB*-Stacks kann nun Byte-weise über die serielle Schnittstelle geschrieben werden. Da es theoretisch insgesamt weltweit 1024 Kanäle gibt und dieser Wert im Programm als 16 Bit-Zahl definiert werden muss, wird diese vor der Übertragung in ein High- und Low-Byte zerlegt. Zusätzlich wird noch ein „Trenn-Byte“ `0xFF` vorangestellt. Das erste Trenn-Byte erhält zudem den Wert `0xFE`, somit kann der Start der

³¹ „Beim High Level Data Link Control (HDLC) handelt es sich um ein strukturiertes Protokoll, das die Mittel bestimmt, mit denen ungleiche Geräte über Datennetze miteinander kommunizieren können.“ [ITW10c]

Signalstärke-Übertragung signalisiert werden. Pro Kanal werden also 4 Byte (Trenn-Byte + 2xKanal + RSSI) gesendet. Die Signalstärkemessung aller Kanäle dauert ca. 7,5 Sekunden³², dadurch werden darauffolgend stets insgesamt 400 Bytes im gleichen Intervall übertragen (siehe Abbildung 5.25).

```

for (i=0; i<TOP_ARFCN_SIZE; i++)
{
  if(i==0) cons_putchar(0xFE);
  else cons_putchar(0xFF);
  cons_putchar(top_arfcn_list[i].arfcn >> 8);
  cons_putchar(top_arfcn_list[i].arfcn);
  cons_putchar(top_arfcn_list[i].rxlev);
}

```

Abbildung 5.25: Übertragen der Liste der signalstärksten Kanäle

Abschließend wird Liste der stärksten Kanäle durch `top_arfcn_empty()` zurückgesetzt und die Messungen beginnen wieder beim ersten Kanal.

Ein größerer Ringpuffer bewirkt für einen Ort im Raum einen konstanteren RSSI-Wert, wogegen ein zu großer Puffer beim Wechsel der Messposition zu einer sehr trägen Änderung der Signalstärke führt. In dieser Arbeit wurde ein Ringpuffer mit 5 Byte Größe eingesetzt.

Abbildung 5.26 zeigt den Programmablauf der Callback-Funktion `l1s_signal_cb()`.

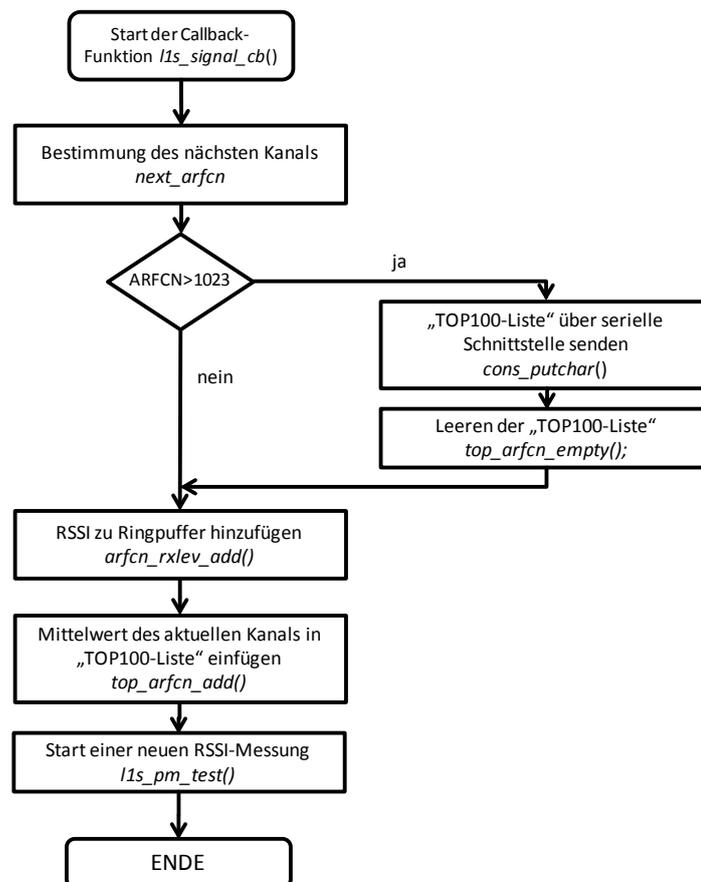


Abbildung 5.26: Programmablaufplan des Programms l1stest

³² Empirisch gemessen

5.2.5.2 Software des RC2301AT-Moduls

Im Folgenden wird die zweite Softwarekomponente der Plattform erläutert.

Location Engine

Wie bereits in der Beschreibung des *Radiocrafts*-Moduls erläutert (siehe Kapitel 5.1.3) enthält der Baustein eine *Location Engine*. Die Berechnung der Position wird durch Signalstärke-Lateration in der Hardware durchgeführt. Die Engine ist durch Register im erweiterten Datenbereich (`xdata`) des 8051-Kerns ansprechbar (`0xDF55-0xDF59`).

Vorbereitungen für die Location Engine

Bevor die Berechnung durchgeführt werden kann, müssen die RSSI-Werte von bis zu 16 Basisstationen vorliegen. Wie bereits im Kapitel 5.2.4 unter *Senden der Position* beschrieben, wird eine RSSI-Anfrage an sämtliche Basisstationen in Reichweite geschickt. Nach Erhalt der Antwort, welche die Signalstärkeinformation beinhaltet, werden diese zusammen mit der Position zu der Liste `LE_liste` (siehe Abbildung 5.27) hinzugefügt bzw. im Falle eines bereits bestehenden Eintrags nur der jeweilige RSSI-Wert aktualisiert.

```
typedef struct
{
    union
    {
        uint16    pos; //je ein byte für x und dann y
        uint8     posxy[2];
    };
    uint8 rssi;
} LocEngine_t;
LocEngine_t LE_liste[16];
```

Abbildung 5.27: Liste für Koordinaten und RSSI-Werte von Basisstationen

Des Weiteren gibt es dazu folgende Anmerkungen:

- Für die Position wurde ein Union definiert, so dass die Kurzadresse des Absenders direkt als X-Y-Koordinate verwendet werden kann, da beide denselben Speicherbereich belegen.
- Der RSSI-Wert wird als Zweierkomplement *signed* ermittelt und übertragen. Die *Location Engine* benötigt jedoch den positiven Absolutwert *unsigned*. Dieser kann durch Invertierung der Variable und Addition des Wertes 1 ermittelt werden (`rssi_usigend = ~(rssi_sigend)+1`).
- Zuletzt wird für die Evaluierung angenommen, dass sich der *Blind Node* für beide Achsen immer in dem Raum zwischen dem kleinsten und dem größten Wert des Koordinatentupels aller Basisstationen befindet (siehe Abbildung 5.28). Diese Ausdehnung stellt die räumliche Begrenzung für ein mögliches Ergebnis der *Location Engine* dar, was den Berechnungsfehler verhindern kann und die Berechnungszeit verringert. Im Detail wird jedes hinzugefügte Koordinatentupel überprüft, ob es ein Minimal- oder Maximalwert der bestehenden Koordinatenliste enthält und dann als `LE_minX`, `LE_minY`, `LE_maxX` und `LE_maxY` gespeichert.



Abbildung 5.28: Räumliche Grenzen für die Location Engine

Positionsberechnung in der Location Engine

Die im vorherigen Abschnitt ermittelten Parameter können nun verwendet werden. Nachdem die *Location Engine* aktiviert wurde, werden zunächst die bekannten Positionen der Referenzbasisstationen nacheinander in das zugehörige Register **REFCOORD** geschrieben. Dafür werden drei bis sechzehn Basisstation benötigt, wobei eine Koordinate aus zwei Mal einem Byte besteht und 0,25 Meter aufgelöst werden können. Das ergibt für die Lokalisierung einen maximalen abdeckbaren Bereich von 64 x 64 Meter, wobei beispielsweise ein Wert von 128 einer Ausdehnung von 32 Meter entlang einer Achse entspricht. Anschließend werden nacheinander Parameter, wie die einzelnen ermittelten Signalstärken, räumliche Abgrenzungskordinaten für die Positionsbestimmung sowie die Parameter *A* und *n* in das Register **MEASPARM** geschrieben. Danach wird die Berechnung durch Setzen des Bit **RUN** gestartet und ist beendet sobald das Bit **DONE** gesetzt wurde. Nun können die Koordinaten aus den Registern **LOCX** und **LOCY** gelesen werden. Der Vorgang kann beliebig oft mit neuen Messwerten bzw. neuen Koordinaten für die Basisstationen wiederholt werden. Abbildung 5.29 zeigt den Programmablauf für die *Location Engine*.

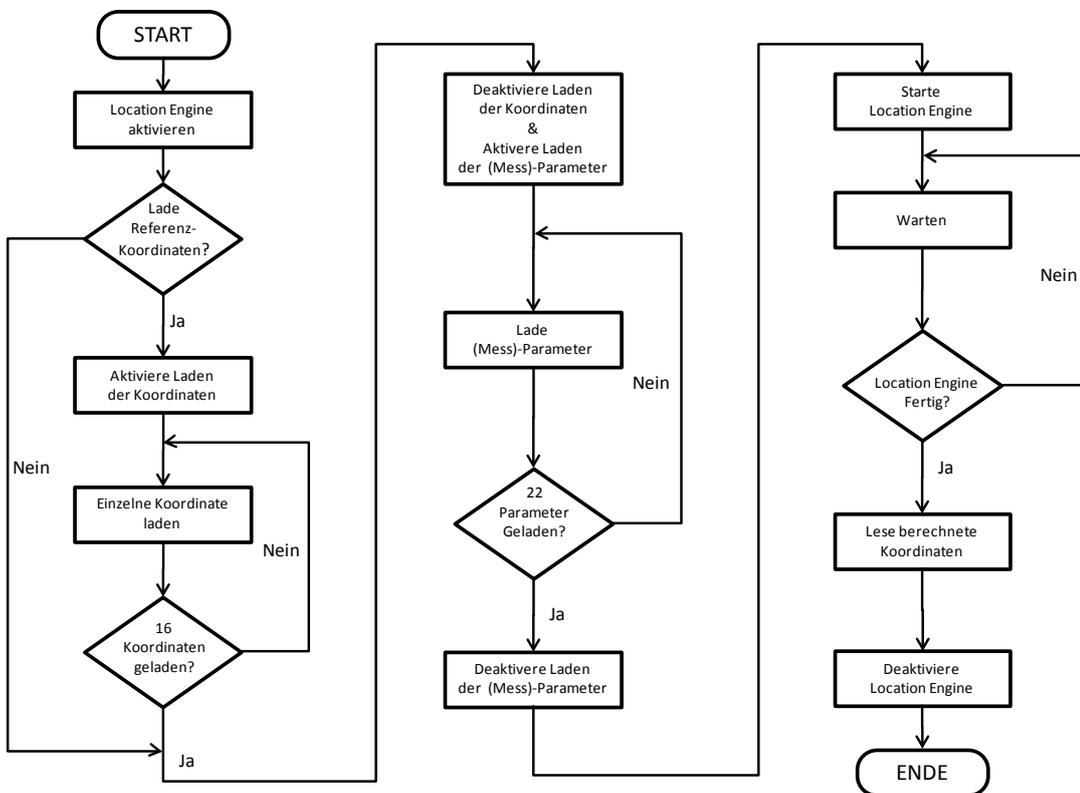


Abbildung 5.29: Programmablaufplan der Location Engine

Kalibrierung der Location Engine

Die Parameter A und n sind Variablen der Signalstärke-Funktion von Sendeleistung und der Distanz zwischen Sender und Empfänger [ANLE] .

$$RSSI = -(10 n \log_{10} d + A)$$

Dabei ist d die Distanz zwischen Sender und Empfänger, A entspricht dem RSSI-Wert in 1 Meter Entfernung und n einer Signalausbreitungskonstante. Die beiden Variablen müssen nun empirisch bestimmt werden. Dafür wird im Abstand von einem Meter der RSSI-Wert ermittelt. Da die auf dem Funkmodul integrierten Antennen nicht isotrop³³ sind, wird eine Messung durch ein weiteres sendendes Funkmodul an mindestens vier Positionen um das sendende Modul durchgeführt. Die Ausrichtung des in der Mitte platzierten Moduls bleibt dabei unverändert und der Mittelwert aus den gemessenen Werten wird für den Parameter A verwendet (siehe Abbildung 5.30). Der Parameter n stellt die Steigung der Gerade zwischen der Signalstärke und dem zehnfachen des Logarithmus der Distanz dar. Dafür werden für verschiedene Distanzen RSSI-Werte gemessen und somit die Steigung bestimmt.

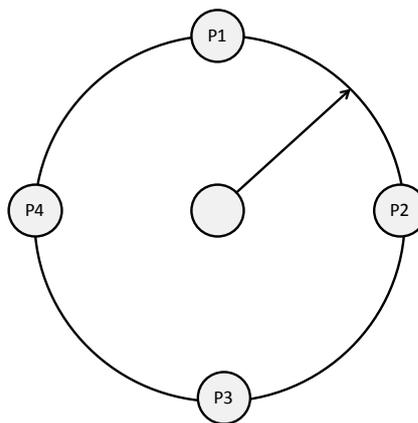


Abbildung 5.30: Einmessen des Parameter A

Versenden der GSM-Daten von der seriellen Schnittstelle an das Gateway

Über die serielle Schnittstelle des *Radiocrafts*-Moduls werden die ermittelten Signalstärkeinformationen des *Motorola C123* gelesen (vgl. Kapitel 5.2.5.1) und per Funk an das Gateway übertragen.

