Studiengang: Informationstechnik

Entwicklung eines externen Sensornetzes mit WLAN Kopplung und Visualisierung

Studienarbeit

Im Rahmen des 5. und 6. Theoriesemesters

Abgabedatum 11.5.2015

Verfasser	Maik Maier, Nicolai Staege
Matrikelnummer	4050846, 4615051
Kurs	TINF12B3
Betreuer	Herr Prof. Hans-Jörg Haubner

Erklärung

Gemäß §5 (3) der „Studien- und Prüfungsordnung DHBW Technik“ vom 22. September 2011.

Ich habe die vorliegende Studienarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Karlsruhe, Datum

Unterschrift

Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

IoT	Internet of Things
SPOT	Small Programmable Object Technology
I/O	Input/Output
USB	Universal Serial Bus
M2M	Machine-to-Machine
MHz	Megahertz
CPU	Central Processing Unit
SRAM	Static Random Access Memory
G	G-Kraft - Gewichtskraft
IEEE	Institute of Electrical and Electronics Engineers
IDE	Integrated Development Environment
USART	Universal Asynchronous Receiver Transmitter
I^2C	InterIntegrated Circuit
mA	Milli-Ampere
EEPROM	Electrically Erasable Programmable Read-Only Memory
LED	Licht-emittierende Diode
RGB	Rot, Grün und Blau

Kapitel 1

Einleitung und Intention

Der Computer ist mittlerweile zum festen Bestandteil im alltäglichen Leben geworden. Mit ihm können viele Aufgaben wie Recherchen, komplexe Rechnungen und Kommunikation vereinfacht und schnellstmöglich erledigt werden. Während vor einigen Jahren noch der Desktop-PC die beliebteste Wahl darstellte, geht der Trend mittlerweile in Richtung der mobilen Endgeräte wie z.B. Smartphone, Laptop oder auch Tablet. Menschen wollen sich nicht an einen Ort binden, an dem sie ihren Computer benutzen können und sehnen sich nach dem Wunsch, dass alle Alltagsgegenstände per Smartphone oder Tablet kontrollierbar werden.

Diese Vernetzung aller elektronischen Geräte in einem Haushalt wird als "Internet of Things"(kurz IoT) bezeichnet. Die grundsätzliche Idee besteht darin, dass alle elektronischen Geräte wie z.B. Kühlschrank, Backofen u.a. miteinander kommunizieren können und der Nutzer über sein mobiles Endgerät alle Daten der vernetzten Geräte einsehen und diese auch auf seinen Wunsch hin steuern kann. Nähere Informationen zu IoT folgen im nächsten Kapitel.

Zur beispielhaften Demonstration des Aufbaus eines solchen Netzes elektronischer Geräte beschäftigt sich diese Studienarbeit mit Oracle SunSpot, einem Sensornetzwerk bestehend aus 2 Sensoren und einer Basisstation. Im Folgenden wird die Inbetriebnahme und Programmierung dieser Sensoren vorgenommen und die darin enthaltene Technik erklärt. Ziel der Studienarbeit ist es, mit Hilfe von SunSpot eine rudimentäre Raumüberwachung zu programmieren, indem bewegte Fenster oder Türen bei Abwesenheit des Besitzers der Wohnung erkannt werden, die Basisstation die Werte sammelt und sie an den Besitzer meldet.

1.1 Ziel dieser Arbeit

Hier werden wir das Ziel dieser Arbeit sowie das erwartete Ergebnis niederschreiben. Im Fazit kann hierauf Bezug genommen werden, um rückblickend den Erfolg dieser Arbeit zu messen.

Kapitel 2

Internet of Things

Als im Februar 1946 ENIAC, der erste elektronische sowie programmierbare Universalrechner vorgestellt wurde, wog dieser 27 Tonnen und füllte einen gesamten Raum. Für private Anwendungen waren diese Rechnersysteme nicht geeignet. Mit der voranschreitenden Entwicklung werden Computer immer kleiner und leistungsfähiger. Es erschließen sich immer neue Anwendungen von Computersystemen, die hauptsächlich den Menschen in seinem Alltagsleben unterstützen sollen.

Rund um 1990, als das Internet kommerzialisiert und somit für jeden zugänglich wurde, begann eine rasante Entwicklung neuer Technologien. Bis heute hat es unsere Arbeitsweise sowie unser Privatleben verändert und dieser Trend schreitet ungebremst voran.

