



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Arno Davids

Positionsbestimmung innerhalb eines Gebäudes
auf Basis eines Motes-Sensornetzwerks in einem
Rescue-Szenario

Arno Davids

Positionsbestimmung innerhalb eines Gebäudes
auf Basis eines Motes-Sensornetzwerks in einem
Rescue-Szenario

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Gunter Klemke
Zweitgutachter: Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 14. Februar 2008

Arno Davids

Thema der Masterarbeit

Positionsbestimmung innerhalb eines Gebäudes auf Basis eines Motes-Sensornetzwerks in einem Rescue-Szenario

Stichworte

Positionsbestimmung, Lokalisation, Drahtloses Sensornetzwerk, Motes, TelosB, TinyOS, IEEE 802.15.4, ZigBee, RSSI

Kurzzusammenfassung

Standortbezogene Dienste (Location-Based Services) werden heute immer wichtiger. Typische Anwendungen sind z. B. Routenplaner, Restaurantfinder oder Notfallrettung. Um diese Dienste anbieten zu können, ist eine Form der Positionsbestimmung notwendig. Da innerhalb von Gebäuden nicht auf GPS zurückgegriffen werden kann, sind alternative Verfahren zu Positionsbestimmung erforderlich.

In dieser Masterarbeit wird, basierend auf einem Rescue-Szenario, eine Positionsbestimmung in einem Gebäude realisiert. Dazu wird ein Gebäude mit Sensorknoten ausgerüstet, die ein drahtloses Sensornetzwerk bilden. Die Position von Rettungskräften, die mit einem Sensorknoten ausgestattet sind, kann mit Hilfe des Sensornetzes ermittelt werden. Die Sensorknoten werden zusätzlich eingesetzt zur Sensorüberwachung des Gebäudes. Sensorwerte und Positionsdaten werden in einer Leitstandanwendung visuell dargestellt.

Arno Davids

Title of the master thesis

Indoor-Localization based on a motes sensornetwork in a rescue scenario

Keywords

Lokalization, Wireless Sensor Network, Motes, TelosB, TinyOS, IEEE 802.15.4, Zig-Bee, RSSI

Abstract

The importance of location-based services is increasing. Everyday applications are travel planning, restaurant finder or emergency rescue. To provide these services location-specific information is necessary. Indoor localization systems cannot use GPS, consequently alternative methods are required.

The indoor localization system, based on a rescue scenario, is the subject matter of this master thesis. Sensor nodes are mounted in a building and provide a wireless sensor network. With this sensor network location data can be determined for those rescue workers equipped with a sensor node. Furthermore the sensor nodes are used for monitoring of the building. Sensor node values along with the locations data will be displayed in the rescue control center.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 8 |
| 1.1 | Rescue-Szenario | 9 |
| 1.2 | Motivation | 10 |
| 1.3 | HAW Rescue-Projekt | 11 |
| 1.4 | Gliederung der Arbeit | 12 |
| 2 | Ziele und Vorhaben | 14 |
| 2.1 | „Sensornetz aus stationären Sensorknoten“ | 14 |
| 2.2 | „Rettungskräfte als mobile Knoten“ | 15 |
| 2.3 | „Sensorkette als Lifeline-Ersatz“ | 16 |
| 2.4 | „Container für den Sensorauswurf“ | 17 |
| 3 | Grundlagen und Analyse | 19 |
| 3.1 | Drahtloses Sensornetzwerk | 19 |
| 3.2 | Sensorkommunikation | 21 |
| 3.2.1 | IEEE 802.15.4 und ZigBee | 22 |
| 3.2.2 | IEEE 802.11/ WLAN | 26 |
| 3.2.3 | Digitaler Mobilfunk (GSM) | 27 |
| 3.2.4 | Bewertung und Analyse | 29 |
| 3.3 | Lokalisation | 31 |
| 3.3.1 | Distanzbestimmung | 32 |
| 3.3.2 | Winkelbestimmung | 34 |
| 3.3.3 | Triangulation | 35 |
| 3.4 | Sensorplattformen | 37 |
| 3.4.1 | Scatterweb | 37 |
| 3.4.2 | BTnode | 38 |
| 3.4.3 | SunSpot | 38 |
| 3.4.4 | Berkeley Motes | 39 |
| 3.4.5 | Bewertung und Analyse | 41 |
| 4 | TinyOS | 44 |
| 4.1 | Komponentenmodell | 46 |
| 4.2 | Nebenläufigkeit | 47 |

| | | |
|----------|---|-----------|
| 4.3 | Split-Phase Operation | 47 |
| 5 | Current Work | 49 |
| 5.1 | Smart Dust | 49 |
| 5.1.1 | Bewertung | 51 |
| 5.2 | intelligent Building Response (iBR) | 52 |
| 5.2.1 | Bewertung | 53 |
| 5.3 | FIRE (Fire Information and Rescue Equipment)-Project | 54 |
| 5.3.1 | SmokeNet | 54 |
| 5.3.2 | Bewertung | 56 |
| 5.4 | wearIT@work | 56 |
| 5.4.1 | Bewertung | 57 |
| 5.5 | RELATE: Relative Positioning of Mobile Objects in Ad hoc Networks | 57 |
| 6 | Entwurf | 59 |
| 6.1 | Anforderungen | 59 |
| 6.2 | Hard- und Softwaregrundlagen | 61 |
| 6.3 | Programmierungsumgebung und Werkzeuge | 62 |
| 6.4 | Aufbau eines Sensornetzes | 62 |
| 6.4.1 | Funktionale Anforderungen | 64 |
| 6.4.2 | Datenflussrichtung | 65 |
| 6.5 | Entwurf einer Leitstand-Applikation | 66 |
| 6.5.1 | Funktionale Anforderungen | 68 |
| 7 | Realisierung und Ergebnisse | 69 |
| 7.1 | Die Sensornetzanwendungen | 69 |
| 7.1.1 | Mobile Unit | 69 |
| 7.1.2 | Detector Unit | 71 |
| 7.1.3 | Base Unit | 72 |
| 7.2 | Die Leitstandanwendung | 72 |
| 7.2.1 | Umwandlung des RSSI-Wertes | 73 |
| 7.2.2 | Positionsberechnung durch Lateration | 73 |
| 7.2.3 | Sensornetzkommunikation | 75 |
| 7.2.4 | Graphical User Interface | 76 |
| 7.3 | Ergebnisse | 78 |
| 8 | Schlussbetrachtungen | 82 |
| 8.1 | Zusammenfassung | 82 |
| 8.2 | Fazit | 83 |
| 8.3 | Ausblick | 84 |
| | Literaturverzeichnis | 87 |

| | |
|----------------------|-----------|
| A Anhang | 91 |
| A.1 nesdoc | 91 |

1 Einleitung

Die Bestimmung des eigenen Standorts ist eine Aufgabe, die schon seit langem die Menschen beschäftigt. Ein Beispiel hierfür ist die antike Navigation, die die Gestirne als natürliche Bezugspunkte für eine Orientierung nutzte.

Heutzutage existieren andere und genauere Verfahren zur Positionsbestimmung. Ortsinformationen werden heute in vielen Anwendungen und Systeme benötigt, um dem Benutzer eines solchen Systems, standortbezogene Dienste (Location-Based Services) anzubieten. Typische Anwendungen sind u. a. Routenplaner, Hotel- oder Restaurantfinder, Verkehrsinformationen oder auch die Nottfallortung.

Um diese Dienste zu nutzen, ist eine Form der Positionsbestimmung notwendig. Ein Beispiel hierfür ist ein Navigationsgerät das sich in einem Auto befindet. Um dem Benutzer die gewünschte Route zum Ziel anzeigen zu können, ist es zunächst nötig, den eigenen Standpunkt und damit den Ausgangspunkt der Routenberechnung zu kennen. Eine Positionsbestimmung ist ebenfalls während der Fahrt erforderlich, damit die aktuelle Position auf einer Landkarte dargestellt werden kann. GPS (Global Positioning System) wird vielfach zur Positionsbestimmung eingesetzt, so auch in dem Beispiel des Navigationsgeräts.

GPS basiert auf der Satellitenortung. Ein GPS-Empfänger braucht den Kontakt zu mehreren Satelliten, die die Erde umkreisen und ständig ihre Position und die aktuelle Zeit aussenden, um daraus den eigenen Standort errechnen zu können. Durch die Notwendigkeit permanent Satellitenkontakt zu benötigen wird ein Nachteil des GPS-Systems deutlich: Innerhalb von Gebäuden ist keine Positionsbestimmung möglich, denn dort besteht größtenteils keine Verbindung zu den Satelliten.

Um auch innerhalb von Gebäuden ein Positionsbestimmungsverfahren, auch als *Indoor-Lokalisierung-System* bezeichnet, anbieten zu können, muss auf alternative Verfahren zurückgegriffen werden. Ein möglicher Ansatz, um in einem Gebäude eine Lokalisation durchzuführen, soll Gegenstand dieser Masterarbeit sein. Es wird untersucht, inwieweit sich ein Motes-Sensornetzwerk dazu eignet, die Position von Objekten innerhalb eines Gebäudes zu bestimmen. Dazu wird ein Bereich in einem Gebäude mit Sensorknoten ausgerüstet und die Position von frei beweglichen Objekten, die ebenfalls mit Sensorknoten ausgestattet sind, bestimmt.

Zusätzlich zum Vorhaben einer Indoor-Lokalisation ist diese Masterarbeit eingebettet in ein Rescue-Szenario. Es werden daher die besonderen Anforderungen, die ein solches Umfeld mit sich bringt, berücksichtigt. Um innerhalb des Rescue-Szenarios möglichst viele Daten von dem Geschehen zu erhalten, sollen das Motes-Sensornetzwerk ebenfalls dazu dienen eine Sensorüberwachung in einem Gebäude durchzuführen.

Innerhalb des Rescue-Szenarios entstehen viele Vorteile durch eine Positionsbestimmung in einem Gebäude. Durch die Lokalisation der Rettungskräfte kann die Einsatzleitung im Leitstand beispielsweise erkennen, welche Rettungskräfte sich an welcher Stelle im Gebäude aufhalten. Damit wird die Übersicht über das Geschehen in dem Gebäude erhöht und die Leitstandbenutzer haben die Möglichkeit diese Informationen zu nutzen, um z. B. Feuerwehrmänner in dem Gebäude umzuverteilen, Rettungskräfte, die sich in gefährdeten Bereichen aufhalten, gezielt zu evakuieren oder verunglückte Feuerwehrkräfte zu bergen. Im nächsten Abschnitt wird das Szenario detailliert erörtert und Gründe aufgezeigt, die für den Einsatz eines Positionsbestimmungsverfahrens sprechen.

1.1 Rescue-Szenario

Die Motivation für diese Arbeit soll anhand eines Szenarios, das jedoch Bezug zur Realität besitzt, definiert werden. In dem Szenario wird von einem Desaster auf einer großen und komplexen Anlage, beispielsweise einem internationalen Flughafen, wie z. B. dem Frankfurter Flughafen (siehe Abbildung 1.1), ausgegangen. Ebenso wäre ein Unglück in einer chemischen Fabrikationsanlage denkbar. Es soll hierbei angenommen werden, dass es zu einem oder mehreren Bränden kommt, durch die auch Gebäude auf dem Gelände betroffen sind.

Innerhalb dieses Szenarios ist die Brandbekämpfung schwierig zu organisieren und durchzuführen. Desaster in großen Gebäudekomplexen, wie beispielsweise Abfertigungshallen eines Flughafens, sind für Feuerwehrleute besonders gefährlich. Innerhalb solcher Gebäude können noch zusätzliche Gefahren und Probleme auftreten, die bei einem Einsatz im Freien nicht oder nur z. T. zum Tragen kommen, wie u. a. Sichtbehinderung durch Rauch, eingestürzte Gebäudeteile oder umherliegende Gegenstände, die das Vorankommen der Rettungskräfte behindern. Außerdem ist es in dieser Umgebung für die Einsatzleitung schwierig, die Übersicht über das Geschehen zu behalten.

Die in dem Szenario beschriebenen Einrichtungen verfügen in der Regel über eine Werksfeuerwehr. Dieses ist insofern von Bedeutung, da Werksfeuerwehren über eine bessere finanzielle und technische Ausstattung verfügen, so dass hier das Interesse an technischen Lösungen, wie sie in dieser Arbeit angestrebt werden, größer ist, besonders in Zusammenhang mit den Gefahren und dem wirtschaftlichen Schaden, der durch ein Unglück auf einer



Abbildung 1.1: Frankfurter Flughafen: Komplexes System aus Gebäuden und Abfertigungshallen [Google Earth]

diesem Szenario zu Grunde gelegten Einrichtungen entstehen würde. Außerdem ist die Ausrüstung eines Gebäudes mit Sensorknoten nur in begrenztem räumlichen Umfang durchführbar, und z. B. für die Feuerwehr einer Stadt nicht so interessant. Ausnahmen können jedoch besonders schützenswerte oder gefährdete Objekte bilden.

1.2 Motivation

Innerhalb des definierten Szenarios ist es das Ziel dieser Arbeit, die Feuerwehr bei ihrer gefährlichen und schwierigen Arbeit zu unterstützen. Es sollen den Feuerwehrkräften vor Ort, aber insbesondere den Einsatzleitern in der Leitstelle, zusätzliche Informationen und Daten zur Verfügung gestellt werden, damit sich die Brandbekämpfung effektiver planen lässt und Risiken und Gefahren früher erkannt werden können.

In der Einsatzleitzentrale existieren zwar in der Regel Raumpläne für die in diesem Szenario betroffenen Gebäude, es sind aber kaum aktuelle Daten aus dem Gebäude verfügbar. So sind oft keine Informationen über die tatsächliche Ausbreitung eines Feuers bekannt, sei es bezogen auf die betroffenen Bereiche als auch genauere räumliche Eingrenzungen, die z. B. innerhalb einer großen Empfangshalle nötig sind. Auch aktuelle Temperaturwerte innerhalb eines Gebäudes wären interessant, um Aussagen über den Verlauf und die Ausbreitungsgeschwindigkeit eines Feuers bzw. über gefährdete Abschnitte treffen zu können. Bei Bränden in chemischen Produktionsanlagen ist es von besonderer Bedeutung, genaue Daten über

die chemische Zusammensetzung eines Produktionsstoffes und den Lagerort zu kennen, da dies für die Art der Brandbekämpfung entscheidend ist. Diese Beispiele zeigen, dass es notwendig ist, so viele Informationen wie möglich über den Einsatzort zu erfassen, um die Brandbekämpfung zu verbessern und die Risiken für die Rettungskräfte zu minimieren.

Ein weiteres Problem für die Rettungskräfte sind enge und verwinkelte oder auch unübersichtliche Bereiche innerhalb von Gebäuden. Dadurch wird ihnen zum einen die Arbeit erschwert, viel gefährlicher ist es zum anderen jedoch, dass die Feuerwehrleute sich dadurch leicht verirren oder die Orientierung verlieren können. Dies stellt eine große Gefahr für die Einsatzkräfte dar, insbesondere da sie sich nur eine sehr begrenzte Zeit in dem Gebäude aufhalten dürfen, beispielsweise auf Grund der ihnen zur Verfügung stehenden Pressatemluft. Diese Gefahr kann durch eine Sichtbehinderung, die durch Rauch entsteht, noch verstärkt werden.

Das Ziel dieser Arbeit soll es sein, durch den Einsatz von technischen Hilfsmitteln die Risiken für die Brandbekämpfer vor Ort zu minimieren, ihre Arbeit zu erleichtern und ihnen die Möglichkeit zu geben, sich stärker auf die eigentliche Brandbekämpfung zu konzentrieren. Dazu sollen den Rettungskräften bzw. der Leitstelle aktuelle Daten und Informationen aus den betroffenen Gebäuden zur Verfügung gestellt werden. Es soll auf technische Hilfsmittel zurückgegriffen werden um diese Werte zu erfassen, da es viel präziser und ungefährlicher für die Retter ist, die Daten durch Sensoren automatisiert zu erfassen und diese an eine zentrale Leitstelle in sicherer Entfernung weiterzuleiten. Dort können die Daten ausgewertet und es kann entsprechend reagiert werden. Es muss allerdings sichergestellt sein, dass die in dieser Arbeit entwickelten Hilfsmittel möglichst intuitiv zu bedienen sind und die Feuerwehrleute, so wenig wie möglich beansprucht und abgelenkt werden.

1.3 HAW Rescue-Projekt

Innerhalb des Masterstudiums an der Hochschule für Angewandte Wissenschaften Hamburg (HAW) wurden in verschiedenen Veranstaltungen bereits erste Schritte und Überlegungen innerhalb des in Abschnitt 1.1 dargestellten Szenarios unternommen (vgl. [Davids \(2006\)](#) und [Davids \(2007\)](#)). Im Rahmen dieser Arbeiten, an der mehrere Studenten der HAW beteiligt waren, wurden bereits erste Ergebnisse erzielt, die auch in diese Masterarbeit mit einfließen werden.

Abbildung 1.2 zeigt eine Darstellung der Themenbereiche des HAW Rescue-Projekts und der dafür verantwortlichen Personen. Der Bereich des Leitstands, in dem ein Rettungseinsatz koordiniert wird und die dort stattfindende Kollaboration der Einsatzleiter wird von Andreas

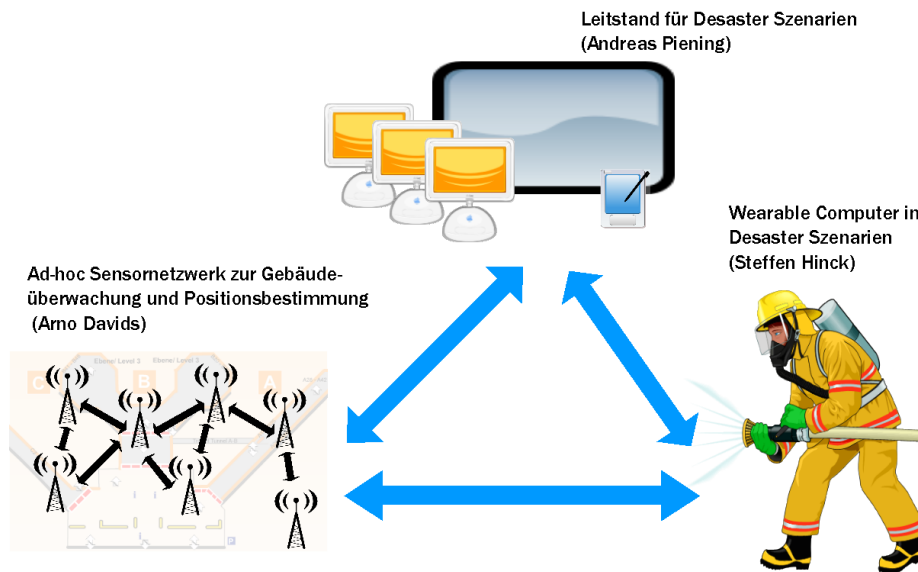


Abbildung 1.2: Rescue Projekt an der HAW Hamburg

Piening untersucht. Steffen Hinck befasst sich mit der Ausrüstung des einzelnen Feuerwehrmanns und der Anwendung von technischen Hilfsmitteln, wie z. B. einem Head-Mounted-Display, die die Rettungskräfte bei der Arbeit unterstützen können. In [Piening \(2006\)](#), [Piening \(2007\)](#), [Hinck \(2006\)](#) und [Hinck \(2007\)](#) werden die Arbeiten in diesen Bereichen des HAW Rescue-Projekts erläutert.

1.4 Gliederung der Arbeit

Nachdem eine Einführung in den Themenbereich dieser Arbeit und eine Einordnung innerhalb des Rescue-Projekts der HAW Hamburg gegeben wurde, sollen im nächsten Kapitel die Ziele und Vorhaben dieser Arbeit erläutert werden. Im dritten Kapitel werden die Grundlagen dieser Arbeit vorgestellt. Es werden u. a. einige Sensorplattformen, die für die Realisierung des Sensornetzes in Betracht kommen, gegenübergestellt und analysiert. Daran anschließend soll im nächsten Kapitel TinyOS vorgestellt werden. Dabei handelt es sich um ein Betriebssystem für Sensorknoten, das auf vielen Plattformen einsetzbar ist. Im Kapitel [Current Work](#) werden einige aktuelle Projekte vorgestellt, die sich ebenfalls mit dem Thema Sensornetze und Rescue befassen. Es erfolgt jeweils eine Bewertung inwieweit es Überschneidungen zu dieser Arbeit gibt und es werden Anregungen, die mit in diese Arbeit einfließen, aufgegriffen. Das sechste Kapitel befasst sich mit der Anforderungsanalyse und dem Entwurf der zu realisierenden Anwendungen und im folgenden Kapitel werden Realisierung und Ergebnisse präsentiert. Es wird dargestellt inwieweit die Ziele dieser Arbeit realisiert werden

konnten. Im Kapitel *Schlussbetrachtungen* wird das erreichte zusammenfassend dargestellt und daran anschließend ein Ausblick gegeben. Es werden Verbesserungsvorschläge und alternative Anwendungsformen untersucht.

2 Ziele und Vorhaben

Das übergeordnete Ziel dieser Masterarbeit ist es, Lösungen zu entwickeln, die die Rettungskräfte bei ihrer Arbeit unterstützen. Es sollen den Feuerwehrleuten vor Ort, aber besonders auch der Einsatzleitstelle zusätzliche Information und Daten zur Verfügung gestellt werden, damit Risiken für die Feuerwehrmänner minimiert und die Brandbekämpfung besser organisiert und koordiniert werden kann.

Dazu sollen zunächst einige Ziele mit Hilfe theseartiger Überschriften formuliert werden. Diese Überschriften fassen Themenbereiche zusammen, die in dieser Arbeit untersucht werden sollen. Es lassen sich insgesamt vier Bereiche identifizieren, die z. T. aufeinander aufbauen:

- „Sensornetz aus stationären Sensorknoten“
- „Rettungskräfte als mobile Knoten“
- „Sensorkette als Lifeline-Ersatz“
- „Container für den Sensorauswurf“

Die ersten beiden Thesen umschreiben dabei die Ziele, die in dieser Arbeit in einer Testumgebung auch praktisch realisiert werden sollen. Es soll somit eine Anwendung entstehen, die die dort genannten Ziele demonstriert. Die letzten beiden Thesen beschreiben ergänzende Bereiche und Anwendungsgebiete, die auf die vorherigen Bereiche aufbauen. Sie können hier auf Grund der Komplexität jedoch nur theoretisch betrachtet werden.

Im Folgenden werden die einzelnen Punkte im Detail beschrieben.

2.1 „Sensornetz aus stationären Sensorknoten“

Um den Grundgedanken, das Erfassen von Daten innerhalb eines Gebäudes, zu realisieren, soll ein drahtloses Sensornetzwerk aufgebaut werden, das flächendeckend Sensorwerte ermittelt. Es können innerhalb des Sensornetzes unterschiedliche Sensortypen genutzt werden, um Umgebungsdaten zu erfassen. In dieser Arbeit soll exemplarisch ein Temperatursensor eingesetzt werden. Dadurch können Feuer in einem Gebäude automatisch erkannt

und an die Leitstelle gemeldet werden und die Ausdehnung des Brandes, sowie die Temperaturwerte in den einzelnen Bereichen des Gebäudes erfasst und auf einem Gebäudeplan in der Leitstelle dargestellt werden. Die Sensorknoten, die die Temperaturwerte messen, werden an definierten Stellen im Gebäude angebracht. So ist es denkbar, sie in die bereits vorhandenen Rauchmelder zu integrieren bzw. die konventionellen Rauchmelder durch diese zu ersetzen.

Die Temperaturwerte innerhalb des Gebäudes sollen periodisch erfasst werden. Dabei soll die Periode, die zwischen zwei Messungen liegt, angepasst werden an den gemessenen Wert, so dass bei erhöhten Umgebungstemperaturen, die auf einen Brand hindeuten, der Zeitraum zwischen zwei Messungen verkürzt wird, um schneller reagieren zu können. Bei „normalen“ Werten, kann dagegen mit einer großen Periode gemessen werden, so dass die Sensorknoten lange Schlafphasen einlegen können. Dadurch wird erreicht, dass die Sensorknoten im Normalzustand, der die meiste Zeit ausmacht, Energie sparen können und damit die Lebenszeit ohne Batteriewechsel verlängert wird.

2.2 „Rettungskräfte als mobile Knoten“

Zusätzlich zu den fest eingebauten Sensorknoten, sollen auch die Feuerwehrmänner, die sich im Gebäude im Einsatz befinden, mit einem Sensorknoten ausgestattet werden, der sich in das vorhandene Sensornetz integriert. Damit kann die Überwachung und Datenerfassung innerhalb des Gebäudes auf die Feuerwehrräfte ausgedehnt werden und es werden viele Vorteile erzielt, die sowohl für die Leitstelle, als auch für die Rettungskräfte von großem Nutzen sind.

Mit Hilfe der fest im Gebäude installiert Sensoren, deren Position wie bereits geschildert bekannt ist, besteht die Möglichkeit, die Position der Feuerwehrmänner zu berechnen und damit in der Leitstelle die Position der Feuerwehrräfte innerhalb des Gebäudes visuell darzustellen. Der Leitstand erhält einen besseren Überblick über die Zahl und den genauen Ort der eingesetzten Feuerwehrmänner und kann so gegebenenfalls weitere Rettungskräfte an einen besonders gefährdeten Ort entsenden oder zusätzliche Einsatzwagen anfordern, falls ein Überlastung der bisher eingesetzten Kräfte erkennbar wird. Außerdem erhöht eine solche visuelle Darstellung die Übersicht und kann Systeme ersetzen, in denen diese bisher manuell durch die Einsatzleitung erstellt wird.

Durch die Ausstattung der Rettungskräfte mit Sensorknoten, besteht die Möglichkeit unterschiedliche Sensorwerte zu erfassen und weiterzuleiten und damit in der Leitstelle die Situation in der direkten Umgebung des Feuerwehrmanns transparent zu machen. So ist denkbar

die Vitaldaten der Feuerwehrmänner aufzuzeichnen, um sie besser vor Gefahren zu schützen und bei zu hoher körperlicher Belastung reagieren zu können. Ebenso wäre es interessant Informationen über die noch vorhandene Pressatmosphäre an die Leitstelle zu übertragen. Wird im Leitstand eine Gefährdung angezeigt, können die Rettungskräfte angewiesen werden sich zurückzuziehen und ausgetauscht werden. Es wird somit in der Einsatzleitung zusätzlich die Übersicht verbessert und die Informationsdichte erhöht.

Um die Überwachung der Rettungskräfte durch Sensoren demonstrieren zu können, sollen die Temperaturwerte in der direkten Umgebung aufgezeichnet und im Leitstand dargestellt werden. Die anderen genannte Überwachungsfunktionen können jedoch aus Zeitgründen nicht praktisch realisiert werden, lassen sich jedoch mit entsprechenden Sensoren in das in dieser Arbeit realisierte System integrieren.

Dadurch, dass der einzelne Feuerwehrmann mit Hilfe des an seinem Körper angebrachten Sensorknotens, Teil des Netzes ist, besteht außerdem die Möglichkeit eine Datenübertragung zwischen Feuerwehrmann und Leitstand aufzubauen. So wäre es denkbar Bilder auf einem Head-Mounted-Display, das in die Atemschutzmaske des Feuerwehrmanns eingebaut ist, anzuzeigen oder Fotos von einer Helmkamera an den Leitstand zu senden. Dies wird jedoch in der Masterarbeit von Steffen Hinck näher untersucht werden und ist daher nicht Bestandteil dieser Arbeit.

2.3 „Sensorkette als Lifeline-Ersatz“

Nachdem in den beiden zuvor präsentierten Abschnitten die Vorhaben dargestellt wurden, die in dieser Arbeit auch praktisch realisiert werden sollen, werden im folgenden ergänzenden Ideen dazu betrachtet. Diese passen inhaltlich in den hier beschriebenen Themenkomplex und bieten sich als Erweiterung zu dem in 2.1 und 2.2 dargelegten an. Aus diesem Grund sollen die Vorhaben an dieser Stelle vorgestellt werden. Eine Einschätzung in wie weit sie sich umsetzen lassen, soll im [Ausblick](#) abgegeben werden.

Ergänzend zu dem bisher beschriebenen besteht die Möglichkeit, zusätzliche Sensorknoten innerhalb des Gebäudes durch die Rettungskräfte auszubringen. Es könnten von den Feuerwehrleuten automatisch in regelmäßigen Abständen Sensorknoten ausgelegt werden, so dass eine Sensorkette entlang des zurückgelegten Wegs vom Eingang des Gebäudes bis zur aktuellen Position des Feuerwehrmannes entsteht.

Die beschriebene Sensorkette dient in erster Linie dazu, die Sicherheit der Rettungskräfte im Gebäude zu erhöhen. Sie kann genutzt werden, um einen Rückweg aus dem Gebäude zu bestimmen. Feuerwehrleuten, die sich im Gebäude verirrt haben, kann anhand der umgekehrten Reihenfolge der ausgelegten Sensoren ein Fluchtweg aus dem Gebäude berechnet

werden. Es soll somit über die Sensorknoten eine Route ermittelt und darüber navigiert werden. Diese Route kann dem Feuerwehrmann z. B. durch Richtungspfeile auf einem Head-Mounted-Display angezeigt werden. Alternativ wäre es aber auch möglich, sie ihm über sein Sprechfunkgerät akustisch mitzuteilen. Die Sensorkette kann somit als Ergänzung bzw. als Ersatz für die bei der Feuerwehr genutzte Lifeline dienen.

