



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Anwendungen I

Arno Davids

RESCUE: Sensorik zur Gebäudeüberwachung

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1. Einleitung / Motivation	3
1.1. Szenario: „Gebäudebrand“	3
1.2. Problemstellung	3
2. Anforderungen	4
2.1. Relevante Daten für die Rettungskräfte	4
2.2. Klassifikation der Sensoren	5
2.3. Sensorknoten	5
2.4. Sensornetz	5
3. Entwurf eines Sensornetzes	6
3.1. MAC-Protokoll	6
3.1.1. S-MAC (Sensor MAC)	6
3.1.2. T-MAC (Timeout MAC)	7
3.1.3. Berkeley-MAC / B-MAC	8
3.1.4. IEEE 802.15.4 und ZigBee	9
3.2. Berkeley Motes	9
3.3. TinyOS	10
4. Kommunikation zum Leitstand	11
5. Meine Ziele für das nächste Semester	11
Abbildungsverzeichnis	13
Literaturverzeichnis	14

1. Einleitung / Motivation

1.1. Szenario: „Gebäudebrand“

In diesem Projekt soll Anhand des Szenarios „Gebäudebrand“ untersucht werden, in wie weit sich die gefährliche und schwierige Arbeit der Rettungskräfte bei einem Gebäudebrand optimieren lässt, um die Risiken zu minimieren und die Arbeit der Helfer zu erleichtern und zu verbessern. Dabei wird bei dem von uns gewählten Szenario davon ausgegangen, dass es sich um ein Desaster beispielsweise in einer großen chemischen Produktionsanlage handelt, bei dem besondere Gefahren auftreten. Auch ist die dort im Allgemeinen vorhandene Werksfeuerwehr ein bedeutender Partner, für den sich den Einsatz, der in diesem Projekt vorgestellten Technik, am meisten lohnen würde.

Das Projekt ist in drei Bereiche unterteilt:

- Sensorik zur Gebäudeüberwachung (Arno Davids)
- Leitstand für Desaster Szenarien (Andreas Piening)
- Wearable Computer in Desaster Szenarien (Steffen Hinck)

Der von mir bearbeitete Teil befasst sich folglich mit der Sensorik, die nötig ist, um alle für die Rettungskräfte wichtigen Informationen zu erfassen und an den Leitstand weiterzureichen. Dies soll mit Hilfe eines drahtlosen Sensornetzwerks (Wireless Sensor Network, WSN) erreicht werden, in dem die benötigten Daten durch Sensoren ermittelt und über ein Sensornetzwerk an die Leitstellen weitergereicht werden.

In den weiteren nicht mehr von mir bearbeiteten Teilen erfolgt dann die Verarbeitung und Darstellung der Daten im Leitstand und am Ende die Weiterleitung und Darstellung ausgewählter Informationen an die einzelnen Rettungskräfte (siehe Arbeiten von Andreas Piening und Steffen Hinck)

1.2. Problemstellung

In der Brandbekämpfung stellen Gebäudebrände für die Rettungskräfte, insbesondere für Feuerwehrleute, eine besondere Gefahrensituation dar, da hier besonders viele Einschränkungen und Unsicherheiten auftreten. So müssen sie auf engem Raum operieren, kennen meist die Gebäudepläne nicht und es ist oft nicht genau bekannt an welchen Stellen es in welchem Umfang brennt. Ebenso sind für die Brandbekämpfer im Allgemeinen keine weiteren Informationen über das Gebäude, wie z.B. die Temperatur in den einzelnen Räumen, verfügbar. Dabei können zusätzliche Daten überaus wichtig sein. So ist es bei Bränden in industriellen Produktionsanlagen, beispielsweise in einer chemischen Fabrik, von besonderer Bedeutung, die genaue chemische Zusammensetzung eines Produktionsstoffes 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 zur Verfügung zu stellen, um die Brandbekämpfung zu verbessern und die Risiken für die Rettungskräfte zu minimieren. Die Datenerfassung kann durch die Brandbekämpfer vor Ort geschehen, beispielsweise indem Personen den Umfang des Brandes augenscheinlich erfassen. Viel besser und ungefährlicher für die Retter ist es jedoch die Daten durch Sensoren zu erfassen, die die gewünschten Informationen sammeln und diese an eine zentrale Leitstelle in sicherer

Entfernung weiterleiten. Dort können sie dann ausgewertet werden und es kann entsprechend reagiert werden.

2. Anforderungen

Um die beschriebene Datenerfassung mit Hilfe von Sensoren durchführen zu können, ist es zunächst wichtig sich darüber Gedanken zu machen, welche Informationen von den Rettungskräften überhaupt benötigt werden und durch welche Arten von Sensoren diese erfasst werden können. Daraus sollen dann die Anforderungen an die einzelnen Sensorknoten, sowie an das Design und den Aufbau des zur Datenerfassung nötigen Sensornetzes abgeleitet werden.