Entsprechend der Einstellung der seriellen Schnittstelle auf Seiten des *Motorola C123* wurde die Baudrate ebenfalls auf 115200 Bits sowie 8 Daten-Bits und 1 Stop-Bit ohne Parität festgelegt.

Die Interrupt-gesteuerte Callback-Funktion `HalUARTCback()` übergibt durch den Parameter `event` um welchen UART-Interrupt es sich handelt. Für die Implementierung wird in der

³³ „Ein Modell einer Referenzantenne, die gleichmäßig in alle Richtungen strahlt und ein optimales Energiefeld aufweist.“ [ITW10b]

Callback-Funktion geprüft, ob der Puffer zur Hälfte gefüllt ist (`HAL_UART_RX_ABOUT_FULL`) oder ob für einen festgelegten Zeitraum (5 ms) keine Daten empfangen wurden (`HAL_UART_RX_TIMEOUT`). Für die beiden Ereignisse wird dann das globale Flag `BufferFull` bzw. `Timeout` gesetzt, welche in dem Anwendungs-Task des zyklischen Ereignisses `UART_EVENT` behandelt werden. Diese Vorgehensweise ist notwendig, da aufwändigere Operationen in der Callback-Funktion zu nicht reproduzierbaren Fehlern führen.

Im Events `UART_EVENT` werden, wenn `BufferFull` gesetzt, mit Hilfe der Funktion `HALUARTread()` alle im Puffer vorhandenen Bytes gelesen und in der Variable `readbuffer` gespeichert. Dieser Vorgang wird so oft wiederholt bis die Größe eines Datenrahmens des Typs `GSM_DATA` (100 Bytes) (vgl. Kapitel 5.2.3) erreicht wird. An dieser Stelle wird nun das Frame mit dem `frametype GSM_DATA` an den `PAN Coordinator` versendet. Das Flag `Timeout` signalisiert das Ende eines Datenstroms, wenn dieses Flag gesetzt ist, wird ohne dass die Größe von 100 Bytes erreicht ist, das Datenframe versendet. Zusätzlich beginnt die Sequenznummer des „GSM-Datenrahmens“ bei jeder Übertragung bei dem Wert 0. So kann ggf. die richtige Reihenfolge der Rahmen wieder hergestellt werden (vgl. Kapitel 5.2.4).

Programmablauf

Der beschriebene Ablauf wird im Anwendertask `Radio` (`radio.c`) der Plattform realisiert. In der folgenden Abbildung 5.31 wird der Programmablaufplan vereinfacht dargestellt.

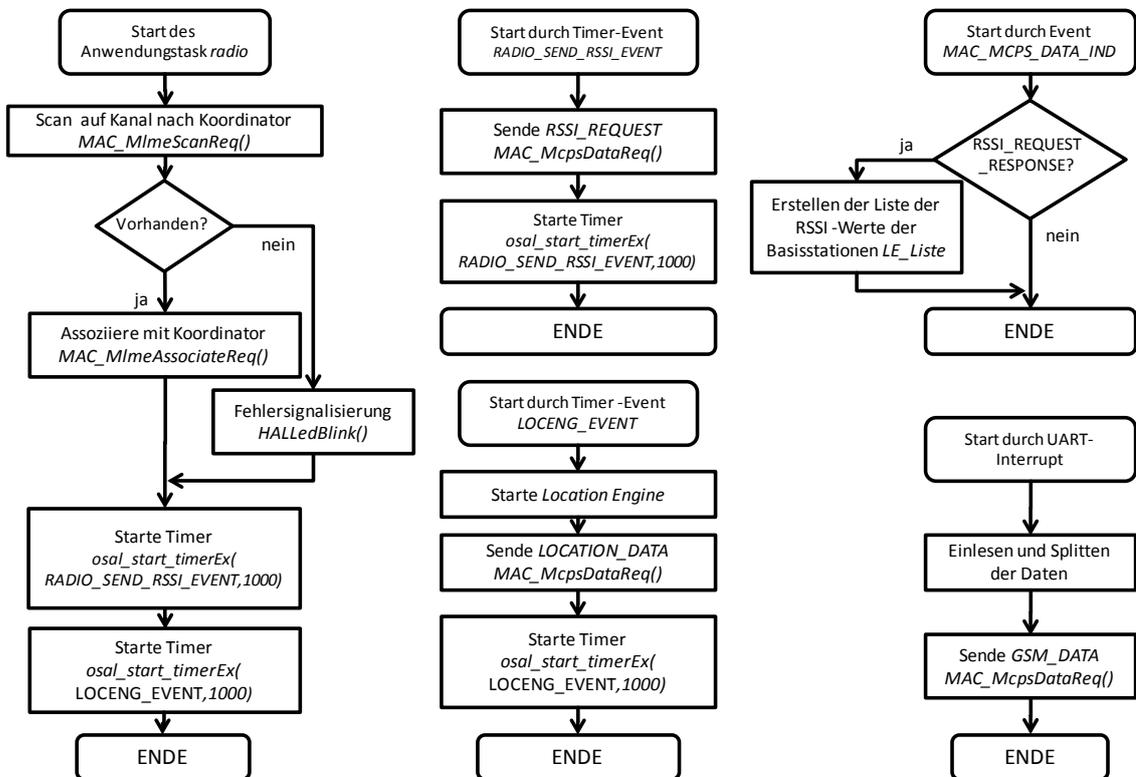


Abbildung 5.31: Programmablauf der Plattform

5.2.6 Software des Host-Computer

Die Software zur Berechnung und Darstellung auf dem Host-Computer wurde mit Hilfe der Programmiersprache C# und dem .NET-Framework 3.5 in Verbindung mit der IDE *Microsoft Visual Studio 2008* entwickelt.

Es gibt drei wichtige *Visual Studio*-Projekte:

- **GSMModule:** Das Projekt beinhaltet für verschiedene GSM-Module eine einheitliche Schnittstelle zur Ansteuerung.
- **GSMDatabase:** In dem Projekt *GSMDatabase* werden die aufgenommenen Daten gespeichert. Des Weiteren stellt es Methoden für die Verarbeitung bereit.
- **GSMLocalization:** Hiermit wird das Benutzerinterface erzeugt, und dem Anwender die ermittelte Position auf einer Karte dargestellt. Außerdem hat er hier die Möglichkeit weitere Einmeßvorgänge durchzuführen.

Der Präfix GSM resultiert daher, dass die Evaluierung mit dem GSM-Teil gestartet wurde. Die Darstellung der ermittelten Position der *Location Engine* wurde dem Projekt später hinzugefügt und die Namensgebung jedoch belassen.

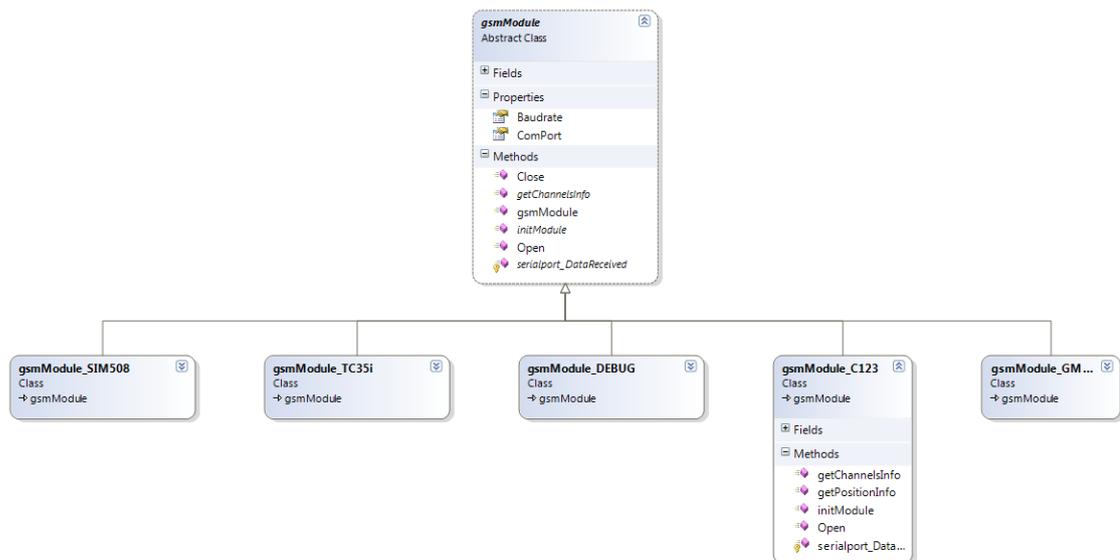
Integration verschiedener GSM-Module

Im Verlauf der Arbeit wurden verschiedene GSM-Module evaluiert, alle wurden über die serielle Schnittstelle angeschlossen, jedoch war die Ansteuerung verschieden. Die GSM-Module *Siemens TC35i*, *SimCom Sim508* und *Sony Ericsson GM29* wurden über Modem-übliche AT-Befehle angesprochen, manche mussten vorher in einen *Engineering Mode* gebracht werden. Das *Motorola C123* verfügt dagegen über eine eigene modifizierbare Firmware (vgl. Kapitel 5.2.5.1) und sendet kontinuierlich Signalstärkeinformationen.

Um alle getesteten Module einheitlich zugreifbar zu machen, wurde das Projekt *GSMModule* erstellt. Es enthält die abstrakte Klasse `gsmModule` als Schnittstelle, von welcher für jedes GSM-Modul eine Klasse angelegt wurde, die von dieser abstrakten Klasse erbt. Das führt dazu, dass die geerbten Methodenaufrufe einheitlich sind und selbst jedoch in der erbenden Klasse implementiert werden müssen.

Die geerbte Methode `getChannelsInfo()` muss implementiert werden und gibt eine Liste mit den GSM-Kanälen zurück.

Für das *Motorola C123* wurde zunächst in diesem Projekt eine neue Klasse mit dem Namen `gsmModule_C123` erstellt. Diese wurde um die Methode `getPositionInfo()` erweitert, da im Rahmen der Arbeit über die serielle Schnittstelle zusätzlich Positionsdaten der *Location Engine* übermittelt werden. In Abbildung 5.32 wird dazu ein Klassendiagramm dargestellt.

Abbildung 5.32: Klassendiagramm für *gsmModule*

Beim Instanzieren dieser Klasse wird eine leere Liste `gsmChannelList` für die Kanäle erzeugt: Die generische Liste vom Typ `Dictionary<Channel, int>` enthält Wertepaare, welche in diesem Fall den Kanal und die dazugehörige Signalstärke repräsentieren. Ebenso gibt es in der Instanz für die übertragene Position die Variable `Point location`.

Bei dem anschließendem Aufruf der Methode `Open()` wird die serielle Verbindung aufgebaut. Die Methode `serialport_DataReceived()` wird beim Erhalt von Bytes über die serielle Schnittstelle Event-basiert aufgerufen. Daraufhin wird der „Byte-Strom“ von der seriellen Schnittstelle dekodiert. Dieser Vorgang wird im Folgenden kurz verbal beschrieben:

- Sämtliche Bytes werden durch die Methode `Read()` in eine generische Liste vom Typ `List<byte>` geschrieben
- Es wird das erste Vorkommen der „Zeichenkette“ `FOKUS` gesucht (vgl. Kapitel 5.2.4).
- Darauf folgt das Byte, welches den Rahmentyp darstellt (`GSM_DATA` bzw. `LOCATION_DATA`).
- Für ein Frame, welches durch die `Location Engine` berechnete Koordinaten enthält, werden diese Koordinaten aus den folgenden Bytes in der Variable `location` gespeichert.
- Bei einem Frame, das hingegen Signalstärkeinformationen beinhaltet, wird für jeden Kanal eine Instanz der Klasse `Channel` (siehe folgenden Abschnitt *GSMDatabase*) erzeugt. Darin wird die aus den beiden Bytes bestehende Kanalnummer wieder zusammengesetzt gespeichert und mit dem RSSI-Wert in der Liste `gsmChannelList` gespeichert.

Beim Aufruf der Methoden `getChannelsInfo()` und `getPositionInfo()` werden damit aktuelle Positionsdaten und Signalstärkewerte zurückgegeben. Es muss sichergestellt werden, dass beim Einmessen die erforderliche Dauer von ~ 7,5 Sekunden verstreicht, sonst könnten

für verschiedene Positionen die gleichen RSSI-Werte verwendet werden. Später könnte hier zur Überprüfung ein Zeitstempel der letzten Datenaufnahme genutzt werden.

Bei Testen ist aufgefallen, dass einige Signalstärke-Werte fehlerhaft übertragen wurden (siehe Kapitel 5.3.6). Da das Problem in der Schnelle nicht eingegrenzt werden konnte, musste für die weitere Evaluierung des *Radiomap*-Verfahrens das *Motorola*-Telefon direkt an den Host-Computer angeschlossen werden. Dafür wurde die Klasse wegen der fehlenden berechneten Positionsdaten der `gsmModule_C123` modifiziert und als `gsmModule_C123_direkt` gespeichert.

GSMDatabase

Die Klasse `gsmDatabase` ist ein Wrapper für die dahinterliegende Datenbank. Sie stellt verschiedene Methoden bereit, um beispielsweise die Signalstärke-Daten zu einer Position zuzuordnen. Dies geschieht mit Hilfe der Methode `addLocation(Channel ch, float x, float x, int rssi)`. `Channel` und `Location` sind Klassen, welche durch das *Entity Framework* (EF)³⁴ automatisch im Code generiert werden.