Die Lifeline besteht aus einem Drahtseil, das die Feuerwehrleute in einem Gebäude hinter sich herführen, um es damit auf demselben Weg, auf dem es betreten wurde, wieder verlassen zu können. Sie wird besonders dann wichtig, wenn es durch Rauchentwicklung zu einer Sichtbehinderung im Gebäude kommt und der Ausgang nicht mehr erkennbar ist. Die Lifeline in ihrer bisherigen Form hat aber den großen Nachteil, dass beim Rückzug exakt derselbe Pfad wie auf dem Hinweg verfolgt werden muss und so z. B. Pfeiler auf derselben Seite wieder umrundet werden müssen. Außerdem kann ein Teil des Weges verschüttet sein oder durch Feuer unpassierbar werden, so dass die Lifeline damit nutzlos werden würde. Wenn die Lifeline durch die oben beschriebene Sensorkette ersetzt wird, können zwar Probleme wie die Unpassierbarkeit des vorgeschlagenen Weges ebenfalls die Navigation erschweren, es wäre aber beispielsweise möglich, dass der Feuerwehrmann sich selbständig einen alternativen Weg um das Hindernis herum sucht und danach wieder der berechneten Route folgt.

Ein weiterer Vorteil, der sich durch das zusätzliche Ausbringen von Sensorknoten durch die Rettungskräfte ergibt ist, dass damit die Feuerwehrmänner besser in das im Gebäude vorhandene stationäre Sensornetz eingebunden sind. Es entstehen durch die ausgelegten Sensorknoten zusätzliche Verbindungsmöglichkeiten zum im Gebäude vorhandenen Sensornetz und daher auch zur Leitstelle. Durch die höhere Knotendichte kann auch der eventuelle Ausfall von stationären Knoten besser kompensiert werden. Außerdem können die ausgelegten Sensorknoten mit Sensoren beispielsweise zur Temperaturüberwachung ausgestattet werden, um so ein besseres Monitoring eines möglichen Fluchtweges durchführen zu können.

Die Routenbestimmung kann zusätzlich optimiert werden, wenn auch die Sensorketten, die von anderen Feuerwehrleuten ausgelegt wurden, mit in die Berechnung einbezogen werden. Dadurch könnte, wenn die Route eines Feuerwehrmannes blockiert ist, eine neue Route über die Sensorknoten anderer Feuerwehrmänner bestimmt werden. Dies wäre dann möglich, wenn zwei Routen sehr nahe beieinander liegen, so dass darauf geschlossen werden kann, dass sich keine Mauern zwischen den Routen befinden.

2.4 „Container für den Sensorauswurf“

Als Ergänzung zu dem bisher beschriebenen System aus fest installierten Sensoren und mobilen Knoten, die die Feuerwehrleute bei sich tragen bzw. von ihnen ausgelegt werden,

ist denkbar dieses System um Container zu erweitern, durch die sich im Brandfall zusätzliche Sensoren im Gebäude ausbringen lassen. Hiermit sind Behälter mit einer Vielzahl von Sensorknoten gemeint, die sich bereits im Gebäude befinden können oder aber erst durch Feuerwehrleute in das Gebäude gebracht werden. Diese könnten bei einem Feuer aktiviert werden und ihre Sensorknoten in einem bestimmten Umkreis um die Station ausbringen.

Die Behälter können beispielsweise in besonders wichtigen Bereichen aufgestellt werden, um diese bei einem Brand noch intensiver und detaillierter überwachen zu können oder aber den Ausfall von festen Knoten damit zumindest teilweise kompensieren zu können. Außerdem wäre es möglich ein Gebäude nachträglich mit Sensoren zur Datenerfassung auszustatten. Durch den Einsatz solcher Container könnte ein Gebäude mit nur wenigen stationären Sensoren ausgestattet werden und nur bei Bedarf würde ein engmaschiges Überwachungsnetz aktiviert.

Zusammenfassung

In dieser Arbeit sollen die in den Abschnitten [2.1](#) und [2.2](#) beschriebenen Vorhaben praktisch umgesetzt werden. Zusammenfassend betrachtet wird ein Sensornetz aus stationären Knoten in einem Gebäude aufgebaut und zunächst dazu genutzt, das Gebäude mit Hilfe von Sensoren zu überwachen. Hierzu wird exemplarisch ein Temperatursensor eingesetzt. Weiterhin soll eine Positionsbestimmung innerhalb des Sensornetzes durchgeführt werden. Dazu wird die Position von Sensorknoten, die die Rettungskräfte an ihrer Ausrüstung tragen, ermittelt. Die Positionsdaten sollen zusammen mit den Sensorwerten an eine Leitstand-Applikation übertragen und dort dargestellt werden.

Auf die in den Abschnitten [2.3](#) und [2.4](#) dargestellten Ideen wird im [Ausblick](#) näher eingegangen.

3 Grundlagen und Analyse

In diesem Kapitel werden zunächst die für diese Arbeit wichtigen Grundlagen betrachtet. Begonnen wird mit einer Einordnung des Begriffs „drahtloses Sensornetzwerk“. Anschließend werden unterschiedliche Kommunikationstechniken, die für die drahtlosen Datenübertragung in Frage kommen, dargestellt und analysierend verglichen. Außerdem soll die Lokalisation betrachtet werden, die einen wesentlichen Punkt für die in dieser Arbeit angestrebten Vorhaben darstellt. Darauf aufbauend werden einige Sensorplattformen, die für die Realisierung des Sensornetzes genutzt werden können, verglichen und ihre Eignung bewertet.

3.1 Drahtloses Sensornetzwerk

Da in dieser Arbeit bereits mehrfach der Begriff „drahtloses Sensornetzwerk“ genannt wurde, soll zunächst näher erläutert werden, anhand welcher Kriterien sich ein drahtloses Sensornetzwerk (Wireless Sensor Network) definiert.

Ein Sensornetzwerk soll in dieser Arbeit, in dem zugrunde liegenden Szenario, eingesetzt werden, um ein Gebäude bzw. die sich darin befindenden Rettungskräfte mit Hilfe verschiedener Sensorknoten zu überwachen. Dazu werden Umgebungsdaten, wie beispielsweise Temperaturwerte, automatisch erfasst. Obwohl die Sensorknoten selber relativ simple Aufgaben übernehmen (Sensorwerte erfassen, Vorauswertung und Weiterleitung der Daten) ergibt sich die Funktionalität des Gesamtsystems aus dem Zusammenspiel der Einzelkomponenten.

Im Folgenden werden die Anforderungen, die an ein Sensornetz und an die einzelnen Sensorknoten gestellt werden und die sich daraus ergebenden Konsequenzen in Bezug auf die Eigenschaften erläutert (vgl. [\[Mattern und Römer \(2003\)\]](#)). Für Sensorknoten, bei denen ein Batteriebetrieb als grundsätzlich gegeben vorausgesetzt wird, sind wichtige Kriterien für die Entwicklung, dass sie möglichst klein, robust und günstig in der Herstellung sind und dass sie eine lange Betriebsdauer haben.

Durch den Anspruch der geringen Größe und der maximalen Betriebsdauer ergibt sich auch die Anforderung eines geringen Energieverbrauchs, um wenig Platz für Batterien zu beanspruchen. Es ist somit entscheidend, dass die Sensorknoten schonend mit den Ressourcen

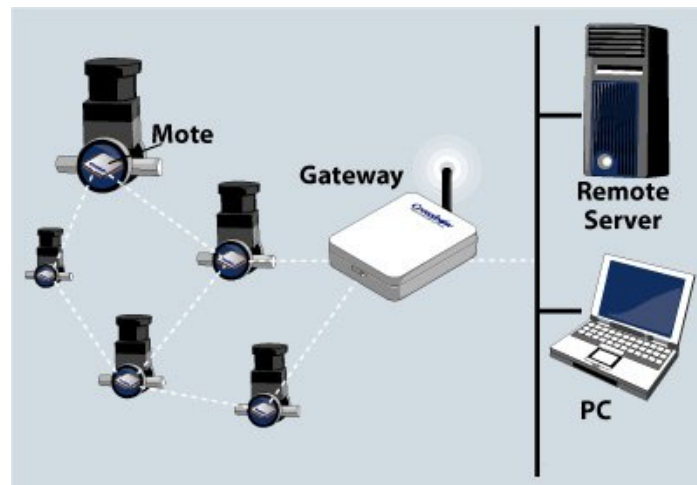


Abbildung 3.1: Möglicher Aufbau einer Sensornetzwerk-Architektur [<http://www.xbow.com>]

umgehen. Da bei den gängigen Sensorknoten, wie sie auch in Abschnitt 3.4 vorgestellt werden, die meiste Energie für die Kommunikation nötig ist, ist die Steuerung des Funkmoduls und damit die Software bzw. das Betriebssystem von großer Bedeutung, um einen minimalen Energieverbrauch zu erzielen. Um den Energieverbrauch zu reduzieren ist es erforderlich, dass sich die Sensorknoten den größten Teil der Zeit in einem Schlafzustand befinden und nur zyklisch aufwachen, um beispielsweise Sensorwerte zu erfassen und weiterzuleiten. Dies stellt aber im Allgemeinen kein Problem dar, da es z. B. oft nicht nötig ist, permanent einen Sensorwert zu erfassen, sondern es ausreichend ist, dass dies periodisch erfolgt. Auch für den in dieser Arbeit geplanten Einsatzzweck innerhalb des Sensornetzes stellt es kein Problem dar, dass die Knoten nur von Zeit zu Zeit aktiv werden, wobei im Normalbetrieb, wenn kein Feuer ausgebrochen ist, längere Inaktivitätsphasen möglich sind. Ist dagegen ein Feuerwehreinsatz nötig, sollten die Schlafphasen kürzer sein, um eine schnellere Reaktion zu gewährleisten.

Um die Herstellungskosten der Sensorknoten gering zu halten, werden die verfügbaren Knoten aus Standardkomponenten gefertigt. Diese werden auch als COTS bezeichnet (COTS = commercial off-the-shelf). Da jedoch nur geringe Stückzahlen hergestellt werden, betragen beispielsweise die Kosten je TelosB-Sensorknoten noch 110 \$. Würden die Sensorknoten, z. B. integriert in Rauchmelder, in großen Stückzahlen hergestellt, sind jedoch wesentlich geringere Kosten zu erwarten, da die einzelnen Hardwarekomponenten zusammen wesentlich günstiger sind. Ein anderer Weg wurde im Smart Dust-Projekt gegangen, bei dem nicht auf Standardkomponenten zurückgegriffen wurde (vgl. Abschnitt 5.1). Dort war jedoch ein hoher Aufwand in Forschung und Entwicklung nötig, um die Sensorknoten zu erstellen.

Auch für das Sensornetz als Gesamtheit ergeben sich einige Anforderungen. Es besteht

der Anspruch, ebenso wie bei den Sensorknoten, die Lebenszeit des Netzes zu maximieren, so dass das Sensornetz in seiner größt möglichen Ausdehnung erhalten bleibt. Damit dieses Ziel erreicht werden kann, muss der Energieverbrauch der einzelnen Sensorknoten möglichst gleich verteilt sein.

Ein weiterer Gesichtspunkt ist die Verlässlichkeit und die Fehlertoleranz auch bei Ausfall einzelner Knoten. Da es sich um ein sehr dynamisches System handelt, bei dem neue Knoten hinzukommen können, Knoten ausfallen oder sich mobile Knoten im Netz bewegen, muss sichergestellt sein, dass diese Ereignisse nicht zum Ausfall oder zur Fehlfunktion des Gesamtsystems führen. Außerdem müssen sich neue Knoten selbstständig in das System einfügen, sich somit spontan vernetzen können (Ad-Hoc-Sensornetz).

Da das Sensornetz aus wenigen ebenso wie aus mehreren tausend Knoten bestehen kann, stellt die Skalierbarkeit eine wichtige Anforderung an das System dar. Es muss sichergestellt sein, dass es auch bei hoher Last beispielsweise noch möglich ist, Nachrichten zwischen den Knoten auszutauschen, ohne dass es zu Kollisionen kommt.

Die genannten Anforderungen sollen bei der Auswahl der Sensorknoten (siehe Abschnitt 3.4) und beim Aufbau des Sensornetzes so gut wie möglich umgesetzt werden. Dabei muss aber beachtet werden, dass sich die Anforderungen teilweise widersprechen können, so sind etwa sehr kleine Sensorknoten, teurer als größere. Da die Sensorknoten einen Großteil ihrer Energie für das Senden und Empfangen von Nachrichten benötigen, sollen im folgenden Abschnitt zunächst verschiedene Kommunikationstechniken diskutiert werden, um die Ergebnisse bei der Auswahl der Sensorknoten berücksichtigen zu können.

3.2 Sensorkommunikation

Im diesem Abschnitt werden Verfahren und Protokolle vorgestellt, um einen Datenaustausch zwischen den Sensorknoten zu ermöglichen und damit eine Grundvoraussetzung für ein drahtloses Sensornetz zu erfüllen. Dabei wird im folgenden vor allem auf Verfahren eingegangen, die auf dem Prinzip der Funkübertragung basieren. Dieses ist das gängigste Verfahren zum drahtlosen Nachrichtenaustausch in der Informatik, dass auch von den meisten verfügbaren Sensorplattformen zur Kommunikation eingesetzt wird. Außerdem können Funkmodule z. T. auch zur Positionsbestimmung genutzt werden. Darauf soll jedoch später detaillierter eingegangen werden.

Andere Kommunikationstechniken

Neben der Funkübertragung gibt es noch andere Verfahren, wie Ultraschall oder die optische Kommunikation. Ultraschall wird in verschiedenen Systemen vor allem zur Positionsbe-

stimmung eingesetzt, es sei hier exemplarisch auf das Cricket Indoor Location System verwiesen (vgl. Priyantha u. a. (2000)). An der HAW Hamburg wurde von Sebastian Gregor das IMAPS-System (Indoor Distance Measurement And Positioning System) entwickelt, das auf Cricket basiert (vgl. Gregor (2006)). Das IMAPS-System wurde in unterschiedlichen Arbeiten an der HAW¹ genutzt und weiterentwickelt, soll aber in dieser Arbeit nicht weiter behandelt werden. Obwohl Cricket Ultraschall nur zur Positionsermittlung einsetzt, kann es theoretisch auch zur Datenübermittlung herangezogen werden. Auch eine optische Signalübertragung kann zur Kommunikation zwischen Sensorknoten genutzt werden. Ansätze in diese Richtung wurden im Smart Dust-Projekt (siehe Abschnitt 5.1) gemacht.

Sowohl Ultraschall als auch eine optische Datenübertragung werden in dieser Arbeit jedoch nicht weiter als Kommunikationsform untersucht, da beide einen sehr gravierenden Nachteil gegenüber der Funkübertragung besitzen. Es ist bei beiden eine direkte Sichtverbindung zwischen Sender und Empfänger für einen reibungslosen Betrieb nötig. Diese Einschränkung ist besonders in der geplanten Einsatzumgebung innerhalb des Rescue-Szenarios problematisch, da dort u. a. durch Rauch und Staubpartikel vor allem die optische Kommunikation erschwert wird, aber auch eine korrekte Ausrichtung eines z. B. am Feuerwehrmann angebrachten Ultraschallmoduls kann während eines Einsatzes nicht gewährleistet werden.

Die am meisten eingesetzte und sicherlich am nächsten liegende Technik zur Kommunikation ist die Funkübertragung. Im folgenden Abschnitt werden einige Verfahren zur drahtlosen Kommunikation vorgestellt. Durch die Auswahl der hier vorgestellten Verfahren wird jeweils eine andere Klasse in Bezug auf die Reichweite und die Datenrate repräsentiert (vgl. Abbildung 3.2). Bei allen hier vorgestellten Kommunikationstechniken handelt es sich um standardisierte Protokolle.

Es werden Verfahren untersucht, mit denen eine Kommunikation innerhalb von Gebäuden möglich ist. Die beiden zuerst genannten Protokolle IEEE 802.15.4/ ZigBee und WLAN werden vor allen in Anwendungen in Gebäuden eingesetzt. GSM-Mobilfunk, das als letztes vorgestellt wird, ist dagegen nur eingeschränkt im Gebäude möglich. Es soll jedoch dennoch betrachtet werden, da hiermit große Distanzen überbrückt werden können und es ist in bestimmten Anwendungsfällen, wenn ein permanenter Empfang garantiert werden kann, eine Option.

3.2.1 IEEE 802.15.4 und ZigBee

Der IEEE 802.15.4-Standard (vgl. [IEEE (2003b)]) beschreibt ein Protokoll zum Aufbau von drahtlosen Nahbereichsnetzwerken (LR-WPAN: Low Rate - Wireless Personal Area Network) mit geringen Datenraten (s. a. [Fischer (2005)]). Der Standard zeichnet sich dadurch

¹Quellen: <http://users.informatik.haw-hamburg.de/~ubicomp/papers.html>

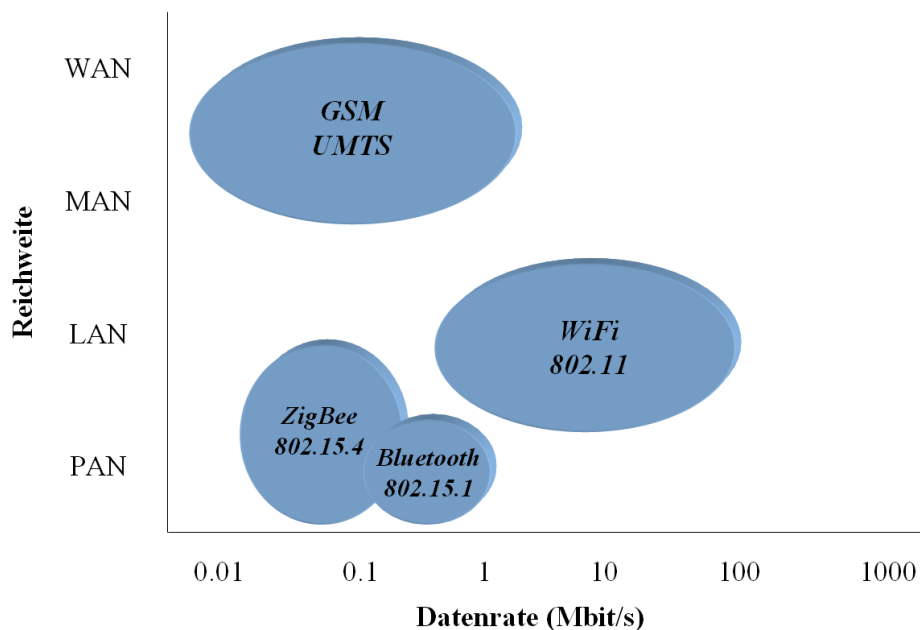


Abbildung 3.2: Unterschiedliche Kommunikationsverfahren im Vergleich

aus, dass Geräte mit limitierten Hardwareressourcen eine drahtlose Kommunikation mit einem niedrigen Stromverbrauch realisieren können. Damit eignen sich insbesondere tragbare Geräte, die batteriebetrieben sind, für die Anwendung einer Kommunikation nach IEEE 802.15.4. Des Weiteren ist der Standard ausgelegt auf Verbindungen mit geringer Reichweite und soll einen zuverlässigen Datentransfer durch ein einfaches und flexibles Protokoll ermöglichen.

Ein LR-WPAN nach IEEE 802.15.4-Protokoll zeichnet sich u. a. aus durch:

- Datenraten von bis zu 250 kbit/s
- geringe Latenzzeiten
- 16 Kanäle im 2,4 GHz-Bereich (ISM-Band)
- Stern und Peer-To-Peer Topologien
- Medienzugriff durch CSMA-CA

Aus dem OSI-Schichtenmodell (Open Systems Interconnection Reference Model) der ISO (International Organization for Standardization) werden innerhalb von IEEE 802.15.4 die untersten beiden Schichten definiert. Es wird die Bitübertragungsschicht (physical layer, PHY) voll implementiert und aus der Sicherungsschicht (data link layer) wird eine der beiden Subschichten, die MAC-Schicht (MAC = Medium Access Control) beschrieben.

Ein LR-WPAN nach IEEE 802.15.4 kann aus zwei verschiedenen Arten von Geräten bestehen. Beschrieben werden dort sowohl FFDs (full-function devices), Einheiten mit vollem Funktionsumfang, wie auch RFDs (reduced-function devices), die über eingeschränkte Fähigkeiten verfügen. RFDs sind sehr einfache Geräte, die nur wenige Daten senden und mit genau einem anderen Gerät kommunizieren müssen und daher nur minimale Anforderungen stellen. Ein RFD wäre z. B. ein Lichtschalter oder ein passiver Infrarotsensor. RFDs können nur mit FFDs kommunizieren, währenddessen FFDs sowohl mit RFDs als auch mit anderen FFDs Daten austauschen können. Ein Netzwerk aus mehreren Geräten muss damit mindestens ein FDD enthalten. In einem WPAN-Netz fungiert dabei ein FDD als PAN-Koordinator.

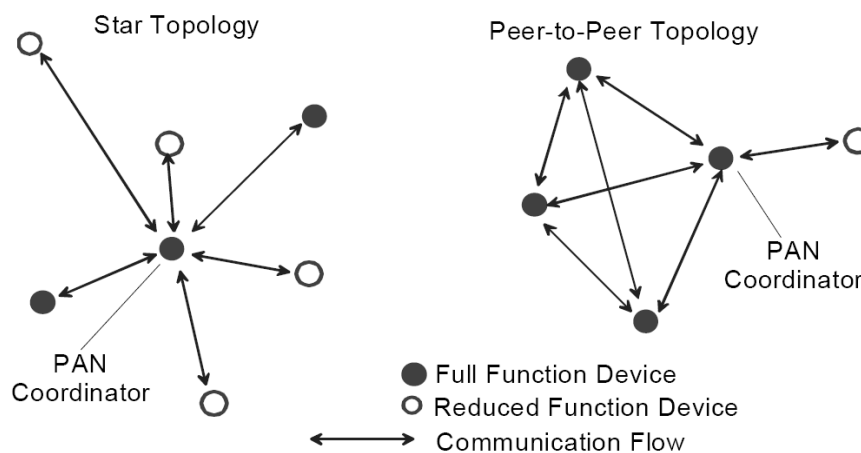


Abbildung 3.3: IEEE 802.15.4 Topologien [IEEE (2003b)]

Der PAN-Koordinator ist die zentrale Instanz in einem WPAN, es existiert in jedem Netz genau einer. Er verwaltet die Adressen der einzelnen Mitglieder. In der Stern-Topologie wird jede Kommunikation vom PAN-Koordinator initiiert bzw. beendet und alle Nachrichten laufen über ihn. Im Unterschied zur Stern-Topologie ist die Peer-To-Peer-Topologie wesentlich flexibler. Hier können alle Teilnehmer (hierbei muss es sich natürlich um FDDs handeln) mit allen anderen kommunizieren, die sich in ihrer Reichweite befinden. Nachrichten werden auf flexiblen Routen durch das Netz geleitet und müssen nicht in jedem Fall über den Koordinator laufen. Die Peer-To-Peer-Topologie erlaubt damit den Aufbau eines Ad-Hoc-Netzes, bei dem sich Nachrichten selbstständig einen Weg durch das WPAN suchen. Damit ist diese Topologie auch am besten geeignet für den Aufbau eines Sensornetzes.

Die Zugriffskontrolle auf das Übertragungsmedium, wurde in IEEE 802.15.4 durch den CSMA-CA-Modus (Carrier Sense Multiple Access-Collision Avoidance) realisiert. Außerdem können gesendete Pakete optional durch den Empfänger bestätigt und der Inhalt kann mit Hilfe einer CRC-Prüfsumme verifiziert werden. In einer Ad-Hoc-Topologie kommt das *unscheduled* CSMA-CA-Verfahren zum Einsatz. Hat ein Teilnehmer einen Sendewunsch, muss er zuvor eine zufällige Zeitspanne warten. Danach darf er, wenn das Medium nicht belegt ist,

senden. Sendet dagegen bereits ein anderes Gerät, muss er erneut eine zufällige Zeitspanne warten und kann danach wiederum prüfen, ob das Medium frei ist.

Eine Datenübertragung nach IEEE 802.15.4 findet in Europa vor allem im lizenzfreien ISM-Band bei 2,4 GHz statt. Dieser Bereich wird wir auch durch den Standard IEEE 802.11 (WLAN) (vgl. Abschnitt 3.2.2) genutzt und damit können beim Parallelbetrieb Interferenzen auftreten. Bei einer maximalen Sendeleistung von 10 mW sind Reichweiten von bis zu 100 m möglich. Diese können allerdings durch Hindernisse erheblich reduziert werden.

ZigBee

Die ZigBee-Technologie wurde von der ZigBee-Alliance², einem Zusammenschluss mehrerer Unternehmen, entwickelt. Mit ZigBee können Geräte kostengünstig und mit geringem Stromverbrauch drahtlos vernetzt werden.

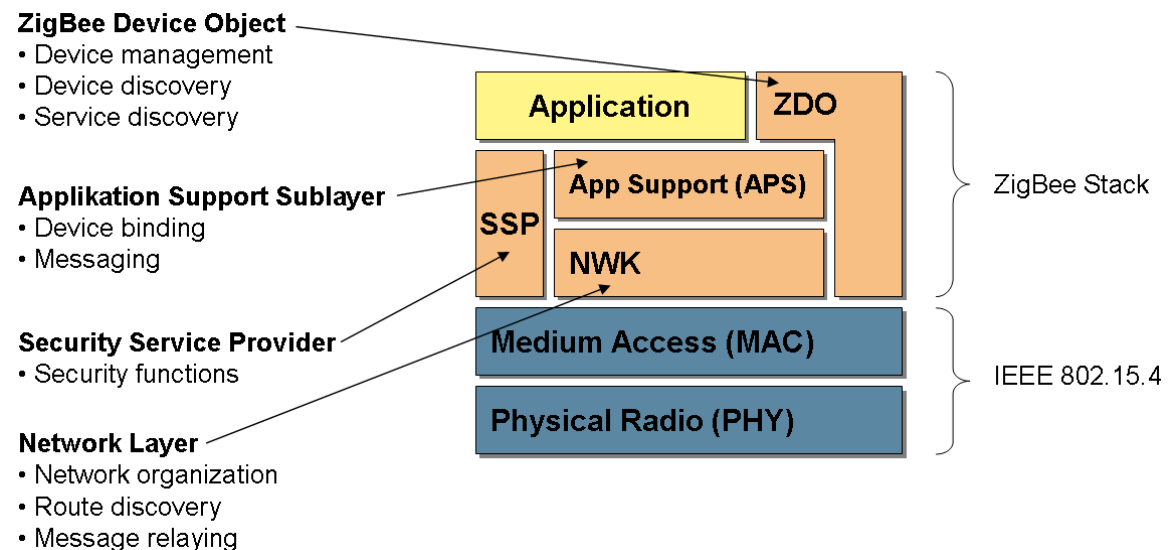


Abbildung 3.4: ZigBee Architektur [Quelle: ZigBee Alliance]

Da die ZigBee-Technologie auf dem IEEE-Standard 802.15.4 aufbaut, damit die PHY- und die MAC-Schicht von IEEE 802.15.4 erbt und um einen eigenen Stack für die höheren Schichten ergänzt, sind die genannten Eigenschaften des IEEE 802.15.4-Standards auch für ZigBee gültig. Abbildung 3.4 zeigt eine Darstellung der einzelnen Schichten des ZigBee-Standards mit einer kurzen Beschreibung ihrer jeweiligen Aufgaben.

ZigBee definiert verschiedene Profile, die die Eigenschaften und Funktionen eines Gerätes für einen bestimmten Anwendungsbereich beschreiben. Ein ZigBee-konformes Gerät kann einen oder mehrere dieser Profile unterstützen und muss alle im Profil definierten Funktionen

²<http://www.zigbee.org>

anbieten. Damit soll die Interoperabilität von Geräten verschiedener Hersteller sichergestellt werden.

Ein großer Vorteil von IEEE 802.15.4/ZigBee sind die geringen Latenzzeiten. Diese sind in Tabelle 3.1 aufgelistet.

Tabelle 3.1: Latenzzeiten von IEEE 802.15.4/ZigBee

| Aktion | Dauer |
|--|-----------|
| Beitritt zu einem Netz | ca. 30 ms |
| Wechsel zwischen Wach- und Schlafphase | ca. 15 ms |
| Zugriff eines aktiven Knotens | ca. 15 ms |

3.2.2 IEEE 802.11/ WLAN

WLAN (Wireless Local Area Network) ist eine drahtlose Netzwerktechnologie innerhalb der IEEE 802-Normenfamilie. WLAN ist in IEEE 802.11 (vgl. [IEEE (2003a)]) beschrieben und wurde entwickelt, um lokale drahtgebundene Netze, wie z. B. Ethernet, um eine Funkübertragung zu ergänzen. Dadurch wird es möglich Geräte drahtlos, in ein bestehendes Netzwerk einzubinden.

Der ursprüngliche Standard IEEE 802.11, in dem eine Datenrate von bis zu 2 MBit/s im lizenzfreien Frequenzband von 2,4 GHz (ISM-Band) vorgesehen war, wurde im Laufe der Zeit immer wieder erweitert und verbessert, um höhere Übertragungsraten zu erzielen. So besitzt heute IEEE 802.11g, das im Jahre 2003 verabschiedet wurde, die größte Verbreitung. Hier sind Datenraten von maximal 54 MBit/s im selben Frequenzband möglich. Es muss jedoch bei den genannten Datenraten beachtet werden, dass diese Werte in der Praxis nicht erzielt werden können, sondern selbst bei idealen Bedingungen nur etwa die Hälfte der genannten Bruttodatenrate erreicht wird. IEEE 802.11, das ebenso aus dem ursprüngliche Standard entwickelt wurde, nutzt jedoch ein anderes Frequenzband, das im Bereich von 5 GHz liegt.