2.1. Relevante Daten für die Rettungskräfte

Die für die Rettungskräfte nützlichen Daten lassen sich in zwei Klassen einteilen. Es existieren sowohl statische als auch dynamische Informationen zu einem Ereignis, beispielsweise einem Gebäudebrand. Als statisch werden Daten bezeichnet, die bereits vor einem Zwischenfall erhoben werden und sich auch nicht ändern. Dynamische Daten werden zum Zeitpunkt des Unglücks erfasst und liefern Informationen über aktuelle Vorgänge im Gebäude. Sie werden über verschiedenen Arten von Sensoren erfasst.

Relevante statische Informationen sind u.a. [Jones-05]:

- Gebäudepläne und -daten (Position der Raumtüren, Fenster, Brandschutztüren, Fahrstühle, Treppen; Anzahl und Ort der Eingänge, Sprinkleranlagen)
- Pläne der Gebäudeinfrastruktur wie z.B. von Klimaanlage, Stromversorgung, Wasseranschlüssen
- Positionen der Sensoren
- Positionen von Hydranten
- Besondere Gefahrenstellen, z.B. Lager explosiver Stoffe
- Lagerort der Gebäudeschlüssel, Kontaktdaten des Gebäudemanagers/Verantwortlichen

Die Bedeutsamen dynamische Informationen umfassen [Jones-05]:

- Status der Brandmelder (Wo brennt es?; In welchem Umfang brennt es?; Wie lange brennt es schon?)
- Temperaturentwicklung des Feuers
- Status der Sprinkleranlagen
- Zusammensetzung der Umgebungsluft (Treten Gase aus?; Wie viel Sauerstoff ist noch in einem brennenden Raum?)
- Anzahl und Ort der Retter im Gebäude
- Anzahl und Ort der sonstigen Personen im Gebäude
- Fahrstuhlpositionen
- Status der Fluchtwege (Sind Fluchtwege blockiert?)
- Gebäudestatus (Ist Klimaanlage, Strom ausgeschaltet?)
- Aktuelle Menge und Einsatzort gefährlicher Produktionsstoffe
- Status der Sensoren (Sind Sensoren ausgefallen?; Erkennung von fehlerhaften Daten)
- Wetterdaten (Wichtig z.B. bei Austritt von gesundheitsgefährdenden Gasen in die Umgebung)

2.2. Klassifikation der Sensoren

Die für die Datenerfassung notwendigen Sensoren lassen sich grob in zwei Klassen unterteilen. So wird ein Grossteil der oben genannten statischen und dynamischen Informationen durch fest im Gebäude installierte Sensoren erfasst. Diese sind bereits vor dem Unglücksfall installiert und werden teilweise auch im „Normalbetrieb“ für Steuerungs- und Überwachungsaufgaben genutzt, beispielsweise zur Kontrolle der Klimaanlage. Des Weiteren muss es bei einem Brand für die Rettungskräfte möglich sein, zusätzliche Sensoren in Betrieb zu nehmen, um zusätzliche Daten zu erfassen oder Werte engmaschiger zu beobachten.

In den beiden Klassen gibt es jeweils zwei Arten von Sensoren. Dabei handelt es sich zum einem um Sensoren die sich fest an einem Ort befinden und auch niemals bewegt werden. Andererseits gibt es Sensoren die mobil sein müssen, beispielsweise Sensoren die sich am Anzug des Feuerwehrmannes befinden (z.B. zur Ortsbestimmung oder Temperaturüberwachung).

2.3. Sensorknoten

Obwohl eine Vielzahl unterschiedlicher Sensoren zum Einsatz kommen soll, kann man einige grundsätzliche Anforderungen feststellen, die alle Sensorknoten so gut wie möglich umsetzen sollen. Dabei ist zu berücksichtigen, dass sich die genannten Eigenschaften durchaus widersprechen können und dürfen, da es auf eine optimale Kombination der Eigenschaften in Abhängigkeit von der späteren Aufgabe und Art der eingesetzten Sensoren ankommt. Es lassen sich folgende Anforderungen für die einzelnen Sensorknoten definieren:

- Möglichst klein
- Robust und widerstandsfähig (bspw. gegen hohe Temperaturen, aber auch gegen Feuchtigkeit durch Löschwasser)
- Zuverlässig
- Lange Lebensdauer
- Geringer Energieverbrauch
- Kostengünstig
- Kurzfristig hohes Verkehrsaufkommen möglich
- Funkbasiert

Die vorletzte Anforderung ist dahingehen zu verstehen, dass es beim Eintreten eines Unglücksfalls nötig werden kann, dass einzelne Sensoren in schnelleren Intervallen Daten erfassen, als sie dies im „Normalmodus“ machen würden, um genauere Beobachtungen durchführen zu können.

