

# Projektbericht 1 - SoSe 2011

Johannes Meyer

„Navigational Lighting“

Lichtunterstütztes Leitsystem auf Basis von  
selbstverortenden Funknetzen

---

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>3</b>
1.1 Vision . . . . .	3
1.2 Projektziele . . . . .	4
1.3 Fazit . . . . .	4
<b>2 Realisierung</b>	<b>5</b>
2.1 Lokalisierung . . . . .	5
2.2 Kommunikation und Steuerung . . . . .	9
<b>3 Fazit und Ausblick</b>	<b>13</b>
<b>Literatur</b>	<b>15</b>

## 1 Einleitung

Diese Arbeit beinhaltet den Projektbericht für das Masterprojekt 1. Das Thema beschäftigt sich mit der lichtgestützten Navigation von Personen innerhalb von Gebäuden. Der Schwerpunkt liegt hierbei auf dem Beschaffen und Testen der Infrastruktur welche für das Masterprojekt 2 und die Masterarbeit benötigt wird. Dies beinhaltet einerseits die Lokalisierungs-Hard- und -Software, andererseits die verteilten Module zur Ansteuerung der Lichtinstallationen. Die Ausarbeitung lässt sich in folgende Abschnitte aufteilen: Zunächst geben die „Projektziele“ einen Überblick auf das Themengebiet, welches umgesetzt werden sollte. Diese werden anschließend in dem Kapitel „Realisierung“ in den Unterpunkten „Lokalisierung“ und „Kommunikation und Steuerung“ näher betrachtet. Es folgt das Fazit in dem die Ergebnisse mit der Zielsetzung verglichen werden.

### 1.1 Vision

Navigationslösungen für den Outdoorbereich sind in der heutigen Zeit alltäglich geworden. Kartenbasierte Endgeräte nutzen die Positionsdaten des GPS<sup>1</sup> um eine Route zum Ziel zu berechnen. Begibt man sich jedoch in den Indoorbereich, stoßen diese Navigationslösungen an ihre Grenzen: Neue Arten der Lokalisierung und Navigation werden benötigt. Es existieren schon u.a. kartenbasierte Lösungen, welche ein mobiles Endgerät zur Orientierung benötigen (13). Diese nutzen bereits bestehenden Ressourcen (WLAN Hotspots oder ähnliches) um die Position zu bestimmen. Um eine möglichst natürliche und intuitive Wegweisung zu realisieren, bietet es sich an, Personen anhand von Landmarken zu führen, da diese erheblich zur Orientierung beitragen (16). Dies begründet sich in der natürlichen Raumkognition des Menschen, der sich intuitiv anhand real existierender Landmarken orientiert (7) (2). Somit ist eine kartenbasierte Navigation nicht optimal, da sie eine Abstraktion der Realität darstellt und nicht intuitiv genutzt werden kann (12). Nachteilig ist zudem die Tatsache, dass für die Navigation ein mobiles Endgerät nötig ist, welches ständig getragen und abgelesen werden muss. Dies ist jedoch nicht immer möglich, da die Person z. B. andere Gegenstände trägt (17).

Es ist daher von Vorteil, eine landmarkengestützte Navigationslösung zu entwickeln, da Landmarken in die Umgebung integriert sind und das Ablesen von z. B. einem Smartphone überflüssig machen. Künstliche Landmarken können fest installierte Objekte wie beispielsweise elektronische Hinweisschilder, Licht- oder auch Tonsignale sein, welche die Personen individuell zum jeweiligen Ziel führen. In diesem Kontext gibt es Arbeiten, welche öffentliche Displays verwenden, um Personen zu führen (14) (20). Daraus lässt sich die Idee ableiten, eine feste Installation zu verwenden, die einerseits flexibel und einfach zu installieren ist, andererseits kostengünstig und intuitiv verständlich ist (15).

---

<sup>1</sup>Global Positioning System

## 1.2 Projektziele

Ziel für das Projekt soll es sein, eine Infrastruktur zu schaffen, mit deren Hilfe die in AW1 (15) skizzierte Navigationslösung realisiert werden kann. Diese basiert einerseits auf der Lokalisierung der zu führenden Person und der Routenberechnung, andererseits auf der Hardwareinstallation vor Ort, welche den Routenpfad visuell darstellt. Die visuelle Route lässt sich am besten durch farbige Lichtleisten (RGB-LEDs) realisieren, welche in bestimmten Abständen zueinander auf den Fluren eines Stockwerkes angebracht werden. Für die Ansteuerung der Lichtleisten kommen Mikroprozessoren zum Einsatz. Diese erhalten von einer zentralen Steuereinheit über ein selbst organisierendes verteiltes Funknetz die entsprechenden Steuerbefehle, in welcher Farbe die jeweilige LED zu leuchten hat.