Wichtig sind weiterhin die Methoden `computeEuclideanDistance()` und `computeKNN()`. Erstere berechnet mit Hilfe einer übergebenen Liste von Kanälen und den zugehörigen Signalstärke-Werten die Position mit der kleinsten euklidischen Distanz von RSSI-Werten aus der Datenbank. Im Detail wird darauf eine Liste mit allen möglichen Positionen sortiert nach der euklidischen Distanz zurückgegeben.

Die zweite Methode kann verwendet werden um die Genauigkeit des Lokalisierungs-Ergebnisses weiter zu erhöhen (vgl. Kapitel 3.2). Aus der übergebenen Liste der Positionen und den zugehörigen euklidischen Distanzen wird der geometrische Mittelpunkt berechnet und zurückgegeben. Durch den Parameter *k* kann die Zahl der einbezogenen „nächsten Nachbarn“ angepasst werden. Abbildung 5.33 zeigt graphisch die verfügbaren Methoden der Klasse.

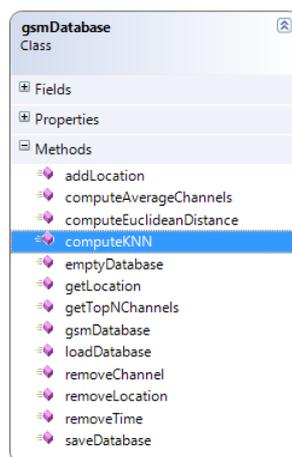


Abbildung 5.33: Klasse `gsmDatabase`

³⁴ Das EF bildet bestehende Entitäten einer Datenbank als Objekte im Programm ab - weitere Informationen dazu unter [Mic10].

GSMLocalization

Das Projekt enthält die eigentliche graphische Benutzerschnittstelle. In der Klasse **Mainform** werden die Klassen **gsmModule** und **gsmDatabase** eingebunden und instanziiert. Der Programmcode liegt auf der CD-ROM bei.

Das Hauptformular ist unterteilt in einen Zeichenbereich auf der linken Seite, in welchem die einzumessene Karte abgebildet ist und Informationen zu den Messwerten auf der rechten Seite. Die Schaltflächen *Add* und *Estimate* dienen dem Position einmessen und Position ermitteln. Die Quadratmeter-großen Gitterflächen, welche bereits eingemessen wurden, werden blau dargestellt und die schließlich ermittelte Position grün.

Die Koordinaten, die durch die *Location Engine* auf der Plattform ermittelt wurden, werden als Text auf Zeichenfläche ausgegeben. Die Darstellung der Position direkt auf der Karte könnte später umgesetzt werden. Aufgrund der komplizierten Skalierung und den verschiedenen Koordinatenursprüngen konnte dies zeitlich nicht realisiert werden.

Abbildung 5.34 stellt die graphische Benutzeroberfläche dar.

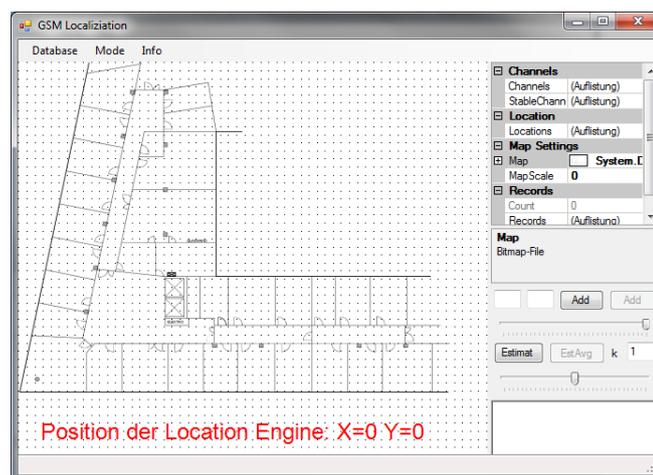


Abbildung 5.34: Graphische Benutzeroberfläche

5.3 Aufgetretene Probleme

5.3.1 SDCC Compiler

Beim Versuch die *Location Engine* mit einem Programm anzusteuern, welches mittels *Small Device C Compiler* (SDCC) [SDCC] übersetzt wurde, kam es zu unerklärlichen reproduzierbaren Fehlern mit der Engine. Nach Beschreiben des für die Berechnung unabdingbaren Schieberegister **MEASPARM** hat sich die Engine selbst deaktiviert. Da dieser Vorgang bei dem Compiler in *Embedded Workbench* nicht auftritt, wurde auch die Entscheidung erleichtert, fortan den bereits für *Embedded Workbench* vorkompilierten Stack **TIMAC** (siehe Kapitel 5.1.5) zu verwenden.

5.3.2 Kompilierung des TIMAC-Stacks mit IAR Embedded Workbench

Bei der Übersetzung der Beispielprogramme muss beachtet werden, dass diese für die Development-Boards von Texas Instruments entworfen wurden. So enthalten diese Boards für den CC2431-Chip einen zusätzlichen externen 32 kHz Taktgeber, welcher in den *Radiocrafts*-Modulen nicht enthalten ist. In den Binärbibliotheken des *TIMAC* wird das Bauteil jedoch für die Verwendung des externen Taktes konfiguriert, was zur Folge hat, dass der Protokollstapel *TIMAC* nicht startet. Nach langer Internetrecherche wurde das Compiler-Flag `OSC32K_CRYSTAL_INSTALLED` gefunden, welches `FALSE` gesetzt werden muss und damit unterbindet, dass der externe Takt verwendet wird.

5.3.3 CC-Debugger mit IAR Embedded Workbench

Ebenfalls undokumentiert ist das nicht reproduzierbare Verhalten des Debuggers in *Embedded Workbench*, wobei der Debugger nicht sinnvoll genutzt werden kann. Abhilfe brachte das Aktivieren der Einstellung „Reduced Interface Speed“ des Plugin-Treibers in der Software *Embedded Workbench*.

Des Weiteren resultierte das Debugging von Passagen, welche Interrupts enthalten, ebenfalls oftmals in nicht reproduzierbarem Verhalten.

5.3.4 Speichern der IEEE-Adresse aus der eigenen Anwendung heraus

Es war zunächst geplant die Position (IEEE Adresse) der Basisstation aus der GUI direkt in das Gerät zu übertragen. Das vorliegende Konsolenprogramm `SmartRFProgConsole.exe` der Software *SmartRF Flash Programmer* verweigerte jedoch das Schreiben der Adresse mit der Meldung `No actions have been specified`. Somit ist der Aufruf aus der eigenen Software nicht möglich.

5.3.5 Download des Programms in das Mobiltelefon schlägt fehl

Gelegentlich kommt es bei dem Download der Software *11test* in das *Motorola C123* zu der Fehlermeldung `Received FTMT00L from phone, ramloader has aborted`. Die Ursache dafür ist nicht bekannt. Der Grund dafür könnte ein sporadisch gestörtes serielles Kabel oder Timing-Probleme der seriellen Schnittstelle in der virtuellen Maschine sein.

5.3.6 Fehlerhafte Übertragungen der RSSI-Werte über IEEE 802.15.4

Bei der Dekodierung der sortierten RSSI-Werte in der Klasse `gsmModule_C123` sind in regelmäßigen, jedoch nicht periodischen, Abständen Kanalnummern (ARFCNs) aufgetreten, welche offensichtlich falsch sind. Die Werte erscheinen sporadisch unter den besten Kanälen, andere ARFCN-Werte liegen jedoch sogar über 1023, dem theoretischen Maximum von GSM-Kanälen. Mit Hilfe der Software *HTerm* und *Wireless Protocol Analyser* konnte dieser Fehler über die serielle bis hin zur Funkschnittstelle verfolgt werden.

Ein Problem in der Software der Basisstation liegt nahe. Es konnte jedoch nach langer Analyse des Programm-Codes kein Fehler festgestellt werden. Wie in Kapitel 5.3.3 beschrieben, ist der Debugger bei manchen Code-Passagen bei Verwendung von Interrupts ohne Funktion, was die Problemfindung enorm erschwert. Aus Zeitgründen musste die Fehlersuche abgebrochen werden.

Bei Überprüfung der Werte direkt an der seriellen Schnittstelle des C123 durch *HTerm* scheinen die Werte jedoch, abgesehen von einigen Ausreißern, korrekt zu sein. Da die falschen RSSI-Werte bei der Evaluierung zu unsinnigen Ergebnissen führen würden, wurde das Mobiltelefon direkt an dem Host-Computer angeschlossen und die *Radiomap*-Lokalisierung durchgeführt.

Dass die Übertragung der RSSI-Datenrahmen an sich jedoch funktioniert wird unter *Funktionsnachweis* im Kapitel 6.1.4 gezeigt.

6 Ergebnis der Evaluierung

6.1 Funktionsnachweis

Aus Zeitgründen konnte die Evaluierung nicht in einer typischen Einkaufsumgebung durchgeführt werden. Daher wurde diese in den Räumlichkeiten der Fraunhofer FOKUS durchgeführt.

6.1.1 Vorbereitungen

Für die Lokalisierung wurde das Foyer der vierten Etage gewählt (Abbildung 6.1). Bei dem vorliegenden Grundriss wurde festgestellt, dass dieser nicht bemaßt ist und zudem die Proportionen sämtlicher Längen mit der Realität nicht genau übereinstimmen – teilweise gab es Abweichungen von bis zu zwei Metern. Aus diesem Grunde wurde das Foyer auf 25 Zentimeter genau vermessen. Die wichtigsten Maße und einige der Informationen aus den folgenden Unterkapiteln wurden im Grundriss eingezeichnet (Abbildung 6.1).

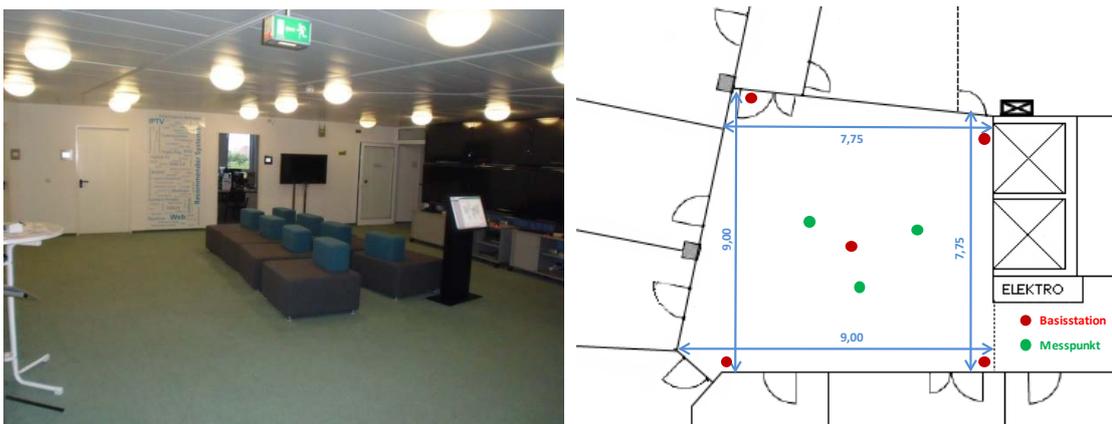


Abbildung 6.1: Foyer und Grundriss des Foyers

6.1.2 Kalibrierung der Location Engine

Die Genauigkeit der Lokalisierung durch IEEE 802.15.4 ist stark von der Umgebung abhängig, daher wird die Kalibrierung im Foyer, dem letztlichen Einsatzort, durchgeführt.

Zunächst wird dazu der Parameter A bestimmt. Die Plattform wird im Raum platziert und im Abstand von einem Meter um die Plattform wird eine Basisstation bewegt und Messungen der Signalstärke durchgeführt (vgl. Kapitel 5.2.5.2 Kalibrierung der Location Engine). Dabei wird die Basisstation stets mit der gleichen Seite der integrierten Antenne in Richtung Plattform ausgerichtet. Die RSSI-Werte können aus den `RSSI_REQUEST_RESPONSE`-Datenrahmen mit Hilfe des *Wireless Protocol Analyser* gelesen werden.

Tabelle 6.1 zeigt die verschiedenen Signalstärke-Werte. Für den Parameter ist daher ein Durchschnittswert von $A=-55$ ermittelt worden.

Winkel [Grad]	0	45	90	135	180	225	270	315	Durchschnitt
RSSI [dBm]	-51	-53	-55	-57	-63	-54	-58	-49	-55

Tabelle 6.1: Messungen der Signalstärke aus verschiedenen Winkeln in 1 Meter Abstand

Im Folgenden wird im Foyer mit der Plattform und Basisstation der Parameter n bestimmt. Hierfür verwendet man die Diagonale als längste Strecke im Raum, wobei jeweils pro Meter vier Messungen der Signalstärke auf der ca. 11 Meter langen Strecke durchgeführt werden. Anschließend wird mit Hilfe des Tabellenkalkulationsprogramms *Microsoft Excel* die Regressionsgerade über sämtliche Werte erzeugt und somit die Steigung, welche den Parameter n für diese Umgebung darstellt, ermittelt (siehe Abbildung 6.2). Die ermittelten Werte $A=55$ und $n=0,6819$ liegen außerhalb des Bereichs, welcher im Datenblatt des CC2431 angegeben wurde ($A = 30 - 50$, $n = 1 - 8$). Daher wurde für die weitere Evaluierung der *Location Engine* der Parameter $A=50$ und $n=1$ verwendet und mit Hilfe des *SmartRF Flash Programmer* in die letzten Byte der IEEE-Adresse geschrieben. Der Parameter n wird nicht als Gleitkommazahl übergeben, sondern als Index-Wert einer 32 Werte umfassenden Liste (siehe [CC2431]). Des Weiteren wurde auch der Hinweis in der technischen Dokumentation [ANLE] verfolgt, dass Werte zwischen $n=3,25$ und $n=4,5$ in der Regel gute Resultate erzielen würden.