Die Sensorknoten sollten, wie in der letzten Anforderung definiert, funkbasiert sein, da es beim Einsatz von mobilen Knoten unerlässlich ist, dass diese frei bewegt werden können. Aber auch bei fest angebrachten Sensoren ist es sinnvoll diese mit einer Funkschnittstelle auszurüsten, um den hohen Aufwand für die Verkabelung zu reduzieren.

2.4. Sensornetz

Die verschiedenen Sensoren, die für die Erfassung der genannten Daten nötig sind, müssen in einem gemeinsamen Sensornetz miteinander kommunizieren. Auch an den Aufbau und das Design des Netzes gibt es einige grundsätzliche Anforderungen:

- Dynamisches Ad-hoc Netzwerk
- Große Anzahl Knoten möglich

- Tolerant gegenüber Knotenausfall (z.B. durch Feuer)
- Stationäre wie auch mobile Knoten vorhanden
- Knoten haben nur begrenzte Rechenleistung
- Knoten haben nur begrenzte Energiereserven

3. Entwurf eines Sensornetzes

Das gemeinsame drahtlose Sensornetz ist ein Schlüsselbereich und essentiell für die Weiterleitung und spätere Verarbeitung der Sensordaten. Ebenso ist das Senden und Lauschen auf dem Funkkanal für die einzelnen Sensoren eines der Bereiche, der am meisten Energie verbraucht. Somit ist die Auswahl eines geeigneten MAC (Medium Access Control)-Layer-Protokolls entscheidend für die Umsetzung der gestellten Anforderungen, insbesondere im Hinblick auf eine möglichst lange Lebensdauer der Knoten. Aber auch die anderen Anforderungen haben Einfluss auf die Auswahl des Protokolls.

3.1. MAC-Protokoll

Die traditionelle MAC Protokolle sind optimiert auf einen hohen Durchsatz, eine möglichst geringe Latenz und sollen einen fairen Mediumzugriff für alle Teilnehmer ermöglichen. Allerdings sind sie häufig in Bezug auf den Energieverbrauch ineffizient, wie beispielsweise im WirelessLAN-Protokoll (802.11). Dort wird das CSMA/CA Verfahren genutzt und somit kann es ständig zu dem Problem des idle listening kommen, wodurch die begrenzten Energievorräte der Sensorknoten unnötig belastet werden. Idle listening meint den Zustand, wenn ein Knoten des Medium abhört, obwohl gar kein Sender aktiv ist.

Es wird somit ein Protokoll benötigt, das sehr sparsam mit den limitierten Energieressourcen der Sensorknoten umgeht. Für den Fall des idle listening bedeutet dies, dass der Knoten sich so wenig wie möglich im Empfangsmodus befindet und er sich möglichst oft ausschalten kann. Ebenso ist aber auch wichtig, dass möglichst wenige Kollisionen auftreten, um unnötige Wiederholungen zu vermeiden. Auch der Kommunikationsoverhead sollte gering sein, um die Pakete möglichst klein zu halten. Es wurden daher Protokolle speziell für Sensornetze entwickelt. Einige sollen im Folgenden vorgestellt werden.

3.1.1. S-MAC (Sensor MAC)

Das S-MAC-Protokoll wurde von Ye, Heidemann, Estrin an der University of Southern California und an der University of California (Los Angeles) entwickelt [Ye-02].

Dieses Protokoll folgt folgendem Grundsatz: Alle Knoten haben einen festen Ablaufplan (schedule). Darin wird die Zeit in Listen und Sleep-Perioden unterteilt. Die Länge der jeweiligen Phasen hängt von der Anwendung ab, längere Sleep-Phasen bedeuten einen geringeren Energieverbrauch, aber auch längere Inaktivitätsphasen, wodurch es bei zeitkritischen Ereignissen zu Problemen kommen kann.