## 1.3 Fazit

Dieses Projekt hat somit mehrere parallele Arbeitsbereiche. Eine zuverlässige und relativ präzise Lokalisierungstechnik ist für das gesamte Konzept unabdingbar und erhält demnach höchste Priorität. Im Zuge der Recherche für AW1 hat sich MagicMap als vielversprechende Software dargestellt und soll im Laufe des Projektes ersten Tests unterzogen werden. Ein weiteren wichtigen Punkt stellt die drahtlose Kommunikation zwischen den einzelnen Steuerelementen dar. Diese muss flexibel, störunanfällig und leicht zu installieren sein. Um eine schnelle Entwicklung eines ersten Prototypen zu erzielen, bietet es sich an, auf ein weitverbreitetes Experimentierboard zurückzugreifen. Für dieses existieren diverse Hardware-Erweiterungen, unter anderem auch eine drahtlose Kommunikation untereinander. Die erste Wahl fiel dementsprechend auf die Arduino-Serie mit seinen unterschiedlichen Shields. Im Laufe des Projektes werden mehrere Arduino Duemilanove Boards zusammen mit xBee-Shields auf ihre Verwendbarkeit hin geprüft.

## 2 Realisierung

In diesem Kapitel werden die Technologien näher betrachtet, welche in den Projektzielen(1.2) angesprochen wurden. Hierbei gibt es zwei parallele Schwerpunkte. Einerseits die „Lokalisierung“, andererseits die „Kommunikation und Steuerung“, welche im folgenden Abschnitt separat betrachtet werden. Anschließend werden jeweils die Vorgehensweise, erste Tests, Problemfälle und deren Lösungsversuche während der Umsetzungen des Projektes dargelegt.

### 2.1 Lokalisierung

Für die Lokalisierung einer zu navigierender Person wird der Fokus für die Testinstallation auf das Gebäudeinnere beschränkt. Grund hierfür ist die einfachere Installation in Bezug auf Witterungseinflüsse, Diebstahlschutz sowie bessere Funkabdeckung durch eine große Anzahl von WLAN-Accesspoints. Im Speziellen dient der 11. Stock des TI-Gebäudes als Versuchsort für die Lokalisierung. Dort herrscht eine große Abdeckung durch unterschiedliche WLAN-Accesspoints, sowie einen weitläufigen Flur mit Querverbindungen und angrenzenden Räumen (siehe Abb. 1) Spätere Testrouten können dann z. B. an den Fahrstühlen beginnen und

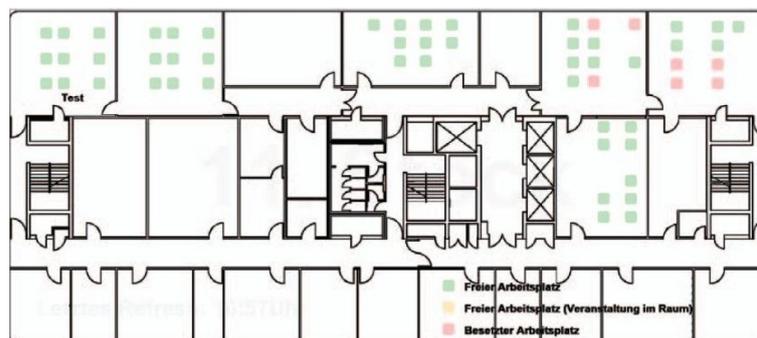


Abbildung 1: Flurplan des 11. Stockwerkes

vier verschiedene Richtungen einschlagen. Für erste Versuche sollte dies zunächst ausreichend sein und im späteren Verlauf könnte der 10. Stock hinzugenommen werden.

### Grundlagen

Es gibt unterschiedliche Herangehensweisen, eine Position zu bestimmen. Hier liegt der Fokus auf der Signalstärkeauswertung von WLAN-Accesspoints (APs). Diese senden im 2,4GHz-Bereich und erreichen eine theoretische Reichweite von ca. 300m. Durch Hindernisse wie z. B.

Wände werden diese Funkwellen jedoch gedämpft. Der Dämpfungswert hängt u.a. vom Material und dessen Dicke ab. Zwei mögliche Lokalisierungstechniken werten die Signalstärke aus. Das sogenannte „Fingerprinting“ basiert auf dem Vergleich mit zuvor gemessenen Signalstärken. Hierbei werden in der Setup-Phase - auf dem gesamten Ortungsbereich verteilt - Probenmessungen durchgeführt. Mit einer Position werden die dazugehörigen Signalstärken zu den unterschiedlichen APs in einer sogenannten Signalstärkenkarte abgespeichert. Je mehr dieser Stützstellen aufgenommen werden, desto fein auflösender ist später die Ortung. In der folgenden Deployment-Phase wird zur Ortung lediglich die aktuelle Signalstärke zu den APs benötigt. Ein Vergleich mit der größten Übereinstimmung aus der Signalstärkenkarte liefert näherungsweise die aktuelle Position. Voraussetzung für einen störungsfreien Betrieb ist ein statisches Umfeld. Dies bedeutet, dass sich weder die Sendeleistung, noch der Standort der APs verändern darf. Ebenso muss der Empfänger aus der Setup- und Deployment-Phase die gleichen Empfangscharakteristika besitzen, damit ein Vergleich möglich wird. In den wenigsten Fällen, ist dies jedoch der Fall und macht somit einen flexiblen und anpassungsfähigen Algorithmus nötig.