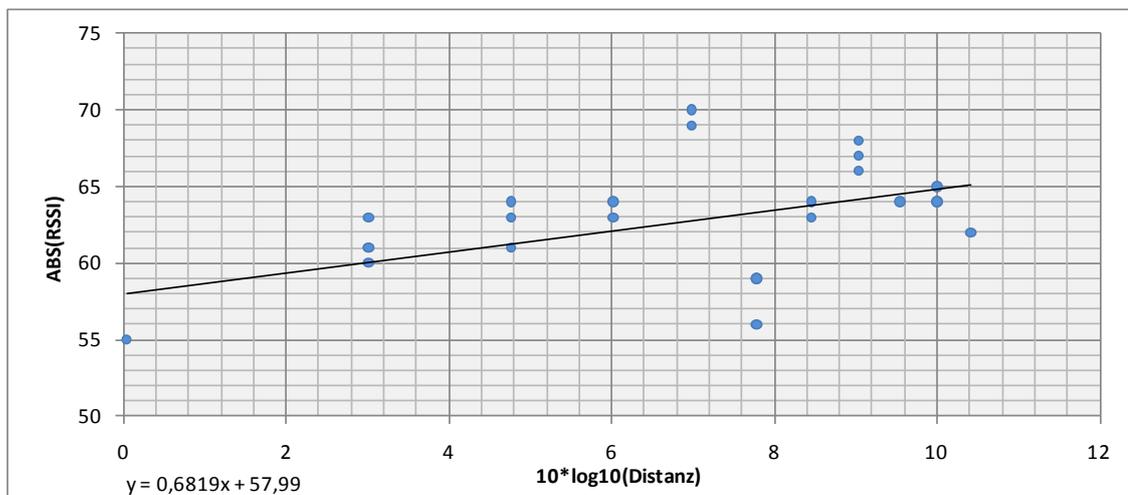


Abbildung 6.2: Bestimmung des Parameter n durch die Regressionsgerade

6.1.3 Installation der Basisstationen und des Host-Computer

Für die Evaluierung stehen insgesamt fünf Basisstationen zur Verfügung. Zunächst muss sichergestellt werden, dass auf allen Basisstationen das gleiche Programm gespeichert wird.

Daraufhin werden im Foyer Positionen ausgewählt, welche in diesem Szenario fast im gesamten Raum eine Sichtverbindung ermöglichen (siehe Tabelle 6.2). Dafür wurden 4 Basisstationen in den Ecken sowie eine Weitere in der Mitte des Raumes mit Klebeband an der

Decke fixiert. Die Basisstationen in den Ecken wurden so angebracht, dass die Antenne in Richtung Raum gerichtet ist. Dies führt damit bei gleicher Distanz zu vergleichbaren Ausbreitungscharakteristiken und damit zu ähnlicheren RSSI-Werten der Basisstationen.

BS Nr.	X[m]	Y[m]	X[HEX]	Y[HEX]
1	8	0,25	0x20	0x01
2	8	7,75	0x20	0x1F
3	1	9	0x04	0x24
4	1	0,25	0x04	0x01
5	3,25	3,75	0x0D	0x0F

Tabelle 6.2: Positionen der Basisstation für das Foyer

Vorher wurden jedoch die Koordinaten in den Basisstationen gespeichert. Da, wie bereits in Kapitel 5.2.6 beschrieben, die Position bzw. die IEEE-Adresse nicht direkt durch die GUI-Anwendung in das Gerät geschrieben werden kann, wird hierfür die Software *SmartRF Flash Programmer* eingesetzt. Es muss jedoch beachtet werden, dass die Koordinate aus zwei 1 Byte-Werten besteht und in hexadezimaler Form in der Software gespeichert werden muss. Des Weiteren muss berücksichtigt werden, dass eine Differenz des Bytes-Wertes von 1, in der Realität einer Strecke von 25 cm entspricht. Am Beispiel der Basisstation 1 (siehe Tabelle 6.2) werden für die Position die Werte 0x20 und 0x01 in den letzten beiden Bytes, also der IEEE-Adresse (vgl. 5.2.2), der Basisstation gespeichert.

Dann werden zunächst die Gateway-Basisstation über die serielle Schnittstelle angeschlossen und gestartet, im Anschluss werden sämtliche Basisstationen und letztendlich die Plattform aktiviert.

6.1.4 Kommunikation über das IEEE 802.15.4-Netzwerk

Die Abbildung 6.3 zeigt einen Ausschnitt der Software *Wireless Protocol Analyser*, welcher einen Teil der mitgeschnittenen Datenrahmen darstellt.

Zunächst wird in (1) die Assoziierung der Plattform am *PAN Coordinator* dargestellt. Die *PAN ID* ist 0x11CC. Darauf folgt der Datenrahmen (2) *RSSI_REQUEST* (0x11) als Broadcast, was der Kurzadresse (0xFFFF) entspricht. Im Anschluss (3) senden sämtliche aktiven Basisstationen eine *RSSI_REQUEST_RESPONSE* (0x22) an die Plattform. Die berechnete Position wird in dem Rahmen des Typs *LOCATION_DATA* an den *PAN Coordinator* gesendet (4). Schließlich sendet die Plattform vier aufeinanderfolgende Datenrahmen vom Typ *GSM_DATA* mit den Signalstärke-Werten (5).

Das gesamte selektierte IEEE 802.15.4-Frame wird im unteren Fenster angezeigt (6). Hier ist die Payload fett dargestellt. Das erste Byte darin enthält den Wert 0x44 für den Rahmentyp *GSM_DATA* darauf folgt die Sequenznummer 0x00 und im Anschluss daran wiederholend jeweils 4 Byte für den Kanal beginnend mit 0xFF bzw. 0xFE für das erste „Trenn-Byte“, gefolgt von dem High- und Low-Byte für den Kanal und schließlich dem Byte-langen RSSI-Wert (vgl. 5.2.5.1).

Wireless Protocol Analyzer 3.4

File View Tools Statistics Settings Help

Decode Type Medium Access Ct View No Filters Channel 11

Summary View of Received Packets

No	Timestamp	Dest PAN	Dest Addr	Src PAN	Src Addr	Detail	Additional Info
1	077.501.535,20	0xFFFF	0xFFFF			Command	BeaconRequest
2	077.503.835,20			0x11CC	0x0120	Beacon	
3	078.012.335,20	0x11CC	0x0120	0xFFFF	0x201FFFFFFF...	Command	AssociationRequest
4	078.013.035,20			1		Acknowledgement	Seq: 0x14
5	078.505.635,20	0x11CC	0x0120		0x201FFFFFFF...	Command	DataRequest
6	078.506.235,20					Acknowledgement	Seq: 0x15
7	078.508.735,20	0x11CC	0x201FFFFFFF...		0x2001FFFFFF...	Command	AssociationRespo...
8	078.509.835,20					Acknowledgement	Seq: 0x23
9	079.509.635,20	0x11CC	0xFFFF	2	0x1F20	Data	Seq: 0x16
10	079.510.335,20					Acknowledgement	Seq: 0x16
11	079.512.235,20	0x11CC	0x1F20		0x0104	Data	Seq: 0x6C
12	079.512.935,20			3		Acknowledgement	Seq: 0x6C
13	079.514.935,20	0x11CC	0x1F20		0x0120	Data	Seq: 0x24
14	079.515.635,20					Acknowledgement	Seq: 0x24
15	080.009.435,20	0x11CC	0x0120	4	0x1F20	Data	Seq: 0x17
16	080.010.135,20					Acknowledgement	Seq: 0x17
17	080.197.235,20	0x11CC	0x0120		0x1F20	Data	Seq: 0x18
18	080.199.835,20			5		Acknowledgement	Seq: 0x18
19	080.210.435,20	0x11CC	0x0120		0x1F20	Data	Seq: 0x19

Actual Packet Data

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Data
0x0000	00	41	9A	5F	73	42	61	88	18	CC	11	20	01	20	1F	44	.A?sBa??.D
0x0010	00	FE	00	11	3F	FF	00	22	32	FF	00	37	2B	FF	00	5A	..??."??+?.Z
0x0020	24	FF	03	2D	21	FF	03	DC	1C	FF	02	A7	1A	FF	03	E6	\$?-??.??.?
0x0030	19	FF	03	FD	19	FF	03	2C	16	FF	03	2E	16	FF	00	59	..??.?..?Y

Abbildung 6.3: Mitschnitt des IEEE 802.15.4-Protokolls im Wireless Protocol Analyser

6.1.5 Erstellen der GSM-Radiomap

Nach der Installation der Basisstationen und Aktivierung des IEEE 802.15.4-Netzwerkes kann nun die *Radiomap* erstellt werden. Dazu wird in der Software die zu messende Position durch einen Klick auf die Zeichenfläche ausgewählt (Gelbe Gitterfläche) und die Plattform in diesem Quadratmeter großem Teilstück im Raum platziert. Durch betätigen der Schaltfläche *Add* wird diese Position eingemessen. Dieser Vorgang wird so oft wiederholt bis jeder Quadratmeter des Foyers eingemessen (Blaue Gitterfläche) ist.

6.1.6 Darstellen der Position in der Software

Schließlich kann durch Betätigen der Schaltfläche *Estimate* die Position bestimmt werden und das Programm stellt die berechnete Position grün dar. Des Weiteren wird die durch die

Location Engine berechnete Position aktualisiert und auf der rechten Seite kann zusätzlich der Wert k für die „Nächste Nachbarn“-Klassifikation eingestellt werden (Abbildung 6.4).

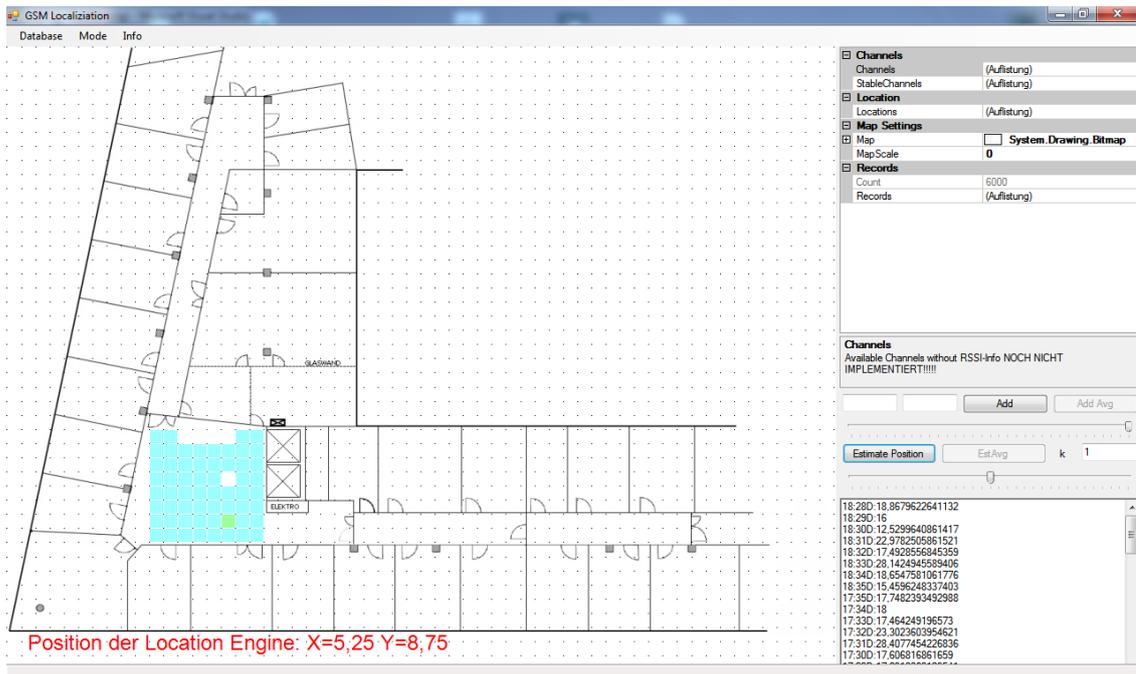


Abbildung 6.4: Darstellung der ermittelten Position durch IEEE 802.15.4 und GSM

6.2 Genauigkeit der Lokalisierung

Die Genauigkeit gibt an, wie weit das gemessene Ergebnis von dem physikalisch exakten Ergebnis abweicht. Da die „exakten“ Messpositionen durch direktes Messen³⁵ selbst nur mit einer Genauigkeit von ± 25 cm ermittelt werden konnten, sind die Fehler der folgenden Lokalisierungsergebnisse zur jeweiligen physikalisch exakten Position womöglich größer als in den folgenden Zahlen dargestellt.

Die folgenden Fehlerbetrachtungen wurden unter Zuhilfenahme von [Sch07] durchgeführt.

Genauigkeit der RSSI-Lateration durch die Location Engine

Für die Ermittlung der Genauigkeit des Lokalisierungsverfahrens wurden drei Messpunkte im Foyer bestimmt (Tabelle 6.3).

Position	X[m]	Y[m]
M1	6,75	4,25
M2	2,5	4,75
M3	4	2,75

Tabelle 6.3: Messpunkte für die Bestimmung der Genauigkeit der Location Engine

³⁵ Gemessen wurde mit einem 2-Meter-Zollstock

Zum Überprüfen der Genauigkeit der berechneten Position wurden mehrere Messreihen im Foyer durchgeführt. Neben dem „kalibrierten“ Signalausbreitungskoeffizient $n=1$ ($n=0,6819$) wurden weitere Werte verwendet. Dabei wurde versucht, dass die Plattform stets in die gleiche Richtung ausgerichtet ist. Ein Einfluss der Ausrichtung auf die Positionsbestimmung konnte festgestellt werden, aber aufgrund Zeitmangels nicht weiter evaluiert werden.