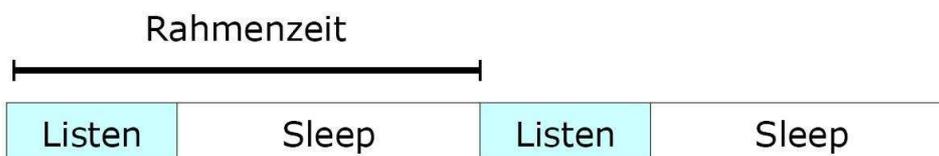


Abbildung 1: S-MAC: Aufteilung in Listen- und Sleep-Phasen

Nach der Aktivierung müssen die Knoten sich in möglichst großen Verbänden (cluster) zusammen finden und einen Koordinator (synchronizer) bestimmen. Dieser legt einen gemeinsamen Ablaufplan fest, dem alle Knoten folgen. In der Listen-Phase findet die Datenübertragung statt. Dabei wird diese Phase nochmals in 2 Abschnitte unterteilt. Es gibt eine Sync- und eine RTS/CTS-Phase. Der erste Bereich ist für den Abgleich der Ablaufpläne bestimmt, die periodisch aufgrund der Uhrendrift abgeglichen werden müssen. In der RTS/CTS-Phase findet die eigentliche Datenübertragung statt. Durch dieses Protokoll lässt sich der Energieverbrauch gegenüber dem 802.11-Protokoll wesentlich senken. Problematisch an diesem Protokoll ist, dass mehrere cluster entstehen können und auch Knoten mehr Verbänden angehören können. Dies ist allerdings auch unbedingt nötig, um einen Datenaustausch zwischen den Verbänden zu ermöglichen. Allerdings müssen diese Knoten die Ablaufpläne aller cluster denen sie angehören berücksichtigen, wodurch ein erhöhter Energieverbrauch entsteht. Somit sind diese wichtigen Knoten früher von Ausfall bedroht.

3.1.2. T-MAC (Timeout MAC)

T-MAC ist eine Weiterentwicklung des S-MAC Protokolls und wurde von van Dam und Langendoen erstellt [van Dam-03]. Eine Veränderung gegenüber S-MAC ist die adaptive Anpassung der Listen-Phase, die beim S-MAC-Protokoll immer gleich lang ist. Die Länge der Listen-Phase kann dem Nachrichtenaufkommen angepasst werden. Werden während einer Listen-Phase keine Nachrichten mehr übertragen, gehen die Knoten in den Sleep-Zustand über, um Energie zu sparen.

Problematisch ist jedoch, dass es bei mehreren sendewilligen Knoten in einer Nachbarschaft zu der Situation kommen kann, dass ein Knoten warten muss, um eine laufende Übertragung nicht zu stören, seinen Empfänger aber nach Ende dieser Übertragung nicht mehr erreichen kann, da sich dieser bereits in der Schlafphase befindet. Die Lösung erfolgt durch die Einführung eines zukünftigen Sendeantrags (FRTS, future request to send), um dem Empfänger den Sendewunsch mitzuteilen (siehe Abbildung 2). In der Abbildung wird gezeigt, wie A ein RTS an B sendet und C gleichzeitig den Wunsch hat an D zu senden. C hat nun die Möglichkeit, nachdem es ein CTS von B gehört hat ein FRTS zu senden, um D anzuzeigen, dass er auch senden möchte. Es kommt zwar bei B zu einer Kollision mit dem ebenfalls neu eingeführten DS-Paket, dies ist jedoch unerheblich, da dieses Paket keine Informationen enthält. Erst danach kommen die Datenpakete.

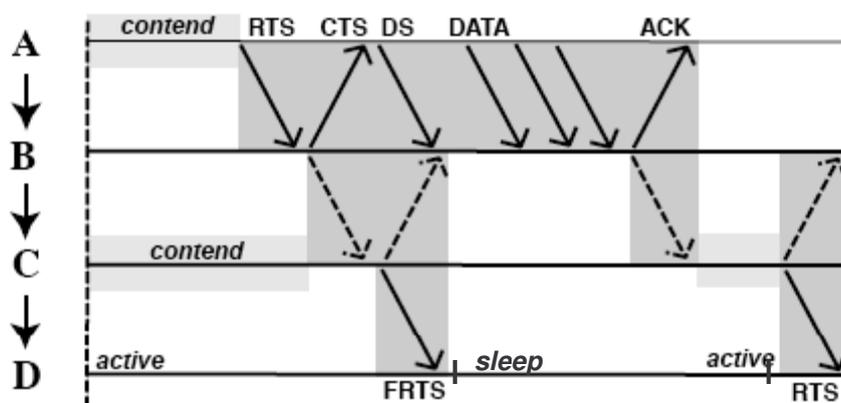


Abbildung 2: Einführung eines future request to send (FRTS)

Abschließend lässt sich feststellen, dass das T-MAC-Protokoll durch die Möglichkeit zur Verkürzung der Listen-Phase, weniger idle listening aufweist, als das S-MAC Protokoll. Allerdings wird dies durch einen erhöhten Verwaltungsaufwand erkauft. In

Vergleichsmessungen (vgl. [van Dam-03]) wurde gezeigt, dass bei gleichmäßigem Nachrichtenaufkommen sowohl T-MAC als auch S-MAC gegenüber den klassischen CSMA ungefähr 98% weniger Energie verbrauchen. Kommt es jedoch zu stark schwankendem Nachrichtenaufkommen ist das T-MAC-Protokoll bis zu 5-mal energieeffizienter als S-MAC.