Eine andere Ortungsmöglichkeit bietet die „Multilateration“. Hierbei wird anhand der Ausbreitungscharakteristika von Funkwellen auf die Distanz zwischen Sender und Empfänger geschlossen. Sind die Standorte der APs bekannt und eine ausreichende Anzahl von Entfernungen, so reicht im einfachsten Fall eine Trilateration (3 Entfernungen reichen aus um einen Punkt im 3-Dimensionalen Raum zu bestimmen) aus um die Position zu bestimmen. Stehen mehr als drei Distanzen zur Verfügung, können Messfehler ausgeglichen und die Position präziser bestimmt werden. Die Schwierigkeit bei diesem Verfahren liegt in der Annahme, dass sich die Funkwellen gleichmäßig ausbreiten. Dies ist in einem Gebäude jedoch nicht der Fall, da zwischen dem Sender und Empfänger unterschiedliche Materialien (Luft, Wände, Inventar, usw.) mit stark abweichenden Dämpfungswerten liegen. Ähnlich wie beim Fingerprinting-Verfahren ist auch hier ohne entsprechende Algorithmen mit Abweichungen zwischen realer Position und vom System geschätzter Position zu rechnen.

## **MagicMap**

Zur Lokalisierung hat sich während der Recherche im Rahmen des AW1 Vortrages, die Software „MagicMap“ der Humboldt-Universität zu Berlin als vielversprechend dargestellt (15). Eine Integration eines bestehenden Systems kann die Entwicklungszeit hin zu einem ersten Prototypen erheblich beschleunigen. MagicMap (8) (unter der Leitung von Peter Ibach) beschäftigt sich schon länger mit der Indoor-Positionsbestimmung. Dementsprechend viel Erfahrung und Entwicklungszeit ist in das Projekt bereits hinein geflossen. Die verwendete Lokalisierungsmethode bedient sich, wie in Kap. 2.1 beschrieben, der Signalstärkenmessung von Funksignalen. Zusätzlich zur Ortung per WLAN können andere Funkquellen integriert werden (ZigBee, Blue-

tooth, GPS, u.a. - siehe Abb. 2). Es existiert auch eine Schnittstelle um IMU<sup>2</sup>-Daten auszuwerten (9). Das WLAN-Ortungsverfahren basiert auf der Distanzschätzung mittels Signalstärken-

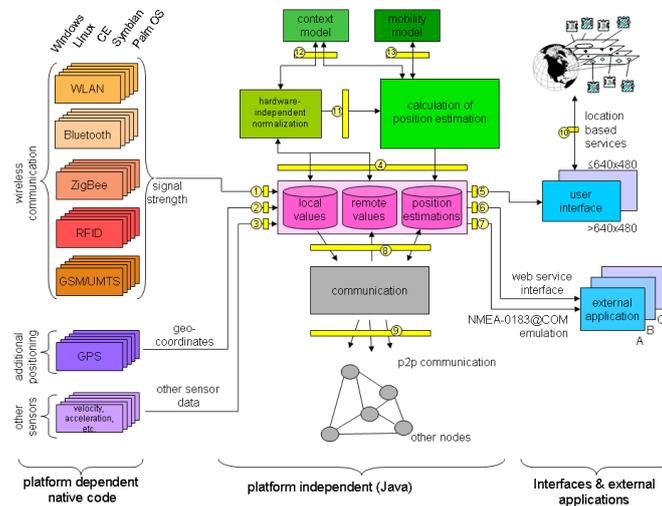


Abbildung 2: MagicMap Schnittstellen

auswertung. Basis ist der MagicMap Server, auf diesem sind die Karten des Gebäudegrundrisses gespeichert. Dort werden ebenfalls die Berechnungen durchgeführt und die Anfragen von den Clients bearbeitet. Ein Client misst die Signalstärken zu den verfügbaren APs und sendet diese Daten an den Server. Die APs sind in dem Kartenmaterial des Servers fest verortet. Anhand der Signalstärken und der daraus berechneten Kantenlängen, wird die Position des Clients bestimmt (vgl. Abb. 3). Es besteht auch die Möglichkeit Stützstellen auf Basis des Fingerprinting-Verfahrens einzumessen. Sofern genug dieser Punkte verortet sind, soll die Positionsbestimmung ebenfalls funktionieren, ohne dass die APs auf der Karte eingetragen sind. In folgendem Kapitel wird das bisherige Vorgehen und dessen Ergebnis vorgestellt.

### Vorgehensweise und Ergebnisse

Zu Beginn des Projektes stand eine mobile Version des MagicMap Clients für Android noch nicht zur Verfügung. Dieser befindet sich aktuell in der Entwicklung und eine erste Betaversion kam erst gegen Ende des Semesters heraus. Um die Lokalisierungsfähigkeit von MagicMap zu testen kam daher der Java-Client zum Einsatz. In Verbindung mit einem Laptop war es so möglich, an verschiedenen Orten Messreihen aufzunehmen und die Position bestimmen zu lassen.