Für jeden n -Parameter wurden an den drei Messpunkten fünf Messungen durchgeführt und die übertragene Position in der GUI zur Auswertung in *Microsoft Excel* übernommen, um damit den Fehler der Positionsbestimmung zu ermitteln. Die fünf Messwerte pro Position wurden gemittelt, und der Fehler in X- und Y-Richtung zu der wahren Position berechnet. Wiederum wurde aus den drei Mittelwerten der Messpunkte ein Gesamtmittelwert für den jeweiligen Signalausbreitungskoeffizient gebildet. Abbildung 6.5 zeigt den Gesamtfehler in X- und Y-Richtung und die Standardabweichung der einzelnen Messwerte einer Koordinatenachse für verschiedene n .

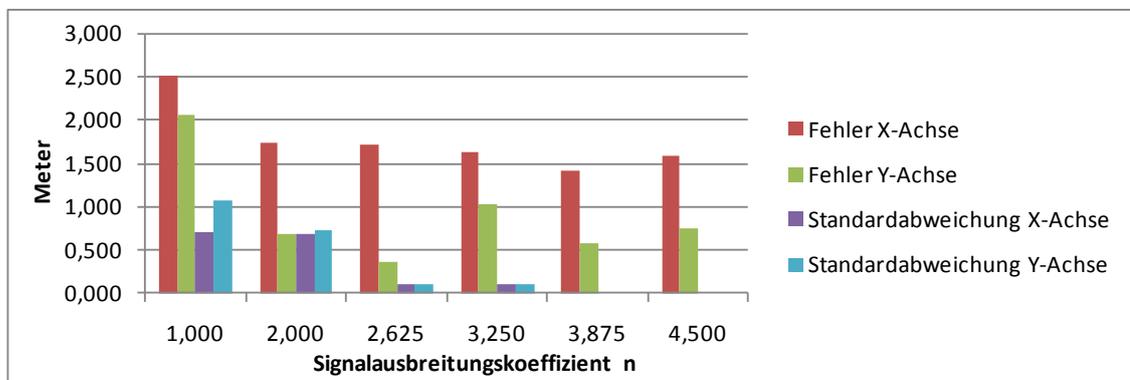


Abbildung 6.5: Lokalisierungsfehler der Location Engine

Wie bereits vorher erwähnt (vgl. Kapitel 6.1.2), werden die besten Ergebnisse bei $n=3,875$ erzielt. Der Fehler liegt hier bei $x=1,41$ m und $y=0,58$ m. Die Werte der Standardabweichungen zeigen zudem, dass mit steigendem Koeffizienten die in den Messreihen ermittelten Koordinaten weniger stark schwanken bzw. konstant sind. Auffällig ist zudem der stets größere Fehler in der X-Richtung. Es könnte dabei ein Zusammenhang zwischen der Ausrichtung der nicht-isotropen Antennen bestehen. Dies müsste später weiter geprüft werden.

Die Messreihen liegen auf der beigefügten CD-ROM als *Microsoft Excel*-Datei vor.

Genauigkeit des GSM-Radiomap-Verfahrens

Für das *Radiomap*-Verfahren wurden ebenfalls drei Messpunkte genutzt, jedoch wird hier systembedingt durch die Auflösung von 1 Quadratmeter der Nachkommaanteil verworfen (Tabelle 6.4).

Zunächst wurde der Raum mit Hilfe der Software und dem *Motorola C123* sorgfältig eingemessen. Bei der Ausrichtung des C123 wurde darauf geachtet, dass es bei den Messungen im Raum stets in die gleiche Richtung zeigte, um Einflüsse einer nicht-isotropen Antenne zu minimieren.

Position	X[m]	Y[m]
M1	7	5
M2	5	3
M3	1	5

Tabelle 6.4: Messpunkte für die Bestimmung der Genauigkeit des Radiomap-Verfahrens

Die Messreihe bestand aus 5 Messung pro Messpunkt für $k=1$, $k=2$, $k=3$ und $k=4$. Analog zur vorherigen Messreihe wird die Genauigkeit in X- und Y-Richtung für verschiedene k in Abbildung 6.6 dargestellt.

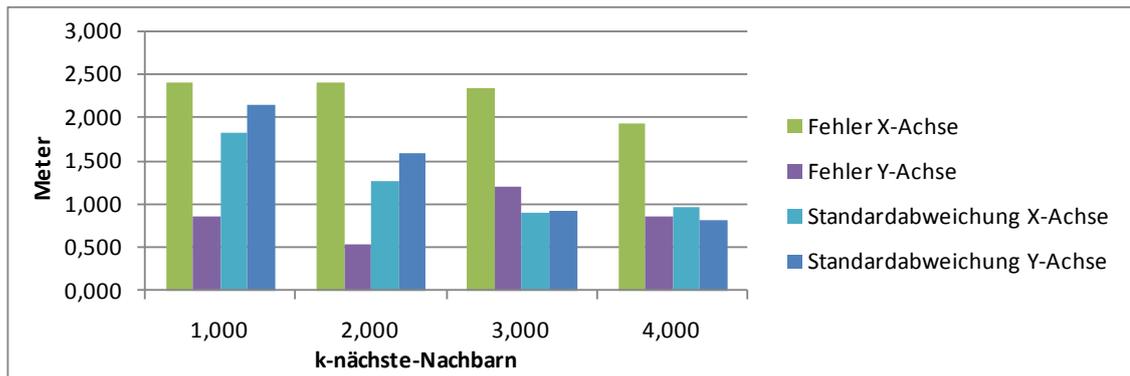


Abbildung 6.6: Lokalisierungsfehler des GSM-Radiomap-Verfahrens

In diesem Diagramm ist wie bereits zuvor der dargestellte Fehler der Durchschnittswert der fünf Messungen in den drei Messpunkten. Die Standardabweichung für $k=1$ zeigt jedoch, dass hohe Schwankungen zwischen den einzelnen ermittelten Positionen vorliegen. Dazu zeigt Abbildung 6.7 exemplarisch die Messwerte für die X/Y-Koordinate im Messpunkt M1 für $k=1$.

Abbildung 6.7: Schwankungen des Positionsfehlers für $k=1$ in Messpunkt M1

Für $k=4$ verringert sich die Standardabweichung und der Positionsfehler, jedoch variieren die Positionen noch immer sehr stark. Eine weitere Erhöhung des Faktors k ist jedoch nicht sinnvoll, da sich der Einfluss von weiter entfernten eingemessenen Positionen erhöht. Für $k=60$ beispielsweise werden alle eingemessenen Positionen (vgl. Abbildung 6.4) verwendet und die daraus berechnete Position wäre fälschlicherweise immer der geometrische Mittelpunkt.

Auch hier liegen die Messreihen auf der beigefügten CD-ROM als *Microsoft Excel*-Datei vor.

Stabilität von GSM-Kanälen

Zusätzlich wurde die Stabilität der GSM-Kanäle untersucht. Für das Prinzip der GSM-Lokalisierung durch eine *Radiomap* ist eine konstante Signalstärke eines GSM-Kanals an einem Ort notwendig. In einem viertägigen Messversuch wurde die Signalstärke an einer Position gemessen. Das Programm im *Visual Studio*-Projekt *GSMDump2Excel* speichert unter Verwendung eines GSM-Moduls und der zugehörigen Klassenbibliothek im Projekt `gsmModule` alle 10 Sekunden die Signalstärke der Kanäle in einer Excel-Datei.

Ein Diagramm im Anhang zeigt sieben Kanäle dieser Messung, wobei sichtbar wird, dass die Signalstärken in der Regel am Wochenende weniger stark bzw. gar nicht schwanken. Der Kanal (ARFCN) mit der Nummer 14 ändert periodisch alle 4 Stunden die Sendeleistung. Allgemein scheint es, als hätten andere Netzteilnehmer Einfluss auf die empfangene Signalstärke, dabei könnte es sich um eine adaptive Anpassung der Sendeleistung einer BTS handeln.

Um trotzdem ein gutes Ergebnis für die Lokalisierung zu erzielen, könnte später zur Verbesserung eine manuelle oder automatische Auswahl der Kanäle für die Lokalisierung geeigneten Kanäle durchgeführt werden.

7 Schlussbetrachtung

In diesem abschließenden Kapitel werden das Arbeitsergebnis sowie die Messergebnisse ausgewertet. Des Weiteren gibt es einen Ausblick über weitere mögliche Erweiterungen der Plattform.

7.1 Zusammenfassung und Fazit

Die Zielstellung dieser Arbeit war es eine Plattform zu entwerfen, welche verschiedene Verfahren zur Indoor-Lokalisierung vereint. Mit Hilfe der Mobilfunktechnologie GSM, in Form des Mobiltelefons *Motorola C123*, und dem Funkmodul *Radiocrafts RC2301AT* aus dem Bereich des IEEE 802.15.4-Standards konnte das Konzept (vgl. Kapitel 3.2) der GSM-*Radiomap* und der Signalstärke-Lateration umgesetzt und evaluiert werden. Die Signalstärke- und Positionsdaten der Evaluierungsplattform werden über eine zusätzliche IEEE 802.15.4-Infrastruktur in Form von Basisstationen an den Host-Computer zur Auswertung und Darstellung übertragen.

Der Vorteil des GSM-*Radiomap*-Verfahrens ist die fast flächendeckende Verfügbarkeit der GSM-Signale - auch innerhalb von Gebäuden. Diesem Vorteil steht der große Einmessaufwand gegenüber. Für den Einsatz von Funkmodulen nach dem IEEE 802.15.4-Standard ist zwar die Installation zusätzlicher Basisstationen notwendig, aber dafür ist die Signalstärke-Lateration weniger anfällig auf Änderung der Signalausbreitungscharakteristik in der direkten Umgebung.

Die Funktionskontrolle dieses Netzwerkes hat gezeigt, dass die Übertragung der Signalstärkedaten nicht immer reibungslos funktioniert hatte (vgl. Kapitel 5.3.6). Sobald die Lösung für dieses Problem gefunden wird, liegt mit der entwickelten Plattform eine vielseitige Evaluierungsmöglichkeit für die beiden Verfahren vor.

Des Weiteren wurde in dieser Arbeit mit Hilfe der Plattform die Genauigkeiten der beiden Verfahren in einem Indoor-Szenario in den Räumlichkeiten des Fraunhofer FOKUS ermittelt (vgl. Kapitel 6.2). Für jedes Verfahren wurde jeweils eine Messreihe durchgeführt, der durchschnittliche Fehler ermittelt und graphisch dargestellt.

Insgesamt ist dabei aufgrund der durchgeführten Messungen das Verfahren der Signalstärke-Lateration durch die *Location Engine* des *Radiocrafts*-Moduls besser zu bewerten. Trotz nicht-isotroper Antennen konnten bei „richtigem“ Signalausbreitungskoeffizienten n hierbei konstante Lokalisierungsergebnisse mit geringem Fehler ermittelt werden (siehe Abbildung 6.5).

Die berechnete Position durch die GSM-*Radiomap* unterlag hierbei jedoch großen Schwankungen, was jedoch auch unter Einbeziehung des k -Nächste-Nachbarn-Algorithmus nicht bedeutend verbessert werden konnte.

7.2 Ausblick

Für die Zukunft besteht die Möglichkeit die beiden Verfahren der Plattform weiter zu optimieren, um diese Verbesserungen gegebenenfalls erneut zu evaluieren.

Im Hinblick auf die Signalstärke-Lateration durch die *Radiocrafts*-Funkmodule dürfte der Einsatz zusätzlicher Rundstrahlantennen statt den integrierten Antennen die größte Verbesserung darstellen. Durch die rundum relativ gleichmäßige horizontale Ausbreitung der Signale und dem zusätzlichen Antennengewinn, sollte eine Verbesserung des Lokalisierungsergebnisses erreicht werden. Des Weiteren stellt neben der Installation der Basisstationen die Kalibrierung der *Location Engine* in der jeweiligen Umgebung einen großen Zeit- und Kostenaufwand dar. Ein möglicher Ansatz um dies zu vermeiden, wäre es die Kalibrierungsparameter durch das Netz von Basisstationen selbst bestimmen zu lassen.

Für das *Radiomap*-Verfahren könnte zukünftig evaluiert werden, inwieweit dessen Anwendung in unterschiedlichen Räumen zu besseren Resultaten der Genauigkeit führt. Aufgrund der Dämpfungseigenschaften der Wände müsste es größere Varianzen der einzelnen Signalstärkepegel geben. Zusätzlich könnte mit Hilfe der Plattform überprüft werden, ob die manuelle oder automatische Selektion von stabilen Kanälen positive Auswirkungen auf das Lokalisierungsergebnis haben würde.

Nach umfangreichen Verbesserungen der einzelnen Verfahren kann auch die Fusion der Sensordaten, bzw. der ermittelten Positionen in Betracht gezogen werden und die gewonnenen Erkenntnisse künftig dazu genutzt werden, die Entwicklung einer Käufer-Lokalisierungslösung zu unterstützen.