Der Zugriff auf das Übertragungsmedium wird mit CSMA-CA realisiert. Die Sendeleistung im 2,4 GHz Bereich ist in Deutschland auf maximal 100 mW begrenzt. Die typische Sendeleistung liegt zwischen 20 und 40 mW. Im Freien lassen sich damit Reichweiten von ca. 100 m erzielen. Innerhalb von Gebäuden ist eine Reichweite von etwa 30 m möglich, in Abhängigkeit von der Art und dem Material von Hindernissen kann diese jedoch eingeschränkt sein.

Die Mittelpunktsfrequenzen der 13 in Europa zugelassenen Übertragungskanäle im Standard IEEE 802.11b/g/n befinden sich im Bereich von 2,412 GHz bis 2,472 GHz jeweils im Abstand von 5 MHz. Für jeden Kanal ist eine Bandbreite von ± 11 Mhz um den Mittelpunkt vorgesehen. Da der Abstand der Kanäle nur 5 Mhz beträgt, aber jeder Kanal eine Bandbreite von 22 Mhz belegt, kommt es damit zu Überlappungen der einzelnen Kanäle. Für eine störungsfreie Kommunikation bei mehreren sich überschneidenden Funkzellen, stehen damit nur drei vollständig getrennte Frequenzbereiche zur Verfügung (z. B. Kanal 1, 6 und 11).

3.2.3 Digitaler Mobilfunk (GSM)

Der digitale Mobilfunk ist in Europa fast flächendeckend im GSM-Standard möglich. Der GSM-Standard wurde 1992 in Deutschland eingeführt und ist aktuell die am meisten genutzte Mobilfunktechnik. GSM soll in Zukunft durch UMTS abgelöst werden, dass in Deutschland bereits in Großstädten verfügbar ist.

Ein GSM-Mobilfunknetz ist aus vielen einzelnen Funkzellen aufgebaut. Eine Funkzelle befindet sich rund um einen Mobilfunkmast. Dies ist die Basisstation (BTS - Base Transceiver Station), an der sich die Teilnehmer mit ihren mobilen Geräten anmelden. Jedes Mobiltelefon (MS - Mobile Station) bucht sich dabei an der nächstgelegenen Basisstation mit der größten Empfangsstärke ein. Bewegt sich der Teilnehmer während des Gesprächs von einer Funkzelle in die nächste wird die bestehende Verbindung von der bisherigen Basisstation zur neuen übergeben, ohne das die Verbindung dabei abbricht. Mobiltelefone, Basissationen und Steuerungseinheiten (Base Station Controller) bilden das Funksubsystem (BSS - Base Station Subsystem), eine der drei großen Einheiten eines GSM-Mobilfunknetzes. Die anderen beiden Bestandteile sind das Network and Switching Subsystem (NSS) und das Operation and Maintenance Center (OMC), auf die jedoch nicht weiter eingegangen werden soll.

Es gibt in Europa zwei unterschiedliche Frequenzbereiche auf denen ein GSM-Netz betrieben werden darf. GSM900 nutzt den Frequenzbereich von 890 MHz bis 960 MHz und GSM1800 ist im Abschnitt 1710 MHz bis 1880 MHz angesiedelt, wobei getrennte Bereiche für Up- und Downlink definiert sind.

Um das Übertragungsmedium möglichst gut auszunutzen, werden die Daten, die bei einer Verbindung zwischen Mobiltelefon und Basisstation gesendet werden, mit Hilfe einer Kombination zweier Multiplexverfahren übertragen. Es kommt sowohl das Frequenzmultiplexverfahren (FDMA - Frequency Division Multiple Access), als auch das Zeitmultiplexverfahren (TDMA - Time Division Multiple Access) zum Einsatz. Durch das FDMA-Verfahren wird der zur Verfügung stehende Frequenzbereich in jeweils 124 Kanäle für den Up- bzw. Downlink aufgeteilt (bei GSM900), wobei die Bandbreite der einzelnen Kanäle jeweils 200 kHz beträgt (siehe Abbildung 3.5). Für eine Verbindung zwischen Mobiltelefon und Basisstation

wird ein getrennter Sende- und Empfangskanal (Duplexbetrieb) genutzt. Dabei ist ein Kanalpaar immer um 45 MHz versetzt angeordnet. Außerdem können optional durch frequency hopping die Kanalpaare periodisch gewechselt werden, um Störungen zu minimieren. Zeitlich wird jeder der 248 Kanäle in acht Zeitschlitze (timeslots) aufgeteilt, hier wird also das TDMA-Verfahren angewandt. Jeder Zeitschlitz hat eine Länge von $577 \mu\text{s}$. Es kann in einem Zeitschlitz eine Übertragungsrate von $33,9 \text{ kbit/s}$ erzielt werden, wovon aber nur $24,7 \text{ kbit/s}$ auf die eigentlichen Nutzdaten entfallen.

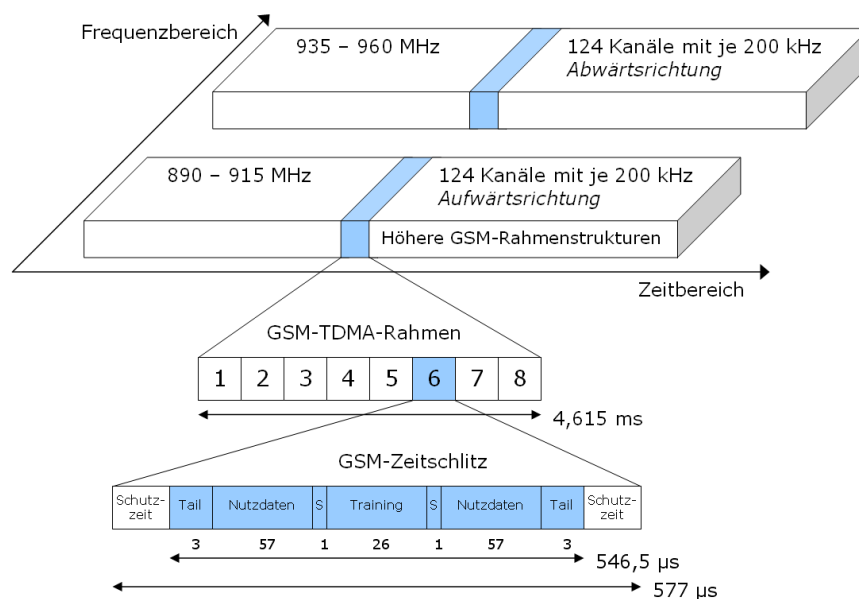


Abbildung 3.5: Aufbau der GSM Rahmenstruktur [<http://www.wikipedia.de/>]

Eine GSM Verbindung kann außerhalb von Gebäuden im Europa fast flächendeckend aufgebaut werden, hier können im Idealfall Entfernungen von bis zu 35 km zwischen Basisstation und Mobiltelefon überbrückt werden. Störungen können jedoch durch große Gebäude oder landschaftliche Gegebenheiten wie z. B. Berge auftreten. Innerhalb von Gebäuden ist eine Kommunikation teilweise möglich. Befindet man sich jedoch innerhalb eines größeren Gebäudes mit vielen bzw. starken Innenmauern, ist oft kein Verbindungsaufbau mehr möglich. Ob eine Kommunikation zustande kommt hängt dabei von vielen Faktoren ab, wie beispielsweise die Entfernung zur nächsten Basisstation, die Gebäudemauerstärke oder auch das Material aus dem das Gebäude errichtet wurde. Ein Nachteil des Mobilfunks innerhalb von Gebäuden ist es, dass eine Kommunikation zwischen zwei Geräten immer über eine Basisstation erfolgen muss und damit beide Teilnehmer über eine Verbindung aus dem Gebäude heraus verfügen müssen.

Eine Lokalisation in einem GSM-Netz ist, sobald das Telefon eingeschaltet ist und sich im Netz eingebucht hat, grundsätzlich möglich und wird beispielsweise von den Strafverfol-

gungsbehörden aktiv genutzt. Auch einige Dienstleister bieten die GSM-Ortung an. Details zur Positionsbestimmung sollen jedoch in einem gesonderten Abschnitt erörtert werden (vgl. 3.3).

3.2.4 Bewertung und Analyse

Über die drei hier vorgestellten Kommunikationsverfahren hinaus existieren noch viele weitere Verfahren. So wurde beispielsweise Bluetooth nicht erwähnt, das jedoch viele Parallelen zu IEEE 802.15.4/ZigBee besitzt. Es können in Rahmen dieser Arbeit aber nicht alle in Frage kommenden Protokolle untersucht werden, so dass die Wahl auf die genannte drei Verfahren, die, wie in Abschnitt 3.2 erläutert, jeweils eine unterschiedliche Klasse repräsentieren.

Um zu einer abschließenden Bewertung zu kommen sollen anhand einiger Kriterien die Vor- und Nachteile der jeweiligen Kommunikationsverfahren verglichen werden.

Reichweite:

Die größte Reichweite bieten die Mobilfunkverfahren GSM und UMTS. Aber auch mit ZigBee und WLAN lassen sich Reichweiten erzielen, die auch im Gebäude bei mindestens 30 Metern liegen. Da in einem Sensornetz, wie es in dieser Arbeit realisiert werden soll, die Abstände der Sensoren gering sind, ist also prinzipiell jedes Verfahren geeignet. Beim Mobilfunk besteht jedoch die Einschränkung, dass dort eine Kommunikation immer über die Basisstation (also den Sendemast) erfolgen muss und nicht direkt zwischen den Teilnehmern erfolgen kann. Somit können Knoten die sich tief im Innern eines Gebäudes befinden nicht erreichbar sein. Damit ist dieses Kommunikationsverfahren nur geeignet für Gebäude in denen Überall eine Netzversorgung besteht. Dieses ist innerhalb des beschriebenen Szenarios ein Ausschlusskriterium. Daher sollen GSM und UMTS im Folgenden nicht weiter berücksichtigt werden.

Datenrate:

Die höchste Datenrate lässt sich mit WLAN erzielen. ZigBee kann eine maximale Datenrate von 250 kbit/s erreichen. Da in einem Sensornetz nur wenig Daten anfallen und die Sensoren die meiste Zeit in Schlafmodus verbringen, ist dieses Kriterium für ein Sensornetz nicht entscheidend, da auch die geringere Datenrate von ZigBee für den Anwendungsfall noch ausreichend ist.

Energieverbrauch:

Der Energieverbrauch ist bei batteriebetriebenen Sensorknoten sehr wichtig, da sie nur über sehr geringe Reserven verfügen und möglichst lange aktiv sein sollen. Hier besitzt IEEE 802.15.4/ZigBee die größten Vorteile, da es besonders auf die Energieeffizienz hin optimiert und entwickelt wurde. Die maximale Sendeleistung beträgt 10 mw. WLAN kann mit einer maximalen Leistung von 100 mW senden. Damit lässt sich feststellen, dass ZigBee/IEEE 802.15.4 in Bezug auf den Energieverbrauch das beste Protokoll darstellt und sich in diesem Punkt am besten für Sensornetzwerke eignet.

Latenzzeiten:

Die Latenzzeiten sind besonders bei ZigBee/IEEE 802.15.4 sehr gering. Dies ist insofern nicht verwunderlich, da dies ein Designaspekt des Protokolls darstellt. Bei der Entwicklung des WLAN-Standards wurden andere Prioritäten gesetzt, so dass auch nach diesem Kriterium eine Lösung nach IEEE 802.15.4 die bessere Wahl ist.

Insgesamt lässt sich abschließend feststellen, dass IEEE 802.15.4/ZigBee von den hier vorgestellten Kommunikationsverfahren am besten für den Einsatz in einem Sensornetz geeignet ist. Es erfüllt alle der diskutierten Kriterien zufriedenstellend. Dies ist jedoch insofern nicht verwunderlich, da es schließlich für den Einsatz in Geräten mit geringen Ressourcen und reduziertem Funktionsumfang konzipiert wurde und in diesem Bereich seine Stärken ausspielt. Da Sensorknoten im Allgemeinen nur über wenig Ressourcen verfügen, bietet sich somit IEEE 802.15.4/ZigBee als Kommunikationsprotokoll innerhalb dieser Arbeit an.

Im folgenden sollen die Bewertungskriterien zusammenfassend in eine Tabelle dargestellt werden.

Tabelle 3.2: Bewertung der Beurteilungskriterien

| | IEEE 802.15.4 / ZigBee | Wlan | GSM |
|------------------|---------------------------|------|-----|
| Reichweite | + | + | ++ |
| Datenrate | + | ++ | o |
| Energieverbrauch | + | o | o |
| Latenzzeiten | + | o | - |

3.3 Lokalisation

Für die im Kapitel *Ziele und Vorhaben* genannten Ziele ist es notwendig die Position von mobilen Sensorknoten innerhalb eines Sensornetzes bestimmen zu können. Für die Lokalisation von Objekten innerhalb eines funkgestützten Kommunikationsnetzes existieren unterschiedliche Verfahren und Techniken. GPS das oft zur Positionsbestimmung dient, kann hier leider nicht genutzt werden, da nur außerhalb von Gebäuden ein reibungsloser Empfang gewährleistet ist. Eine Lokalisation lässt sich mit unterschiedlichen Prämissen und Kriterien durchführen, die im folgenden diskutiert werden sollen (vgl. [Hightower und Borriello \(2001\)](#)):

Lage im Koordinatensystem:

Die Position eines Objektes kann zum einem als absolute Koordinate in einem lokalen bzw. globalen Koordinatensystem angegeben werden, zum anderen aber auch als eine relative Position in Bezug auf einen fixen Punkt.

Genauigkeit der Position:

Unterschiedliche Verfahren zur Lokalisation erzielen unterschiedliche Genauigkeiten. Diese kann von wenigen Zentimetern bis zu mehreren Metern reichen. Eine höhere Genauigkeit bedeutet in der Regel auch einen höheren Aufwand. Auch ist es für einige Anwendungen nicht nötig über sehr genaue Positionsdaten zu verfügen, sondern es ist ausreichend nur die ungefähre Position zu kennen.

Ort der Positionsberechnung:

Die Lokalisation eines Objektes kann an zwei unterschiedlichen Stellen erfolgen. Die Berechnung kann einerseits lokal im Gerät selber durchgeführt werden. Damit ist die eigene Position nur dort bekannt und kann beispielsweise für die Anzeige auf einer Umgebungskarte genutzt werden, wie dies z. B. bei einem Navigationsgerät der Fall ist. Auch aus Sicherheits- oder Persönlichkeitsschutzgründen kann es sinnvoll sein so vorzugehen, da keine Datenübermittlung an andere Geräte notwendig ist und die Positionsdaten nur dem Gerät selber bekannt sind. Alternativ kann die Positionsermittlung aller lokalisierbaren Objekte zentral erfolgen. Dies hat den Vorteil, dass dort die Lage aller Objekte bekannt ist und weiterverarbeitet werden kann. Wenn die Geräte nur über begrenzte Ressourcen verfügen, wie dies bei einem Sensorknoten der Fall ist, kann es ebenfalls sinnvoll sein die Berechnung an zentraler Stelle durchzuführen.

Kosten:

Es entstehen bei der Lokalisation unterschiedliche Arten von Kosten. Zum einen verursacht die Hardware, die für die Lokalisation benötigt wird, Kosten. So wird oft zusätzliche Hardware nötig, die die Positionsermittlung durchführt. Teilweise kann aber auch auf vorhandene Komponenten zurückgegriffen werden. (z.B. Nutzung des CC2420-Funkmoduls und Messung des RSSI-Wertes (siehe Abschnitt 3.3.1.1)). Andererseits bedeutet auch der nötige Rechenaufwand, der für die Lokalisation betrieben werden muss, einen Kostenfaktor, insbesondere wenn die Geräte (also in diesem Fall die Sensorknoten) nur über begrenzte Ressourcen verfügen.

Eine Lokalisation innerhalb eines Gebäudes mit den in Abschnitt 3.2 beschriebenen Kommunikationsverfahren (z. B. IEEE 802.15.4), kann in folgende zwei Schritte unterteilt werden:

1. Distanz- oder Winkelbestimmung
2. Positionsberechnung durch Triangulation

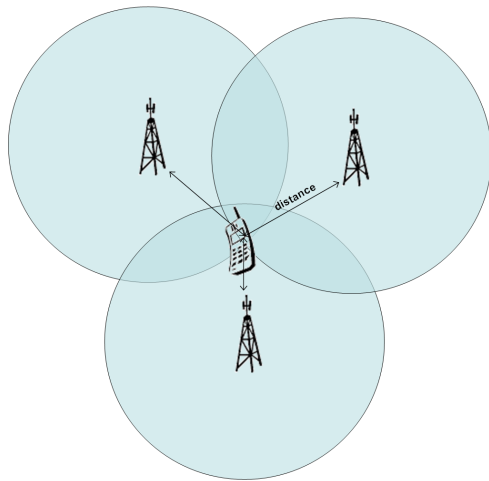
Um eine Lokalisation durchführen zu können ist es zunächst nötig, eine Distanz oder einen Empfangswinkel zwischen zwei Funkeinheiten zu bestimmen. Ohne eine Distanz bzw. einen Winkel ist eine Berechnung nicht möglich. Weiterhin müssen Referenzpunkte zur Verfügung stehen, deren Position bekannt ist, um davon ausgehend die Positionsermittlung einer Einheit mit unbekanntem Standort durchzuführen. Diese Referenzpunkte bilden in dieser Arbeit die fest installierten Sensorknoten und sie können daher genutzt werden, um die Position der mobilen Knoten zu bestimmen.

Im nächsten Abschnitt sollen zunächst verschiedene Verfahren zur Distanz- bzw. Winkelbestimmung beschrieben werden. Bis auf die Bestimmung des RSSI-Wertes (siehe Abschnitt 3.3.1.1) sind die anderen hier vorgestellten Verfahren vor allem aus dem Mobilfunkbereich bekannt. Sie lassen sich jedoch durchaus auch in einem Sensornetz einsetzen.

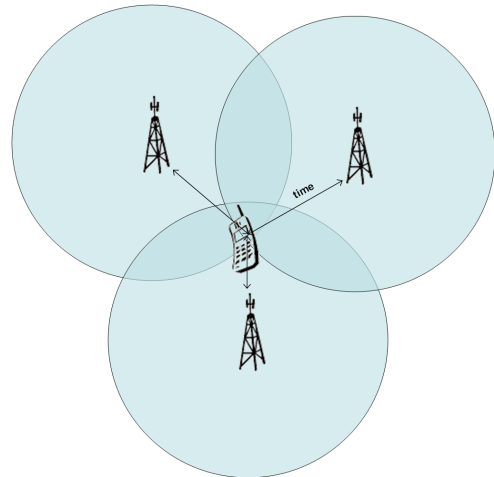
Welches der genannten Verfahren zur Lokalisation genutzt werden kann, ist auch abhängig von der Wahl der Sensorknoten, die im nächsten Abschnitt betrachtet werden sollen. Nicht alle Verfahren sind mit jedem Knoten durchführbar bzw. es besteht nicht die Möglichkeit ein entsprechendes Modul mit jeder Plattform zu nutzen.

3.3.1 Distanzbestimmung

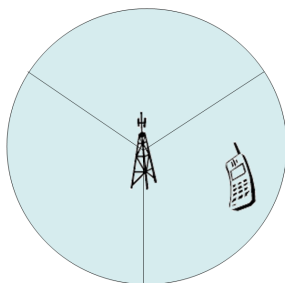
Um die Distanz zwischen zwei Funkeinheiten zu bestimmen, sollen hier die Methoden der Signalstärkemessung sowie der Laufzeitmessung erläutert werden und jeweils ein Verfahren aus den Bereichen vorgestellt werden.



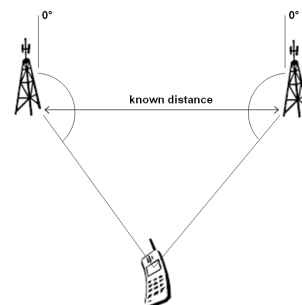
(a) Received Signal Strength Indicator (RSSI)



(b) Time of Arrival (ToA)



(c) Cell of Origin (CoO) with Cell Sector



(d) Angle of Arrival (AoA)

Abbildung 3.6: Verfahren zur Positionsbestimmung

3.3.1.1 Received Signal Strength Indicator (RSSI)

Der RSSI-Wert beschreibt die Empfangsstärke einer gesendeten Nachricht. Aus einer übertragenen Nachricht zwischen zwei Stationen kann der Empfänger den RSSI-Wert bestimmen. Da die Signalstärke quadratisch zur Entfernung abnimmt, kann damit auf die Entfernung geschlossen werden. Es kann jedoch bei gleichen Abständen zu unterschiedlichen RSSI-Werten kommen und damit zu Differenzen zwischen den berechneten und den tatsächlichen Entfernungswerten. Ursächlich dafür sind zum einen Hindernisse in der Funklinie, wie z. B. Wände aber auch sich bewegende Personen. Zum anderen hat auch die räumliche Lage des Funkmoduls bzw. der Antenne Einfluss auf den RSSI-Wert, da die Antenne nicht in alle Richtungen gleichmäßig ausstrahlt.

Der RSSI-Wert lässt sich beispielsweise mit dem CC2420-Funkmodul, das von vielen Sensorknoten genutzt wird (s. a. den folgenden Abschnitt 3.4), einfach und mit geringen Kosten ermitteln. Damit ist dieses Verfahren besonders energieeffizient und gut für Sensorknoten geeignet. Außerdem wird keine zusätzliche Hardware benötigt.

3.3.1.2 Time of Arrival (ToA)

Beim Time of Arrival-Verfahren wird die Laufzeit einer Nachricht zwischen Sender und Empfänger gemessen und daraus unter Zuhilfenahme der Signalausbreitungsgeschwindigkeit eine Entfernung berechnet. Um die Laufzeit einer Nachricht ermitteln zu können, ist es jedoch erforderlich, dass alle Geräte über exakt synchronisierte Uhren verfügen. In einem Sensornetz mit sehr vielen Knoten, ist die Uhrensynchronisation jedoch sehr aufwendig, insbesondere da wegen des geringen Abstands der Sensoren nur sehr kurze Laufzeiten zu erwarten sind und die Synchronisation damit sehr genau sein muss. Ein Vorteil dieses Verfahrens ist jedoch, dass die Nachrichtenlaufzeit nicht durch Hindernisse beeinflusst wird, wie es z. B. beim RSSI-Wert der Fall ist. Außerdem lässt sich aus der Laufzeit sehr genau eine Entfernung ermitteln. Das Time of Arrival-Verfahren kann nicht nur im Bereich der Funkübertragung eingesetzt werden, sondern eignet sich beispielsweise auch für Ultraschallverfahren.

Eine Weiterentwicklung dieses Verfahrens ist das Time Difference of Arrival (TDoA)-Verfahren, das nach dem selben Prinzip der Laufzeitmessung arbeitet, jedoch genauere Ergebnisse erzielt.

3.3.2 Winkelbestimmung

Zur Lokalisation kann ebenso eine Winkelbestimmung zwischen verschiedenen Geräten genutzt werden. Das im folgenden vorgestellte Cell of Origin-Verfahren passt nur z. T. in diese Kategorie, da es maximal einen Winkelabschnitt als Ergebnis zurück gibt, soll jedoch

trotzdem genannt werden, da es mit wenig Aufwand eingesetzt werden kann und damit bei geringen Anforderungen an die Genauigkeit bereits ausreichend ist.

3.3.2.1 Cell of Origin (CoO)

Das Cell of Origin-Verfahren ist vor allem aus dem Mobilfunkbereich bekannt. Es ist das einfachste Verfahren, um eine Aussage über die Position eines Teilnehmers innerhalb des Netzes zu treffen. Als Positionsangabe wird die Mobilfunkzelle, in die sich das Telefon eingebucht hat, angegeben. Damit ist die Positionsangabe sehr ungenau und hängt von der Größe der Zelle ab. Außerdem kann bei mehreren sich überlappenden Zellen nicht davon ausgegangen werden, dass sich das mobile Gerät immer in der am nächsten gelegenen Station einbucht. Ein Vorteil des Cell of Origin-Verfahrens ist, dass es einfach und mit wenig Aufwand zu implementieren ist.

Verbessern lässt sich dieses Verfahren durch eine Sektorerkennung (Cell of Origin with Cell Sector). Dazu wird die Zelle beispielsweise in drei Sektoren von je 120 Grad eingeteilt und mit mehreren Antennen an der Basisstation wird eine Zuordnung zu den Sektoren möglich.

Das Cell of Origin-Verfahren ist ressourcenschonend und ausreichend, wenn nur ein ungefährender Aufenthaltsort bestimmt werden soll. Es wird z. Z. im GSM-Mobilfunk genutzt, um die Position eines Mobilfunknutzers für die Sicherheitsbehörden zu orten.

3.3.2.2 Angle of Arrival (AoA)

Beim Angle of Arrival-Verfahren wird durch spezielle Antennen der Empfangswinkel bei einer Datenübertragung zwischen Mobileinheit und Basisstation gemessen. Durch eine Bestimmung des Empfangswinkels an zwei Basisstationen lässt sich mit Hilfe des Triangulationsverfahrens (siehe. Abschnitt [3.3.3.2](#)) die Position der Mobileinheit berechnen.

3.3.3 Triangulation

Der Begriff Triangulation fasst zwei Verfahren, die Lateration und die Angulation, zusammen. Beide nutzen geometrische Eigenschaften, um daraus eine Position zu berechnen. Bei der Lateration wird eine Berechnung der Schnittpunkte von Kreisen durchgeführt, während die Angulation die geometrischen Eigenschaften eines Dreiecks gebraucht.

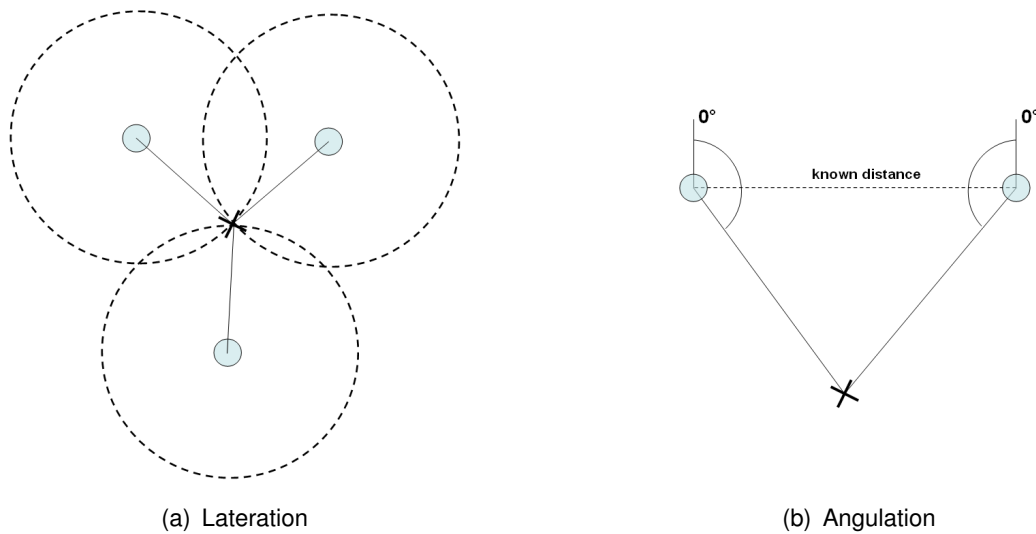


Abbildung 3.7: Darstellung der Triangulationsverfahren

3.3.3.1 Lateration

Bei der Triangulation sind mindestens drei Referenzknoten, deren Position bekannt sein muss, und der Abstand des gesuchten Knotens zu diesen nötig, um eine eindeutige Position bestimmen zu können. Die Distanzen entsprechen den Radien der drei Kreise in Abbildung 3.7 (a). Die Schnittpunkte aller Kreise bilden einen eindeutigen Punkt, der damit der gesuchten Position entspricht.

Für die Distanzbestimmung eignen sich u. a. die in 3.3.1 beschriebenen Verfahren.

3.3.3.2 Angulation

Für eine Positionsbestimmung mit der Angulationsmethode reichen zwei Referenzknoten mit bekannter Position. Zur Berechnung wird der jeweilige Winkel zum gesuchten Punkt (vgl. Abbildung 3.7 (b)) benötigt. Mit Hilfe der trigonometrischen Funktionen eines Dreiecks kann damit die Position des gesuchten Knotens berechnet werden.

Die Winkelbestimmung kann z. B. mit dem Angle of Arrival-Verfahren erfolgen.

Nachdem in den vorherigen Abschnitten unterschiedliche Kommunikationsverfahren präsentiert und die Grundlagen der Lokalisation erläutert wurden, können darauf aufbauend verschiedene Sensorplattformen betrachtet werden. Diese sollen im folgenden Abschnitt vorgestellt werden.

3.4 Sensorplattformen

Es existieren unterschiedliche Sensorplattformen, die zum Aufbau eines Sensornetzes geeignet erscheinen. Es sollen in diesem Kapitel einige der in Frage kommenden Sensorknoten vorgestellt werden und daran anschließend eine Entscheidung für eine der Plattformen getroffen werden.

In diesem Abschnitt sollen die Scatterweb-Plattform, die BTnode, die Sun SPOT-Plattform und die Berkeley Motes beschrieben werden. Eine weitere interessante Plattform, die Intel iMote2, die allerdings erst seit Februar 2007 verfügbar ist, kann hier leider nicht mehr berücksichtigt werden, da die Entscheidung schon im Herbst 2006 getroffen werden musste, um rechtzeitig mit den ersten Versuchen für diese Arbeit beginnen zu können. Andere Sensorplattformen wie die μ AMPS des MIT bleiben leider aus Zeitgründen unerwähnt.

3.4.1 Scatterweb

Die ScatterWeb-Plattform ist eine Entwicklung der FU Berlin ([[ScatterWeb \(2007\)](#)]). Es wurden verschiedene Knotentypen realisiert, wie Sensor- oder Gateway-Knoten für die Anbindung des Sensornetzes an ein PC oder ein LAN. Als Sensorknoten entstand u. a. die Scatterweb ESB, wobei ESB für „Embedded Sensor Board“ steht.