<sup>2</sup>Inertial Measurement Unit



Abbildung 3: MagicMap GUI (10)

In einem Gespräch mit einem Kommilitonen, der die Software bereits genutzt hat, wurde darauf hingewiesen, dass einige Cisco APs im TI Gebäude eine automatische Signalstärkenanpassung besitzen. Dies hätte natürlich erheblichen Einfluss auf die Ortung, da diese auf einer konstanten Signalstärke aufbaut. Aus diesem Grunde wurden vier APs der Marke „WL530gV2“ von ASUS für das Projekt angeschafft, deren Sendeleistung konstant und Installationsort frei wählbar und somit bekannt war. Die anschließenden Versuche wurden in Zusammenarbeit mit Sascha Möhlmann durchgeführt. Dieser benötigt im Rahmen seiner Bachelorarbeit ebenfalls ein Ortungssystem und so bot sich eine Kooperation an. Die ersten Versuche waren jedoch wenig erfolgversprechend, da keinerlei APs von der Software erkannt wurden. Es stellte sich heraus, dass die WLAN-Karte in Kombination mit dem Betriebssystem vom Client als solche nicht identifiziert wurde. Ein Wechsel auf Windows 7 ermöglichte daraufhin erste Messungen. Die vom Server berechnete Position stimmte jedoch auch nach mehreren Kalibrierungen nicht mit der realen Position überein, sondern zeigte nur den Schwerpunkt zwischen den vier fixierten APs. Nach mehrmaliger Mailkorrespondenz stellte sich heraus, dass der aktuell auf der Webseite verfügbare Client gar nicht mit der Serverversion kompatibel war. Bis die neue Clientversion verfügbar war, wurde dieses Teilprojekt zurückgestellt und an der Entwicklung des verteilten Stuenetzes (2.2) gearbeitet. Die anschließend erschienene Version (0.9.9) erwies sich als äußerst ressourcenverbrauchend und fror nach wenigen Sekunden der Benutzung ein. Zudem ließen sich die anderen in Reichweite befindlichen APs (außer den vier für das Projekt fixierten) nicht zuverlässig aus der Berechnung ausschließen. Ein Zusammenhang zwischen dem Ressourcenverbrauch und der Vielzahl von APs lag nahe. Nach erneuter Rücksprache mit dem Entwickler wurde uns eine speziell kompilierte Version zur Verfügung gestellt, welche lediglich die hart encodierte MAC-Adressen der vier APs an den Server weiterleitete. Diese Version lief anschließend stabiler und weitere Messungen konnten durchgeführt werden,

welche nun eine veränderliche Position zeigte, diese wanderte aber beliebig über die Karte und darüber hinaus. Bis zum Ende des Semesters konnte keine zufriedenstellende Lösung in Zusammenarbeit mit den Entwicklern für dieses Problem gefunden werden. Eine persönliche Einweisung direkt bei den Entwicklern in Berlin im kommenden Semester scheint hier erfolgversprechend. Parallel dazu sollte die Möglichkeit für eine Alternative gesucht werden. Eine erste - zu Testzwecken erstellte - Fingerprinting-Anwendung von Sascha Möhlmann lieferte korrekte Ortungsergebnisse und ist als Proof-of-Concept anzusehen. Inwieweit diese Anwendung in größeren Testinstallationen nutzbar ist, und auf sich in ändernden Umgebungen bewährt, gilt es noch zu klären. Der Fokus liegt aber weiterhin auf MagicMap und seinen möglichen Erweiterungen (Plugins), welche eine präzisere Lokalisierung aufgrund von Pfadanalysen und MapMatching bieten soll.

## 2.2 Kommunikation und Steuerung

Wie bereits in 1.1 erwähnt, bietet es sich an, aufgrund der natürlichen Raumkognition des Menschen, für die Wegweisung künstliche Landmarken zu benutzen. Im Speziellen sollen für dieses Projekt farbige LED-Lichtleisten als Orientierungshilfe verbaut werden. Zudem ist es durch farbliche Unterschiede möglich, mehrere Routen parallel darzustellen. Bei überlappenden Routenabschnitten ist eine periodisch abwechselnd angezeigte Farbe möglich. Für die Ansteuerung von RGB-LEDs sind drei Ausgänge pro Lichtleiste am Mikroprozessor nötig. Um ein großes farbiges Spektrum abzudecken, sollten dies PWM-Ausgänge<sup>3</sup> sein.

### Grundlagen

Um die Lichtleisten anzusteuern, müssen die Knoten miteinander kommunizieren. Dieses Problem lässt sich in zwei Teilbereiche aufgliedern:

#### Infrastruktur-Ebene:

Zunächst muss eine physikalische Verbindung zwischen den Knoten sichergestellt werden. Eine Verkabelung ist hierfür zu aufwändig und würde den Installationsaufwand unnötig erschweren. Daher soll die Kommunikation per Funk stattfinden. Im Rahmen der AW1 Recherchen bietet es sich an, die Funkknoten in einem sogenannten Mesh-Netzwerk zu organisieren.