8 Literaturverzeichnis

- [6loWPAN] *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. Abgerufen am 20. Juni 2010 von RFC4944: <http://tools.ietf.org/html/rfc4944>
- [ANLE] Texas Instruments. Application Note AN042 - CC2431 Location Engine.
- [Atm10] Atmel. *AVR Studio 4*. Abgerufen am 11. Juli 2010 von http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725
- [Bah100] Bahl, V., & Padmanabhan, V. N. (2000). *RADAR: An In-Building RF-based User Location and Tracking System*. Abgerufen am 03. Juni 2010 von <http://research.microsoft.com/en-us/people/padmanab/infocom2000.pdf>
- [Bal00] Balzert, H. (2000). *Lehrbuch der Software-Technik – Software Entwicklung*. Heidelberg; Berlin: Spektrum Akademischer Verlag.
- [C123] Testberichte.de. *Motorola C123*. Abgerufen am 12. Juli 2010 von Testberichte.de: <http://www.testberichte.de/p/tchibo-motorola-tests/c123-testbericht.html>
- [Cad10] Cadsoft Computer GmbH. *CadSoft*. Abgerufen am 20. Mai 2010 von <http://www.cadsoft.de/info.htm>
- [CALYPSO] Texas Instruments. *CALYPSO-Chip*. Abgerufen am 08. Juli 2010 von <http://cryptome.org/ti-calypso1.pdf>
- [CC2430] Texas Instruments. (30. Mai 2007). *CC2430*. Abgerufen am 29. Mai 2010 von System-on-Chip (SoC) Solution for ZigBee/IEEE 802.15.4 Wireless Sensor Network: <http://focus.ti.com/docs/prod/folders/print/cc2430.html>
- [CC2431] Texas Instruments. (15. Juni 2007). *CC2431*. Abgerufen am 29. Mai 2010 von System-on-Chip (SoC) Solution for ZigBee/IEEE 802.15.4 Wireless Sensor Network: <http://focus.ti.com/docs/prod/folders/print/cc2431.html>
- [CCDEBUG] Texas Instruments. *Debugger and Programmer for RF System-on-Chips*. Abgerufen am 20. Juni 2010 von CC-Debugger: <http://focus.ti.com/docs/toolsw/folders/print/cc-debugger.html>
- [CCFLASHER] Kuhar, P. (2008). *CC Flasher - Open Source TI/Chipcon SOC Flash programmer*. Abgerufen am 13. April 2010 von http://www.modula.si/cc_flasher/en
- [DH10] Dodel, H., & Häupler, D. (2010). *Satellitennavigation*. Berlin - Heidelberg: Springer Verlag.
- [DONGLE] *Zigbee USB Dongle*. Abgerufen am 22. Juni 2010 von <http://www.adaptivem2m.com/zigbee-technology/zigbee-usb-dongle.htm>

- [Ele10] Elektronik Kompendium. *GPS*. Abgerufen am 12. Juli 2010 von <http://www.elektronik-kompendium.de/sites/kom/1201071.htm>
- [EuAb] Wikipedia. *Euklidischer Abstand*. Abgerufen am 30. Mai 2010 von Euklidischer Abstand: http://de.wikipedia.org/w/index.php?title=Euklidischer_Abstand&oldid=74708051
- [FOKUS] Fraunhofer FOKUS. Abgerufen am 10. Mai 2010 von Was ist FOKUS: http://www.fokus.fraunhofer.de/de/fokus/institut/was_ist_fokus/index.html
- [Gat99] Gatliff, B. (November 1999). *Embedding with GNU: the gdb Remote Serial Protocol*. Abgerufen am 20. Juni 2010 von <http://leadingedgeinnovations.biz/EmbeddedSystemProgramming/files/99/9911/f-gatlif.pdf>
- [GCCARM] *GNU ARM Toolchain*. Abgerufen am 20. Juni 2010 von <http://www.gnuarm.com>
- [Goo10] Goodspeed, T. *CC2430 Debug Protocol, First Notes*. Abgerufen am 20. Juni 2010 von Travis Goodspeed's Blog: <http://travisgoodspeed.blogspot.com/2009/10/cc2430-debug-protocol-first-notes.html>
- [HBW00] Hightower, J., Borriello, G., & Want, R. (2000). *SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength*. Washington: University of Washington.
- [HUB10] Humboldt Universität Berlin. *MagicMap*. Abgerufen am 05. Juni 2010 von <http://www2.informatik.hu-berlin.de/rok/MagicMap/index.htm>
- [IAREW] IAR. *IAR Embedded Workbench® for 8051*. Abgerufen am 20. Juni 2010 von <http://www.iar.com/website1/1.0.1.0/244/1/>
- [IEEE] IEEE. Abgerufen am 29. April 2010 von Institut of Electrical and Electronics Engineers: <http://standards.ieee.org/>
- [IEEE06] IEEE 802.15 Working Group. (2006). Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
- [ITW10a] ITWissen. *IEEE 802.15.4*. Abgerufen am 04. Mai 2010 von <http://www.itwissen.info/definition/lexikon/802-15-4-IEEE-802-15-4.html>
- [ITW10b] ITWissen. *Isotrop*. Abgerufen am 14. Juni 2010 von <http://www.itwissen.info/definition/lexikon/Isotrop-isotropic.html>
- [ITW10c] ITWissen. *HDLC*. Abgerufen am 17. Juni 2010 von <http://www.itwissen.info/definition/lexikon/high-level-data-link-control-HDLC-HDLC-Protokoll.html>
- [Kar05] Karl, H., & Willing, A. (2005). *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, Ltd.

- [Kli10] Klingbeil, L. (5. Mai 2010). Fusion verschiedener Sensoren und zusätzlicher Informationen zur Indoor Lokalisierung. Berlin.
- [KNN] Wikipedia. (28. Mai 2010). *Nächste-Nachbarn-Klassifikation*. Abgerufen am 7. Juni 2010 von <http://de.wikipedia.org/w/index.php?title=N%C3%A4chste-Nachbarn-Klassifikation&oldid=74874117>
- [Kup07] Kupris, G., & Sikora, A. (2007). *Zigbee - Datenfunk mit IEEE 802.15.4 und Zigbee*. Poing: Franzis Verlag.
- [Lan00] Lange, C. (November 2000). *Untersuchung von Algorithmen zur visuellen Roboterlokalisierung*. Abgerufen am 05. Juni 2010 von Untersuchung von Algorithmen zur visuellen Roboterlokalisierung: <http://www.techfak.uni-bielefeld.de/~clange/diplom.pdf>
- [MAC] Texas Instruments. (12. Juni 2009). 802.15.4 MAC Application Programming Interface.
- [Mic10] Microsoft. *Getting Started (Entity Framework)*. Abgerufen am 10. Juli 2010 von <http://msdn.microsoft.com/en-us/library/bb386876.aspx>
- [MSTACK] Kuhar, P. *m-Stack, a 6loWPAN stack for CC2430 with DYMO routing support*. Abgerufen am 20. Juni 2010 von <http://www.modula.si/mstack/>
- [Mul01] Muller, N. J. (2001). *Bluetooth*. MITP Verlag.
- [Nan10] Nanotron. Abgerufen am 12. Mai 2010 von Technology - Symmetrical Double-Sided Two Way Ranging: http://www.nanotron.com/EN/CO_techn-sdstwr.php
- [Neu10] Neumann, D. T. *GSM Kanäle*. Abgerufen am 7. Juli 2010 von http://www.roaminginfo.de/html/kanale_und_frequenzen.html
- [OSAL] Texas Instruments. (9. April 2009). OSAL-API.
- [OSALHAL] Texas Instruments. *OSAL and HAL*. Abgerufen am 22. Mai 2010 von OSAL and HAL: http://processors.wiki.ti.com/images/a/a9/06_-_OSAL_and_HAL.pdf
- [OSMOCOM] *OsmocomBB*. Abgerufen am 21. April 2010 von Open source mobile communications BaseBand: <http://bb.osmocom.org/trac/>
- [OSMOLOAD] *OsmocomBB. The Compal RAM Loader*. Abgerufen am 16. Juni 2010 von <http://bb.osmocom.org/trac/wiki/CompalRamloader>
- [PCB00] Priyantha, N. B., Chakraborty, A., & Balakrishnan, H. (2000). The Cricket Location-Support System. *Proceedings of the 6th annual international conference on Mobile computing and networking* (S. 32-43). Boston: ACM.
- [RC2301AT] Radiocrafts. (2007). *RC2300/01/02/04*. Abgerufen am 09. Juni 2010 von http://www.radiocrafts.fr/uploads/rc230x_data_sheet_1_2.pdf

- [Red95] Redl, S., Weber, M., & Oliphant, M. (1995). *An Introduction To GSM*. Artech House Publishers.
- [Rot02] Roth, J. (2002). *Mobile Computing*. dpunkt.verlag.
- [Sch07] Schwee, C. (25. April 2007). Abgerufen am 11. Juli 2010 von http://www.physik.uni-marburg.de/fileadmin/user_upload/veranstaltungen/praktika/physik/FehlerfolienSchwee.pdf
- [SDCC] *Small Device C Compiler*. Abgerufen am 14. Juni 2010 von Small Device C Compiler: <http://sdcc.sourceforge.net/>
- [Sky10] Skyhook Wireless. *How It Works*. Abgerufen am 05. Juni 2010 von <http://www.skyhookwireless.com/howitworks/>
- [Sza10] Szabo, Andrei; Siemens. (2010). Received Signal Strength localization, from simple models to automated learning. *AVS Workshop*. Berlin.
- [Tan00] Tanenbaum, A. S. (2000). *Computernetzwerke*. Pearson Studium.
- [TIFLASH] Texas Instruments. *SmartRF Flash Programmer*. Abgerufen am 8. Juli 2010 von <http://focus.ti.com/lit/ug/swru069d/swru069d.pdf>
- [TIMAC] Texas Instruments. *TIMAC*. Abgerufen am 20. Mai 2010 von IEEE802.15.4 Medium Access control (MAC) software stack: <http://focus.ti.com/docs/toolsw/folders/print/timac.html>
- [TINYOS51] *TinyOS 8051 working group*. Abgerufen am 20. Juni 2010 von <http://www.tinyos8051wg.net/>
- [TUD10] TU Dortmund. *Passive Infrarot-Lokalisierung*. Abgerufen am 05. Juni 2010 von Passive Infrarot-Lokalisierung: http://www.irf.tu-dortmund.de/cms/de/IT/Forschung/Embedded_Systems/Projekte/PIL/index.html
- [USB2BOT] c't Magazin. *Aufbauhinweise zum USB-2-Bot-Adapter*. Abgerufen am 22. Juni 2010 von <http://www.heise.de/ct/projekte/machmit/ctbot/wiki/USB2Bot>
- [VLHLO07] Varshavsky, A., de Lara, E., Hightower, J., LaMarca, A., & Otsason, V. (Dezember 2007). GSM indoor localization. *Computing, Pervasive and Mobile Volume 3 , Issue 6 , S. 698-720*.
- [VS2008] Microsoft. *Microsoft Visual Studio 2010*. Abgerufen am 20. Juni 2010 von <http://www.microsoft.com/germany/visualstudio/>
- [Wan07] Wanner, N. (September 2007). *Nutzung der Sensorik des MSB430 zur Lokalisierung und Detektierung von sich bewegendenden Personen*. Abgerufen am 05. Juni 2010 von http://page.mi.fu-berlin.de/wanner/publications/wanner07_slides.pdf

- [Wel10] Welte, H. *Typical TI Calypso baseband modem design* ¶. Abgerufen am 08. 07 2010 von <http://bb.osmocom.org/trac/wiki/TypicalCalypsoModemDesign>
- [WHFG92] Want, R., Hopper, A., Falcão, V., & Gibbons, J. (Jan 1992). The Active Badge Location System. *ACM Transactions on Information Systems*, S. 91-102.
- [WIPS00] Royal Institut of Technology, Schweden. *WIPS Technical Documentation*. Abgerufen am 03. Juni 2010 von <http://www.tslab.ssvl.kth.se/csd/projects/0012/technical.pdf>
- [WJH97] Ward, A., Jones, A., & Hopper, A. (Oct 1997). A new location technique for the active office. *IEEE Personal Communications; Volume: 4 Issue:5*, S. 42-47.