3.1.3. Berkeley-MAC / B-MAC

B-MAC (vgl. [Woo-01] und [Polastre-04a]) wurde von Woo und Culler an der University of California (Berkeley) im Jahre 2001 entwickelt. Beim Design dieses Protokoll wurde davon ausgegangen, dass kurzfristig ein hohes Verkehrsaufkommen entstehen kann und das die übermittelten Pakete relative kurz sind (ca. 30 Bytes). Weiterhin wurde angenommen, dass es eine eindeutige Basisstation (Datensenke) gibt. Jede Kommunikation zwischen den Knoten hat diese Basisstation zum Ziel, d.h. es existiert eine Baumstruktur und jeder Knoten kennt seinen Vater und weiß somit an wen er seine Nachrichten senden muss.

Berkley-MAC basiert auf dem CSMA-Prinzip, 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 mehrerer Knoten dasselbe Ereignis zur selben Zeit erkennen. Berkeley-MAC verwendet keinerlei Kontrollpakete, stattdessen wird die Technik des parenting genutzt (siehe Abbildung 3).

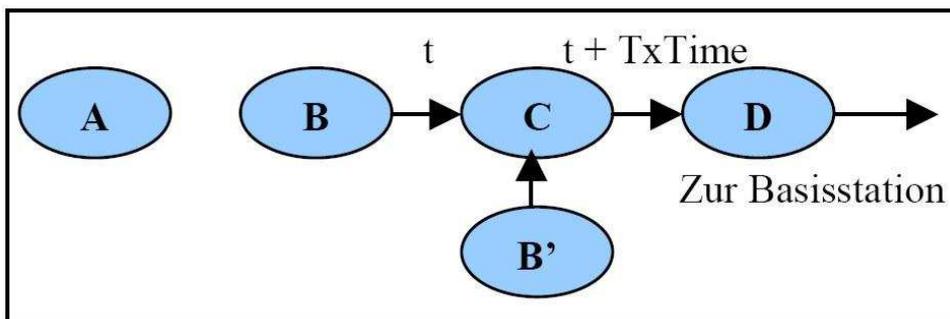


Abbildung 3: Parenting

In dem in der Abbildung gezeigten Ablauf sendet B an C und gleichzeitig möchte A an B senden. A hört das B an C sendet und weiß, dass C das Paket an D weitersenden wird. Nun wartet A $t + TxTime$ ab bis C die Übertragung abgeschlossen hat, um eine Kollision bei B zu vermeiden und kann danach selber senden. Durch diesen Ablauf ist auch gleichzeitig eine indirekte Empfangsbestätigung eingebaut, denn wenn B an C sendet und danach C die Nachricht an D weiterleitet, hört B dies und weiß somit, dass seine Nachricht angekommen ist. Kollisionen können jedoch weiterhin entstehen, wenn beispielsweise B', das außerhalb des Empfangsbereichs von A liegt, an C sendet und gleichzeitig A beginnt eine Nachricht an B zu senden, kommt es zu einer Kollision bei B.

Negativ lässt sich feststellen, dass durch das parenting viel Energie durch die ständige Überwachung des Mediums verbraucht wird und außerdem bei dem geschilderten Ablauf eine gewisse Symmetrie der Knotenabstände vorausgesetzt wird. Dagegen ist positiv am Berkeley-MAC-Protokoll, dass kein Overhead durch Kontrollpakete erzeugt wird.

In [Woo-01] wird gezeigt, dass der Einsatz von Berkeley-MAC sinnvoll ist bei relativ hohem Übertragungsvolumen, da dann das idle listening-Problem nicht so zum tragen kommt. In [Polastre-04a] werden Verbesserung und Erweiterung von Berkeley-MAC beschrieben durch die die genannten Nachteile aufgehoben bzw. verringert werden.

3.1.4. IEEE 802.15.4 und ZigBee

Im IEEE 802.15.4 Standard wird sowohl die MAC-Schicht als auch die physikalische Bitübertragungsschicht definiert (vgl. [Zhao-04] und [Fischer-05]). ZigBee verwendet diesen Standard und erweitert ihn um weitere Schichten. In der ZigBee-Allianz sind viele Hardware-Hersteller vereint, deren Ziel es ist, einfach zu handhabende, standardisierte und universell einsetzbare Geräte zu entwickeln. ZigBee ist optimiert auf geringen Energieverbrauch und auf geringen Datendurchsatz ausgelegt. Es können sowohl Stern- als auch Peer-to-Peer-Topologien gebildet werden, in denen es jeweils einen Koordinator gibt. Der Koordinator fungiert in einem Netz als Datensenke, da über ihn ein Großteil der Kommunikation läuft, ist er in der Regel über eine feste Verbindung an des Stromnetz angebunden. Die anderen Knoten befinden sich einen Großteil ihrer Laufzeit im Ruhezustand und lauschen regelmäßig, ob Nachrichten an sie gerichtet sind oder auf sie warten. Bei der Datenübertragung nutzt die ZigBee MAC-Schicht den CSMA/CA-Algorithmus zur Kollisionsvermeidung.