Die Scatterweb ESB nutzt als Microcontroller den MSP430 der Firma Texas Instruments. Auf dem Sensorknoten ist das Funkmodul RFM TR1001 verbaut, das eine Reichweite von 300 m außerhalb und 100 m im Gebäude haben soll. Damit ist eine Kommunikation zu anderen Sensorknoten im 868MHz-Bereich mit bis zu 115.2kbps möglich. Bereits in die Plattform integriert sind u. a. ein Temperatur-, Helligkeits-, Infrarot-, Bewegungs- und Erschütterungssensor, ebenso wie ein Mikrofon. Es sind noch weitere externe Module verfügbar, wie z. B. ein GSM-Modul oder ein Solarzellenmodul, um den Sensorknoten ohne Batterien zu betreiben. Außerdem besteht die Möglichkeit externe I/O-Module anzuschließen.

Die Scatterweb ESB wird mit 3 AAA Batterien betrieben und es ist laut [[ScatterWeb \(2007\)](#)] in einer Demo-Anwendung bei einem duty-cycle von 1%, in dem die Datenverarbeitung und Kommunikation stattfindet, eine Lebenszeit von 5 Jahren möglich. Die Scatterweb ESB hat einen sehr niedrigen Stromverbrauch von nur $8\mu\text{A}$ im Schlafzustand. Zur Programmierung besitzt sie eine serielle Schnittstelle. Es besteht außerdem die Möglichkeit das TinyOS-Betriebssystem, auf das in Kapitel 4 näher eingegangen werden soll, auf der Scatterweb ESB einzusetzen.

3.4.2 BTnode

Die BTnode, an der ETH Zürich entwickelt, ist eine drahtlose Kommunikations- und Berechnungsplattform, die über keine integrierten Sensoren verfügt ([BTnode (2007)] und [Beutel u. a. (2007)]). Eine weitere Besonderheit dieser Plattform ist es, dass sie mit zwei unterschiedlichen Systemen zur Kommunikation ausgestattet ist.

Die BTnode rev3 ist mit einem 8 MHz-Microcontroller Atmel ATmega 128L ausgestattet. Sie verfügt über 64+180 kByte RAM und 128 kByte Flashspeicher. Die BTnode ist mit Bluetooth-Funk der Klasse 2 und einem Chipcon CC1000 Funkmodul ausgestattet. Durch diese beiden Funkmodule soll eine Kommunikation sowohl zu den älteren BTnodes, die ein Bluetooth-Modul besitzen, als auch zu einigen Berkeley-Motes, auf die in Abschnitt 3.4.4 näher eingegangen wird, sichergestellt werden. Es ist möglich beide Systeme gleichzeitig zu betreiben, wodurch es jedoch zu einem erhöhten Energieverbrauch kommt. Da allerdings für die Funkkommunikation die meiste Energie benötigt wird, ist der gleichzeitige Betrieb beider Funkmodule nur in Ausnahmefällen sinnvoll. Die BTnode hat, verglichen mit der Scatterweb ESB oder der TelosB-Mote, einen relativ hohen Stromverbrauch von 50 μA im Schlafzustand. Sie besitzt, wie bereits erwähnt, keine eigenen Sensoren. Es besteht jedoch die Möglichkeit, über verschiedene Schnittstellen externe Sensoren anzuschließen.

Auf den BTnodes lässt sich sowohl das TinyOS Betriebssystem (siehe Kapitel 4) einsetzen als auch eine angepasste Variante des Nut/OS³, einem Open Source Realtime-Betriebssystem.

3.4.3 SunSpot

Von der Firma Sun Microsystems stammt die Sun SPOT Sensorplattform ([Sun (2007)]). Einem Sensorboard auf dem, im Gegensatz zu vielen anderen Plattformen, eine Java Virtual Machine (VM) eingesetzt werden kann. Dabei handelt es sich um die Squawk VM, eine Java 2 Micro Edition (J2ME) VM. Applikationen für diese Sensorplattform können damit in Java entwickelt werden, mit der Möglichkeit die Sensoren, die I/O-Pins oder das Funkmodul mit Hilfe von Java Bibliotheken anzusprechen.

Die Sun SPOT Plattform ist modular aufgebaut. Das so genannte Main-Board realisiert die Grundfunktionalität der Plattform. Es enthält eine mit 180 MHz getaktete 32-Bit ARM920T CPU, die mit 512 kByte RAM und 4 MB Flashspeicher ausgerüstet ist. Als Kommunikationseinheit wird ein Chipcon CC2420-Modul verwendet, mit dem eine Kommunikation nach IEEE 802.15.4 mit 2,4 GHz im ISM-Band möglich ist. Weiterhin wurde ein Sensor-Board entwickelt, das mit einem Beschleunigungs-, Temperatur- und einem Helligkeitssensor ausgestattet ist

³egnite Software GmbH: Nut/OS – <http://www.ethernut.de/en/software/index.html>

und zusammen mit dem Main-Board in einem Gehäuse ausgeliefert wird. Zusätzlich ist es möglich an die Sun SPOT externe I/O-Module anzuschließen.

Die Sun SPOT Plattform wird mit einem 750 mAh Lithium-Ionen-Akku betrieben. Über die Lebensdauer sind leider keine Informationen verfügbar, obwohl dies von großen Interesse wäre, da bei der wesentlich leistungsstärkeren Hardwareausstattung des Main-Boards gegenüber den anderen in diesem Kapitel vorgestellten Plattformen, ein höher Energieverbrauch zu erwarten ist.

3.4.4 Berkeley Motes

Bei den so genannten Berkeley Motes handelt es sich um mehrere Sensorplattformen, die an der University of California in Berkeley entstanden sind ([Hill u. a. (2000)], [Polastre u. a. (2004b)]) und von der Firma Crossbow Technology Inc.⁴ vertrieben werden. Die Entwicklung der Motes an der UC Berkeley basiert auf dem Smart Dust-Projekt (siehe Abschnitt 5.1). Dort ging es in erster Linie darum, neue Techniken zum Bau von Sensorknoten zu erforschen und damit Knoten von einer Größe im Bereich weniger Kubikmillimeter zu entwickeln. Dieses Designziel wurde bei den Motes nicht verfolgt, sondern hier sollten mit Standardkomponenten (COTS = commercial off-the-shelf) und den damit verbundenen Vorteilen Sensorknoten erschaffen werden.

Auf den Motes sind ein Mikrocontroller, eine Funkkommunikationseinheit, ein Flash-Speicher und ein Interface, an das z. B. Sensoren angeschlossen werden können, als Hauptbestandteile enthalten. In der Zeit von 1998 bis 2004 sind verschiedene Sensorplattformen entstanden (siehe Abbildung 3.8), die kontinuierlich optimiert wurden. Es wurde beispielsweise die Leistungsfähigkeit des Microcontrollers zwischen den verschiedenen Generationen verbessert. Dabei ging die Entwicklung vom Atmel AT90LS8535 über den Atmel ATmega163 und den ATmega128 bis zum Texas Instruments MSP430 bei der Telos-Mote.

Bei der Telos-Mote in der Revision B (als TelosB bezeichnet), handelt es sich um die z. Z. neueste und am weitesten entwickelte Mote der UC Berkeley. Sie besitzt einen MSP430 Microcontroller, 10 kByte RAM, 1MByte Flashspeicher und ist mit dem Chipcon CC2420-Modul, zur drahtlosen Kommunikation mit bis zu 250 kbit/s nach dem IEEE 802.15.4 Standard, ausgerüstet (s. a. Abbildung 3.9). Es wird eine in die Platine integrierte Antenne zur Kommunikation genutzt, mit der laut Datenblatt⁵ eine Reichweite von bis zu 100 m außerhalb und 30 m innerhalb eines Gebäudes möglich ist. Zur Verbesserung der Reichweite kann eine

⁴<http://www.xbow.com/>

⁵http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf








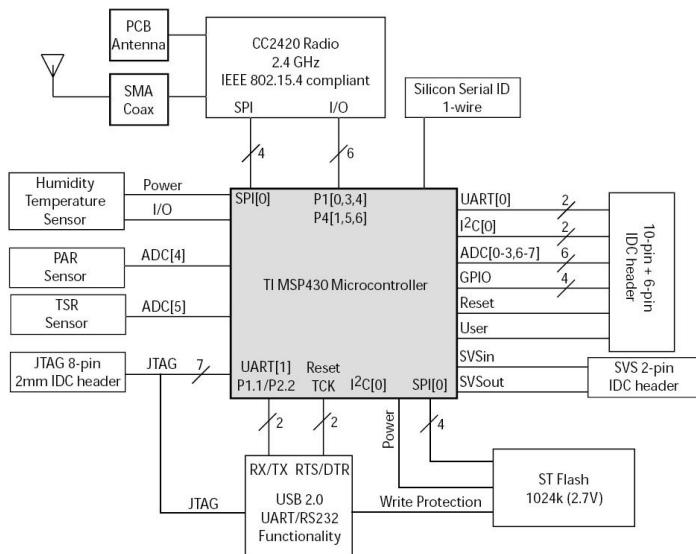
| Mote Type Year | WeC 1998 | René 1999 | René 2 2000 | Dot 2000 | Mica 2001 | Mica2Dot 2002 | Mica 2 2002 | Telos 2004 | |
|----------------------------------|---|---|----------------|---|--|---|---|---|--|
| |  |  | |  |  |  |  |  | |
| Microcontroller | | | | | | | | | |
| Type | AT90LS8535 | | ATmega163 | | ATmega128 | | TI MSP430 | | |
| Program memory (KB) | 8 | | 16 | | 128 | | 60 | | |
| RAM (KB) | 0.5 | | 1 | | 4 | | 2 | | |
| Active Power (mW) | 15 | | 15 | | 8 | | 33 | | |
| Sleep Power (μ W) | 45 | | 45 | | 75 | | 75 | | |
| Wakeup Time (μ s) | 1000 | | 36 | | 180 | | 180 | | |
| Nonvolatile storage | | | | | | | | | |
| Chip | 24LC256 | | | AT45DB041B | | | ST M24M01S | | |
| Connection type | I ² C | | | SPI | | | I ² C | | |
| Size (KB) | 32 | | | 512 | | | 128 | | |
| Communication | | | | | | | | | |
| Radio | TR1000 | | | TR1000 | | CC1000 | | CC2420 | |
| Data rate (kbps) | 10 | | | 40 | | 38.4 | | 250 | |
| Modulation type | OOK | | | ASK | | FSK | | O-QPSK | |
| Receive Power (mW) | 9 | | | 12 | | 29 | | 38 | |
| Transmit Power at 0dBm (mW) | 36 | | | 36 | | 42 | | 35 | |
| Power Consumption | | | | | | | | | |
| Minimum Operation (V) | 2.7 | | 2.7 | | 2.7 | | 1.8 | | |
| Total Active Power (mW) | 24 | | | 27 | | 44 | | 89 | |
| Programming and Sensor Interface | | | | | | | | | |
| Expansion | none | 51-pin | 51-pin | none | 51-pin | 19-pin | 51-pin | 10-pin | |
| Communication | IEEE 1284 (programming) and RS232 (requires additional hardware) | | | | | | | USB | |
| Integrated Sensors | no | no | no | yes | no | no | no | yes | |

Abbildung 3.8: Übersicht der Berkeley Motes [Polastre u. a. (2004b)]

externe Antenne angeschlossen werden. Der eingesetzte MSP430 Microcontroller, ein 16-Bit-Microcontroller mit von Neumann-Architektur, ist im Hinblick auf einen geringen Energieverbrauch optimiert. Es ist bereits ein Temperatur-, Feuchtigkeits- und ein Helligkeitssensor in die Mote integriert, weitere Sensoren können über eine externe Schnittstelle angeschlossen werden. Angesprochen und programmiert werden kann die TelosB vom PC über einen USB-Port. Dieser emuliert eine serielle Schnittstelle (UART).

Die TelosB-Mote zeichnet sich durch besonders geringen Energieverbrauch in der Schlafphase und durch eine besonders kurze Aufwachzeit aus (sleep: $2.4 \mu\text{A}$; wakeup-time: max. $6 \mu\text{s}$) (vgl. [Polastre u. a. (2004b)]). So soll laut [Polastre u. a. (2004b)] in einer Beispielanwendung mit einem duty-cycle von $<1\%$ eine Lebensdauer von 945 Tagen möglich sein, erzielt mit 2 handelsüblichen AA Batterien.

Parallel zu den Motes wurde an der UC Berkeley das TinyOS Betriebssystem entwickelt, dass auch auf der TelosB zum Einsatz kommt. Auf TinyOS wird im Kapitel 4 *TinyOS* näher eingegangen.



(a) Blockschaltbild



(b) TelosB-Mote mit Halterung für 2 AA-Batterien

Abbildung 3.9: TelosB-Mote der UC Berkeley

3.4.5 Bewertung und Analyse

Eine Entscheidung für eine der genannten Sensorplattformen, soll anhand einiger maßgeblicher Kriterien getroffen werden. Dazu werden die Sensorknoten anhand dieser Gesichtspunkte bewertet, aber auch ihre spezifischen Vor- und Nachteile sollen mit in die Entscheidung einbezogen werden.

Die wesentlichen Entscheidungskriterien sind:

- Energieeffizienz
- Größe
- Kommunikationsmodul
- Zuverlässigkeit
- Erweiterbarkeit der Plattform
- Programmierumgebung

Energieeffizienz

Wenn man die vier hier vorgestellten Sensorknoten im Hinblick auf die Energieeffizienz beurteilt, dann sind besonders bei der Scatterweb ESB und der TelosB-Mote die lange Lebensdauer und die Leistungsfähigkeit interessant. Bei beiden ist die prognostizierte Lebensdauer

sehr lang und damit sind diese Plattformen im Hinblick auf dieses für die Auswahl wichtige Kriterium am besten geeignet.

Größe

Die Größenunterschiede zwischen den unterschiedlichen Plattformen sind gering und damit ist dieser Punkt für eine Beurteilung nachrangig zu bewerten. Der meiste Raum wird in der Regel durch die Batteriehalterung beansprucht.

Kommunikationsmodul

Zur Kommunikation benutzen zwei Plattformen, TelosB und Sun SPOT, das Chipcon CC2420-Modul mit dem eine Kommunikation nach dem IEEE 802.15.4 Standard möglich ist. Damit kann prinzipiell eine Verbindung zu weiteren Geräten, die den gleichen Protokoll-Stack implementieren, aufgebaut werden. Außerdem bieten diese Module die höchsten Datenraten mit 250kbps.

Zuverlässigkeit

Die Zuverlässigkeit der Sensorknoten lässt sich ohne Praxistest schwer beurteilen. Besonders von der BTnode und der TelosB-Mote kann man jedoch eine hohe Zuverlässigkeit und Stabilität erwarten, da es bereits mehrere Generationen bzw. Modelle dieser Sensorknoten hergestellt wurden und daher weniger Fehler oder Probleme zu erwarten sind.

Erweiterbarkeit der Plattform

Da die Sensorknoten in dieser Arbeit dazu eingesetzt werden sollen, Temperaturwerte zu erfassen, muss die Möglichkeit bestehen, über entsprechende Schnittstellen einen Sensor anschließen und auswerten zu können. Einige der hier vorgestellten Plattformen besitzen einen integrierten Temperatursensor, so dass für die Vorhaben dieser Arbeit damit keine externe Schnittstelle benötigt wird. Diese wäre jedoch für zukünftige Erweiterungen wünschenswert. Bereits einen eingebauten Temperatursensor besitzen die TelosB-Mote, die Scatterweb ESB und die Sun SPOT. Bei der Scatterweb ESB ist erwähnenswert, dass zusätzlich viele weitere Sensoren integriert sind und außerdem interessante Erweiterungsmodule zur Verfügung stehen.

Programmierungsumgebung

Die Software-Umgebung für den Betrieb der Sensorknoten ist auch ein Entscheidungskriterium. Hier sollte am besten ein Open Source-Produkt zum Einsatz kommen, da dies die meisten Eingriffs- und Erweiterungsmöglichkeiten gewährleistet. Dazu bietet sich TinyOS

(s. a. Kapitel 4) an, dass auf vielen Sensorplattformen einsetzbar ist. So kann es, mit Ausnahme der Sun SPOT-Plattform, mit allen anderen hier vorgestellten Sensorknoten genutzt werden.

Die Besonderheit der Sun SPOT, die Fähigkeit Java-Code auszuführen, ist sehr interessant. Jedoch ist dafür auch ein wesentlich leistungsstärkerer Microcontroller unerlässlich und es ist mehr Aufwand nötig, um eine Java Virtual Maschine darauf auszuführen, womit wiederum ein höherer Energieverbrauch verbunden ist. Zur genaueren Beurteilung dieser Plattform wären daher genauere Daten nötig.

Nachdem in diesem Abschnitt mehrere Plattformen gegenübergestellt und die Vor- und Nachteile der einzelnen Sensorknoten erläutert wurden, sind die am Besten geeigneten Plattformen die TelosB-Mote und die Scatterweb ESB. Tabelle 3.3 zeigt zusammenfassend die Kriterien und die jeweilige Tauglichkeit der Plattform. Bei mit einem Fragezeichen beurteilten Punkten, standen nicht genügend Informationen für eine Bewertung zur Verfügung.

Tabelle 3.3: Bewertung der Beurteilungskriterien

| | Scatterweb ESB | BTnode | Sun SPOT | TelosB |
|-------------------------|----------------|--------|----------|--------|
| Energieeffizienz | + | - | ? | ++ |
| Größe | o | o | o | o |
| Kommunikationsmodul | + | + | ++ | ++ |
| Zuverlässigkeit | ? | + | ? | + |
| Erweiterbarkeit | ++ | + | - | + |
| Programmierungsumgebung | ++ | ++ | ++ | ++ |

Nach Abwägung aller Aspekte fiel die Wahl auf die Berkeley Motes und dabei auf die TelosB-Mote. Sie erfüllt die meisten der genannten Kriterien und soll daher dazu genutzt werden ein Sensornetz aufzubauen.

4 TinyOS

TinyOS ist ein Real-Time-Betriebssystem (RTOS) für Sensorknoten, das an der University of California in Berkeley entwickelt wurde ([Hill u. a. (2000)]). Es entstand ursprünglich parallel zu den Berkeley Motes (vgl. Abschnitt 3.4.4), ist aber inzwischen als Open Source-Projekt¹, auf vielen weiteren Plattformen, zusätzlich zu den in dieser Arbeit vorgestellten (vgl. Abschnitt 3.4), einsetzbar. Seit Anfang November 2006 ist TinyOS in der Version 2.x verfügbar, die in diesem Projekt auch eingesetzt werden soll. Gegenüber dem 1.x-Zweig wurde einige konzeptionelle Nachteile verbessert. Damit sind allerdings Anwendungen, die für TinyOS 1.x entwickelt wurden, unter der neuen Version z.T. nicht mehr lauffähig und müssen angepasst werden.

Die Nutzung von TinyOS hat viele Vorteile bei der Anwendungsentwicklung. Es kann auf viele bereits fertige Komponenten und Schnittstellen zurückgegriffen werden. So lässt sich z. B. die Komponente für das Senden einer Nachricht direkt aufrufen und der Benutzer braucht sich um das konkrete Senden eines Paketes mit Hilfe der Hardwarekomponente nicht zu kümmern. Damit wird beispielsweise die Nutzung eines bestimmten MAC-Protokolls komplett vor dem Anwender verborgen. Tatsächlich wird als Grundeinstellung das B-MAC (auch Berkeley-MAC genannt) Protokoll genutzt.

B-MAC wurde von Woo und Culler an der UC Berkeley im Jahre 2001 entwickelt (vgl. [Woo und Culler (2001)] und [Polastre u. a. (2004a)]). Beim Design dieses Protokolls wurde davon ausgegangen, dass kurzfristig ein hohes Verkehrsaufkommen entstehen kann und dass die übermittelten Pakete relativ kurz sind (ca. 30 Bytes). B-MAC nutzt das CSMA-Prinzip für den Medienzugriff, allerdings wird vor dem Lauschen auf dem Medium eine zufällige Zeitspanne im Schlafmodus gewartet. Durch die Wartephase sollen Kollisionen vermieden werden, da es häufig vorkommt, dass mehrere Knoten dasselbe Ereignis zur selben Zeit erkennen.

TinyOS wurde laut Hill entwickelt, da die vorhandenen Betriebssysteme für Embedded-Geräte nicht ausreichend waren für die Anforderungen der Sensorknoten. Beim Design bisheriger Real-Time-Betriebssysteme wie QNX oder VxWorks stand im Vordergrund eine Ausführungsumgebung ähnlich zu traditionellen PC-Betriebssystemen bereitzustellen, so dass bereits existierender Programmcode leicht weiterverwendet werden kann und möglichst viele

¹<http://sourceforge.net/projects/tinyos/>

Techniken und Programmierparadigmen erhalten bleiben. Beide Betriebssysteme implementieren daher auch den POSIX-Standard und sind vom Umfang und von der Größe mehr für Embedded-PCs geeignet als für Sensorknoten. Außerdem werden bei diesen Real-Time-Betriebssystemen die Energieanforderungen an Sensornetze nicht ausreichend berücksichtigt.

Bei der Entwicklung von TinyOS standen folgende Gesichtspunkte im Vordergrund ([Hill u. a. (2000)] und [Levis u. a. (2004)]):

- Betriebssystem-Architektur, die möglichst viele aktuelle und auch zukünftige Sensorknoten mit unterschiedlicher Hardware unterstützen kann
- In TinyOS entwickelte Software soll auf unterschiedlichen Hardwareausstattungen und Sensorknoten lauffähig sein, ohne den Programmcode anpassen zu müssen
- Die speziellen Aspekte eines Sensornetzwerks und der Sensorknoten beim Design berücksichtigen, wie beschränkte Ressourcen, einen hohen Grad an nebenläufigen Operationen und Robustheit der Anwendung
- Effektiver Umgang mit der Kommunikationseinheit, da für senden, empfangen und lauschen auf dem Medium besonders viel Energie verbraucht wird
- Skalierbarkeit: Ein Sensornetz aus einer Vielzahl von Sensorknoten soll möglich sein

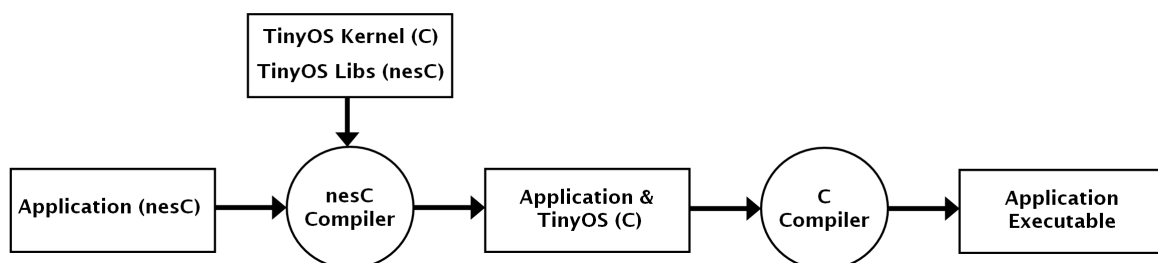


Abbildung 4.1: TinyOS Build-Prozess

Um die genannten Anforderungen zu erfüllen, wurde daher bei der Entwicklung von TinyOS ein, gegenüber anderen RTOS, anderer Ansatz gewählt. TinyOS ist modular aufgebaut und beim Design von TinyOS wurde eine Komponentenarchitektur gewählt, bei der sowohl die Anwendungen, als auch das Betriebssystem aus einzelnen Komponenten bestehen. Betriebssystem und Anwendung sind eng miteinander verzahnt. Beim Build-Prozess eines Programms werden nur die benötigten Abschnitte des Betriebssystems zusammen mit der Anwendung integriert und kompiliert und können am Schluss in einem lauffähiges Programm auf die Sensorknoten übertragen werden (vgl. Abbildung 4.1). Anwendungen werden in nesC, eine an C angelehnte Sprache, die speziell für Sensorknoten entwickelt wurde, programmiert (siehe [Gay u. a. (2003a)], [Gay u. a. (2003b)] und [Gay u. a. (2005)]).

4.1 Komponentenmodell

Wie bereits dargelegt, besteht ein TinyOS-Programm aus verschiedenen Komponenten. Im folgenden soll der innere Aufbau einer Komponente erläutert werden. Eine Komponente enthält zwei Bereiche. Sie besitzt einen *Interface*- und einen *Implementation*-Abschnitt (vgl. Abbildung 4.2).

Im Interface-Bereich wird der Komponententyp festgelegt. Die beiden möglichen Typen sind: *Module* oder *Configuration*. Eine Komponente kann Schnittstellen zu anderen Komponenten besitzen. Daher kann in diesem Bereich noch festgelegt werden, welche *Commands* und *Events* die Komponente benutzt (*uses*) bzw. zur Verfügung stellt (*provides*).

Im Implementation-Abschnitt der Komponente befindet sich, wenn es sich bei der Komponente um ein Modul handelt, der eigentliche Programmcode, der das Verhalten festlegt. Ist die Komponente dagegen vom Typ Configuration findet dort das so genannte *Wiring* statt. Beim Wireing werden die Schnittstellen verschiedener Komponenten verbunden. Dabei werden immer Events einer Komponente Commands einer anderen zugeordnet.

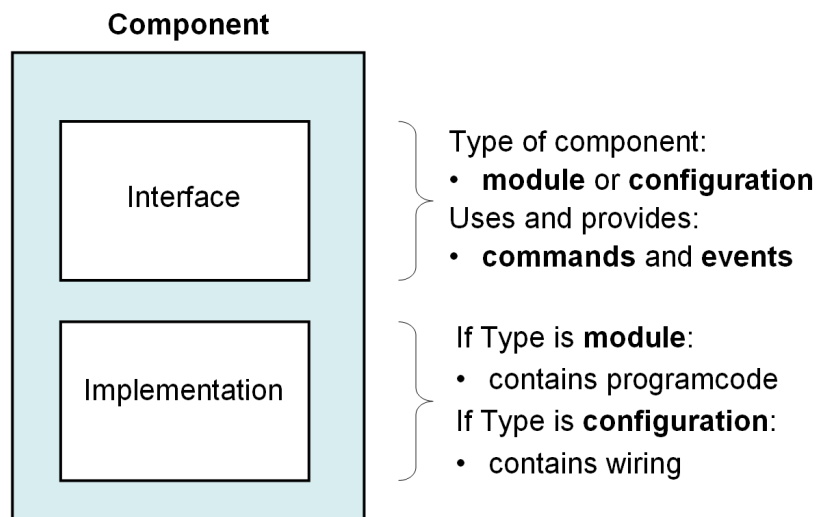


Abbildung 4.2: TinyOS Komponentenmodell

Durch das Wireing erfolgt, wie erwähnt, die Verknüpfung mit anderen Komponenten. Die Spezifikation einer Komponente muss im Interface-Abschnitt veröffentlicht werden, damit sie von anderen Komponenten genutzt werden kann. Eine Komponente kann dort verschiedene Commands zur Verfügung stellen, die vom Benutzer aufgerufen werden können und kann darüberhinaus vom Aufrufer verlangen bestimmte Event-Handler zu implementieren, damit diese von der Komponente angesprochen werden können (weiteres dazu im Ab-

schnitt 4.3). Komponenten müssen in ihrem Implementation-Abschnitt die Commands, die von ihnen zur Verfügung gestellt werden, implementieren.

4.2 Nebenläufigkeit

Da Sensorknoten oft mehrere Dinge gleichzeitig tun müssen, wie z. B. verschiedene Sensoren auswerten, Berechnungen durchführen oder Nachrichten weiterleiten, ist ein nebenläufiges Ausführungsmodell am besten dafür geeignet. Um möglichst wenig Energie zu verbrauchen und oft in den Schlafzustand übergehen zu können, wurde bei TinyOS ein ereignisgesteuertes Ausführungsmodell gewählt. Nebenläufige Operationen können in TinyOS mit *Tasks* und *Events* modelliert werden.

Tasks sind für nicht zeitkritische Berechnungen gedacht. Sie werden in eine FIFO-Queue eingereiht und nacheinander abgearbeitet. Tasks können sich nicht gegenseitig unterbrechen, so dass eine später aufgerufene Task auf das Ende der vorherigen warten muss. Wenn die Task-Queue leer ist und damit alle Tasks beendet sind, geht TinyOS in einen Schlafzustand über, um Energie zu sparen. Der Schlafzustand kann durch einen Hardware-Interrupt, der ein Event auslöst, wieder verlassen werden. Komponenten können Tasks mit dem Schlüsselwort *post* starten. Die Task wird dadurch in die Queue eingestellt und danach kann die Programmausführung fortgesetzt werden, später wird die Task dann vom Scheduler gestartet.

Events werden ausgelöst durch einen Hardware-Interrupt, z. B. von einem Timer oder Kommunikationsmodul. Außerdem wird das Ende einer *Split-Phase Operation* (siehe Abschnitt 4.3) durch ein Event signalisiert (mit Hilfe des *signal*-Schlüsselworts). Events können selbst nicht unterbrochen werden, jedoch laufende Tasks unterbrechen. Sie sollten nur kurze Berechnungen enthalten und dann ggf. Tasks aufrufen. Da nur durch Events ein aufwachen aus dem Schlafzustand möglich ist, spielen sie eine große Rolle beim Anwendungsdesign.

4.3 Split-Phase Operation

Mit Hilfe der so genannten *Split-Phase Operations* werden in TinyOS alle langlaufenden Operationen modelliert. Dadurch wird der Start und das Ende einer Operation in zwei getrennte Abschnitte aufgeteilt. Wird eine Split-Phase Operation gestartet, kehrt danach die Ablaufsteuerung sofort wieder an den Aufrufer zurück und das Programm kann dort fortgesetzt werden. Ist die Split-Phase Operation beendet, signalisiert sie dies durch auslösen eines Events beim Aufrufer.