9 Abbildungsverzeichnis

Abbildung 2.1: Tracking und Positioning.....	3
Abbildung 2.2: Angulation und Lateration	4
Abbildung 2.3 Graphische Darstellung einer <i>Radiomap</i> [Sza10]	4
Abbildung 2.4: Kombination verschiedener Sensordaten [Kli10]	9
Abbildung 2.5: OSI-Referenzmodell	11
Abbildung 2.6: PHY- und MAC-Datenrahmen [IEEE06]	15
Abbildung 2.7: Topologien nach IEEE 802.15.4 - vgl. [IEEE06]	16
Abbildung 2.8: Anmeldung eines Knotens im Netzwerk - vgl. [Kup07] S. 71	16
Abbildung 2.9: Beacon-Enabled Modus	17
Abbildung 2.10: IEEE 802.15.4 und ZigBee im Überblick - vgl. [Kup07] S. 21	18
Abbildung 2.11: Bluetooth-Pico-Netz.....	19
Abbildung 2.12: Wireless LAN Modi.....	20
Abbildung 2.13: Einfache Darstellung der Zellstruktur von GSM.....	21
Abbildung 3.1: Blockschaltbild der Plattform.....	23
Abbildung 3.2: Positionsermittlung der Plattform durch Lateration mit IEEE 802.15.4	24
Abbildung 3.3: Positionsermittlung der Plattform durch GSM-Radiomap.....	25
Abbildung 3.4: Verbesserung der Ortung durch Mittelung der Koordinaten	26
Abbildung 3.5: Softwarestruktur des Systems als vereinfachtes Komponentendiagramm	26
Abbildung 4.1: Übersichtsdiagramm für die Evaluierungsplattform.....	29
Abbildung 5.1: Motorola C123 [C123]	31
Abbildung 5.2: Radiocrafts-Modul RC2301AT	32
Abbildung 5.3: CC-Debugger von Texas Instruments und Minimalbeschaltung [CCDEBUG] ..	33
Abbildung 5.4: Aufbau des TIMAC Stack.....	34
Abbildung 5.5: Die integrierte Entwicklungsumgebung IAR Embedded Workbench	35
Abbildung 5.6: Wireless Protocol Analyser	35
Abbildung 5.7: ZigBee USB Dongle von Integration [DONGLE]	36
Abbildung 5.8: SmartRF Flash Programmer - IEEE Adresse	36
Abbildung 5.9: CC Flasher - Modifizierte Platine.....	37

Abbildung 5.10: Struktur einer OSAL-Anwendung.....	39
Abbildung 5.11: Das Speicherbank-Mapping.....	39
Abbildung 5.12: Auslesen der IEEE Adresse.....	40
Abbildung 5.13: Sequenzdiagramm beim Assoziieren einer Basisstation oder Plattform	41
Abbildung 5.14: Programmausschnitt zum Senden und Empfangen von Datenrahmen	42
Abbildung 5.15: Rahmenstruktur der MAC-Payload.....	42
Abbildung 5.16: Basisstation.....	43
Abbildung 5.17: Kodierte Position in der IEEE-Adresse.....	43
Abbildung 5.18: Ablauf der Signalstärkeermittlung	44
Abbildung 5.19: Datenstruktur beim Erhalt eines Datenrahmens	44
Abbildung 5.20: Programmablaufplan einer Basisstation beim Start	45
Abbildung 5.21: Evaluierungsplattform.....	46
Abbildung 5.22: Handshake des RAM-Bootloaders	47
Abbildung 5.23: Aufruf und verkürzte Ausgabe des Tools <i>osmocom</i>	47
Abbildung 5.24: Array mit Ringpuffer für RSSI-Werte	48
Abbildung 5.25: Übertragen der Liste der signalstärksten Kanäle	49
Abbildung 5.26: Programmablaufplan des Programms <i>l1test</i>	49
Abbildung 5.27: Liste für Koordinaten und RSSI-Werte von Basisstationen	50
Abbildung 5.28: Räumliche Grenzen für die Location Engine	51
Abbildung 5.29: Programmablaufplan der Location Engine.....	51
Abbildung 5.30: Einmessen des Parameter A	52
Abbildung 5.31: Programmablauf der Plattform	53
Abbildung 5.32: Klassendiagramm für <i>gsmModule</i>	55
Abbildung 5.33: Klasse <i>gsmDatabase</i>	56
Abbildung 5.34: Graphische Benutzeroberfläche.....	57
Abbildung 6.1: Foyer und Grundriss des Foyers.....	60
Abbildung 6.2: Bestimmung des Parameter <i>n</i> durch die Regressionsgerade	61
Abbildung 6.3: Mitschnitt des IEEE 802.15.4-Protokolls im Wireless Protocol Analyser	63
Abbildung 6.4: Darstellung der ermittelten Position durch IEEE 802.15.4 und GSM.....	64
Abbildung 6.5: Lokalisierungsfehler der Location Engine.....	65

Abbildung 6.6: Lokalisierungsfehler des GSM-Radiomap-Verfahrens	66
Abbildung 6.7: Schwankungen des Positionsfehlers für $k=1$ in Messpunkt M1	66
Abbildung A.1: Schaltplan der Basisstation	81
Abbildung A.2: Schaltplan cc-flasher	82
Abbildung A.3: Blockschaltbild des CC2431 [CC2431]	83
Abbildung A.4: Blockschaltbild des Motorola C123 [Wel10]	84
Abbildung A.5: Blockschaltbild des CALYPSO-Chip [CALYPSO]	84
Abbildung A.6: Langzeitmessung der GSM-Signalstärke-Werte	85

10 Tabellenverzeichnis

Tabelle 2.1: Einige wichtige Indoor-Lokalisierungsverfahren	9
Tabelle 2.2: Kanäle im 2,4 GHz-Band	14
Tabelle 5.1: Verschiedene Rahmentypen.....	43
Tabelle 6.1: Messungen der Signalstärke aus verschiedenen Winkeln in 1 Meter Abstand.....	61
Tabelle 6.2: Positionen der Basisstation für das Foyer	62
Tabelle 6.3: Messpunkte für die Bestimmung der Genauigkeit der Location Engine.....	64
Tabelle 6.4: Messpunkte für die Bestimmung der Genauigkeit des Radiomap-Verfahrens.....	66

11 Abkürzungen

AoA	Angel of Arrival
ARFCN	Absolut Radio Frequency Channel Number
ARQ	Automatic Repeat reQuest
API	Application Programming Interface
APS	Application Support Sublayer
BTS	Base Transceiver Station
CDMA	Code Division Multiple Access
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CoO	Cell of Origin
DC	Debug Clock
DD	Debug Data
D-GPS	Differential Global Positioning System
DSSS	Direct Sequence Spread Spectrum
DSP	Digitaler Signalprozessor
EEPROM	Electrically Erasable Programmable Read Only Memory
EF	Entity Framework
EUI	Extension Unified Identifier
FCCH	Frequency Correction Channel
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FFD	Full Function Device
FHHS	Frequency Hopping Spread Spectrum
FOKUS	Fraunhofer-Institut für Offene Kommunikationssysteme
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
IDE	Integrated Development Environment
IEEE	Institute of Electric and Electronic
ISM	Industrial, Scientific and Medical
ISO	International Standards Organisation
LAN	Local Area Network
LED	Light Emitting Diode
MAC	Medium Access Control
MPDU	MAC protocol data unit

MSDU	MAC service data unit
MSB	Most Significant Bit
NWK	Network Layer
OSI	Open Standards Interconnection
OSAL	Operation System Abstraction Layer
PAN	Personal Area Network
PC	PAN Coordinator
PHY	Physical Layer
PoS	Point of Sale
RFD	Reduced Function Device
RFID	Radio Frequency Identification
RTof	Roundtrip Time of Flight
SANE	Sensor Applications and Networks
SAP	Service Access Point
SMD	Surface Mounted Device
SDS-TWR	Symmetrical Double-Sided Two Way Ranging
SFR	Special Function Register
SoC	System on Chip
TDMA	Time Division Multiple Access
TDoA	Time Difference of Arrival
ToA	Time of Arrival
TI	Texas Instruments
TG	Task Group
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VM	Virtuelle Maschine
WLAN	Wireless LAN
WPA	Wireless Protocol Analyser
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Networks
ZDO	ZigBee Device Object