Hierbei ist es nicht nötig, dass der Zielknoten direkt in Funkreichweite des Auftraggebers ist, da das Signal über dazwischenliegende Funkknoten automatisch weitergeleitet wird. Ein weiterer Vorteil liegt in der selbstständigen Organisation und damit Flexibilität in der Installation der Knoten. Für den Embedded-Bereich existieren sogenannte Xbee-Module (6). Diese bieten von sich aus bereits eine eigenständige Organisation in Mesh-Netzwerkstrukturen, welches einen Vorteil hinsichtlich des Entwicklungsaufwandes für einen Prototypen darstellt. Für diese

---

<sup>3</sup>Pulsweitenmodulation

Module gibt es unterschiedliche Protokolle.

Einerseits gibt es den offenen Standard ZigBee (21). Hierbei ist ein sogenannter „Coordinator“ für die Organisation des Netzwerkes zuständig (Abb. 4). Dieser muss ständig verfügbar sein und stellt somit einen „Single Point of Failure“ dar. Weiterhin gibt es „Router“, die für die Weiterleitung innerhalb des Netzwerkes zuständig sind und „End Devices“ welche nur für den Empfang von Daten verwendet werden. Diese können durch ihren geringeren Stromverbrauch batteriebetrieben sein.

Andererseits gibt es das proprietäre Protokoll DigiMesh (4). Hierbei sind alle Knoten gleichwertig. D. h. es gibt nur einen Knotentyp (Abb. 5). Der Vorteil hierbei ist eine einfachere Konfiguration des Netzwerkes und die Möglichkeit, jedes Modul in einen Stromsparmodus zu versetzen (3). Als nachteilig ist die eingeschränkte Plattformverfügbarkeit anzusehen.

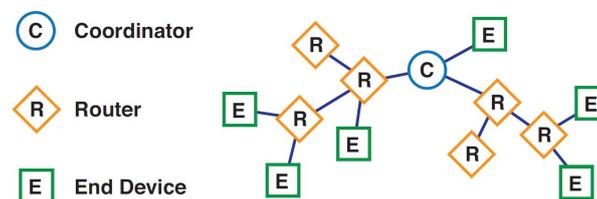


Abbildung 4: ZigBee Knoten

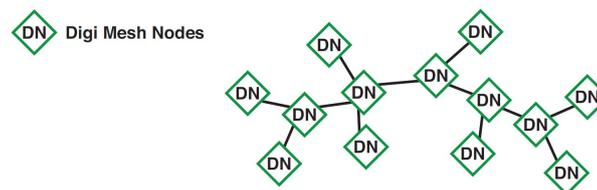


Abbildung 5: DigiMesh Knoten

Dies ist auch der Grund, warum im Verlauf des Projektes das ZigBee Protokoll verwendet wurde. Bekräftigt wurde diese Entscheidung durch Gespräche mit Kommilitonen, welche bereits Erfahrung mit diesem Protokoll gesammelt haben und mir Tipps bei der Entwicklung geben konnten.

#### Logische Ebene:

Um die Routeninformationen auf die Licht-Installation im Gebäude abbilden zu können, müssen die Steuersignale an die jeweilig richtigen Knoten gesendet werden. Durch die Verortung der Knoten im Gebäudemodell und der im selben Modell gebildeten Route kann festgestellt werden, welcher davon auf dieser Route liegt. Mit einer eindeutigen Adressierung können somit die Lichtleisten entsprechend angesteuert werden. (15). Das entsprechende Protokoll

beinhaltet eine eindeutige ID und eine Intensität für jeden Farbwert. Für spätere Weiterentwicklungen ist es angedacht, ein Array für die Farbwerte zu verwenden, um mehrere Routen im blinkenden Wechsel parallel darstellen zu können.

### Arduino und xBee

Für die Ansteuerung der verteilten Steuereinheiten der Lichtleisten kommen Arduino Duemilano Experimentierboards (Abb. 6) zum Einsatz. Diese basieren auf dem Atmel Chip ATmega168 bzw. ATmega328. In der an der HAW verwendeten Variante ist Zweites verbaut. Wie in 2.2 erwähnt, werden drei PWM-Ausgänge pro LED Lichtleiste benötigt.