Jeder Teilnehmer im Netzwerk kann entweder eine vollständig funktionsfähige Einheit (FFD) oder eine beschränkt funktionsfähige Einheit (RFD) sein. Der Koordinator muss allerdings eine vollständig funktionsfähige Einheit (FFD) darstellen.

3.2. Berkeley Motes

Bei den so genannten Motes handelt es sich um kleine Sensorknoten, die an der University of California in Berkeley in Zusammenarbeit mit den Firmen Crossbow¹ und MoteIV² entwickelt wurden.

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 4: Übersicht über die verschiedenen Motes

Auf den Motes sind ein Mikrocontroller, eine Funkübertragungseinheit, ein Flash-Speicher, in dem Daten z.B. für spätere Auswertungen abgelegt werden können, und ein

¹ <http://www.xbow.com/Products/productsdetails.aspx?sid=3>

² <http://www.moteiv.com/>

Interface, an das verschiedene Arten von Sensoren angeschlossen werden können, als Hauptbestandteile enthalten (vgl. [Polastre-04b] und [Jakubowsky-05]).

In Abbildung 4 werden die verschiedenen Typen von Motes gezeigt, wobei besonders die Mica, die Mica2Dot, die Mica2 und die Telos-Mote am interessantesten sind, da diese am leistungsfähigsten und am besten für den wirklichen Einsatz geeignet scheinen. Die Telos-Mote (siehe Abbildung 5), die mit dem Funkmodul CC2420³ der Firma Chipcon ausgerüstet ist, ist zusammen mit einer, in der Abbildung nicht aufgeführten, Variante der Mica2-Mote (bezeichnet als MicaZ⁴), die einzige Mote, die eine Datenübertragung nach dem IEEE 802.15.4-Standard bietet und somit die Möglichkeit mit anderen ZigBee konformen Geräten zu kommunizieren.

Damit ist besonders die Tolos-Mote auch aufgrund des leistungsfähigeren Mikrocontrollers MSP430⁵ der Firma Texas Instruments besonders attraktiv, zumal sie laut einem in [Polastre-04b] angestellten Vergleich eine fast 3 Mal so lange Lebensdauer hat als MicaZ-Mote und eine mehr als doppelte Lebensdauer gegenüber der Mica2-Mote besitzt.

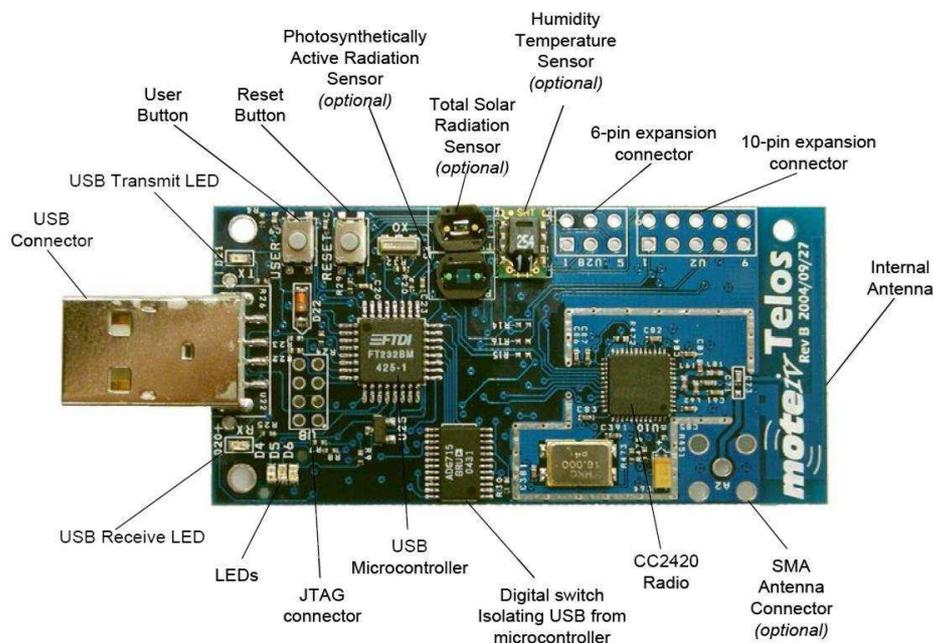


Abbildung 5: Telos-Mote (Forderseite)