Mit dem Web 2.0 wurden Webseiten interaktiv sowie Videos und Bilder im Internet eine Selbstverständlichkeit. Soziale Netzwerke verbinden die Nutzer durch verschiedenste Arten der Kommunikation. Dieses sich immer weiter aufspannende Kommunikations- und Informationsnetz, erreicht nun auch unsere kleinsten elektronischen Geräte.
Das Haus wird durch ein komplexes Sicherheitssystem überwacht, die Tür benötigt nur den Fingerabdruck um sich automatisch zu öffnen, der Fernseher reagiert auf Spracheingaben und in der Zukunft erstellt der Kühlschrank autonom den Einkaufszettel.

All diese Informationen werden über das Internet zu einer zentralen Sammelstelle oder dem Menschlichen Akteur zugespielt, das Internet of Things (IoT) ist entstanden.

2.1 Geschichte

Etwa um 1982 ärgerten sich drei Studenten der School of Computer Science, an der Carnegie Mellon University über den Getränkeautomaten ihres Instituts. An manchen Tagen liefen sie zu dem Automaten und erhielten entweder keine Getränke, oder zu warme, da diese erst kürzlich nachgefüllt wurden. Um diese Problematik zu lösen entwickelten die Studenten John Zsarnay neue Hardware die mit Software von David Nichols und Ivor Durham die Füllstände, sowie die Temperatur der Getränke überwachte [Car98].
Über den damals an der Universität vorhandenen Vorgänger des Internets, das sogenannte Arpanet konnte direkt beim Automaten der aktuelle Status nachgefragt werden. Dieser antwortete zum Beispiel mit:

```
1 >                    EMPTY    EMPTY    1h 3m
2 >                    COLD     COLD     1h 4m
```

Hiermit informierte der Automat darüber, dass kalte Getränke in der mittleren sowie linken Schiene vorhanden seien, die Getränke der rechten Schiene jedoch noch warm seien. Die angegebene Zeit informierte darüber, wie lange die Getränke sich bereits im Automat befanden. Nach drei Stunden nahm der Automat an Getränke seien ausreichend gekühlt.

Bereits 1991 schrieb der Amerikanische Informatiker Mark Weiser eine Vision, wie technische Geräte der Zukunft untereinander vernetzt sein könnten [Wei91]. Den Namen IoT erhielt das ganze jedoch erst 1999.

2.2 Ziele und Anwendungsbeispiele

Aus Spielereien und purem Erfindergeist wurden in wenigen Jahren eine ganze Industrie, die sich heute nur mit Produkten des IoT beschäftigt. Es entstanden bereits viele Projekte, denen man im Alltag begegnet, ohne sie Wahrzunehmen. Diese lassen sich in 3 Hauptkategorien unterteilen, die gleichzeitig die Ziele des IoTs darstellen:

- Automatisierung
- Informationsgewinnung über bessere Vernetzung
- Entertainment

In der folgenden Tabelle haben sind einige der Erfolgreichsten davon Zusammengestellt.

- Umweltsensoren (Temperatur Feuchtigkeit Erschütterung Lautstärke Luftzusammensetzung)
- Lichtsteuerung
- Haushaltshilfen
- Bestandsaufnahme / Nachfuhrkontrolle
- Überwachungsfunktionen
- „Smart Signs“ - Autobahn
- Entertainment
- Haussteuerung
- Prozessüberwachung (Ventile, Flussraten usw.)
- Diagnose / Lebensüberwachung usw.

2.3 Hardware für Privatanwender

In den letzten Jahren sind viele Privatanwender aufgrund der sich bietenden Möglichkeiten dazu übergegangen, eigene Anwendungen mit kleinen IoT-Systemen selbst zu entwickeln. Hierzu werden meist kleine Einplatinencomputer verwendet, da diese sehr günstig zu erhalten sind. Zwei der erfolgreichsten Einplatinen Computer sind die Modelle der Raspberry Pi und Arduino Familie.

2.3.1 Arduino-Plattform

Arduino ist ein Unternehmen, dass all seine Produkte in enger Zusammenarbeit mit zugehörigen Community entworfen hat. Alle Entwicklungen basieren auf den Ideen und Konzepten, die die Gemeinschaft um das Unternehmen entwickelt hat.