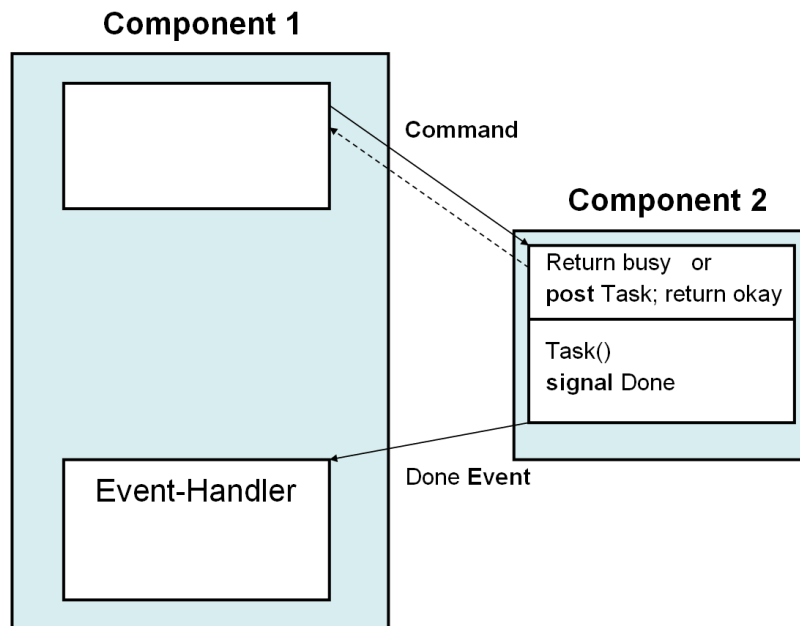


Abbildung 4.3: TinyOS: Ablaufmodell einer split-phase operation

Ein Beispiel für eine Split-Phase Operation ist das Senden einer Nachricht in TinyOS. In einer Komponente wird die `send()`-Funktion der Kommunikationskomponente aufgerufen. Wenn das Senden beendet ist, signalisiert die Kommunikationskomponente dies durch auslösen des `sendDone()`-Events. Dadurch, dass die aufrufenden Komponenten das Kommunikationsmodul benutzt, muss sie das `sendDone()`-Event implementieren. Abbildung 4.3 zeigt ein Ablaufmodell einer Split-Phase Operation.

5 Current Work

In diesem Kapitel sollen einige maßgebliche Forschungsprojekte im Themenbereich Sensornetze und Rescue vorgestellt und diese in Bezug zu dieser Arbeit bewertet werden.

5.1 Smart Dust

Das Smart Dust-Projekt wurde an der University of California in Berkeley mit Unterstützung der DARPA (Defense Advanced Research Projects Agency)¹ bereits im Jahr 1998 gestartet [Warneke u. a. (2001)]. Obwohl der Projektabschluss schon eine Weile zurück liegt, soll es hier auch aus dem Grund erwähnt werden, da es als Ausgangspunkt für viele Technologien, die im Bereich Sensornetzwerke entwickelt wurden, bezeichnet werden kann. Der Begriff Smart Dust wurde durch dieses Projekt geprägt und steht seitdem für die dort erforschten Themengebiete. Außerdem wurden viele Technologien untersucht oder entwickelt, die von grundlegender Wichtigkeit sind.

Der Untertitel des Smart-Dust Projekts – *Autonomous sensing and communication in a cubic millimeter* – weist bereits auf die Ziele hin. So sollte zum einen eine Sensor- und Kommunikationsplattform entwickelt und zum anderen ein Sensornetz daraus aufgebaut werden. An die Sensorknoten wurden dabei einige grundlegende Anforderungen gestellt. So sollten sie eine Größe von nur 1 mm³ haben, aber in der Produktion nur geringe Kosten verursachen, so dass es möglich sein sollte, so die Projektverantwortlichen, tausende davon ausbringen zu können. Des weiteren sollten viele unterschiedliche Sensortypen realisiert werden, die mindestens eine Recheneinheit, eine bidirektionale Kommunikationseinheit und eine Stromquelle besitzen. Um diese Ziele zu erreichen, waren weitreichende Fortschritte in den Bereichen Integration der verschiedenen Bauteile, Miniaturisierung und Energie-Management nötig (vgl. [Warneke u. a. (2002)]). In Abbildung 5.1 ist der konzeptionelle Aufbau eines Smart Dust-Sensorknotens dargestellt.

Eine Besonderheit dieses Projekts stellt die Art des Nachrichtenaustausches dar. Es soll zum Nachrichtenaustausch anstatt der sonst üblicherweise benutzten Funkübertragung eine optische Kommunikationseinheit eingesetzt werden. Als Trägermedium dient ein Laser-

¹DARPA MEMS Programm - <http://www.darpa.mil/mto/mems/index.html>

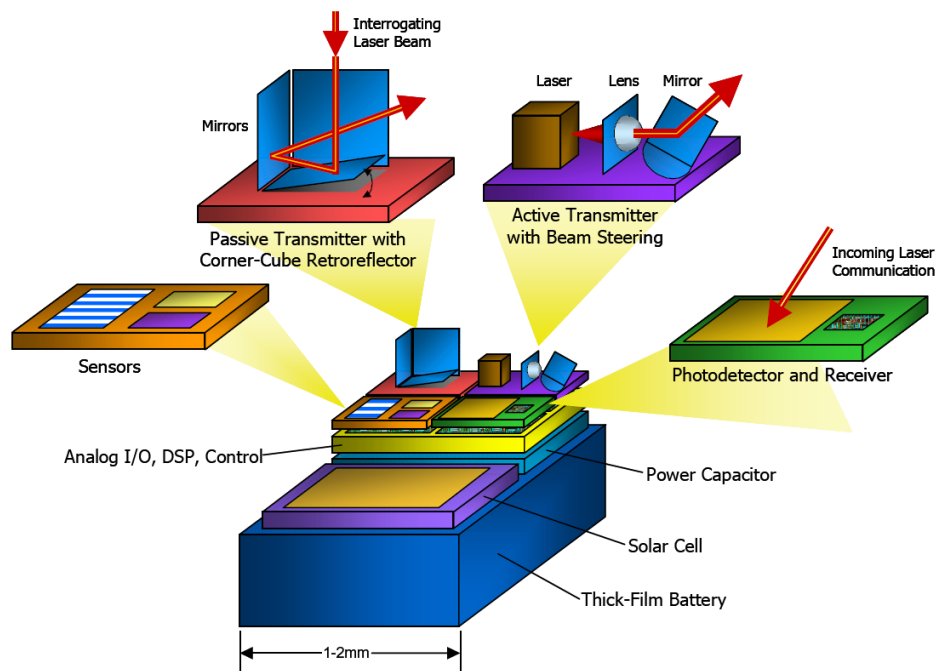
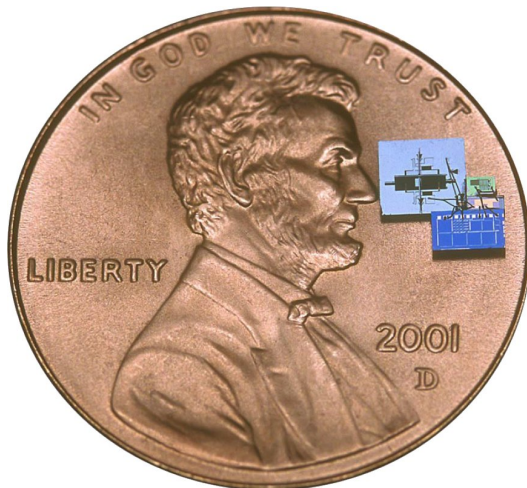


Abbildung 5.1: Smart Dust - konzeptioneller Aufbau einer Mote [Warneke u. a. (2002)]

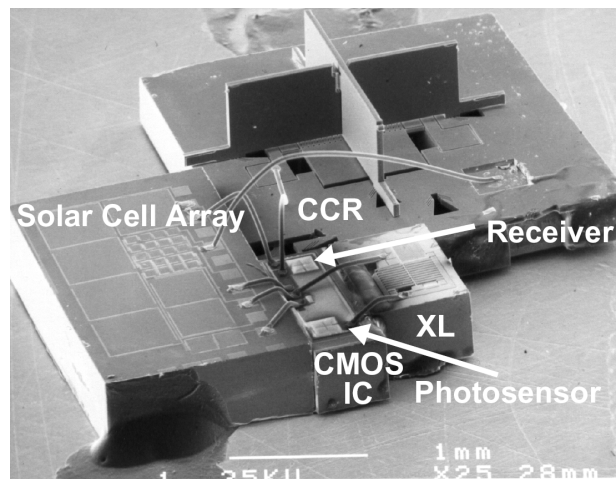
strahl. Dies wird damit begründet, da durch die optische Kommunikation wesentlich weniger Energie verbraucht wird, höhere Datenraten erzielt werden können und kleinere Strukturen möglich sind. Ein erheblicher Nachteil der optische Kommunikation ist es jedoch, dass eine direkte Sichtverbindung und eine genau Ausrichtung des Laserstrahls zum Nachrichtenaustausch nötig ist. Zur Kommunikation wurden sowohl passive wie auch aktive Komponenten entwickelt. Motes, die das passive System zur Kommunikation nutzen, besitzen keine eigene Lichtquelle. Durch drei z. T. bewegliche Spiegel wird ein Laserstrahl, der auf den Sensorknoten gerichtet ist reflektiert (entspricht einer 1) oder eben nicht (entspricht einer 0). Mit diesem System, als *Corner-Cube Retroreflector (CCR)* bezeichnet, ist es möglich ca. tausend Mal pro Sekunde zwischen den beiden Zuständen zu wechseln. Ein Nachteil dieser passiven Art der Kommunikation ist, dass damit keine aktive Datenübertragung möglich ist, sondern sie muss erst von außen initiiert werden. Aus diesem Grund wurde auch eine aktive Kommunikationskomponente realisiert. Dazu wurde ein Laser-Transmitter entwickelt, mit dem Motes selbstständig Nachrichten versenden können (vgl. [Warneke u. a. (2001)]).

Es wurden in dem Projekt tatsächlich verschiedene Sensorknoten mit unterschiedlichen Sensortypen entwickelt, wie z.B Helligkeits-, Temperatur-, Vibrations- und Mag. Feldstärkesensoren. Das Ziel die Größe der Sensorknoten auf 1 mm^3 zu begrenzen wurde fast erreicht, so dass die kleinste Smart Dust-Mote eine Größe von 6.6 mm^3 aufweist. Die Golem Dust-Mote (siehe Abbildung 5.2), die hier stellvertretend näher beschreiben werden soll, besitzt

eine Größe von 16 mm^3 . Sie besteht aus einem Photosensor, einem ADC (Analog-Digital Converter), einem optischen Empfänger, einem CCR für optische Übertragungen, einem Solarzellenfeld und einem einfachen FSM(Finite State Machine)-Controller.



(a) Größenvergleich: Mote – 1 Cent Münze



(b) Rasterelektronenmikroskop-Aufnahme

Abbildung 5.2: Golem Dust [Warneke u. a. (2002)]

5.1.1 Bewertung

In dem Smart Dust-Projekt wurden viele interessante Aspekte entwickelt, durch die sich zahlreiche mögliche Anwendungsgebiete ergeben. Auch war es Ursprung für einige weitere Projekte, in denen beispielsweise die in dieser Arbeit genutzten so genannten Berkeley-Motes wie die TelosB-Mote oder das TinyOS-Betriebssystem entwickelt wurden (s. a. Kapitel 4). Damit können die Erkenntnisse des Smart Dust-Projekts als Grundlage und wichtige Voraussetzung auch für diese Masterarbeit angesehen werden. So ist z. B. die Miniaturisierung von Sensorknoten, auch in Bezug auf den in dieser Arbeit zu Grunde liegenden Anwendungsfall, ein wichtiger Gesichtspunkt. Auch hier sind kleine Knoten nötig, beispielsweise wenn sich möglichst viele Sensoren in einem Behälter am Körper eines Feuerwehrmannes befinden sollen.

Die optische Kommunikation, eine Besonderheit dieses Projekts, eignet sich dagegen nicht für den in dieser Arbeit vorgesehenen Einsatzzweck. Da bei einem Brand im Allgemeinen auch Rauch und Russpartikel entstehen, wäre damit keine zuverlässige Verbindung mehr zwischen den Motes gegeben.

5.2 intelligent Building Response (iBR)

Das intelligent Building Response (iBR)-Projekt wird am NIST-BFRL (National Institute of Standards and Technology-Building and Fire Research Laboratory) durchgeführt ([NIST (2006)]). Ziel ist es einen Standard zu definieren, um Feuerwehr, Polizei oder anderen Sicherheitsbehörden bei einem Notfall einen Zugang zu Gebäudeinformationssystemen zu ermöglichen.

Kontroll- und Überwachungssysteme moderner Gebäude verfügen über eine Vielzahl von Sensoren innerhalb des Gebäudes, nutzen aber oft unterschiedliche Bussysteme und Datenstrukturen und sind inkompatibel zueinander. Sie werden meist im Gebäude in einem Kontrollzentrum zusammengeführt und sind von dort steuerbar. Außerhalb des Gebäudes sind diese Systeme nicht nutzbar und es kann nicht auf sie zugegriffen werden, obwohl diese Informationen bei einem Feuer für die Rettungskräfte sehr hilfreich wären. Durch das iBR-Projekt soll den Rettungskräften ein Zugriff von Außen auf die Gebäudedaten ermöglicht werden und dies bereits bei der Anfahrt bzw. schon in der Leitstelle, nachdem dort ein Feuer gemeldet wurde. Außerdem soll die Leitstelle die Möglichkeit haben beispielsweise eine Gegensprechanlage, die zur Brandmeldung dient, zu nutzen und Kontakt mit der Person vor Ort aufzunehmen.

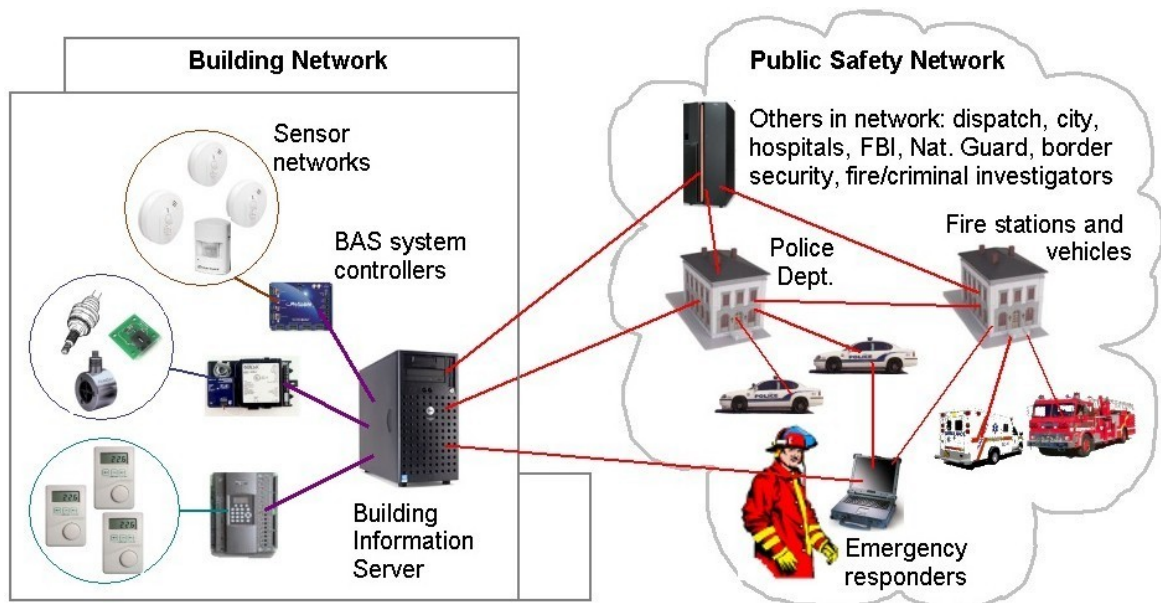


Abbildung 5.3: Vernetzung der relevanten Rettungskräfte mit den Gebäudesystemen durch BACnet [NIST (2006)]

Für die Rettungskräfte hätten u. a. folgende Informationen der Kontroll- und Überwachungssysteme große Bedeutung:

- Wo brennt es?
- Wo gibt es Rauch?
- Wo befinden sich Personen im Gebäude?
- Welche Gebäudesysteme arbeiten noch?
- In welchen Räumen sind die Lichter an?
- Welche Türen sind geöffnet?

Durch diese Informationen könnten sich die Rettungskräfte bereits vor der Ankunft über die Gegebenheiten vor Ort informieren und damit schneller mit der Rettung von Personen im Gebäude und der Brandbekämpfung beginnen.

Als einheitlicher Standard für die Gebäudekommunikation wurde BACnet (Building Automation and Control Network) durch das NIST mitinitiiert. BACnet definiert eine Kommunikationsprotokoll für die Gebäudeautomation und ist ein ISO Standard. Durch BACnet soll die Interoperabilität zwischen verschiedenen Geräten unterschiedlicher Hersteller gewährleistet werden.

5.2.1 Bewertung

Das intelligent Building Response-Projekt zeigt wie wichtig es ist, den Rettungskräften so früh wie möglich Informationen über die Lage am Einsatzort zur Verfügung zu stellen, um die Brandbekämpfung so effektiv wie möglich durchzuführen. Dieses Konzept findet sich auch in dieser Arbeit wieder, denn auch hier stehen, durch die Ausrüstung von Gebäuden mit Sensoren und die Darstellung dieser Daten in der Leitstelle, schon Gebäudeinformationen zur Verfügung, bevor die Retter am Unglücksort eingetroffen sind.

Eines der Kernpunkte des iBR-Projekts, die Nutzung von Standards, zeigt erneut wie wichtig es ist, auf Standards und offenen Schnittstelle zu achten, um Probleme durch inkompatible Systeme zu vermeiden. Dieser Grundsatz soll auch in dieser Arbeit berücksichtigt werden.

Zusammenfassend lässt sich feststellen, dass dieses Projekt viele wertvolle Hinweise und Ideen für die Planung und den Entwurf dieser Masterarbeit liefert. Dagegen werden zur praktischen Umsetzung und zu den verwendeten Technologien, kaum hilfreiche Informationen gegeben. Zu beachten bleibt auch der Punkt der Datensicherheit und informationellen Selbstbestimmung. Außerdem muss gewährleistet sein, dass keine permanente Überwachung der Personen in dem Gebäude stattfindet.

5.3 FIRE (Fire Information and Rescue Equipment)-Project

Das FIRE-Projekt wird im wesentlichen in einer Kooperation zwischen der UC Berkeley und dem Chicago Fire Department durchgeführt ([Wright u. a. (2006)]). In diesem Projekt sollen Hilfsmittel zur Verbesserung der Information und Kommunikation entworfen und entwickelt werden, um die Sicherheit und Effizienz eines Notfalleinsatzes zu verbessern. Der Fokus wird dabei auf die Bekämpfung von Desastern in großen innerstädtischen Gebäuden, wie z. B. Kaufhäusern oder in Industrieanlagen gelegt.

In Chicago wurde nach den Vorkommnissen des 11. September entschieden, dass alle Gebäude über 80 ft (entspricht ca. 24 m) den Sicherheitsbehörden digitale Gebäudepläne zur Verfügung stellen müssen. Außerdem müssen diese Gebäude ein einfaches Informations- und Kommunikationssystem besitzen. Aufbauend auf dieser Grundlage sollen Hilfsmittel für die Rettungskräfte entwickelt werden.

Im FIRE-Projekt wird mit Hilfe eines drahtlosen Sensornetzwerks, als *SmokeNet* bezeichnet, die Position und die Vitaldaten von Feuerwehrmännern in einem Gebäude ermittelt und damit dem Einsatzleiter vor Ort die Möglichkeit gegeben, einen Überblick, über den aktuellen Status seiner Kräfte zu bekommen. Außerdem soll im FIRE-Projekt erkannt werden, wie weit sich das Feuer im Gebäude ausgebreitet hat. Dem Feuerwehrmann soll auf einem *FireEye* genannten Head-Mounted-Display (HMD) ein Gebäudeplan eingeblendet werden, auf dem seine eigene Position und die Position der anderen Rettungskräfte im Gebäude angezeigt wird. Das SmokeNet-Sensornetzwerk wird im folgenden detaillierter dargestellt, da es viele Parallelen zu dem Zielen dieser Arbeit aufweist.

5.3.1 SmokeNet

In dem FIRE-Projekt wird davon ausgegangen, dass das drahtlose Sensornetzwerk im Idealfall bereits im Gebäude installiert ist, bevor es zu einem Notfall kommt. Ist dies nicht der Fall, sollen Sensorknoten durch die Feuerwehrkräfte im Gebäude ausgelegt werden und dadurch nach und nach ein im gesamten Gebäude verfügbares Sensornetz entstehen. Es wird allerdings darauf hingewiesen, dass für die volle Funktionalität ein bereits vorhandenes Sensornetzwerk benötigt wird.

Das Sensornetz ist aufgebaut aus den Berkeley-Motes Telos und Mica2. Es gibt stationäre und auch mobile Sensorknoten. Es werden im FIRE-Projekt, anstelle der konventionellen Rauchmelder, Rauch- und Temperatursensorknoten (siehe Abbildung 5.4 (a)) im Gebäude angebracht. Außerdem sollen „Ampel“-Sensorknoten (siehe Abbildung 5.4 (b)) auf beiden Seiten jeder Tür installiert werden. Diese Sensoren bestehen aus drei farbigen LEDs und



(a) Rauch- und Temperatursensor



(b) „Ampel“-Sensor

Abbildung 5.4: SmokeNet: stationäre Sensorknoten [Steingart u. a. (2007)]

sind zur visuellen Anzeige für die Rettungskräfte, aber auch für Personen, die sich noch im Gebäude befinden, bestimmt. Sie zeigen den Status des dahinter liegenden Raums an. So signalisiert eine rote LED, dass im Raum Gefahr droht, eine gelbe bedeutet, dass der Status des Raums nicht bekannt ist und wenn die grüne LED leuchtet, kann der Raum gefahrlos betreten werden. Durch dieses System werden Fluchtwege einfach und schnell wahrnehmbar signalisiert.

Die mobilen Sensorknoten werden durch die Feuerwehrleute und den Einsatzleiter gebildet, die sich irgendwo im Gebäude aufhalten können. Die Feuerwehrmänner sind mit einem System aus einer Mica2-Mote, einem kleinen wearable Computer und einem HMD ausgerüstet. Über das Sensornetz wird die Position der Feuerwehrmänner im Gebäude ausgetauscht und auf dem HMD angezeigt, ebenso wie die Ausbreitung des Feuers. Außerdem sollen über das Sensornetz Daten zur Überwachung des Gesundheitszustands des Feuerwehrmannes, sowohl an den Einsatzleiter, als auch an die anderen Feuerwehrleute, übermittelt werden. So sollen die Feuerwehrkräfte mit Sensoren für ein Monitoring der Restatmosphäre und der Herzfrequenz ausgestattet werden. Des Weiteren dient das Netz dazu, Nachrichten mit dem Einsatzleiter auszutauschen. Dazu gehören Anweisung des Einsatzleiters an den Feuerwehrmann, die auf dem HMD dargestellt werden und es soll außerdem eine einfache Ja/Nein-Kommunikation zwischen Feuerwehrmann und dem Einsatzleiter realisiert werden.

5.3.2 Bewertung

Das FIRE-Projekt besitzt einige Ähnlichkeiten und Parallelen zu dieser Masterarbeit. So soll hier ebenfalls ein Sensornetzwerk aus Berkeley-Motes aufgebaut werden. Auch der Einsatz eines HMD im Rescue-Umfeld ist an der HAW Hamburg geplant (vgl. [Hinck (2007)]). Die Idee der „Ampel“-Sensorknoten ist ein interessanter Ansatz, der sich auch in das in dieser Arbeit geplante Sensornetz integrieren ließe.

Leider werden in den zur Verfügung stehenden Unterlagen viele Realisierungsdetails unklar gelassen. So wird beispielsweise über die Art der Positionsberechnung nichts mitgeteilt, es könnte aber hier ein Triangulationsverfahren zum Einsatz kommen. Da an jeder Tür zwei „Ampel“-Sensorknoten und dazu die Rauch- und Temperatursensoren in den Räumen angebracht sind, wären genügend Sensoren vorhanden, um mit Hilfe des Abstands zu diesen festen Knoten eine Position zu berechnen. Ebenso wären mehr Informationen darüber wünschenswert wie z. B. die Ja/Nein-Kommunikation über das Sensornetz realisiert werden soll, auch in Bezug auf die dafür zu erwartende Datenmenge.

Zusammenfassend kann festgestellt werden, dass das FIRE-Projekt viele interessante Aspekte besitzt, die auch für die Realisierung dieser Masterarbeit wertvoll wären. Es sind jedoch kaum Detailinformationen verfügbar, inwieweit das hier beschriebene umgesetzt werden konnte und welche Techniken dafür benutzt wurden.

5.4 wearIT@work

Ziel des wearIT@work-Projekts ist es wearable Technologien in unterschiedlichen Arbeitsumgebungen zu entwickeln und zu testen (vgl. [wearIT@work (2007)]). Im Bereich *Emergency Rescue*, einem von vier Anwendungsbereichen, sollen Technologien im Rescue-Umfeld erforscht und konkrete Lösungen entwickelt werden, die die Einsatzkräfte bei Notfall- und Rettungseinsätzen in ihrer Arbeit unterstützen.

Es sollen, in Zusammenarbeit mit der Pariser Feuerwehr, Technologien entwickelt werden, um die Kommunikation, Kollaboration und den Informationsaustausch der Rettungskräfte zu verbessern. Dazu wird hauptsächlich an wearable Technologien im Bereich der Ausrüstung des einzelnen Feuerwehrmanns geforscht. Parallelen zu dieser Arbeit ergeben sich insofern, da auch Sensorknoten für den Aufbau eines Ad-Hoc Sensornetzes genutzt werden sollen.

Ein Sensornetz soll eingesetzt werden, um die bisher genutzte Lifeline der Feuerwehr zu verbessern bzw. zu ersetzen. Die Lifeline ist, wie in Kapitel 2 beschrieben, ein Drahtseil, das ein Feuerwehrmann hinter sich her führt, wenn er sich in einem Gebäude befindet. Damit ist es ihm möglich den Weg zurück zum Ausgang wiederzufinden, wenn er sich verirrt hat oder wenn die Sicht im Gebäude sehr stark eingeschränkt ist. Von den Feuerwehrleuten, die

sich im Gebäude befinden, sollen automatisch Sensorknoten in regelmäßigen Abständen im Gebäude ausgelegt werden. Über das damit aufgebaute Sensornetz soll der Rückweg berechnet und dem Feuerwehrmann in Form von Richtungspfeilen auf einem kleinen wearable Computer oder auf einem Head-Mounted-Display angezeigt werden.

5.4.1 Bewertung

Da konkrete Lösungen im wearIT@work-Projekt erstellt werden sollen und damit Technologien, die sich im Rahmen der Zusammenarbeit mit der Pariser Feuerwehr einsetzen lassen, gibt es einige Unterschiede zu dem in dieser Arbeit gewählten Szenario. So umfasst der Einsatzbereich der Pariser Feuerwehr das gesamte Stadtgebiet von Paris und es ist daher wenig sinnvoll, Gebäude schon im Vorhinein mit einem Sensornetz auszustatten. Dies kann zwar für besonders wichtige Gebäude, wie z. B. den Louvre zweckmäßig sein, jedoch sind Einsätze in diesen so vorbereiteten Gebäuden im Vergleich zum normalen Einsatzalltag eines Pariser Feuerwehrmannes sehr selten. Aus diesem Grund wird hier erst durch die Feuerwehrleute ein Sensornetz aufgebaut, wenn sie am Einsatzort eintreffen.

Parallelen zu dieser Masterarbeit ergeben sich in Bezug auf die Auslegung von Sensorknoten im Gebäude durch Feuerwehrleute. Auch die dort zum Einsatz kommenden Sensorknoten sind die gleichen wie die in dieser Arbeit genutzten.

5.5 RELATE: Relative Positioning of Mobile Objects in Ad hoc Networks

Es soll im folgenden kurz auf das RELATE-Projekt eingegangen werden, da es einen ähnlichen Ansatz wie diese Arbeit besitzt, jedoch eine andere Technik nutzt. Außerdem hat dieses Projekt als ein von der EU gefördertes internationales Forschungsprojekt einen wesentlich größeren Umfang.

Beim RELATE-Projekt liegt der Schwerpunkt, wie im Titel bereits zu erkennen, auf der Positionsbestimmung. Es soll die relative Position von mobilen Objekten in einem Raum berechnet werden. Allerdings wird in diesem Projekt ein anderes System zur Ortsbestimmung, als in den bisher vorgestellten Arbeiten genutzt. Es wird mit Hilfe von Ultraschallimpulsen die Entfernung von Objekten zueinander bestimmt. Dieses Verfahren zur Positionsbestimmung wurde, wie bereits erwähnt, innerhalb der HAW Hamburg von Sebastian Gregor ([Gregor \(2006\)](#)) untersucht und ebenfalls zur Positionsbestimmung eingesetzt. Im HAW Rescue-Projekt hat sich auch Steffen Hinck damit befasst (vgl. [[Hinck \(2006\)](#)] und [[Hinck \(2007\)](#)]).