3.3. TinyOS

TinyOS⁶ ein Open-Source Betriebssystem für Motes wurde ebenfalls an der University of California in Berkeley entwickelt [Hill-00]. Es werden die in Kapitel 3.2 genannten Berkeley-Motes, aber auch weitere Sensoren, unterstützt, wie u.a. Eyes (EU-Projekt mehrerer europ. Hochschulen⁷), TMote Sky (MoteIV) und iMote (Intel). Da die Motes nur sehr begrenzte Ressourcen aufweisen (Programmspeicher, Arbeitsspeicher, Prozessor, Energie, etc.) war es Ziel von TinyOS ein Betriebssystem zu entwickeln, dass sehr sparsam mit den Ressourcen umgeht. TinyOS ist somit, im Gegensatz zu anderen bereits für Embedded-Systeme entwickelten Betriebssystemen, speziell auf diese Anforderungen zugeschnitten. Die beiden wichtigsten Designaspekte beim

³ http://www.chipcon.com/index.cfm?kat_id=2&subkat_id=12&dok_id=115

⁴ <http://www.xbow.com/Products/productsdetails.aspx?sid=101>

⁵ www.ti.com/msp430

⁶ <http://www.tinyos.net/>

⁷ <http://www.eyes.eu.org/>

Aufbau des Betriebssystems waren es, einen hohen Grad an Nebenläufigkeit zu unterstützen und eine effektive Modularität zu ermöglichen. Bei TinyOS sind Betriebssystem und Anwendung eng miteinander gekoppelt. Dazu wurde eine Komponentenarchitektur gewählt, bei der sowohl die Anwendungen als auch das Betriebssystem aus einzelnen miteinander verbundenen Komponenten besteht und nur die jeweils benötigten Teile gemeinsam kompiliert werden. Als Programmiersprache kommt nesC⁸ zum Einsatz, eine an C angelehnte Sprache, die speziell für kleine vernetzte Embedded-Systeme entwickelt wurde [Gay-03].

4. Kommunikation zum Leitstand

Nachdem zuvor die verschiedenen wesentlichen Komponenten und Bereiche, die für den Aufbau eines drahtlosen Sensornetzes erforderlich sind beschrieben wurden, sollen hier nun der weitere Verlauf dargestellt werden, nachdem ein solches Sensornetz in Betrieb genommen worden ist. Ziel des gesamten Projekts soll es, wie in Kapitel 1 beschreiben, sein, die Arbeit der Rettungskräfte bei der Brandbekämpfung zu erleichtern und zu verbessern. Um dieses Ziel zu erreichen, müssen die ermittelten Sensordaten an einen Leitstand übertragen werden, in dem dann die weitere Auswertung geschieht (siehe Arbeit von Andreas Piening).

Für die Übertragung der Sensordaten muss eine Schnittstelle aus dem Sensornetz zur Außenwelt bestehen. Es muss somit eine Datensinke existieren, die alle ermittelten Sensorwerte zum Ziel hat und über die die Kommunikation zum Leitstand aufgebaut wird. Es ist allerdings wichtig, dass dieser Endpunkt des Netzes redundant ausgelegt wird, insbesondere aus dem Grund, da es bei dem hier zugrunde gelegten Szenario „Gebäudebrand“ zum Ausfall von Teilen des Sensornetzes beispielsweise aufgrund von Feuer kommen kann. Ein weiterer Punkt den es abzusichern gilt, ist die tatsächliche Übertragung der Daten aus dem überwachten Gebäude heraus an die Leitstelle. Diese kann beispielsweise über das Internet erfolgen. Es ist jedoch sicherzustellen, dass auch der Netzzugangspunkt redundant und abgesichert ausgelegt wird. Hierbei ist auch die Möglichkeit eines Ausfalls des Stromnetzes in dem Gebäude zu berücksichtigen. Bei der Kommunikation über das Internet muss die Datenintegrität, auch gegenüber böswilligen Angriffen sichergestellt werden.

5. Meine Ziele für das nächste Semester

Im nächsten Semester möchte ich mich als erstes mit den tatsächlichen Realisierungsmöglichkeiten, des hier nur theoretisch beschriebenen Aufbaus eines Sensornetzwerks, beschäftigen. Dabei wird zum einen wichtig sein, welche Arten von Sensorenknoten in welcher Form realisierbar sind. Hier wäre eine Möglichkeit sich eingehender die Berkeley Motes anzuschauen, z.B. die Telos-Mote, die bereits über integrierte Sensoren für Temperatur, Feuchtigkeit und Helligkeit verfügt, und sich parallel dazu mit dem Betriebssystem TinyOS vertraut zu machen. Dieser Bereich würde bereits die Möglichkeit bieten, sich eingehender mit den einzelnen Komponenten der Motes und deren Zusammenspiel vertraut zu machen. Ebenso wie mit dem Aufbau des Betriebssystems TinyOS. Aber auch die konkrete Entwicklung von TinyOS-Anwendungen wäre ein Arbeitsgebiet.