A. Anhang

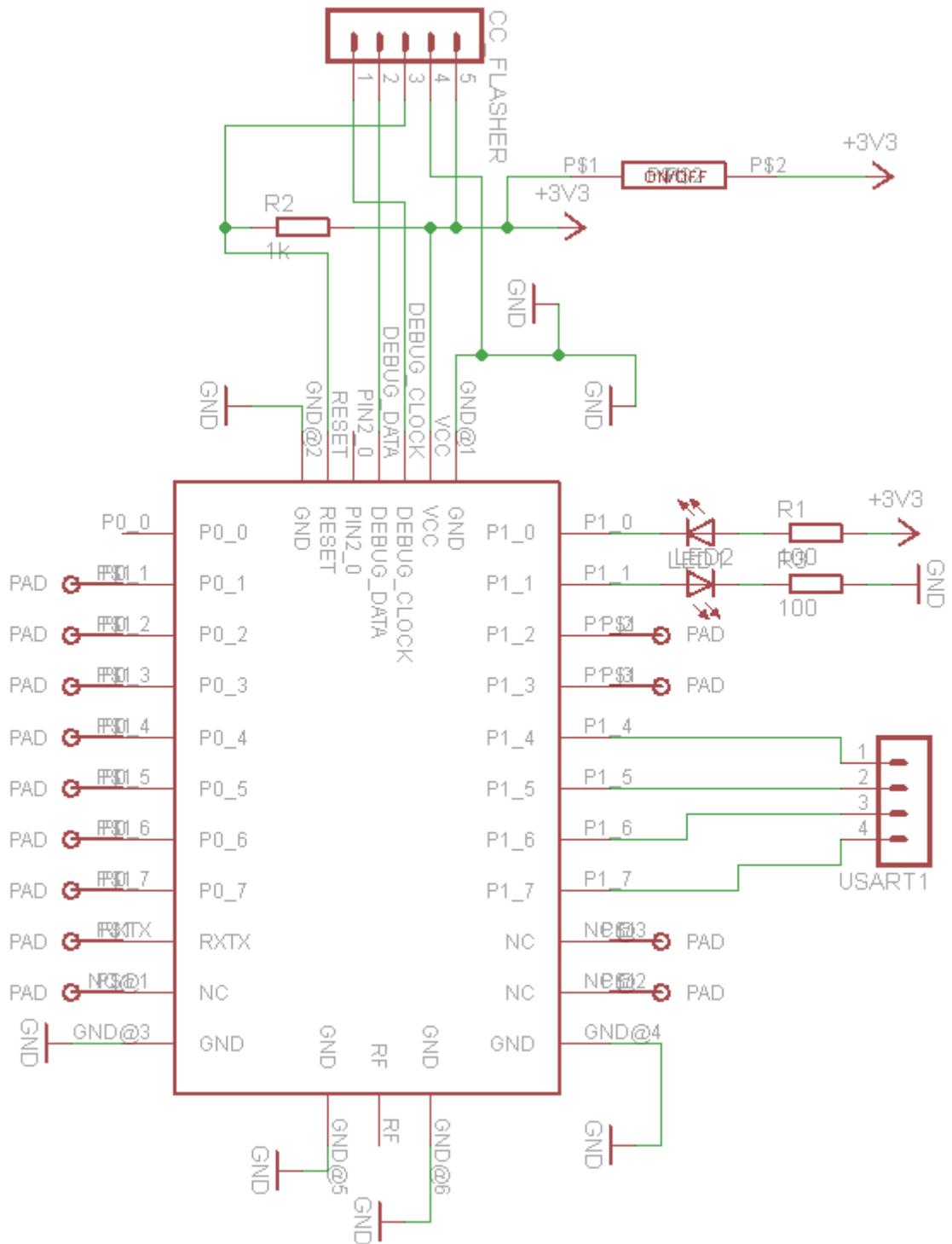


Abbildung A.1: Schaltplan der Basisstation

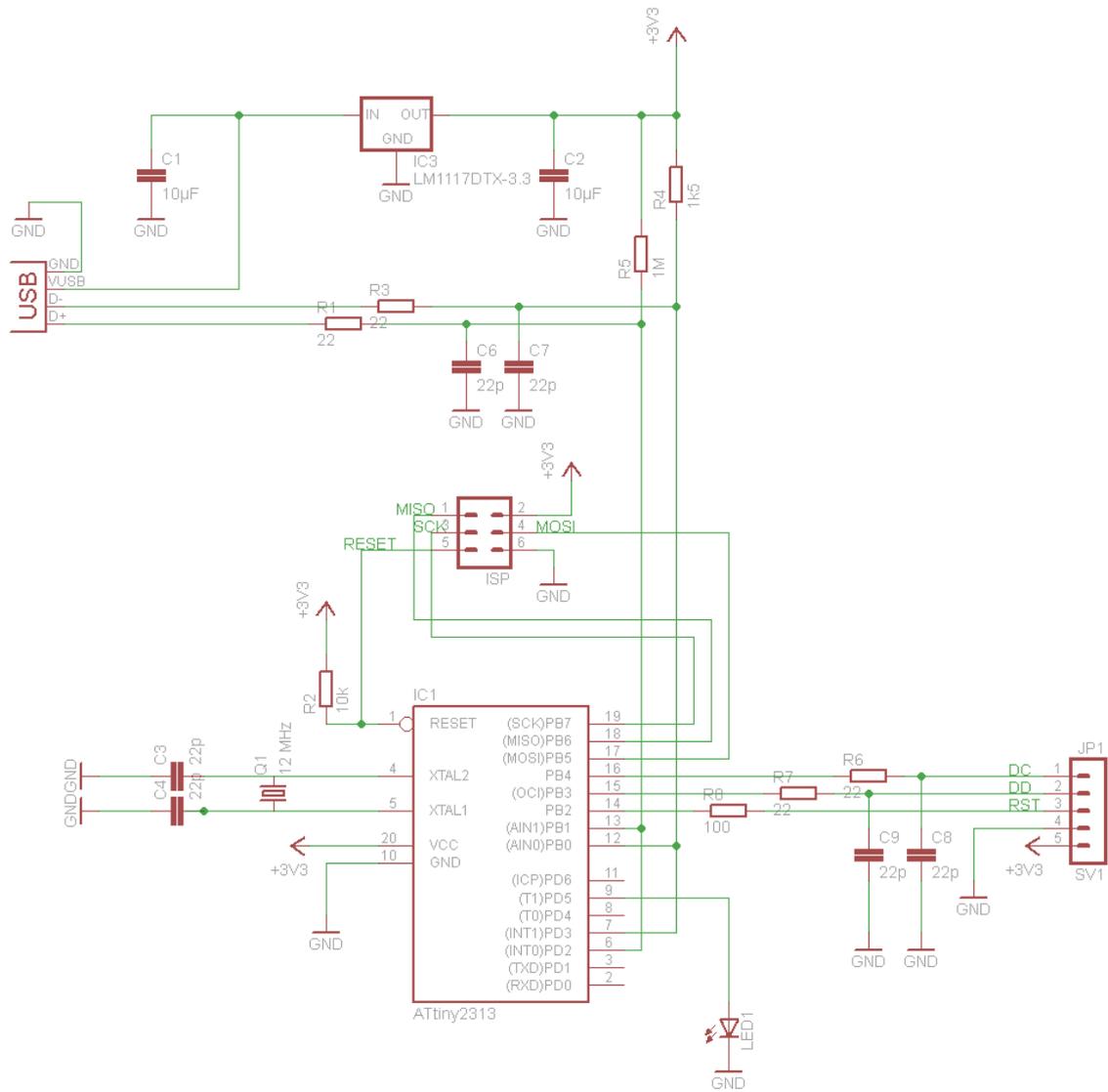


Abbildung A.2: Schaltplan cc-flasher

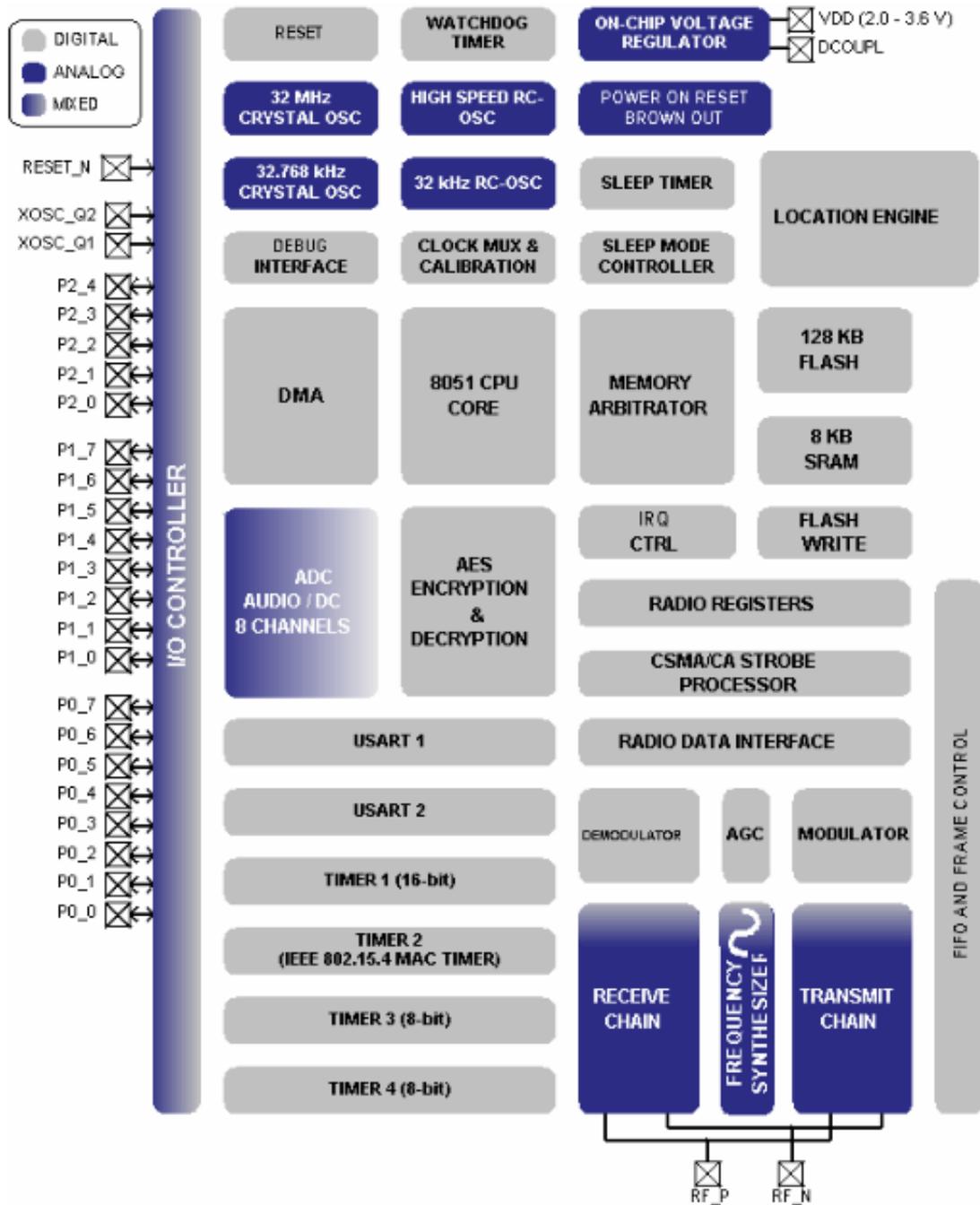


Abbildung A.3: Blockschaltbild des CC2431 [CC2431]

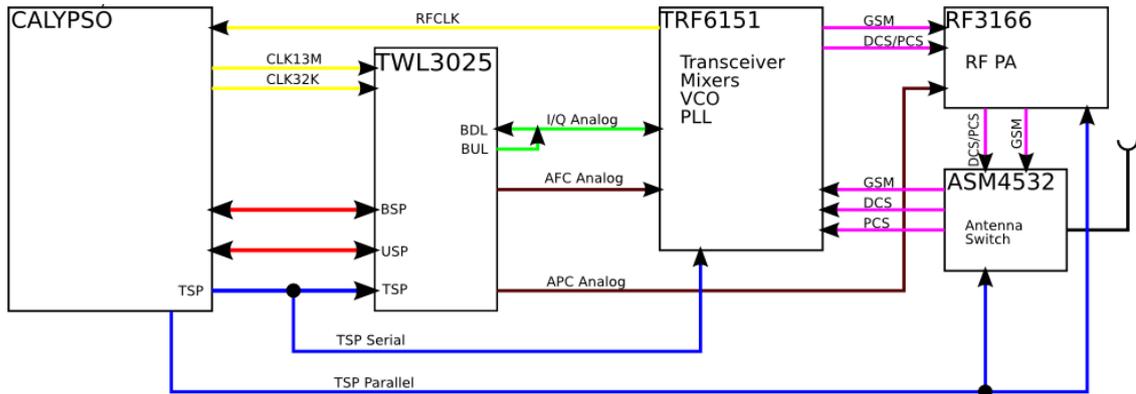


Abbildung A.4: Blockschaltbild des Motorola C123 [Wei10]

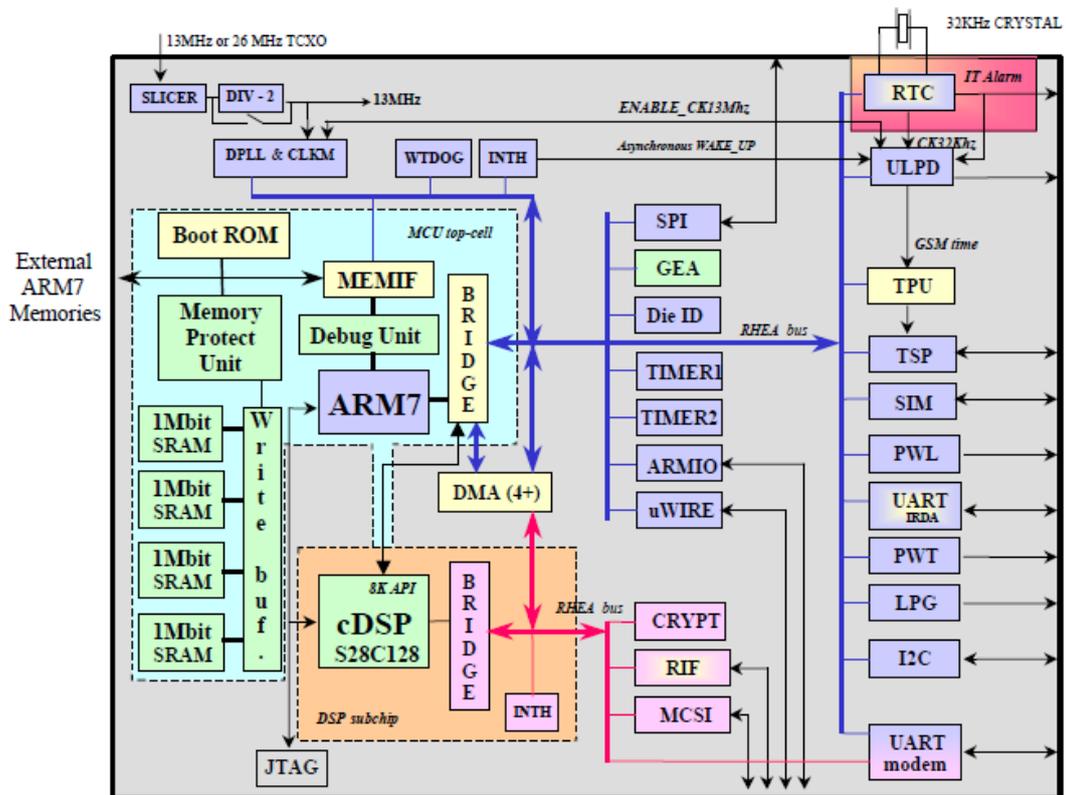


Abbildung A.5: Blockschaltbild des CALYPSO-Chip [CALYPSO]

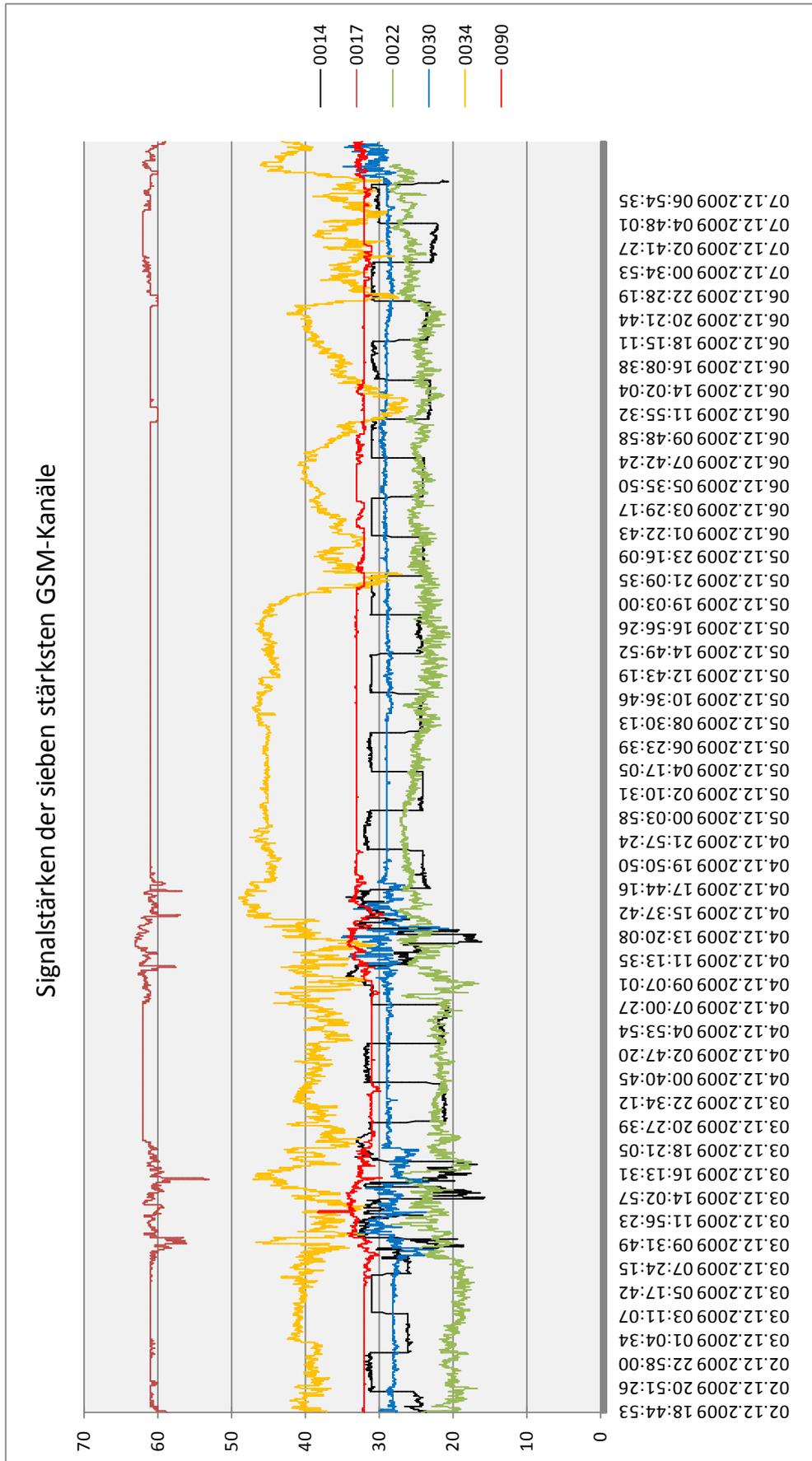


Abbildung A.6: Langzeitmessung der GSM-Signalstärke-Werte

B. Inhalt der CD

Auf der beigefügten CD-ROM befindet sich im Wurzelverzeichnis die Diplomarbeit im PDF-Format sowie folgender Inhalt:

Datenblätter der verwendeten Hard- und Software

\Datenblätter\

Wichtige Quellen der Arbeit

\Quellen\

Messergebnisse der Evaluierung

\Messungen\

Software des *Radiocrafts* RC2301AT-Moduls für die Basisstationen sowie der Plattform

\Software\Basisstation_Plattform\TIMAC-CC2430-1.3.0\Projects\mac\sample\

Software des *Motorola* C123

\Software\Motorola_C123\l1test\

Virtuelle Maschine mit kompletter Toolchain des OsmocomBB-Projekts für *Motorola* C123

\Software\Motorola_C123\VM\

Software für den Host-Computer

\Software\Host_Computer\GSMLocalization