Abbildung 2.1: Das Arduino-Logo [Arda]

Das Unternehmen entwickelt Mikrokontrollerplatinen, die ihre Umwelt durch Sensoren wahrnehmen und auf äußere Einflüsse reagieren können. Alle Produkte sind vorgefertigt oder als Selbstbau-Kit erhältlich. Da die Baupläne aller Platinen durch Arduino quelloffen im Internet erhältlich sind, können die Produkte auch von Grund auf nachentworfen werden.

Das erste Arduino-Board, dass 2005 vorgestellt wurde, hat bereits viele Revisionen erhalten. In der aktuellsten Version ist es mit einem 16Megahertz (MHz) 8-Bit Microcontroller ATmega328 der Firma Atmel bestückt. Es besitzt zusätzlich 32 Kilobyte Flash-Speicher von dem 0.5 Kilobyte durch den Bootloader belegt werden. Zur Ein- und Ausgabe, sowie zum anschließen von Sensoren besitzt die Platine 14 digitale Input- oder Output-Pins, sowie 6 analoge input Pins und einen Universal Serial Bus (USB)-Anschluss. Betrieben wird die Platine mit einem 7-12V Netzteil [Ardb]. Die Platine wird von dem italienischen Unternehmen SmartProjects produziert, ist jedoch aufgrund des quelloffenen Platinenplans auch selbst zusammensetzbar.

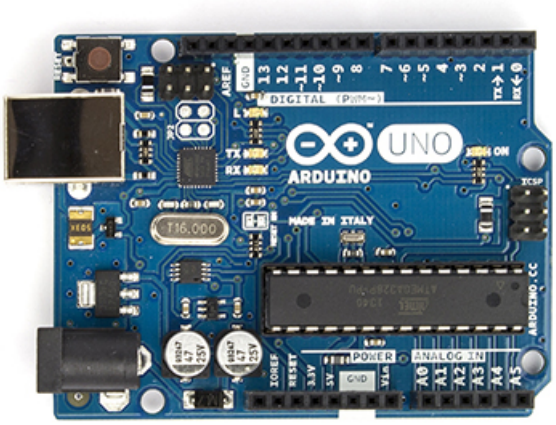

Abbildung 2.2: Die Front der dritten Arduino Uno Revision [Ardc]

Die Arduino-Plattform besitzt eine eigene Integrated Development Environment (IDE), in der die Software für den Microcontroller entwickelt werden kann. Der Mikrocontroller wird mit einer stark vereinfachten Form der Programmiersprachen C und C++ angesprochen. Insbesondere technische Informationen wie Header-Files werden vor dem Programmierer verborgen. Für Anfänger bietet die IDE mehrere Einstiegshilfen, die das Programmieren erleichtern.
Ein Beispiel für die Einstiegshilfen sind die strukturgebenden Funktionen setup(), sowie loop(). Während in der Setup-Methode alle Aktionen definiert werden müssen, die beim Programmstart durchgeführt werden sollen, wird der Code in der Loop-Methode unbegrenzt oft wiederholt. Ein Beispielcode, der eine an Pin 13 angeschlossene LED blinken lässt, lautet wie folgt:

Listing 2.1: Simpler Arduino-Code, der eine LED blinken lässt

```
1  int led = 13; //LED an Pin 13
2
3  void setup() {
4     pinMode(led, OUTPUT);
5  }
6
7  void loop() {
8     digitalWrite(led, HIGH);
9     delay(1000);
10    digitalWrite(led, LOW);
11    delay(1000);
12 }
```

Seit der Entwicklung der Arduino-Boards erfreuen sich diese großer Beliebtheit. So wurden bis 2013 insgesamt 700.000 zusammengesetzte Platinen verkauft [Cua13].

Viele der Projekte, die mit Arduino-Platinen durchgeführt werden beinhalten Benutzerinteraktion. Auch für Kunstinstallationen werden die Einplatinencomputer gerne verwendet [Ele15].

2.3.2 Raspberry Pi

Die in Großbritannien angesiedelte Wohltätigkeitsorganisation "Raspberry Pi Foundation"hat es sich zur Aufgabe gemacht, Einplatinencomputer für Menschen zu entwickeln, die Hardware- oder Programmierkenntnisse erwerben wollen. Um auch für Jugendliche und Studenten ansprechend zu sein, wurde ein besonders geringer Verkaufspreis angestrebt. Die verschiedenen Modelle des Raspberry Pis kosten zwischen 20 und 35 US-$ und sind somit sehr günstig.