Das RELATE-System wird außerdem genutzt im LifeNet-Konzept, an dem die HAW Hamburg durch Mirko Gerling vertreten ist. Bei LifeNet wird ebenfalls der Ansatz verfolgt, die Lifeline der Feuerwehr durch ein Sensornetz zu ersetzen. Innerhalb des Sensornetzes bestimmen die Sensorknoten ihre Position mit Hilfe des RELATE-Systems (vgl. [\[Klann u. a. \(2007\)\]](#)).

6 Entwurf

In diesem Kapitel soll ein Systementwurf der zu realisierenden Ziele erarbeitet werden (vgl. auch Kapitel *Ziele und Vorhaben*). Das gesamte Vorhaben ist eingebettet in ein Rescue-Szenario, so dass die besonderen Bedingungen beispielsweise eines Gebäudebrands auf einem Flughafen mit zum Tragen kommen.

Es soll mit Hilfe von Motes ein drahtloses Sensornetzwerk in einem Gebäude aufgebaut werden. Das Sensornetz kann zum einen genutzt werden, um ein Gebäude mit Hilfe von Temperatursensoren zu überwachen und zum anderen, um die Position von Rettungskräften im Gebäude, die ebenfalls mit einem Sensorknoten ausgestattet sind, zu ermitteln. Die berechneten Positionen der mobilen Sensorknoten sowie die Sensorwerte sollen in einer Leitstandanwendung visuell dargestellt und für die Einsatzplanung genutzt werden.

Die Entwicklung des Gesamtsystems lässt sich trennen in die beiden Bereiche:

1. Aufbau eines Sensornetzes
2. Entwurf einer Leitstand-Applikation

Dieses Aufteilung wurde gewählt, da es sich um nur lose gekoppelte Teile handelt, die daher auch getrennt betrachtet werden können. Die Funktionalität des Gesamtsystems ergibt sich jedoch aus der Verknüpfung beider Bereiche.

Bevor ein Systementwurf durchgeführt wird, soll im nächsten Abschnitt auf die allgemeinen Anforderung des zu implementierende System eingegangen werden. Daran anschließend werden kurz die Grundlagen erläutert, die sich aus der Entscheidung für eine konkrete Sensorplattform zur Realisierung der geplanten Vorhaben ergeben. Nachdem die Hard- und Softwaregrundlagen feststehen, kann darauf aufbauend eine Anforderungsanalyse und ein Systementwurf der einzelnen Anwendungsteile durchgeführt werden.

6.1 Anforderungen

Es sind zunächst die Anforderungen an das zu erstellende System zu definieren. Um die gesteckten Ziele zu erreichen und eine Gebäudeüberwachung und Positionsbestimmung

durchführen zu können, müssen bestimmte Anforderungen erfüllt sein. Dazu wird im folgenden zunächst auf die allgemeinen Anforderungen eingegangen und daran anschließend werden die Anforderungen an das Gesamtsystem dargestellt.

In Abschnitt 3.1 wurden bereits die grundsätzlichen Anforderungen an die Sensorknoten und das Sensornetz insgesamt beschrieben.

Anforderungen an die Sensorknoten:

- geringe Größe
- robust
- günstig in der Herstellung (und damit eine Fertigung aus Standardkomponenten)
- lange Betriebsdauer
- Zuverlässiges Kommunikationsverhalten

Anforderungen an das Sensornetz:

- maximale Lebenszeit
- Verlässlichkeit
- Fehlertoleranz
- Skalierbarkeit

Für die zu erstellende Leitstandanwendung ergeben sich ebenfalls einige allgemeine Anforderungen. Die Anwendung soll eine Aufteilung zwischen Auswertung der Sensornachrichten auf der einen Seite und visuelle Darstellung im Leitstand auf der anderen Seite erkennen lassen. Somit wird eine Trennung von Daten und Darstellung erreicht.

Dieser Ansatz wird im Entwurfsmuster Model View Controller (MVC) beschrieben und bietet viele Vorteile. Durch die Implementierung dieses Designmusters besteht u. a. die Möglichkeit, das Sensornetz und die Datenauswertung der Anwendung in eine andere umfangreichere Leitstandanwendung, wie sie in der Masterarbeit von Andreas Piening (vgl. [Piening (2007)]) entstehen soll, zu integrieren. Außerdem fällt es so leichter, beide Teile auf unterschiedlichen Rechnern zu verteilen und die Daten beispielsweise über eine Webservice-Schnittstelle auszutauschen.

Nachdem die allgemeinen Anforderungen an die einzelnen Systemteile dargestellt wurden, sollen darauf aufbauend die Anforderungen an das Gesamtsystem definiert werden, um die Ziele dieser Masterarbeit zu erreichen.

Für die Realisierung eines Sensornetzes zur Gebäudeüberwachung mit Sensoren und für die Durchführung einer Positionsbestimmung innerhalb des Netzes sind zunächst die dargestellten Anforderungen an das Sensornetz und die einzelnen Knoten zu erfüllen. So ist beispielsweise ein zuverlässiger Nachrichtenaustausch unerlässlich, besonders in dem dieser Arbeit zu Grunde gelegten Rescue-Szenario. Um die Sensorüberwachung durchführen zu können, muss gewährleistet sein, dass die in dieser Arbeit exemplarisch herangezogenen Temperaturwerte zuverlässig ausgelesen und über das Sensornetz an die Leitstandanwendung weitergeleitet werden können. Da mit Hilfe der Sensorwerte auch eine Branderkennung durchgeführt werden soll, würden fehlerhafte Werte erhebliche Konsequenzen bewirken.

Um eine Positionsbestimmung von mobilen Sensorknoten innerhalb des Sensornetzes durchführen zu können, ist es wiederum wichtig, einen zuverlässigen Nachrichtenaustausch zwischen den Sensorknoten zu gewährleisten. Die Lokalisation soll auf Basis des Triangulationsverfahren mit Hilfe des RSSI-Werts (Details dazu in nächsten Abschnitt) durchgeführt werden. Um die Position im Leitstand nutzen zu können, darf die Abweichung zwischen berechneter und realer Position nicht zu groß sein. Es hat sich durch Tests gezeigt, dass eine Abweichung, die weniger als 3 Meter beträgt, noch hingenommen werden kann, ohne in der Leitstandanwendung eine zu große Differenz zwischen Realität und Darstellung zu bekommen. Die Abweichungen bei der Distanzmessung ergeben sich aus den Schwankungen beim RSSI-Wert, wie in Abschnitt 3.3.1.1 erläutert.

Nachdem die Anforderungen an das Gesamtsystem feststehen, kann darauf aufbauend mit ein Systementwurf entwickelt werden. Zur Spezifikation des Sensornetzes, soll auf Basis der gestellten Ziele, ein System entworfen und die funktionalen Anforderungen an die jeweiligen Einheiten herausgestellt werden. Im Anschluss daran wird für den Teil der Leitstandanwendung ebenso verfahren. Zuvor soll in den folgenden beiden Abschnitten kurz auf die Grundlagen eingegangen werden, die sich aus der Entscheidung für eine konkrete Sensorplattform zur Realisierung der geplanten Vorhaben ergeben.

6.2 Hard- und Softwaregrundlagen

Für die Realisierung der Ziele dieser Arbeit, muss zunächst das Grundgerüst – ein Sensornetz – aufgebaut werden. Das dazu nötige Grundwissen wurde im Kapitel *Grundlagen und Analyse* diskutiert. Die Sensorknoten wurden, basierend auf dem in diesem Kapitel durchgeführten Vergleich verschiedener Sensorplattformen (vgl. Abschnitt 3.4.5) und den allgemeinen Anforderung (siehe 6.1), ausgewählt. Auf Grund dieses Wissens wurde eine Entscheidung zugunsten der Plattform mit den meisten Vorteilen, der TelosB-Mote der UC Berkeley getroffen. Diese zeichnet sich u. a. durch ihre Energieeffizienz und ein IEEE 802.15.4 konformes Funkmodul aus (vgl. Abschnitt 3.4.4).

Durch die Entscheidung für eine Plattform entstehen weitere Vorgaben für die Umsetzung. Die TelosB-Mote besitzt bereits einen integrierten Temperatursensor, so dass dieser für die Sensorüberwachung genutzt werden kann. Außerdem ist auf dem Sensorknoten das CC2420-Funkmodul verbaut, mit dem eine Messung des RSSI-Wertes möglich ist. Damit bietet sich die Nutzung dieses Verfahrens zur Distanzbestimmung (vgl. Abschnitt 3.3.1) an, da keine zusätzliche Hardware benötigt wird. Mit Hilfe des Triangulationsverfahrens kann daraus eine Position berechnet werden.

Aus der Wahl der TelosB-Mote als Sensorplattform, ergibt sich ein weiterer wichtiger Punkt für die Realisierung. Auf diesen Motes lässt sich das TinyOS-Betriebssystem einsetzen, das in Kapitel 4 vorgestellt wurde. Damit kann auf viele Basisfunktionen zurückgegriffen werden, die TinyOS bereits implementiert.

6.3 Programmierumgebung und Werkzeuge

Die Entwicklung der nesC-Applikationen für die Sensorknoten findet am PC in einer Linux Umgebung statt. Es bietet sich dazu das XubunTOS¹-System an. XubunTOS ist eine auf Ubuntu basierende Distribution, die Xubuntu 7.04 mit TinyOS 2.0.2 vereint. Xubuntu nutzt XFCE als leichtgewichtige Desktop-Umgebung. Damit steht dem Benutzer ein vorkonfiguriertes System zur Verfügung, das auch in Form einer Live-CD erhältlich ist, bei dem bereits alle benötigten Pakete für die Softwareentwicklung mit TinyOS installiert sind.

Die Entwicklung der Java-Applikation wird ebenfalls unter der XubunTOS-Distribution mit der Eclipse IDE² durchgeführt.

6.4 Aufbau eines Sensornetzes

Für die Realisierung der genannten Ziele soll ein Systementwurf erstellt werden. Es wird zunächst der Bereich des Sensornetzes betrachtet.

Das Basis-Sensornetz soll aufgebaut werden aus fest im Gebäude installierten Sensorknoten (bezeichnet als „Detector Unit“). Dazu werden die bereits beschriebenen TelosB-Motes genutzt. Die Motes befinden sich innerhalb des Gebäudes an vorher festgelegten Orten. Sie besitzen damit einen fest stehenden Standort, der später genutzt werden soll, um daraus die Position eines mobilen Knotens berechnen zu können.

¹<http://toilers.mines.edu/Public/XubunTOS>

²<http://www.eclipse.org/>

Die Sensorknoten des Basis-Sensornetzes werden zusätzlich genutzt, um die Temperaturwerte in ihrer Umgebung zu ermitteln. Dazu soll der integrierte Temperatursensor der TelosB-Mote angesprochen werden. In regelmäßigen Abständen werden die aktuellen Temperaturdaten gemessen und weitergeleitet. Die Abstände zwischen zwei Messungen wird variabel gehalten, so dass im Brandfall mit einer kleineren Periode gemessen werden kann, um schneller reagieren zu können. Dagegen sollen in Normalbetrieb Temperaturmessungen mit einer großen Periode erfolgen, so dass die Motes möglichst lang andauernde Schlafphasen einlegen können.

Die Sensorknoten senden die ermittelten Werte mit Hilfe ihres Funkmoduls an die Basisstation („Base Unit“). Bei der Basisstation handelt es sich ebenfalls um eine TelosB-Mote, die jedoch durch ihre USB-Schnittstelle (emuliert eine serielle Schnittstelle) über eine direkte Verbindung zu einem PC verfügt. Damit findet über die Basisstation der Übergang zwischen Sensornetz auf der einen und PC-Leitstandapplikation zur Auswertung und Visualisierung auf der anderen Seite statt (dazu siehe Abschnitt 6.5).

Ergänzend zum fest installierten Sensornetz sollen sich zusätzlich mobile Sensorknoten („Mobile Unit“) innerhalb des Netzes bewegen. Diese befinden sich am Körper des Rettungspersonals, das sich innerhalb des Gebäudes frei bewegt. Es soll im Leitstand die aktuelle Position der Sensorknoten und damit der Aufenthaltsort der Rettungskräfte dargestellt werden. Die Sensorknoten bestehen wiederum aus TelosB-Motes.

Um den Aufenthaltsort berechnen zu können, wird der RSSI-Wert (vgl. Abschnitt 3.3.1.1) des Chipcon CC2420-Funkmoduls der TelosB ausgewertet und mit Hilfe des Triangulationsverfahrens (vgl. Abschnitt 3.3.3) die Lokalisation durchgeführt. Für die Positionsbestimmung sind mindestens drei weitere (feste) Knoten nötig. Dazu sende die mobilen Knoten in regelmäßigen Abständen eine Beacon-Nachricht. Wird diese Nachricht von den fest installierten Sensorknoten empfangen, können diese den RSSI-Wert bestimmen und an die Basisstation weiterleiten. Dort soll dann die Positionsberechnung erfolgen.

Zusätzlich sollen analog zu den festen Knoten auch die Temperaturwerte der mobilen Sensorknoten ermittelt und zusammen mit der Beacon-Nachricht weitergeleitet werden, so dass die Daten schließlich im Leitstand neben der Position der mobilen Einheiten angezeigt werden können.

Abbildung 6.1 zeigt eine Übersicht des Gesamtsystems. Es werden dort die drei unterschiedlichen Aufgabenbereiche der Sensorknoten gezeigt, wobei jeder der Sensorknoten innerhalb des Sensornetzes genau eine dieser Aufgaben wahrnimmt.

Für die Abbildung der drei unterschiedlichen Knotentypen sollen jeweils eigenständige TinyOS-Anwendungen erstellt werden, da dadurch jede Applikation so wenig Ressourcen wie möglich beansprucht und die Motes möglichst lange Schlafphasen einlegen können, um Energie zu sparen. Die Applikationen müssen in nesC programmiert werden. Im Folgenden

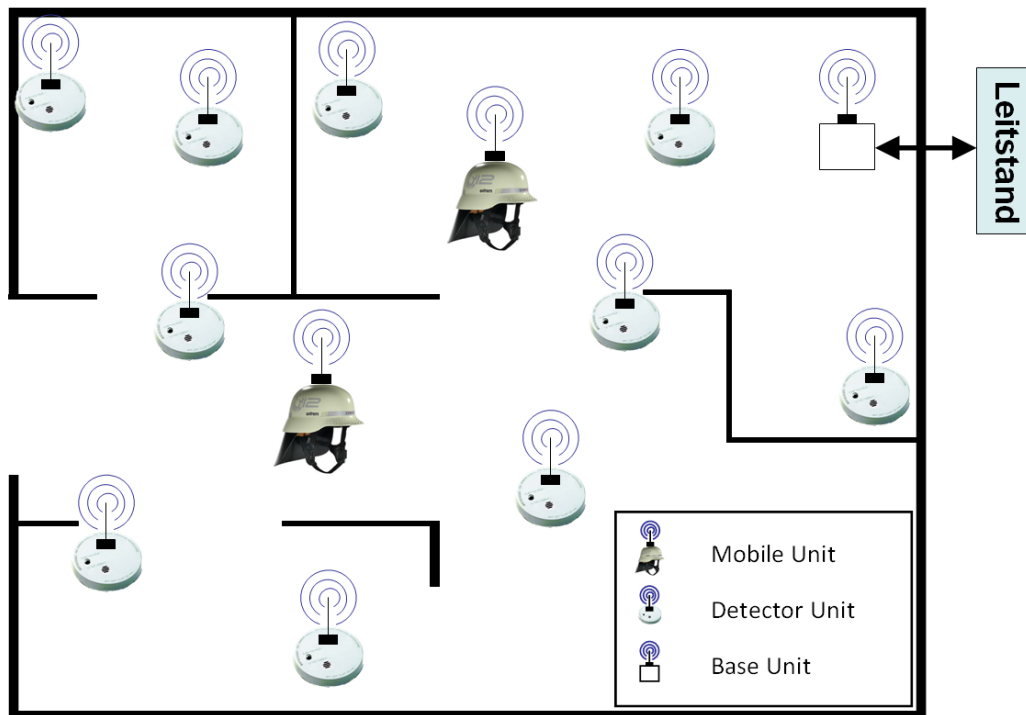


Abbildung 6.1: Topologie des Sensornetzes: System aus mobilen und festen Sensorknoten

sollen nun die funktionalen Anforderungen an die jeweiligen Knotentypen spezifiziert werden.

6.4.1 Funktionale Anforderungen

Nachdem die allgemeinen Anforderungen in Abschnitt 6.1 beschrieben worden sind, können, aufbauen auf dem Systementwurf, die funktionalen Anforderungen an die einzelnen Knotentypen spezifiziert werden. Die mobilen Sensorknoten (Mobile Units) können sich innerhalb den Sensornetzes frei bewegen. Die Position dieser Knoten soll ermittelt und der Temperaturwert im Leitstand angezeigt werden. Daraus lassen sich folgende funktionale Anforderungen ableiten:

Anforderungen an die Mobile Unit:

- in regelmäßigen Abständen ein Beacon-Signal senden
- in regelmäßigen Abständen den eigenen Temperaturwert senden

Die fest installierten Sensorknoten (Detector Units) bilden die Basis des Sensornetzes. Ihre Position ist bekannt und mit ihrer Hilfe soll die Lokalisation der mobilen Sensorknoten durchgeführt werden. Durch die Detector Units erfolgt die Sensorüberwachung des Gebäudes.

Anforderungen an die Detector Unit:

- Beacon-Signal eines mobilen Knotens verarbeiten
 - horchen auf ein Beacon-Signal
 - ermitteln des RSSI-Werts
 - senden des ermittelten RSSI-Werts an die Basisstation
- in regelmäßigen Abständen den eigenen Temperaturwert senden
 - Periode in Abhängigkeit von der gemessenen Temperatur

Die Basisstation (Base Unit) ist die Schnittstelle zwischen Sensornetz und PC. Nachrichten des Sensornetzes sollen an die Leitstand-Applikation weiterleitet werden. Folgende funktionale Anforderungen lassen sich aus dem Systementwurf ableiten:

Anforderungen an die Base Unit:

- entgegennehmen von Nachrichten (der festen Knoten)
 - RSSI-Wert der mobilen Knoten
 - Temperaturwert der mobilen Knoten
 - Temperaturwert der festen Knoten
- weiterleiten der Nachrichten des Sensornetzwerks an die PC-Applikation

6.4.2 Datenflussrichtung

Der Informationsfluss innerhalb des Sensornetzes zwischen diesen drei Sensortypen, ist in Abbildung 6.2 dargestellt. Es werden somit Daten von den mobilen Knoten (Mobile Unit) nur an die festen Knoten (Detector Unit) weitergereicht. Ebenso kommunizieren die festen Sensorknoten nur mit der Basisstation (Base Unit), die die Schnittstelle zur PC-Applikation bildet.

Diese Reihenfolge der Informationsweitergabe entsteht durch die Art des Sensornetzes und die Anzahl der jeweiligen Knotentypen. Das Verhältnis der Knotentypen untereinander stellt sich folgendermaßen dar: Es gibt sehr viele feste Sensorknoten und demgegenüber nur eine Basisstation. Dazu sei jedoch angemerkt, dass diese redundant ausgelegt werden sollte, um einen *Single Point of Failure* zu vermeiden. Zusätzlich existieren noch einige mobile

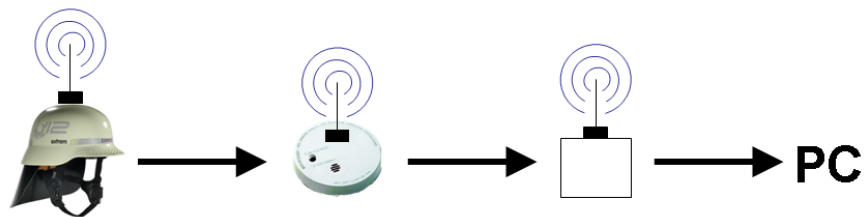


Abbildung 6.2: Richtung des Informationsflusses

Sensorknoten, die erst im Brandfall aktiv werden, da sie erst mit den Rettungskräften in das Gebäude gelangen, jedoch die Anzahl der festen Knoten nicht erreichen werden.

Damit bilden die fest installiert Sensorknoten durch ihre große Zahl das Rückgrat des Systems. Mobile Knoten sollten daher immer versuchen über die festen Sensorknoten Nachrichten an das Sensornetz zu übertragen und nicht direkt mit der Basisstation kommunizieren.

6.5 Entwurf einer Leitstand-Applikation

Nachdem im vorherigen Abschnitt der Systementwurf im Bereich des Sensornetzes beschrieben wurde, soll nun daran anschließend auf die Datenauswertung und Darstellung der Sensornetznachrichten eingegangen werden.

Nachdem mit Hilfe der Basisstation eine Schnittstelle zwischen Sensornetz und PC geschaffen worden ist, kann die Weiterverarbeitung der Sensordaten dort erfolgen. Dazu soll eine Applikation entwickelt werden, die alle Nachrichten des Sensornetzes entgegen nimmt, auswertet und grafisch darstellt. Die Anwendung soll in einem Leitstand eingesetzt werden können und dort den Einsatzleitern einen Überblick über das Geschehen innerhalb des Gebäudes ermöglichen.

Die Leitstand-Applikation soll in Java entwickelt werden. Java bietet sich deshalb als Programmiersprache an, da TinyOS bereits Basisklassen zur Kommunikation mit den Motes zur Verfügung stellt. Außerdem ist auch die Plattformunabhängigkeit und Erweiterbarkeit von Java ein gewichtiges Argument, das für eine Sprache wie Java spricht.

Es soll eine grafische Oberfläche entwickelt werden, auf der dem Benutzer alle relevanten Informationen präsentiert werden können. Die GUI besteht zum größten Teil aus einem Gebäudegrundriss auf dem die Sensoren dargestellt werden. Es sind zunächst einmal die fest im Gebäude installierten Sensorknoten anzuzeigen. Diese sollen mit ihrer Position auf einem Gebäudegrundriss dargestellt werden. Die Positionsdaten sind bekannt und ändern sich nicht. Zusätzlich sollen auch die Temperaturwerte der festen Sensorknoten angezeigt

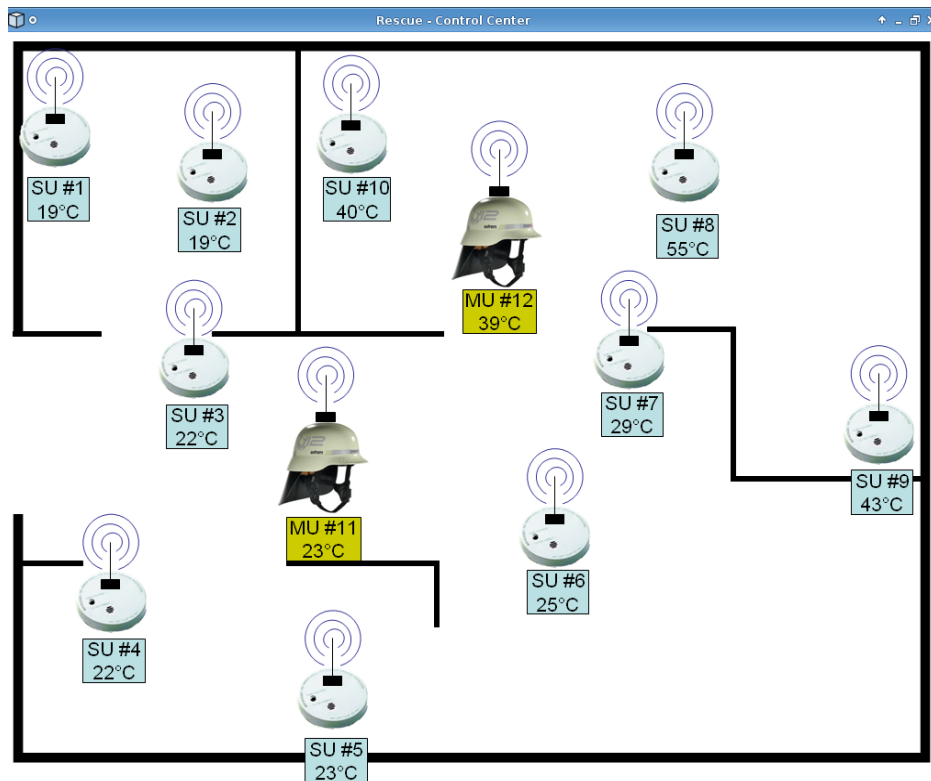


Abbildung 6.3: Entwurf der Leitstand-Anwendung: Visuelle Darstellung der Sensoren und der Rettungskräfte auf einem Gebäude-Grundriss

werden. Diese werden periodisch abgefragt und an die Basisstation übertragen. Die Sensorwerte sind neben den Sensorknoten darzustellen und regelmäßig zu aktualisieren.

Weiterhin sollen auch die mobilen Sensorknoten, die wie erwähnt das Rettungspersonal repräsentieren, auf der Karte abgebildet werden. Da sich diese innerhalb des Gebäudes bewegen und somit nicht über eine feste und bekannte Position verfügen, ist eine Lokalisation mit Hilfe des Triangulationsverfahrens der Lateration (vgl. Abschnitt 3.3.3.1) durchzuführen. Um die Position eines mobilen Knotens berechnen zu können sind die Distanzwerte – mit Hilfe des RSSI-Werts bestimmt – zu mindestens drei festen Knoten, deren Standort bekannt ist, nötig. Ist dies der Fall, kann die Lokalisation erfolgen und die Position des mobilen Knotens ebenfalls auf dem Gebäudegrundriss angezeigt werden. Zusätzlich soll auch der übertragene Temperaturwert dargestellt werden. Diese Werte müssen regelmäßig aktualisiert werden bzw. es muss periodisch eine Neuberechnung der Position erfolgen, weil sich sowohl der Temperaturwert, wie auch die Position verändern können.

6.5.1 Funktionale Anforderungen

Zur Visualisierung im Leitstand soll eine Java-Anwendung entwickelt werden. In der Anwendung findet eine Auswertung der Nachrichten des Sensornetzes statt. Es soll mit Hilfe der RSSI-Werte eine Lokalisation der mobilen Sensorknoten durchgeführt und die mobilen Knoten an der berechneten Position auf einem Gebäudegrundriss visuell dargestellt werden. Außerdem sind die Sensorwerte der festen Sensorknoten anzuzeigen.

Funktionale Anforderungen an die Leitstand-Applikation:

- empfangen und auswerten der Nachrichten des Sensornetzes
- Darstellung der Positionen der festen Knoten
- Anzeige der Temperaturwerte der festen Knoten
- Berechnung der Position der mobilen Sensorknoten
- Darstellung der Positionen der mobilen Knoten
- Anzeige der Temperaturwerte der mobilen Knoten

Das Gesamtsystem ergibt sich aus den beiden in Abschnitt [6.4](#) und [6.5](#) erläuterten Teilbereichen. Nachdem ein Systementwurf erfolgt ist und die jeweiligen Aufgaben spezifiziert wurden, kann darauf aufbauend mit der Realisierung begonnen werden.

7 Realisierung und Ergebnisse

Für die Umsetzung des Gesamtsystems soll im folgen zunächst auf den Realisierung des Sensornetzes und daran anschließend auf die Entwicklung einer Leitstand-Anwendung eingegangen werden. Am Schluss diese Kapitels wird dargestellt, in wieweit die Ziele dieser Arbeit erreicht werden konnten.

7.1 Die Sensornetzanwendungen

Für die drei Sensortypen sollen zur Modellierung, der im Entwurf beschriebenen Aufgaben, eigenständige Applikationen erstellt werden. Damit ist je eine TinyOS-Anwendung für die Mobile Unit (mobiler Sensorknoten), Detector Unit (fester Sensorknoten) und die Base Unit (Basisstation) zu erstellen. Begonnen werden soll mit der Mobile Unit, die wie in Abbildung 6.2 gezeigt, den Anfang des Informationsflusses innerhalb des Sensornetzes bildet.

7.1.1 Mobile Unit

Die Mobile Unit, die die Rettungskräfte bei sich tragen, soll zum einen die Temperaturwerte in der Umgebung des Feuerwehrmanns aufnehmen und an den Leitstand weiterleiten und zum anderen soll die Position des Feuerwehrmanns ermittelt und im Leitstand dargestellt werden.

Für die genannte Funktionalität ist eine nesC-Anwendung zu erstellen. Um die Lokalisation der Mobile Units durch die Detector Units zu ermöglichen, senden die mobilen Sensorknoten in regelmäßigen Abständen ein Beacon-Signal aus. Diese Nachricht kann von allen Motes im Sendebereich der Mobile Unit empfangen werden und es kann daraus eine Bestimmung des RSSI-Wertes erfolgen, mit dem im Leitstand eine Positionsrechnung durchgeführt werden soll. Die RSSI-Bestimmung wird von der im folgenden Abschnitt beschriebenen Detector Unit durchgeführt.

Um den Temperaturwert auszulesen, ist es zunächst erforderlich, den speziellen Temperatursensor der TelosB-Mote, den Sensirion SHT11¹ einzubinden. Dieses ist nötig, da durch den modularen Aufbau von TinyOS zwar eine allgemeine Komponente zur Ansteuerung von Sensoren vorhanden ist, der konkrete Sensortyp der Mote muss jedoch noch eingebunden werden. Es ist laut Datenblatt erforderlich, den vom Sensor ausgelesenen „rohen“ Wert in Grad Celsius umzuwandeln. Dies kann jedoch, um Ressourcen zu sparen, am besten in der Leitstand-Applikation am PC geschehen.

Der Sensorwert soll regelmäßig mit einer kurzen Periode abgefragt und an den Leitstand gesendet werden. Dies ist erforderlich, da die Mobile Units nur bei einem Feuerwehreinsatz genutzt werden und daher eine Überwachung der Rettungskräfte mit einer hohen Frequenz sinnvoll ist.