Ebenso ist es wichtig, sich über den Aufbau des Sensornetzwerks Gedanken zu machen. Hier spielt die Art der Kommunikation der Knoten untereinander und die Topologie des

⁸ <http://nesc.sourceforge.net/>

Netzwerks eine Rolle, die es zu untersuchen gilt. Dies muss in Abhängigkeit mit den für die Realisierung gewählten Sensorknoten geschehen.

Für das Voranschreiten des gesamten Projekts muss es darum gehen, die Schnittpunkte und Überschneidungen der einzelnen Bereiche der beteiligten Personen festzustellen. Hier ist es aus der Sicht meines Anwendungsbereichs beispielsweise nötig, eine Schnittstelle zum Leitstand zu definieren, über die die Daten des Sensornetzes an die Leitstellen weitergegeben werden können.

Über diese Arbeiten an der Realisierung hinaus, wird es ebenso nötig sein, sich differenzierter mit den Aufgaben und Problem in einem drahtlosen Sensornetzwerk zu beschäftigen. Hierzu zählen auch Thema, die bereits in meinem Vortrag angedeutet, aber in dieser Ausarbeitung noch gar nicht erwähnt sind, wie z.B. die Positionsbestimmung innerhalb des Netzwerks oder die Zeitsynchronisation.

Zusammenfassend lässt sich feststellen, dass dieses Projekt noch genügend Raum lässt sich intensiver damit zu beschäftigen. Da es aber meiner Meinung nach ein sehr spannendes Thema ist, freue ich mich bereits darauf, mich damit näher auseinander setzen zu können.

Abbildungsverzeichnis

Abbildung 1: S-MAC: Aufteilung in Listen- und Sleep-Phasen	6
Abbildung 2: Einführung eines future request to send (FRTS)	7
Abbildung 3: Parenting	8
Abbildung 4: Übersicht über die verschiedenen Motes	9
Abbildung 5: Telos-Mote (Forderseite)	10

Literaturverzeichnis

- [Fischer-05] Fischer, C.: **Entwicklung von ZigBee-Modulen für spontane Funknetzwerke**; Bachelorarbeit an der HAW Hamburg; 2005
- [Gay-03] Gay, D.; Levis, P.; von Behren, R.; Welsh, M.; Brewer, E.; Culler, D.: **The nesC Language: A Holistic Approach to Networked Embedded Systems**; Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation; San Diego, CA, USA; ACM Press; 2003
- [Hill-00] Hill, J.; Szewczyk, R.; Woo, A.; Hollar, S.; Culler, D.; Pister, K.: **System architecture directions for networked sensors**; ACM SIGOPS Operating Systems Review; ACM Press; 2000
- [Jakubowsky-05] Jakubowsky, F.: **MOTEs – Evaluation und exemplarische Anwendung einer neuen Technologie**; Diplomarbeit an der HAW Hamburg; 2005
- [Jones-05] Jones, W; Holmberg, D; Davis, W.; Evans, D.; Bushby, S.; Reed, K.: **Workshop to Define Information Needed by Emergency Responders during Building Emergencies**; National Institute of Standards and Technology; 2005
<http://fire.nist.gov/bfrlpubs/fire05/PDF/f05017.pdf> (Zugriff: 14.07.2006)
- [Karl-05] H. Karl, A. Willig: **Protocols and Architectures for Wireless Sensor Networks**, John Wiley & Sons; 2005
- [Polastre-04a] Polastre, J.; Hill, J.; Culler, D.: **Versatile low power media access for wireless sensor networks**; Proceedings of the 2nd international conference on Embedded networked sensor systems; Baltimore, MD, USA; ACM Press; 2004
- [Polastre-04b] 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
<http://webs.cs.berkeley.edu/papers/hotchips-2004-motes.ppt> (Zugriff: 17.07.2006)

- [van Dam-03] van Dam, T.; Langendoen, K.: **An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks**; Proceedings of the 1st international conference on Embedded networked sensor systems; Los Angeles, CA, USA; ACM Press; 2003
- [Woo-01] Woo, A.; Culler, D.: **A transmission control scheme for media access in sensor networks**; Proceedings of the 7th annual international conference on Mobile computing and networking, Rome, Italy; ACM Press; 2001
- [Ye-02] Ye, W.; Heidemann, J.; Estrin, D.: **An Energy-Efficient MAC protocol for Wireless Sensor Networks**; INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE; 2002
- [Zhao-04] Zhao, F.; Guibas, L.: **Wireless Sensor Networks**; Elsevier; 2004