Der Name setzt entstand durch die aus der Kombination zweier Gedanken. Viele Computer wurden nach Früchten benannt (z.B. Apple), aus diesem Grund entschieden sich die Gründer für Raspberry. Der Namenszusatz Pi entstand aus dem Ursprünglichen plan, die Raspberry Pi Einplatinencomputer mit einem Phyton Interpreter auszuliefern, daher das die Abkürzung PI. Als Ergebnis eines öffentlich ausgeschrieben Wettbewerbs wurde eine silisierte Himbeere als Logo für die Wohltätigkeitsorganisation gewählt.

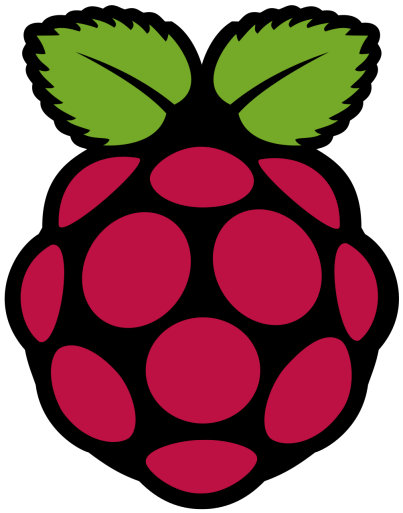

Abbildung 2.3: Das Logo der Raspberry Pi Foundation [Ras]

Insgesamt bietet die Raspberry Pi Foundation sechs Modelle des Computers an. Eine sehr grundlegende ausführung, das Compute Module, dass nur in Stückzahlen über 100 verkauft wird, sowie die günstigen Modelle A und A+, die mit 25 und 20 US-$ am unteren Preisende liegen. Die Modelle B, B+, sowie das überarbeitete Modell Generation 2 B sind am oberne Preisende, mit jeweils 35 US-$ angesiedelt.

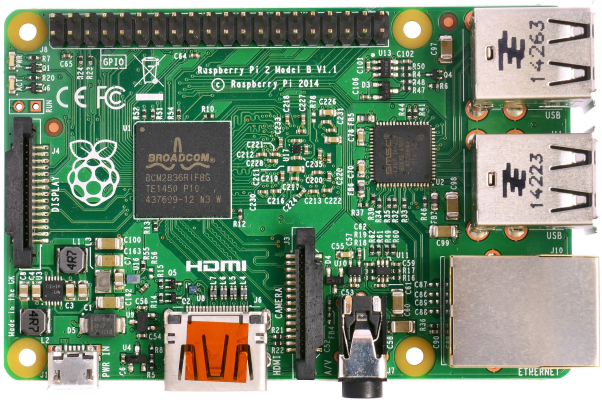

Abbildung 2.4: Die Oberseite des Raspberry Generation 2 B-Modells [Mul15]

Mit dem Anfang 2015 vorgestellten Generation 2 B - Modell wurde ein Performancesprung realisiert. Mit dem System-on-a-Chip Broadcom BCM2836, das sowohl CPU als auch GPU des Einplatinencomputers darstellt sind auf vier Kernen berechnungen mit bis zu 900 **Mhz!** (**Mhz!**) möglich. Dies wird mit 1 Gigabyte Arbeitsspeicher zu einem, für den geringen Preis, leistungsstarken System kombiniert. Die Hardwarekomponenten des Generation 2 B -Modells sind so leis-

tungsstark, dass das ausführen von Windows 10 möglich ist.

Auch in Sachen I/O-Anschlüsse ist der Raspberry Pi gut ausgerüstet. Er stellt insgesamt 17 Pins bereit, die sowohl als Input, sowie Output genutzt werden können. Zusätzlich verfügt der Raspberry Pi über 4 USB-Ports, einen HDMI-Anschluss und einen MicroSD-Karten Slot. Dieser wird als Speicher für das Betriebssystem und alle Programme verwendet.
Um den Raspberry Pi mit Netzwerken zu vebrinden besitzt der Raspberry ein 100Mbit Netzwerkanschluss.