Um möglichst kurz das Medium zu belegen wird zusammen mit der Beacon-Nachricht der Temperaturwert übertragen. Damit enthält das Paket die ID des Sensorknotens, eine fortlaufende Sequenznummer und den Temperaturwert. Die Sequenznummer wird in der Leitstandanwendung für die zeitliche Ordnung, der aus dem Beacon-Nachrichten bestimmten RSSI-Werte, genutzt. Darauf wird im nächsten Abschnitt detaillierter eingegangen.

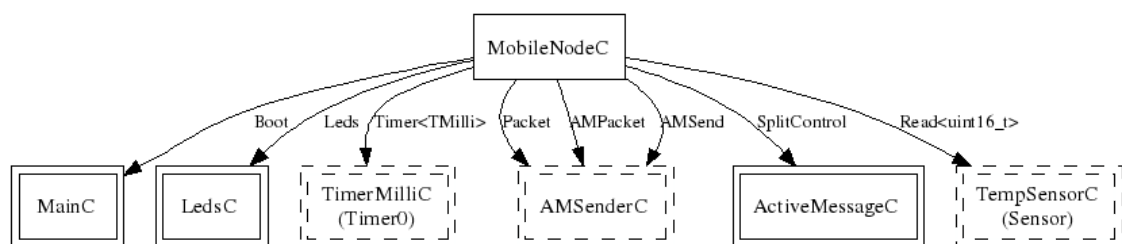


Abbildung 7.1: nesdoc-Darstellung der MobileUnit-Anwendung

Abbildung 7.1 zeigt eine mit nesdoc erzeugte Darstellung, der von der BaseStation-Anwendung genutzten Konfigurationen und Interfaces. Details zu nesdoc finden sich im Anhang A.1. Es werden die Konfigurationen MainC, LedsC, TimerMilliC, AMSenderC, ActiveMessageC und TempSensorC und die Interfaces Boot, Leds, Timer, Packet, AMPacket, AMSend, SplitControl und Read von der Anwendung genutzt. Dabei wird durch das Read-Interface von TempSensorC die Abfrage des Sensirion Temperatursensor implementiert. Auf die Funktionalität der anderen genutzten Interfaces soll hier nicht weiter eingegangen werden, sie ist jedoch größtenteils bereits aus dem Namen ersichtlich bzw. in der TinyOS-Dokumentation beschrieben (vgl. [Levis (2006)]).

¹Datenblatt: http://www.sensirion.com/en/pdf/product_information/Data_Sheet_humidity_sensor_SHT1x_SHT7x_E.pdf

7.1.2 Detector Unit

Die Detector Units werden in einem Gebäude fest installiert. Mit Hilfe dieser Sensorknoten soll zunächst die Temperatur innerhalb des Gebäudes überwacht werden. Dazu wird in regelmäßigen Abständen der Sensorwert ausgelesen und weitergeleitet, so dass dieser über die Basisstation im Leitstand auf einem Gebäudeplan dargestellt werden kann. Die Abfrage des Temperaturwerts erfolgt analog zu den zuvor beschriebenen Mobile Units. Ebenso soll dieser Wert periodisch ermittelt werden.

Es sollen jedoch bei diesem Knotentyp, der permanent im Gebäude installiert ist, zwei unterschiedliche Perioden für die Abfrage des Temperatursensors genutzt werden. Im Normalbetrieb, d. h. wenn kein Brand vorliegt, braucht die Temperatur nicht so häufig ermittelt zu werden als im Brandfall und die Mote kann damit längere Schlafphasen einlegen. In der Mote erfolgt eine Unterscheidung zwischen den beiden Fällen über die gemessene Temperatur. Dieses ist bei den Mobilsenknoten nicht notwendig, da diese am Feuerwehrmann angebracht sind und nur relativ kurz bei einem Einsatz genutzt werden. Daher muss bei diesen Sensortypen immer mit hoher Intensität gemessen werden. Außerdem spielt auch der Energieaspekt hier nicht eine so große Rolle.

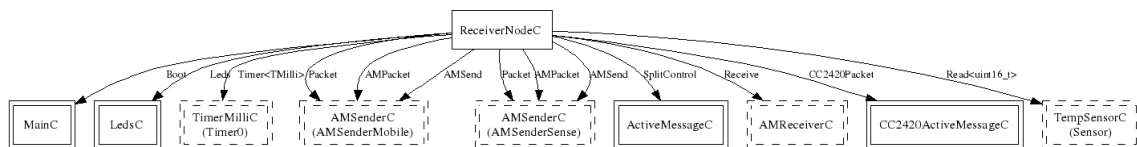


Abbildung 7.2: nesdoc-Darstellung der DetectorUnit-Anwendung

Zusätzlich soll mit Hilfe der Detector Units eine Lokalisation der mobilen Knoten durchgeführt werden. Dazu horchen diese Knoten auf die Beacon-Nachrichten der Mobile Units. Wird eine Nachricht dieses Typs empfangen, kann daraus mit Hilfe des CC2420Packet-Interfaces (siehe Abbildung 7.2) der RSSI-Wert bestimmt werden. Der ermittelte RSSI-Wert wird zusammen mit der ID der Detector Unit und dem Inhalt der Beacon-Nachricht des mobilen Knotens (ID, Sequenznummer und Temperaturwert) in eine neue Nachricht verpackt und gesendet, so dass diese Nachricht über die Base Unit zum Leitstand gelangen kann. Es bleibt also die ursprüngliche Sequenznummer der Mobile Node in dieser Nachricht erhalten. Damit ist es im Leitstand möglich, die berechneten RSSI-Werte von einer Beacon-Nachricht, die von unterschiedlichen Detector Units empfangen wurde, einem Zeitpunkt zuzuordnen und damit, wenn mindestens drei RSSI-Werte vorliegen, eine Lokalisation durchzuführen.

Die Aussendung des eigenen Temperaturwerts der Detector Unit erfolgt unabhängig davon, so dass von dieser Mote zwei unterschiedliche Nachrichtentypen ausgehen, auf die in der Leitstandsanwendung entsprechend reagiert werden kann (näheres in Abschnitt 7.2.3).

7.1.3 Base Unit

Für die Realisierung der Base Unit, die die Schnittstelle vom Sensornetz zum PC bildet, kann auf eine bereits im TinyOS-Paket enthaltene Demo-Anwendung zurückgegriffen werden. Es kann die in Abbildung 7.3 dargestellte Applikation *BaseStation* auf der Mote eingesetzt werden. Dadurch wird bereits die im Entwurf beschriebene Spezifikation implementiert.

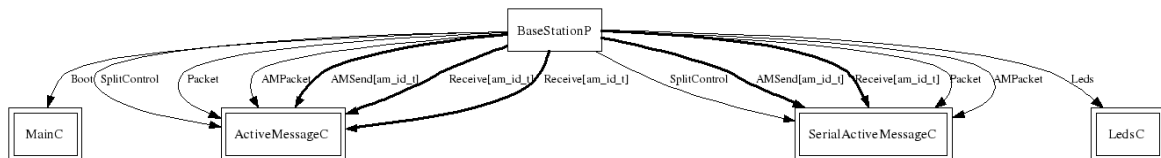


Abbildung 7.3: nesdoc-Darstellung der BaseStation-Anwendung

BaseStation ist eine TinyOS-Hilfsanwendung. Sie arbeitet als Brücke zwischen der seriellen Schnittstelle und dem Funknetz. TinyOS-Nachrichten, die die Mote auf dem seriellen Port erreichen, werden von der Anwendung über die Funkschnittstelle versendet und umgekehrt werden alle Nachrichten, die die Mote per Funk empfängt, an die serielle Schnittstelle weitergeleitet.

Die Mote auf der die BaseStation-Anwendung installiert wird, muss die ID 0 bekommen. Dies ist für eine korrekte Funktion des Routing-Algorithmus des Sensornetzes erforderlich.

Mit der BaseStation-Anwendung wird damit eine direkte Kommunikation zwischen PC und Sensornetz möglich. (Um Nachrichten am PC empfangen bzw. versenden zu können, muss die Mote über den USB-Anschluss mit dem PC verbunden sein.)

7.2 Die Leitstandanwendung

In diesem Abschnitt soll die Entwicklung der Leitstand-Anplikation beschrieben werden. Um das im Systementwurf spezifizierte Verhalten abzubilden, sind besonders in Bezug auf die Positionsberechnung der mobilen Knoten einige Probleme zu lösen. Bevor beschrieben wird, wie die Leitstandanwendung realisiert wurde, soll zunächst auf die Umwandlung des RSSI-Werts in eine Entfernung und auf die mathematischen Details der Positionsbestimmung eingegangen werden.

7.2.1 Umwandlung des RSSI-Wertes

Für die Positionsbestimmung der mobilen Sensorknoten ist es nötig, die Entfernung des Knotens zu anderen festen Sensorknoten zu ermitteln. Dazu soll, wie bereits dargestellt, der RSSI-Wert genutzt werden. Der RSSI-Wert wird als Leistungspegel in dBm angegeben. Aus diesem Wert kann leider keine direkte Umrechnung in einen Entfernungswert erfolgen. Außerdem kann es bei gleichen Distanzen zu unterschiedlichen RSSI-Werten kommen (vgl. Abschnitt 3.3.1.1). Um dennoch einen Entfernungswert angeben zu können, wurden durch Tests Durchschnittswerte für die Zuordnung konkreter RSSI-Wertebereiche gegenüber bestimmten Entfernungen ermittelt.

Das beschriebene Vorgehen bei der Umwandlung des RSSI-Wertes in eine Distanz ist möglich, da es nicht notwendig ist, die exakte Position der mobilen Knoten zu kennen. Für die Darstellung auf dem Gebäudegrundriss reicht es, wenn die ungefähre Position bestimmt werden kann, da auch dies schon einen deutlichen Gewinn für die Übersicht im Leitstand erzielt. Kleine Abweichungen bei der Genauigkeit würden die Übersicht nicht wesentlich verschlechtern.

7.2.2 Positionsberechnung durch Lateration

Um die Position eines mobilen Knotens mit Hilfe des Triangulationsverfahrens (vgl. Abschnitt 3.3.3) berechnen zu können, sollen die ermittelten Distanzwerte der festen Knoten zu dem mobilen Knoten genutzt werden. Es sind mindestens drei feste Knoten und ihr Abstand zum mobilen Knoten nötig, um eine eindeutige Position bestimmen zu können.

Die Positionsbestimmung lässt sich zurückführen auf das mathematische Verfahren der Berechnung des Schnittpunktes zweier Kreise. Die beiden Kreise können sich in keinem, einem oder zwei Punkten schneiden. Die möglichen Schnittpunkte stellen die Position des mobilen Knotens dar. Um bei zwei Schnittpunkten ermitteln zu können, welches der richtige ist, ist ein dritter Referenzpunkt nötig.

Die Schnittpunkte zweier Kreise R_1 und R_2 berechnen sich dabei folgendermaßen:

$$R_1 : (x - x_1)^2 + (y - y_1)^2 = r_1^2 \quad (7.1)$$

$$R_2 : (x - x_2)^2 + (y - y_2)^2 = r_2^2 \quad (7.2)$$

Abbildung 7.4 zeigt eine Darstellung der beiden Kreise, ebenso wie die im weiteren Verlauf gebrauchten Hilfslinien, die für die Berechnung notwendig sind. Zunächst ist der Abstand d zwischen den beiden Kreismittelpunkten M_1 und M_2 zu bestimmen:

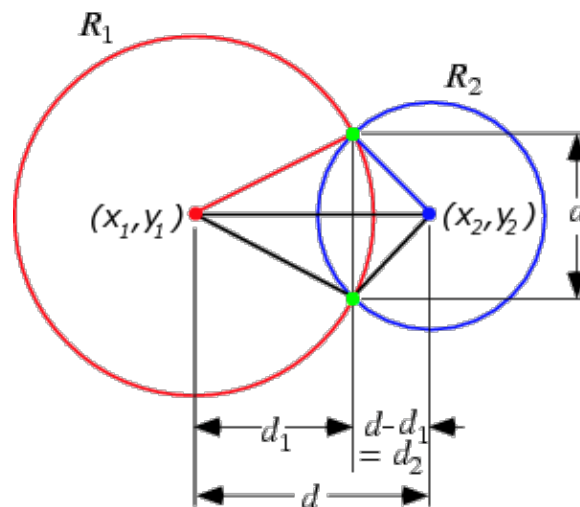


Abbildung 7.4: Kreisschnitt

$$d = \sqrt{|x_1 - x_2|^2 + |y_1 - y_2|^2} \quad (7.3)$$

d kann in d_1 und d_2 aufgeteilt werden, wobei

$$d_1 = \frac{d^2 + r_1^2 - r_2^2}{2d} \quad (7.4)$$

entspricht.

Die Gerade a , die sich zwischen den beiden möglichen Kreisschnittpunkten befindet, steht senkrecht zu d . Dabei schneidet d die Gerade a in der Mitte, somit ergibt sich:

$$\frac{1}{2}a = h \quad (7.5)$$

$$h = \sqrt{r_1^2 - d_1^2} \quad (7.6)$$

Mit Hilfe des Schnittpunktes P_3 von a und d , der im folgenden bestimmt werden soll, lassen sich die beiden gesuchten Kreisschnittpunkte P_4 und P_5 berechnen. Die Koordinaten von $P_3 : (x_3, y_3)$ lauten:

$$x_3 = x_1 + \frac{d_1 * (x_2 - x_1)}{d} \quad (7.7)$$

$$y_3 = y_1 + \frac{d_1 * (y_2 - y_1)}{d} \quad (7.8)$$

Die beiden möglichen Schnittpunkte der Kreise $P_4 : (x_4, y_4)$ und $P_5 : (x_5, y_5)$ sind damit:

$$x_4 = x_3 + \frac{h * (y_2 - y_1)}{d} \quad (7.9)$$

$$y_4 = y_3 - \frac{h * (x_2 - x_1)}{d} \quad (7.10)$$

$$x_5 = x_3 - \frac{h * (y_2 - y_1)}{d} \quad (7.11)$$

$$y_5 = y_3 + \frac{h * (x_2 - x_1)}{d} \quad (7.12)$$

Die Punkte P_4 und P_5 entsprechen den möglichen Positionen des gesuchten mobilen Knotens. Um zu entscheiden welcher der beiden Punkte der richtige ist, ist ein dritter fester Knoten und seine Entfernung zu dem mobilen Knoten nötig. Diese gemessene Distanz kann verglichen werden mit der Distanz zwischen P_4 und dem dritten festen Knoten bzw. der Distanz zwischen P_5 und dem dritten festen Knoten. Sind die Distanzen im Idealfall gleich, ist dies die gesuchte Position des mobilen Knotens. Da es, wie beschrieben zu Abweichungen zwischen der berechneten Distanz aus dem RSSI-Wert und der realen Entfernung kommt, entstehen ebenso Ungenauigkeiten bei der Positionsberechnung.

7.2.3 Sensornetzkommunikation

Durch die auf der Base Unit installierte Anwendung BaseStation werden alle ankommenden Nachrichten des Sensornetzes auf die serielle Schnittstelle übertragen (vgl. Abschnitt 7.1.3). Die Nachrichten sollen in einer Leitstand-Applikation entgegengenommen und ausgewertet werden. Dieser Anwendungsteil ist für die Datenverarbeitung und -haltung innerhalb des Model View Controller-Entwurfsmusters verantwortlich.

Die Java-Anwendung muss dazu eine Instanz der MotIF-Klasse erzeugen und einen MessageListener zum Nachrichtenempfang registrieren. MotIF bietet Methoden für das Empfangen und Senden von Nachrichten über die serielle Schnittstellen und somit kann mit der BaseStation-Anwendung auf der Base Unit kommuniziert werden. MessageListener ist ein Java-Interface. Über die Methode messageReceived() von diesem Interface wird signalisiert, dass eine neue Nachricht eingetroffen ist.

Für jeden Nachrichtentyp der empfangen bzw. versendet werden soll, muss ein Listener in der MotelF-Instanz registriert werden. Damit ist je ein Listener für die beiden Nachrichtentypen der Detector Unit zu registrieren (1. Weitergeleitete Beacon-Nachricht + RSSI-Wert; 2. Temperatur-Nachricht der Detector Unit). Immer dann, wenn die `messageReceived()`-Methode ausgelöst wird, kann je nach Nachrichtentyp reagiert werden und entsprechende Knotenobjekte können angelegt bzw. Membervariablen aktualisiert werden.

7.2.4 Graphical User Interface

Die graphische Darstellung des Gebäudegrundrisses ist der sichtbare Teil der Leitstand-Anwendung. Hier ist eine klare und übersichtliche Darstellung wichtig, um dem Benutzer innerhalb des beschriebenen Szenarios schnell einen Überblick über das Geschehen in dem Gebäude verschaffen zu können. Auch Aspekte der Usability spielen eine große Rolle, würde jedoch den Rahmen dieser Arbeit überschreiten und können daher leider nicht näher untersucht werden. Es sei jedoch auf das Usability-Labor² der HAW Hamburg hingewiesen, an dem umfangreiche Tests diesbezüglich möglich sind.

Zur Darstellung des Gebäudegrundrisses wurde das GEF-Framework (Java Graph Editing Framework) genutzt (vgl. [Tarling und Robbins](#)). Das GEF-Framework ist vor allem durch das ArgoUML³ CASE-Tool bekannt, das GEF z.B. für die Abbildung von UML-Klassendiagrammen einsetzt. GEF stellt ein Java Framework bereit, mit dem grafische Elemente erstellt, editiert und annotiert werden können. Das Graph Editing Framework bietet u. a. einen einfachen und übersichtlichen Aufbau, der leicht erweitert werden kann und ein Model-View-Controller-Design, basieren auf Java-Swing-Bibliotheken, um mit geringem Aufwand eine Integration in bestehende Datenstrukturen zu ermöglichen.

Das GEF-Framework wurde deshalb gewählt, da es ein offenes Framework ist und viele Freiheiten in der Implementierung lässt. Es besteht damit die Möglichkeit, die Anwendung später noch zu erweitern und zu verändern. Obwohl es keine direkte Unterstützung für eine Kartendarstellung besitzt, können die dafür nötigen Objekte erstellt werden.

Mit dem GEF-Framework lässt sich der in Abschnitt 6.5 aufgestellte Entwurf realisieren. Das Framework bietet dabei die Möglichkeit in den Gebäudegrundriss hinein- und hinauszuzoomen und den dargestellten Ausschnitt durch Scroll-Balken zu verschieben. Die mit GEF modellierten Objekte, lassen sich u. a. verschieben, aktualisieren und in der Größe verändern. So sollen beispielsweise die Temperaturwerte der Sensorknoten aktualisiert und die Position der mobilen Sensorknoten auf der Karte verschoben werden.

²<http://users.informatik.haw-hamburg.de/~use-lab/>

³<http://argouml.tigris.org/>

Um einen Grundriss darstellen zu können, wurden Klassen für die Anzeige von Gebäudestrukturen, wie Wände, Türen und Fenster erstellt, so dass damit eine einfache aber für diesen Zweck ausreichende Modellierung eines Gebäudegrundrisses möglich ist.

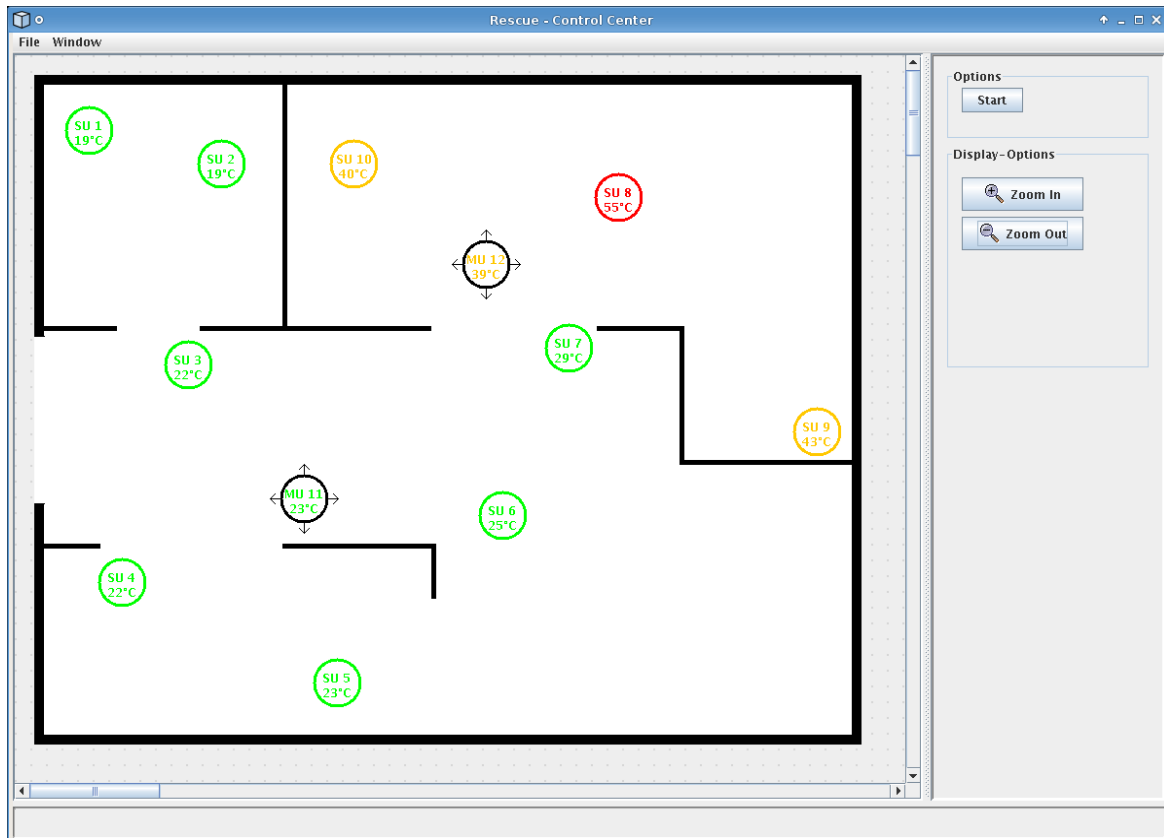


Abbildung 7.5: Leitstand-Anwendung: Darstellung der Detector Units und der Mobile Units auf dem Gebäudegrundriss

Weiterhin sollen die festen, sowie die mobilen Sensorknoten dargestellt werden. Beide werden durch Kreissymbole auf dem Grundriss abgebildet. Um die mobilen von den fest installierten Sensorknoten unterscheiden zu können, sind diese mit Pfeilen in die vier Himmelsrichtungen gekennzeichnet, um anzudeuten, dass sie sich im Raum bewegen können (vgl. Abbildung 7.5). Innerhalb eines Kreises werden jeweils die Knoten-ID und die aktuelle Temperatur des Sensorknotens angezeigt. Die farbliche Gestaltung der Symbole wurde so gewählt, dass diese bereits einen Hinweis auf die aktuelle Gefahrenlage im Bereich des Sensors geben. Ist die Umgebungstemperatur des Sensors kleiner als 30° Celsius, ist dieser in grüner Farbe abgebildet. Steigt die Temperatur über diesen Wert bis hin zu 45° Celsius, dann erfolgt eine Darstellung in orange. Danach wird das Symbol mit roter Farbe dargestellt, um deutlich zu machen, dass hier eine akute Gefahrensituation vorliegt.

Die Temperaturwerte, sowie die Position der mobilen Knoten, müssen fortlaufend aktualisiert

werden. Durch einen Hintergrund-Thread in der Anwendung, werden die Daten der Knotenobjekte (Temperatur, Position) wiederholt abgefragt und aktualisiert. Dieser Ansatz wurde gewählt, um eine Aufteilung zwischen Darstellung im Leitstand und Auswertung der Sensornachrichten und damit eine Trennung von Daten und Darstellung zu erreichen. Durch dieses Vorgehen wird das Model View Controller (MVC) Entwurfsmuster in der Anwendung umgesetzt.

7.3 Ergebnisse

Nachdem zuvor die Realisierung der einzelnen Teile des Sensornetzes und der Leitstandanwendung beschrieben und erklärt wurde, soll im folgenden bewertet werden, inwieweit die Ziele und Anforderungen umgesetzt werden konnten.

Grundsätzlich kann festgestellt werden, dass ein Sensornetz zur Gebäudeüberwachung und zur Positionsbestimmung aufgebaut werden konnte. Es wurden, wie beschrieben, drei Motetypen realisiert. Mobile Sensorknoten (Mobile Units) senden ein Beacon-Signal aus, dass von fest installierten Knoten (Detector Units) empfangen und ausgewertet wird. Durch die Basisstation (Base Unit) wird die Datensinke und Schnittstelle zur Leitstandapplikation implementiert. Mit Hilfe des RSSI-Werts kann eine Distanzbestimmung vorgenommen werden und im Leitstand erfolgt eine Lokalisation. Die mobilen sowie die festen Sensorknoten werden zusätzlich zur Temperaturüberwachung der Umgebung genutzt.

Es wurde eine Leitstandanwendung erstellt, die dazu dient die Sensordaten auf einem Grundriss darzustellen. Ebenso werden die mobilen Sensorknoten an der berechneten Position auf dem Gebäudeplan abgebildet.

Im folgenden soll untersucht werden, ob die in Abschnitt 6.1 aufgestellten Anforderungen erfüllt werden konnten. Einige der dort definierten Anforderungen konnten bereits durch die Entscheidung für TinyOS als Betriebssystem der Sensorknoten erfüllt werden. Die Punkte Verlässlichkeit, Fehlertoleranz und Skalierbarkeit werden bereits durch das Betriebssystem abgedeckt, denn diese Anforderungen sind identisch mit den Zielen aufgrund dessen TinyOS entwickelt wurde. TinyOS ermöglicht somit bereits die Erfüllung dieser Anforderungen.

Das wichtigste Ziel dieser Arbeit ist die Distanzbestimmung und die daraus folgende Positionsberechnung. Aus dem RSSI-Wert kann nicht direkt auf eine Entfernung geschlossen werden. Durch umfangreiche Tests konnten über eine Mittelwertbildung eine Zuordnung eines RSSI-Werts zu einer Distanz erfolgen. Mit diesen so ermittelten Distanzen wurde die Positionsbestimmung durchgeführt.

Mit Hilfe der Positionsdaten erfolgt die Darstellung der mobilen Sensorknoten in der Leitstandanwendung. Um dort die Positionsdaten sinnvoll nutzen zu können, ist in der Leitstandanwendung eine Darstellung erforderlich, die nicht zu große Unterschiede zur Realität besitzt, d. h. die mobilen Sensorknoten werden auf dem Grundriss ungefähr an der Position dargestellt an der sie sich tatsächlich befinden. Durch die stark verkleinerte Abbildung der Gebäudestrukturen in der Applikation und die im Verhältnis dazu relativ große Darstellung der Sensorknoten (um diese gut erkennen und die angezeigten Daten lesen zu können) (vgl. [Abbildung 7.5](#)), hat sich gezeigt, dass eine Abweichungen bei der Lokalisation von bis zu 3 Metern noch hingenommen werden kann.

Mit der in dieser Arbeit durchgeführten Positionsberechnung können Positionsdaten erzielt werden, die maximal 2-3 Meter von der Wirklichkeit abweichen und damit unter dem genannten Schwellenwert bleiben. [Tabelle 7.1](#) zeigt eine Darstellung einiger Messdaten. Es wurde, bei gleich bleibendem Abstand der Sensoren zueinander, wiederholt eine Positionsbestimmung durchgeführt. Während der Messungen wurde die räumliche Lage der Sensorknoten variiert, d. h. sie wurden zwischen den einzelnen Aufnahmen gedreht und gewendet. Die [Tabelle](#) zeigt für drei Sensorknoten die Abweichungen zwischen der realen und der berechneten Distanz zu einem mobilen Sensorknoten. Mit Hilfe der berechneten Entfernung und den Positionsdaten der festen Sensorknoten wurde eine Lokalisation des mobilen Knoten durchgeführt. Die letzte Spalte der [Tabelle](#) zeigt die Abweichung der ermittelten von der realen Position.

Es entstehen, wie der [Tabelle](#) zu entnehmen ist, Differenzen zwischen Messung und tatsächlicher Entfernung, die zunächst relativ groß erscheinen. Für einen Leitstand und die dort handelnden Personen ist es jedoch schon ein deutlicher Gewinn, überhaupt Positionsdaten zu besitzen. So ist es beispielsweise oft schon ausreichend, beurteilen zu können, ob sich eine Person in einem bestimmten Raum aufhält bzw. in einer größeren Halle ist schon viel gewonnen, wenn bekannt ist, dass sich die Person in einem eingegrenzten Bereich von z. B. 4×4 Metern aufhält.

Bei der Durchführung des Messungen wurde deutlich, dass die räumliche Lage der Sensorknoten zueinander einen Einfluss auf den RSSI-Wert hat. Wird der Knoten beispielsweise um 90 Grad gedreht, so hat dies bereits eine Änderung des RSSI-Werts zur Folge. Dieses Verhalten lässt sich damit begründen, dass die in die Plattform integrierte Antenne, nicht in alle Richtungen gleichmäßig abstrahlt. Somit könnten evtl. mit einer externe Antennen weniger starke Schwankungen erreicht und damit genauere Positionsdaten erzielt werden. Auf diese Möglichkeit soll im [Ausblick](#) noch genauer eingegangen werden.