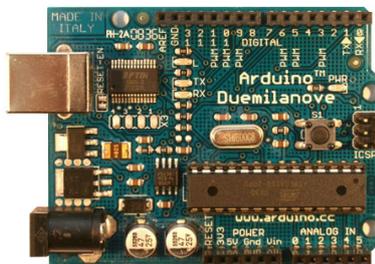


Abbildung 6: Arduino Duemilano (1)



Abbildung 7: xBee Shield mit xBee Modul (11)

Das Arduino Duemilano Board (zukünftig nur noch Arduino genannt) verfügt über 6 PWM-Ausgänge. Ohne Multiplexing können somit zwei Lichtleisten unabhängig voneinander angesteuert werden. Dank des bereits onboard verbauten FTDI-Chips und eines vorinstallierten Bootloaders ist die Programmierung des ATmega per USB schnell und problemlos möglich. Als Programmierumgebung stellt Arduino eine eigene Software zur Verfügung. Die Programmiersprache ist - bis auf kleine Anpassungen - mit C/C++ vergleichbar und wird intern von einem avr-gcc Compiler übersetzt. Sogenannte Shields erweitern die Grundfunktionalitäten und können durch ihre standardisierten Anschlüsse einfach auf den Arduino aufgesteckt werden. Für dieses Projekt wird die Funkkommunikation mittels xBee Modulen realisiert. Hierfür gibt es ein eigenes Shield, welches als Adapterstück zwischen Arduino und xBee dient (Abb. 7). Durch einen zusätzlichen Akkupack kann das komplette Knotenmodul für ersten Installationen und Reichweitentests frei im Raum positioniert werden (Abb. 8).

### Vorgehensweise und Ergebnisse

Für den ersten Prototypen wurden jeweils vier Arduinoboards, Shields mit xBee Modulen und Akkupacks angeschafft und zu Knotenmodulen zusammengebaut. Ein Modul verblieb stets



Abbildung 8: Knotenmodul

als Koordinator bei einem PC und diene diesem als Relais um Steuerbefehle an die übrigen Module zu senden, welche für Testzwecke frei im Raum verteilt werden konnten. Bis auf den Koordinator waren alle Module als „Router“ konfiguriert. Dies hatte zur Folge, dass Sender und Empfänger nicht in direktem Funkkontakt stehen mussten, sondern Nachrichten über dazwischenliegende Module weitergeleitet wurden. Da der spätere Aufbau eine zentrale Steuerung durch einen PC vorsah, wurden Nachrichten über ein Java Programm im PC generiert, über die serielle Schnittstelle an den Koordinator geschickt. Von dort werden die Nachrichten über das Meshnetzwerk drahtlos an den Empfänger gesendet. Die Module wurden zunächst alle im AT-Modus geflasht. Somit konnten die xBee Module direkt über das Java Programm per AT-Befehle<sup>4</sup> konfiguriert werden. Eine typische Nachricht wurde nach folgendem Schema abgeschickt:

- xBee Modul in den Programmiermodus versetzen. (Durch 1,1 Sek. Pause „+++“ 1,1 Sek. Pause)
- Empfänger „R1“ als Ziel speichern (Routendiscovery anstoßen und Zieladresse einspeichern durch „ATDNR1“)
- Auf Bestätigung warten, dass das Ziel bekannt ist, übernommen wurde und der Programmiermodus verlassen wurde.
- Nun können die eigentlichen Daten an das Ziel gesendet werden.

Dieser Ablauf war jedes Mal notwendig, sobald ein neuer Empfänger angesprochen werden sollte. Pro Nachricht dauerte das Senden einer Nachricht somit im Idealfall ca. 5 Sekunden. Für eine große Anzahl an unterschiedlichen Empfängern hätte dies zu erheblichen Verzögerungen geführt. Als weiterer Nachteil hat sich die Unzuverlässigkeit der Programmierung der Module erwiesen. In diesem Fall reagierten die Module nicht auf AT-Befehle und lange Timeouts

---

<sup>4</sup>Steuerbefehle für Modems

waren die Folge. Erst nach einem Reset ließ sich eine neue Zieladresse eingeben. Aufgrund dieses Verhaltens wurde nach einer alternativen Steuerung gesucht. Dies war möglich indem man die Module im sogenannten API-Modus flasht. Hierbei wurden anstatt AT-Befehlen, Pakete gesendet. Die Kontrollstruktur besitzt einen Header mit der Zieladresse, die Länge, die zu sendenden Daten und eine Checksumme (Abb. 9). Weiterhin erlaubte dieser Modus eine bessere Kontrolle, da der Empfang bestätigt und ein entsprechendes Paket zurückgegeben wurde. Das Zusammenstellen des Paketes mit der Checksummenberechnung, sowie der Auswertung

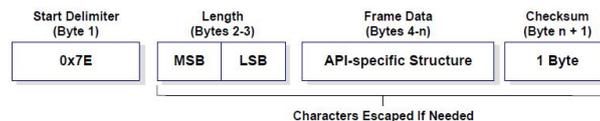


Abbildung 9: API Frame (5)