Im Vergleich mit IMPAS, dass, wie beschrieben, an der HAW Hamburg von Sebastian Greger vgl.. [[Gregor \(2006\)](#)] entwickelt wurde, sind einige Unterschiede zu dem in dieser Arbeit entwickelten System aus Motes erkennbar. Gemeinsam haben beide Systeme, das Sie mit

Tabelle 7.1: Messdaten zur Positionsbestimmung (Angaben in cm)

| Messung | Abweichung 1. Knoten | Abweichung 2. Knoten | Abweichung 3. Knoten | Abweichung Position |
|---------|-------------------------|-------------------------|-------------------------|------------------------|
| 1 | 50 | 25 | 175 | 83,03 |
| 2 | 50 | 25 | 175 | 83,03 |
| 3 | 0 | 25 | 150 | 156,11 |
| 4 | 0 | 25 | 125 | 282,84 |
| 5 | 50 | 25 | 175 | 83,03 |
| 6 | 50 | 50 | 75 | 77,02 |
| 7 | 100 | 25 | 25 | 29,85 |
| 8 | 50 | 25 | 25 | 29,85 |
| 9 | 50 | 25 | 25 | 29,85 |
| 10 | 50 | 50 | 150 | 189,31 |
| 11 | 50 | 25 | 125 | 151,04 |
| 12 | 50 | 25 | 125 | 151,04 |
| 13 | 50 | 25 | 150 | 189,31 |
| 14 | 50 | 25 | 25 | 29,85 |
| 15 | 50 | 50 | 150 | 189,31 |
| 16 | 100 | 300 | 125 | 290,93 |
| 17 | 100 | 300 | 75 | 290,93 |
| 18 | 100 | 50 | 150 | 172,47 |
| 19 | 50 | 25 | 175 | 83,03 |
| 20 | 50 | 25 | 175 | 83,03 |

Hilfe von Sensoren an der Decke bzw. mobilen Knoten arbeiten, um damit eine Indoor-Lokalisation durchzuführen. Zunächst setzten beide auf unterschiedlich Verfahren zur Distanzbestimmung. IMAPS nutzt Ultraschall zur Entfernungsbestimmung, laut Gregor sind damit Messungen mit einer Genauigkeit von wenigen Zentimetern möglich. Damit ist IMAPS wesentlich genauer als das hier genutzte Sensornetz. Allerdings ist das Ultraschallverfahren viel anspruchsvoller in Bezug auf die Lage der Sensoren, da im Gegensatz zum Funksignal der Ultraschallimpuls sich nicht in alle Richtungen ausbreitet, sondern Sender und Empfänger sich gegenüberliegen müssen (mit gewissen Toleranzen). Dies ist ein großer Vorteil der Motes, da die Ausrichtung der Sensorknoten keine Rolle spielt. Dieser Unterschied ist besonders im Hinblick auf das gewählt Szenario von Bedeutung, da dort z. B. durch die Bewegung der Rettungskräfte kein reibungsloser Betrieb gewährleistet werden kann.

Nachdem die Ergebnisse dieser Arbeit präsentiert wurden, soll im folgenden Kapitel nach einer kurzen Zusammenfassung, ein Fazit über das Erreichte gezogen werden.

8 Schlussbetrachtungen

In diesem Kapitel soll zunächst das in dieser Arbeit erreichte zusammengefasst und ein Fazit daraus gezogen werden. Daran anschließend wird ein Ausblick gegeben, in dem Vorschläge zur Optimierung gemacht und weitere Anwendungsgebiete betrachtet werden.

8.1 Zusammenfassung

Mit dem in dieser Arbeit realisierten Sensornetz und der Leitstandanwendung, wurden die im Kapitel [Ziele und Vorhaben](#), in den Abschnitten [2.1](#) und [2.2](#), beschriebenen Vorhaben umgesetzt.

Es wurde ein statisches Sensornetz aus fest installierten Sensorknoten in einem Gebäude aufgebaut. Durch das Sensornetz kann ein Gebäude mit Sensoren überwacht werden. Damit ist es zum einen möglich Brände zu erkennen und zum anderen können in einer Leitstandanwendung, die ebenfalls entwickelt wurde, die Sensorwerte an der tatsächlichen Position auf einem Grundriss dargestellt werden. Zur Branderkennung wird exemplarisch ein Temperatursensor eingesetzt.

Darüber hinaus wird die Position von Rettungskräften, die ebenfalls mit einem Sensorknoten ausgestattet sind und sich innerhalb des Sensornetzes bewegen, berechnet. Die Lokalisation kann mit Hilfe der fest installierten Sensorknoten, deren Position bekannt ist, durchgeführt werden. Zur Positionsbestimmung wird auf das Triangulationsverfahren, für das mindestens drei Knoten nötig sind, zurückgegriffen. Die Distanzen zwischen den Knoten werden über die Empfangsstärke von Beacon-Nachrichten bestimmt. Die Position der Rettungskräfte kann im Leitstand auf dem Gebäudegrundriss dargestellt werden. Zusätzlich werden auch die Temperaturwerte der Sensorknoten, und damit die Werte an der Position der Rettungskräfte, an den Leitstand übertragen. Die Personen im Leitstand erhalten dadurch einen detaillierten Überblick über das Geschehen in dem Gebäude und können die aktuelle Lage dort besser beurteilen.

Der erste Teil, der Aufbau eines Sensornetzes, konnte wie vorgenommen umgesetzt werden. Es wurde mit TelosB-Motes ein auf TinyOS basierendes Sensornetz realisiert. Dazu wurden drei unterschiedliche Motetypen definiert, die alle auf derselben Hardware basieren,

jedoch unterschiedliche Aufgaben wahrnehmen und damit auch eine eigene Software besitzen. Die Temperaturwerte der Sensorknoten können zuverlässig ausgelesen und über das Sensornetz an eine PC-Applikation weitergeleitet und dort ausgewertet werden. Es wurde zur Demonstration eine Leitstandanwendung erstellt. In der Anwendung werden auf einem Grundriss die Sensorknoten und die Sensorwerte visuell dargestellt.

Auch der zweite Teil, die Positionsberechnung der mobilen Sensorknoten, konnte realisiert werden. Der RSSI-Wert einer Beacon-Nachricht wird von den fest installierten Sensorknoten bestimmt und somit kann in der Leitstand-Applikation eine Lokalisation durchgeführt werden. Da vom RSSI-Wert nicht direkt auf eine Distanz geschlossen werden kann und die RSSI-Werte bei gleichen Entfernungen, aber unterschiedlichen räumlichen Lagen zweier Sensorknoten, variieren können, treten Fehler bei der Positionsbestimmung auf. Die Abweichung zwischen berechneter und realer Distanz kann, wie Tests gezeigt haben, 2-3 Meter betragen.

Dadurch kommt es ebenfalls zu einem Fehler bei der Darstellung der Position der Rettungskräfte auf dem Gebäudegrundriss. Verglichen mit der Situation ohne ein solches System, ergibt sich dennoch ein großer Gewinn für die Personen im Leitstand, da es beispielsweise oft bereits ausreichend ist zu wissen, ob sich eine Person in einem bestimmten Bereich aufhält. Eine zentimetergenaue Lokalisation ist somit nicht erforderlich und es wird kein Informationsgewinn erzielt.

8.2 Fazit

Rückblickend kann festgestellt werden, dass die gestellten Ziele erreicht wurden. Es konnte ein Sensornetz aufgebaut und zur Gebäudeüberwachung und Distanzbestimmung erfolgreich genutzt werden. Darüber hinaus konnte neben der durchgeführten Lokalisation eine Leitstandanwendung zur Visualisierung der Ergebnisse entwickelt werden.

Obwohl, wie geschildert, die gestellten Ziele umgesetzt werden konnte, muss jedoch angemerkt werden, dass der RSSI-Wert stark variieren kann, d. h. es kann bei gleichen Abständen zu unterschiedlichen RSSI-Werten kommen. Damit ist eine direkte Umrechnung des RSSI-Werts in eine Distanz nur schwer möglich bzw. es muss eine Fehlertoleranz mit eingerechnet werden.

Die Streuung der RSSI-Werte bei gleicher Entfernung entsteht nicht nur, wie im Grundlagenkapitel geschildert, durch Hindernisse wie Mauern, größeren Gegenständen oder Personen, sondern auch die räumliche Lage der Sensorknoten und damit der integrierten Antenne spielt eine große Rolle. So bewirkt z. B. eine Drehung der Mote um 90 Grad bereits eine erhebliche Änderung des RSSI-Werts. Durch eine externe Antenne ließen sich evtl. bessere Werte erzielen (näheres im [Ausblick](#)).

Trotz dieses Nachteils kann der RSSI-Wert zur Distanzmessung herangezogen werden, es muss jedoch eine Abweichung von bis zu 2-3 Meter zwischen berechneter und realer Position in Kauf genommen werden. Für den geplanten Einsatzzweck innerhalb des Rescue-Szenarios ist es jedoch zunächst von Bedeutung, überhaupt eine Darstellung des Geschehens innerhalb eines Gebäudes zu bekommen und es ist in diesem Kontext nicht so sehr von Nachteil, wenn die auf der Karte dargestellte Position um wenige Meter von der realen abweicht.

Der Einsatz des TinyOS-Betriebssystem auf den Sensorknoten hatte viele Vorteile für die Entwicklung und den Aufbau des Sensornetzes. TinyOS bietet für die Moteprogrammierung bzw. Anwendungsentwicklung bereits viele Basisfunktionen, so dass ein einfaches Sensornetzwerk schnell implementiert werden kann. So sorgt TinyOS bereits für das Routing der Nachrichten durch das Netz und der Benutzer kann sich ganz auf die Anwendungsentwicklung konzentrieren. Außerdem stehen einige Hilfsanwendungen zur Verfügung, wie z. B. die auch in dieser Arbeit eingesetzte BaseStation-Applikation, die für den Datenaustausch zwischen serieller Schnittstelle und Funkschnittstelle sorgt.

Es war jedoch sehr aufwändig sich in die Details von TinyOS einzuarbeiten und beispielsweise den RSSI-Wert der TelosB-Mote auslesen zu können. Da TinyOS darauf ausgelegt ist, möglichst viele verschiedene Plattformen mit unterschiedlicher Hardwareausstattung zu unterstützen, ist es dementsprechend schwierig, Informationen zu einem Problem mit einem konkreten Hardwaremodul zu bekommen. Zusammenfassend kann festgestellt werden, dass TinyOS sicherlich viele Vorteile besitzt, jedoch eine Hürde darstellt, die zunächst überwunden werden muss.

8.3 Ausblick

In diesem Abschnitt soll ein Ausblick gegeben werden, in wieweit sich das in dieser Arbeit Realisierte um weitere Komponenten ergänzen oder in anderen Bereichen einsetzen ließe. Außerdem sollen Verbesserungsvorschläge gemacht werden.

Neben den umgesetzten Vorhaben, welche im Kapitel [Ziele und Vorhaben](#) in den Abschnitten [2.1](#) und [2.2](#) beschreiben wurden, enthalten die Abschnitte [2.3](#) und [2.4](#) bereits Ideen, wie sich das bisher Erreichte ergänzen ließe.

Der Ersatz der in [2.3](#) beschriebene Lifeline durch Sensoren, die durch die Feuerwehrleute entlang der zurückgelegten Route ausgelegt werden, wäre prinzipiell möglich. Es könnten zusätzliche Sensoren durch einen geeigneten Mechanismus entlang des Wegs automatisch abgelegt werden und es wäre möglich diese in das bestehende Sensornetz zu integrieren. Realisiert werden könnte diese Funktion beispielsweise dadurch, dass die Feuerwehrleute einem kleinen Behälter auf dem Rücken tragen, aus dem automatisch in regelmäßigen

Abständen Sensorknoten herausfallen. Das zusätzliche Gewicht hierfür wäre heute schon relativ gering und durch die weitere Miniaturisierung der Sensorknoten, die in Zukunft zu erwarten ist, wäre auch der Platzbedarf nicht sehr groß.

Um jedoch mit Hilfe der Sensorknoten eine Route für den Rückweg berechnen zu können, ist die in dieser Arbeit genutzte Methode zur Positionsbestimmung nicht geeignet bzw. mit den genutzten Sensorknoten zu ungenau. Die Abweichung der ermittelten von der realen Position ist zu groß, um damit zuverlässig eine Route zu berechnen und beispielsweise Wände und Durchgänge richtig zu passieren. Zusammenfassend lässt sich feststellen, dass sich eine Lifeline und damit eine Navigation, mit den in dieser Arbeit genutzten Lokalisationsmechanismen, nicht sinnvoll umsetzen lässt, bei genaueren Positionswerten wäre es jedoch möglich dieses Vorhaben umzusetzen.

Ein ganz ähnlicher Ansatz wird im bereits erwähnten LifeNet-Konzept verfolgt, in dem von der HAW Hamburg Mirko Gerling mitwirkt (vgl. [Klann u. a. \(2007\)](#)). Damit könnte bei der Durchführung dieses Vorhabens auf Erfahrungen, die dort gemacht wurden zurückgegriffen werden.

In Abschnitt 2.4 wurde vorgeschlagen, Behälter zu entwickeln mit denen bei Bedarf weitere Sensoren ausgebracht werden könnten. Diese Sensorknoten dürften sich problemlos in das bisherige System integrieren lassen, sobald man die technischen Schwierigkeiten einer geeigneten Konstruktion eines solchen Behälters überwunden hat.

Damit bestände die Möglichkeit, ein Gebäude nachträglich mit einem Sensornetz auszustatten und damit auch Gebäude ohne ein stationäres Sensornetz überwachen zu können. Da jedoch von diesen Sensoren keine Positionen bekannt sind, wäre es ohne fest installierte Sensorknoten mit bekannter Position nicht möglich, eine Positionsbestimmung durchzuführen.

Optimierung der Sensorknoten

Zusätzlich zu den beschriebenen Erweiterungen, wäre es sinnvoll andere Lösungen zur Distanzbestimmung zu untersuchen, um genauere Ergebnisse bei der Lokalisation erzielen zu können. Hier wäre es zum einen möglich andere Verfahren zur Entfernungsmessung zu nutzen, wie z. B. das bereits angesprochene Ultraschallverfahren, das jedoch die beschriebenen Nachteile besitzt. Andererseits wäre es denkbar, durch Motes mit besserer Hardware genauere Ergebnisse zu erreichen.

Eine Möglichkeit, die Abweichung bei der Messung des RSSI-Werts zu reduzieren, wäre eine andere Antenne einzusetzen. Mit einer externen Antenne die gleichmäßigere Abstrahlungswerte besitzt, könnten bessere Positionsdaten erreicht werden. Außerdem könnten die Ausrichtungen der Antennen der einzelnen Sensorknoten zueinander optimiert werden. Wenn

davon ausgegangen wird, dass sich die festen Sensorknoten alle an der Decke eines Gebäudes befinden, ließe sich durch ein geeignetes Gehäuse bewerkstelligen, dass sich alle Antenne dieser Sensorknoten in der gleiche Lage befinden. Die Antennen der mobilen Sensorknoten könnten z. B. alle in eine vertikale Ausrichtung gebracht werden, auch wenn sich durch die Bewegung der Rettungskräfte Schwankungen ergeben würden. Durch diese Verbesserungen könnten die Lokalisationsergebnisse sicherlich optimiert werden.

Neben der Ausstattung der Motes mit anderen Antennen wäre evtl. auch der Einsatz des erst kürzlich vorgestellten Texas Instruments CC2431¹, der seit Herbst 2007 verfügbar ist, handelt es sich um eine *System on Chip* (SoC)-Lösung, die ein IEEE 802.15.4/ ZigBee konformes Funkmodul, basierend auf dem auch in dieser Arbeit genutzten CC2420, mit einem 8051-Mikrocontroller vereint. Interessant ist jedoch vor allem, dass der Chip eine Hardware Location Engine besitzt. Damit kann bereits auf dem Chip eine Positionsbestimmung mit Hilfe des RSSI-Werts vorgenommen werden. Der Chip findet Platz auf einer Flächen von 7×7 mm, so dass damit eine weitere Verkleinerung der Sensorknoten möglich wäre.

Weitere Anwendungsgebiete

Neben dem in dieser Arbeit diskutierten Einsatz des Sensornetzes in einem Rescue-Umfeld, sind weitere Anwendungsgebiete dieses Systems denkbar. Ein Bereich in dem ein Sensornetz und die Positionsbestimmung anwendbar wäre, ist die Gebäudeautomatisierung und -überwachung. So könnte z. B. in sicherheitsrelevanten Bereichen zur Überwachung der sich dort aufhaltenden Personen dieses System eingesetzt werden. Eine solche Nutzung würde jedoch Probleme mit Datenschutzbestimmungen hervorrufen und wäre nur möglich, wenn sichergestellt ist, dass keine gezielte Überwachung einzelner Personen stattfindet. Es soll jedoch nicht unerwähnt bleiben, dass auch diese Verwendung denkbar ist und bedacht werden muss.

Eine weitere Möglichkeit das in dieser Arbeit realisierte System einzusetzen, böte sich in einem Krankenhaus. Ärzte und Schwestern könnten mit Sensorknoten ausgestattet werden, um beispielsweise bei einem Notfall immer diejenigen zu rufen, die die kürzesten Wege haben und damit am schnellsten einem Patienten helfen können.

Es sind sicherlich noch weitere Anwendungsgebiete denkbar, denn eine Positionsbestimmung innerhalb eines Gebäudes bietet viele Vorteile und eröffnet Möglichkeiten diese in verschiedene Systeme und Anwendungen zu integrieren.

¹Datenblatt: <http://focus.ti.com/lit/ds/symlink/cc2431.pdf>

Literaturverzeichnis

- [Beutel u. a. 2007] BEUTEL, J. ; BLUM, P. ; DYER, M. ; MOSER, C. ; YÜCE, M. ; STADELMANN, P.: *BTnode Programming - An Introduction to BTnut Applications*. Revision 1.5. ETH Zürich Computer Engineering and Networks Laboratory (Veranst.), 2007. – URL http://www.btnode.ethz.ch/pub/uploads/Documentation/btnodetutorial_1.5.pdf
- [BTnode 2007] BTNODE: *ETH Zürich: BTnodes - A Distributed Environment for Prototyping Ad Hoc Networks*. 2007. – URL <http://www.btnode.ethz.ch/>. – Zugriffsdatum: 25.08.2007
- [Davids 2006] DAVIDS, A.: *RESCUE: Sensorik zur Gebäudeüberwachung*. Hochschule für Angewandte Wissenschaften Hamburg. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/davids/abstract.pdf>
- [Davids 2007] DAVIDS, A.: *Entwicklung eines Sensornetzes zur Gebäudeüberwachung im Rescue-Umfeld*. Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-proj/davids/report.pdf>
- [Fischer 2005] FISCHER, C.: *Entwicklung von ZigBee-Modulen für spontane Funknetzwerke*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2005. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/fischer.pdf>
- [Gay u. a. 2003a] GAY, D. ; LEVIS, P. ; BEHREN, R. von ; WELSH, M. ; BREWER, E. ; CULLER, D.: The nesC language: A holistic approach to networked embedded systems. In: *PLDI '03: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*. New York, NY, USA : ACM Press, 2003, S. 1–11. – ISBN 1-58113-662-5
- [Gay u. a. 2005] GAY, D. ; LEVIS, P. ; CULLER, D.: Software design patterns for TinyOS. In: *LCTES '05: Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems*. New York, NY, USA : ACM Press, 2005, S. 40–49. – ISBN 1-59593-018-3

- [Gay u. a. 2003b] GAY, D. ; LEVIS, P. ; CULLER, D. ; BREWER, E.: *nesC 1.1 Language Reference Manual*, 2003. – URL <http://nesc.sourceforge.net/papers/nesc-ref.pdf>
- [Gregor 2006] GREGOR, S.: *Entwicklung einer Hardwareplattform für die Ermittlung von Positionsdaten innerhalb von Gebäuden*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gregor.pdf>
- [Hightower und Borriello 2001] HIGHTOWER, J. ; BORRIELLO, G.: Location Systems for Ubiquitous Computing. In: *Computer* Vol. 34 (2001), S. 57–66. – ISSN 0018-9162
- [Hill u. a. 2000] HILL, J. ; SZEWCZYK, R. ; WOO, A. ; HOLLAR, S. ; CULLER, D. ; PISTER, K.: System architecture directions for networked sensors. In: *ASPLOS-IX: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*. New York, NY, USA : ACM Press, 2000, S. 93–104. – ISBN 1-58113-317-0
- [Hinck 2006] HINCK, S.: *RESCUE: Wearable Computer in Disaster-Szenarien*. Hochschule für Angewandte Wissenschaften Hamburg. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/hinck/abstract.pdf>
- [Hinck 2007] HINCK, S.: *Rescue Szenario*. Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-proj/hinck/report.pdf>
- [IEEE 2003a] IEEE: *ANSI/IEEE Std 802.11, 1999 Edition (R2003) - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE Computer Society (Veranst.), 2003. – URL <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>
- [IEEE 2003b] IEEE: *IEEE Standards 802.15.4 - Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Computer Society (Veranst.), 2003. – URL <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>
- [Klann u. a. 2007] KLANN, M. ; RIEDEL, T. ; GELLERSEN, H. ; FISCHER, C. ; OPPENHEIM, M. ; LUKOWICZ, P. ; PIRKL, G. ; KUNZE, K. ; BEUSTER, M. ; BEIGL, M. ; VISSER, O. ; GERLING, M.: LifeNet: an Ad-hoc Sensor Network and Wearable System to Provide Firefighters with Navigation Support, UbiComp2007, 2007. – URL <http://eis.comp.lancs.ac.uk/fileadmin/eis/publication/2007-LifeNet.pdf>
- [Levis 2006] LEVIS, P.: *TinyOS Programming*. Rev. 1.3, 2006. – URL <http://www.tinyos.net/tinyos-2.x/doc/pdf/tinyos-programming.pdf>

- [Levis u. a. 2004] LEVIS, P. ; MADDEN, S. ; GAY, D. ; POLASTRE, J. ; SZEWCZYK, R. ; WOO, A. ; BREWER, E. ; CULLER, D.: The Emergence of Networking Abstractions and Techniques in TinyOS. In: *Proceedings of the First Symposium on Networked Systems Design and Implementation*, USENIX Association, 2004, S. 1–14. – URL <http://www.cs.brown.edu/courses/cs295-1/tinyos-nsdi04.pdf>
- [Mattern und Römer 2003] MATTERN, F. ; RÖMER, K.: Drahtlose Sensornetze. In: *Informatik-Spektrum* Vol. 36 (2003), Nr. 3, S. 191–194. – URL <http://www.inf.ethz.ch/~mattern/papers/sensornetze.pdf>. – ISSN 0170-6012
- [NIST 2006] NIST: *NIST Building and Fire Research Laboratory - intelligent Building Response*. 2006. – URL <http://www.bfrl.nist.gov/ibr/>
- [Piening 2006] PIENING, A.: *RESCUE: Leitstand für Disaster-Szenarien*. Hochschule für Angewandte Wissenschaften Hamburg. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/piening/abstract.pdf>
- [Piening 2007] PIENING, A.: *RESCUE Leitstand für Disaster-Szenarien*. Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-proj/piening/report.pdf>
- [Polastre u. a. 2004a] POLASTRE, J. ; HILL, J. ; CULLER, D.: Versatile low power media access for wireless sensor networks. In: *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA : ACM Press, 2004, S. 95–107. – ISBN 1-58113-879-2
- [Polastre u. a. 2004b] POLASTRE, J. ; SZEWCZYK, R. ; SHARP, C. ; CULLER, D.: *The Mote Revolution: Low Power Wireless Sensor Network Devices*. Proceedings of Hot Chips 16: A Symposium on High Performance Chips. 2004. – URL <http://webs.cs.berkeley.edu/papers/hotchips-2004-motes.ppt>
- [Priyantha u. a. 2000] PRIYANTHA, N. B. ; CHAKRABORTY, A. ; BALAKRISHNAN, H.: The Cricket location-support system. In: *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, 2000, S. 32–43. – URL <http://nms.lcs.mit.edu/papers/cricket.pdf>. – ISBN 1-58113-197-6
- [ScatterWeb 2007] SCATTERWEB: *ScatterWeb*. 2007. – URL <http://scatterweb.mi.fu-berlin.de/>. – Zugriffsdatum: 24.10.2007
- [Steingart u. a. 2007] STEINGART, D. ; WILSON, J. ; REDFERN, A. ; PATTON, C. ; WATTS, W. ; WRIGHT, P.: *Improving Fire Rescue with Robust Information Technology*. 2007. – URL http://www-bsac.eecs.berkeley.edu/scripts/show_pdf_publication.php?pdfID=1102530635

- [Sun 2007] SUN: *Sun SPOT*. 2007. – URL <http://www.sunspotworld.com/>. – Zugriffsdatum: 25.10.2007
- [Tarling und Robbins] TARLING, B. ; ROBBINS, J.: *Graph Editing Framework (GEF)*. – URL <http://gef.tigris.org/>. – Zugriffsdatum: 20.09.2007
- [Warneke u. a. 2001] WARNEKE, B. ; LAST, M. ; LIEBOWITZ, B. ; PISTER, K.: Smart Dust: Communicating with a Cubic-Millimeter Computer. In: *Computer* Vol. 34 (2001), Nr. 1, S. 44–51. – URL http://www-bsac.eecs.berkeley.edu/~mattlast/papers/final_layout.pdf. – ISSN 0018-9162
- [Warneke u. a. 2002] WARNEKE, B. ; SCOTT, M. ; LIEBOWITZ, B. ; ZHOU, L. ; BELLEW, C. ; CHEDIAK, J. ; KAHN, J. ; BOSER, B. ; PISTER, K.: An autonomous 16 mm³ solar-powered node for distributed wireless sensor networks. In: *Sensors, 2002. Proceedings of IEEE* Bd. Vol. 2. Orlando, FL, USA : IEEE, 2002, S. 1510–1515. – URL <http://www-bsac.eecs.berkeley.edu/~warneke/pubs/49-4.pdf>. – ISBN 0-7803-7454-1
- [wearIT@work 2007] WEARIT@WORK: *wearIT@work*. 2007. – URL <http://www.wearitatwork.com/>. – Zugriffsdatum: 21.10.2007
- [Woo und Culler 2001] WOO, A. ; CULLER, D.: A transmission control scheme for media access in sensor networks. In: *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA : ACM Press, 2001, S. 221–235. – ISBN 1-58113-422-3
- [Wright u. a. 2006] WRIGHT, P. ; WILSON, J. ; STEINGART, D. ; REDFERN, A. ; BHARGAVA, V.: *Fire Information and Rescue Equipment (FIRE)*. 2006. – URL <http://fire.me.berkeley.edu/>. – Zugriffsdatum: 21.02.2007

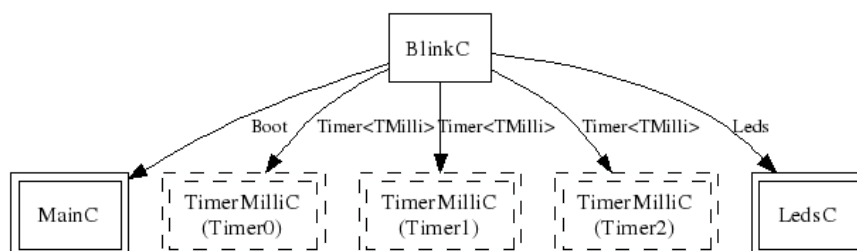
A Anhang

A.1 nesdoc

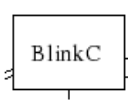
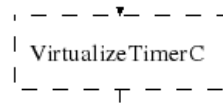

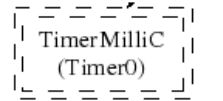
Carefully engineered TinyOS systems often have many layers of configurations, each of which refines the abstraction in simple way, building something robust with very little executable code. Getting to the modules underneath – or just navigating the layers – with a text editor can be laborious. To aid in this process, TinyOS and nesC have a documentation feature called nesdoc, which generates documentation automatically from source code. In addition to comments, nesdoc displays the structure and composition of configurations. [...]

Once you've generated the documentation, go to `tinyos-2.x/doc/nesdoc`. You should see a directory for your platform: open its `index.html`, and you'll see a list of the components and interfaces for which you've generated documentation. For example, if you generated documentation for Blink on the `telosb` platform, you'll see documentation for interfaces such as `Boot`, `Leds`, and `Timer`, as well as some from the underlying hardware implementations, such as `Msp430TimerEvent` and `HpIMsp430GeneralIO`.

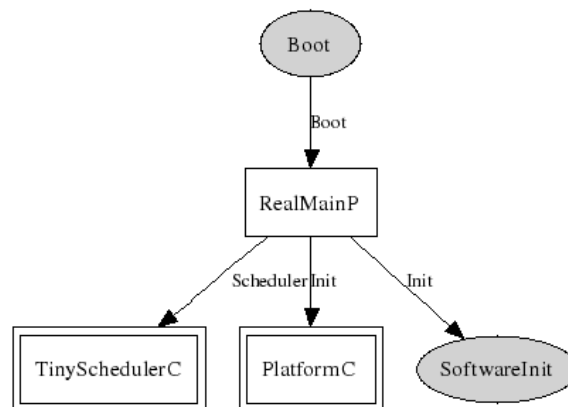
In the navigation panel on the left, components are below interfaces. Click on `BlinkAppC`, and you should a figure like this:



In nesdoc diagrams, a single box is a module and a double box is a configuration. Dashed border lines denote that a component is a generic:

| | Singleton | Generic |
|---------------|---|--|
| Module |  |  |
| Configuration |  |  |

Lines denote wirings, and shaded ovals denote interfaces that a component provides or uses. You can click on the components in the graph to examine their internals. Click on MainC, which shows the wirings for the boot sequence:



Shaded ovals denote wireable interfaces. Because MainC provides the Boot interface and uses the Init (as SoftwareInit) interface, it has two shaded ovals. Note the direction of the arrows: because it uses SoftwareInit, the wire goes out from RealMainP to SoftwareInit, while because it provides Boot, the wire goes from Boot into RealMainP. The details of MainC aren't too important here [..], but looking at the components you can get a sense of what it does: it controls the scheduler, initializes the hardware platform, and initializes software components.

Quelle: <http://www.tinyos.net/tinyos-2.x/doc/html/tutorial/lesson1.html>

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 14. Februar 2008

Ort, Datum

Unterschrift