der Statuspakete war dementsprechend aufwändiger. Es existierte jedoch ein entsprechendes OpenSource-Projekt „xbee-api“ (18), welches speziell für diese xBee Module geschrieben wurde. Nach einiger Einarbeitung in den Code und der Dokumentation war es möglich aus der Entwicklungsumgebung Eclipse ein Paket zu generieren, es an das xBee Modul zu senden und dessen Antwort korrekt auszuwerten. Damit die Empfangsknoten das empfangene Paket korrekt auswerten und entsprechende Antworten senden, ist ein entsprechendes Programm zu schreiben, da die Pakete vom xBee Modul lediglich als String über die serielle Schnittstelle eintreffen. Auch hierfür fand sich ein Projekt (vom selben Autor), welches entsprechend der Dokumentation des Funkmodulherstellers die Frames auswertete. Bei dem „xbee-Arduino“ (19) Projekt handelte es sich um eine Bibliothek, welche die Standardfunktionalität wie z. B. eine Empfangsqueue oder Checksummenberechnung enthielt. Basierend auf dem resultierenden xBee-Objekt konnten Funktionen aufgerufen werden und die höher stehende Funktionalität entwickelt werden. Somit war es möglich vom PC einen Farbwert an einen Empfänger zu senden und dieses Paket dort entsprechend auswerten zu lassen. Die automatische Routerkennung und Weiterleitung der Pakete in Form eines Mesh-Netzwerkes funktionierte auf Anhieb. Das Verhalten bei z. B. Nichterreichbarkeit eines Knotens wurde dem Entwickler durch die Bibliothek freigestellt. In diesem Falle wurde ein Fehler ausgegeben, da sich eine spätere abgebildete Route auf die Lichtinstallation nur durch alle Knoten abbilden lässt.

### 3 Fazit und Ausblick

Abschließend werden in diesem Kapitel die Ergebnisse des Projektes 1 dargelegt und ein Ausblick für Projekt 2 vorgestellt. Die in AW1 gesetzten Ziele für das Projekt, wurden teilweise erreicht. Das selbst organisierende Meshnetzwerk, die Ansteuerung der RGB-LEDs und die Kontrolle über ein PC-Programm konnten erfolgreich realisiert werden. Einerseits wurde viel

---

Entwicklungszeit durch die fertig implementierte Meshnetzwerk-Organisation gespart, auf der anderen Seite kam es durch die zuerst gewählte Kommunikationsvariante mittels AT-Befehlen zu Rückschlägen und machte eine Neuorientierung nötig. Eine zufriedenstellende Lokalisierung konnte bis zum Ende des Semesters jedoch nicht erreicht werden. Bei der Risikoanalyse in AW1 wurde die Möglichkeit einer nicht funktionsfähigen Lokalisierungssoftware nicht bedacht, sondern ausgehend auf den Aussagen und technischen Details der Webseite vertrauend die Zeitplanung erstellt. Für die Einarbeitung in die Software waren wenige Tage bis Wochen vorgesehen und basierend darauf, die höheren Anwendungsentwicklungen. Diese konnten wegen der fehlenden Ortungsbasis noch nicht angegangen werden. Aufgrund dieser Erfahrungen, verschiebt sich der Fokus nun für das Projekt und die kommenden Semester in Richtung Lokalisierung und dessen Verbesserungsmöglichkeiten. Eine eigenständige Entwicklung eines Meshnetzwerkes ist wegen der bisher gemachten guten Erfahrungen mit dem bestehenden System nicht notwendig und wird zur Realisierung verwendet.

## Literatur

- [1] ARDUINO: *Arduino Duemilanove*. Webseite. – URL <http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>. – Letzter Aufruf am 27. August 2011
- [2] BEEHAREE, Ashweeni K. ; STEED, Anthony: A natural wayfinding exploiting photos in pedestrian navigation systems. In: *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. New York, NY, USA : ACM, 2006 (MobileHCI '06), S. 81–88. – URL <http://doi.acm.org/10.1145/1152215.1152233>. – ISBN 1-59593-390-5
- [3] DIGI INTERNATIONAL INC.: *Wireless Mesh Networking ZigBee vs. DigiMesh*. Webseite. 2008. – URL [http://www.digi.com/pdf/wp\\_zigbeevsdigimesh.pdf](http://www.digi.com/pdf/wp_zigbeevsdigimesh.pdf). – Letzter Aufruf am 27. Februar 2011
- [4] DIGI INTERNATIONAL INC.: *The DigiMesh Networking Protocol*. Webseite. 2010. – URL <http://www.digi.com/technology/digimesh/>. – Letzter Aufruf am 27. Februar 2011
- [5] DIGI INTERNATIONAL INC.: *Product Manual v1.x.4x - ZigBee Protocol*. Webseite. 2010. – URL [http://www.google.de/url?sa=t&source=web&cd=1&sqi=2&ved=0CBwQFjAA&url=http%3A%2F%2Fftp1.digi.com%2Fsupport%2Fdocumentation%2F90000866\\_D.pdf&rct=j&q=90000866\\_D&ei=5mNaToPbIIPFswa4m82FCw&usg=AFQjCNF62-S1Ydr54zUvMtqD\\_01PEnz\\_HA&cad=rja](http://www.google.de/url?sa=t&source=web&cd=1&sqi=2&ved=0CBwQFjAA&url=http%3A%2F%2Fftp1.digi.com%2Fsupport%2Fdocumentation%2F90000866_D.pdf&rct=j&q=90000866_D&ei=5mNaToPbIIPFswa4m82FCw&usg=AFQjCNF62-S1Ydr54zUvMtqD_01PEnz_HA&cad=rja). – Letzter Aufruf am 28. August 2011
- [6] DIGI INTERNATIONAL INC.: *XBee® & XBee-PRO® 802.15.4 OEM RF Modules*. Webseite. 2010. – URL <http://www.digi.com/products/wireless/point-multipoint/xbee-series1-module.jsp#overview>. – Letzter Aufruf am 27. Februar 2011
- [7] HILE, Harlan ; VEDANTHAM, Ramakrishna ; CUELLAR, Gregory ; LIU, Alan ; GELFAND, Natasha ; GRZESZCZUK, Radek ; BORRIELLO, Gaetano: Landmark-based pedestrian navigation from collections of geotagged photos. In: *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*. New York, NY, USA : ACM, 2008 (MUM '08), S. 145–152. – URL <http://doi.acm.org/10.1145/1543137.1543167>. – ISBN 978-1-60558-192-7
- [8] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap*. Webseite. – URL <http://www.magicmap.de>. – Letzter Aufruf am 27. Februar 2011
- [9] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap Architektur*. Webseite. – URL <http://wiki.informatik.hu-berlin.de/nomads/index.php/Bild>:

- [MagicMapArchitektur\\_Schnittstellen.png](#). – Letzter Aufruf am 19. August 2011
- [10] HUMBOLDT-UNIVERSITÄT ZU BERLIN: *MagicMap GUI*. Webseite. – URL [http://www2.informatik.hu-berlin.de/rok/MagicMap/\\_images/screenshots/client\\_20-12-2004\\_3.png](http://www2.informatik.hu-berlin.de/rok/MagicMap/_images/screenshots/client_20-12-2004_3.png). – Letzter Aufruf am 25. August 2011
- [11] JACOBS, Robert: *xBee Shield*. Webseite. – URL <http://robertjacobs.eu/wp-content/uploads/2011/07/arduino-xbee-shield-c-xbee.jpg>. – Letzter Aufruf am 27. August 2011
- [12] JENSEN, B. ; KRUSE, R. ; WENDHOLT, B.: Application of indoor navigation technologies under practical conditions. In: *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*, 2009, S. 267 –273
- [13] KALLIOLA, Kimmo: *Bringing Navigation Indoors*. September 2008. – URL [http://www.nokia.com/NOKIA\\_COM\\_1/Press/Press\\_Events/The\\_Way\\_We\\_Live\\_Next\\_2008/presentations/TWVLN08\\_Kimmo\\_Kalliola.pdf](http://www.nokia.com/NOKIA_COM_1/Press/Press_Events/The_Way_We_Live_Next_2008/presentations/TWVLN08_Kimmo_Kalliola.pdf). – Letzter Aufruf am 27. Februar 2011
- [14] KOGAN, Borys: *Indoor Navigationssystem mit dynamischer Beschilderung - Entwicklung und Simulation in einer virtuellen 3D-Umgebung*, Hochschule für Angewandte Wissenschaften (HAW) Hamburg, Diplomarbeit, 2009. – URL [http://opus.haw-hamburg.de/volltexte/2009/755/pdf/masterarbeit\\_kogan.pdf](http://opus.haw-hamburg.de/volltexte/2009/755/pdf/masterarbeit_kogan.pdf). – Letzter Aufruf am 27. Februar 2011
- [15] MEYER, Johannes: *Ausarbeitung Anwendungen 1 - Navigational Lighting*. Webseite. 2011. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-aw1/meyer/bericht.pdf>. – Letzter Aufruf am 22. August 2011
- [16] MILLONIG, Alexandra ; SCHECHTNER, Katja: Developing Landmark-Based Pedestrian-Navigation Systems. In: *Intelligent Transportation Systems, IEEE Transactions on* 8 (2007), Nr. 1, S. 43 –49. – ISSN 1524-9050
- [17] NAPITUPULU, Jan: *Multimediale Fluggastführung*. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/napitupulu/report.pdf>. – Letzter Aufruf am 27. Februar 2011
- [18] RAPP, Andrew: *xbee API - A Java API for Digi XBee/XBee-Pro OEM RF Modules*. Webseite. – URL <http://code.google.com/p/xbee-api/>. – Letzter Aufruf am 28. August 2011

- 
- [19] RAPP, Andrew: *xbee Arduino - Arduino library for communicating with XBees in API mode*. Webseite. – URL <http://code.google.com/p/xbee-arduino/>. – Letzter Aufruf am 28. August 2011
- [20] RUPPEL, Peter ; GSCHWANDTNER, Florian ; SCHINDHELM, Corina K. ; LINNHOFF-POPIEN, Claudia: Indoor Navigation on Distributed Stationary Display Systems. In: *Computer Software and Applications Conference, Annual International 1* (2009), S. 37–44. – ISSN 0730-3157
- [21] ZIGBEE ALLIANCE: *ZigBee Specifications*. Webseite. 2010. – URL <http://www.zigbee.org/Specifications.aspx>. – Letzter Aufruf am 27. Februar 2011