Über Micro-USB wird die Platine mit

2.4 Sicherheitsaspekte

Über das Internet werden immer mehr Informationen versendet. Ein kleiner Bestandteil hiervon sind auch die Informationen die verschiedene Geräte des IoTs untereinander austauschen. Diese automatisch abgewickelte Machine-to-Machine (M2M)-Kommunikation stellt ein hohes Sicherheitsrisiko für Unternehmen und Privatpersonen dar. Durch einen gezielten Angriff könnten personenbezogene oder sicherheitsrelevante Informationen an die Öffentlichkeit gelangen. Aufgrund dessen müssen sich die Verantwortlichen die Frage stellen, wie man mit dieser Herausforderung in der Zukunft umgeht.

Sicherheitsexperten sind sich bereits heute einig, dass das IoT ein enormes Risiko darstellt [Wil15]. Für Privatpersonen äußert sich das hauptsächlich in der Sicherheit ihrer Elektronischen Geräte. So musste der Automobilhersteller BMW kürzlich ein Softwareupdate für seine Automobile mit dem ConnectedDrive-System ausliefern, da es Hackern ohne Spuren zu hinterlassen gelungen war, die Türen mittels Smartphones zu öffnen [ZEI15].

Zusätzlich zur Beschädigung oder Entwendung von Eigentum durch Hacker besteht ein Risiko private Informationen zu verlieren. Gerade Informationen wie E-Mail-Adressen oder Passwörter stehen in großem Interesse der Hacker. Solche könnten über ein privates IoT entwendet werden und für weitere Cyber-Kriminelle Aktionen genutzt werden.

Auch Firmen müssen sich die Risiken bewusst machen, die sie durch die Benutzung von IoT-Geräten eingehen. Oftmals nutzen Unternehmen IoT-Systeme als Infrastrukturkomponenten über die sie einen Service betreiben. Solche Netze

stellen einen Idealen Angriffspunkt für Wirtschaftsspionage oder Denial of Service - Attacken dar. Bricht ein Hacker in solche Systeme ein, wird es ihm schnell möglich sein einen Millionenschaden anzurichten.

Da die Entwicklung des IoT derzeit noch an fahrt aufnimmt, muss jedem Nutzer bewusst sein, dass auch die Sicherheit noch nicht vollständig Entwickelt ist. Viele Aspekte werden sich in den nächsten Jahren weiterentwickeln, die Sicherheit wird jedoch zunehmend eine der wichtigsten Komponenten des IoTs.

Jede am IoT teilnehmende Komponente muss einen Zugang zur aufgebauten Netzwerkumgebung besitzen um ihre Metadaten an andere Teilnehmer zu kommunizieren. Über diese zusammengesammelten Informationen trifft das System eine Entscheidung zu handeln. Hierbei entsteht ein Sicherheitsrisiko.
Mit netzwerkfähigen Geräten ist es möglich, sich in das aufgebaute Netzwerk einzuklinken, und sich ebenfalls als Komponente auszugeben. Dies ermöglicht einem Angreifer Daten abzufangen oder zu seinen Zwecken zu manipulieren. Dies kann das verhalten des Gesamten Systems verändern.
Ebenfalls ist es möglich, dass ein Angreifer ein Teilnehmer des IoTs übernimmt und somit Zugriff auf das Netzwerk erhält.

Kapitel 3

Theoretische Grundlagen

In diesem Kapitel erklären wir kurz, dass wir in den folgenden Unterkapiteln Grundlagen erklären. Worum es sich hierbei handelt können Sie den folgenden Kapiteln entnehmen.

3.1 Wireless Sensor Networks

Hier wird alles zu Wireless Sensor Networks erklärt. Bisher steht das erst als Gerüst.

3.1.1 Ubiquitäres Rechnen

1988 verwendete Mark Weiser erstmals den Begriff 'ubiquitous computing' (dt. ubiquitäres Rechnen), um seine Vision nach einem stets verfügbaren Rechensystem, welches dem Nutzer unsichtbar erscheinen soll, zum Ausdruck zu bringen. Der Computer soll sich so in den Alltag integrieren, dass die Menschen ihn gar nicht mehr bemerken. Nach seiner Vorstellung verbessere das ubiquitäre Rechnen die Erfahrungen, die man mit Computern macht, da die Rechner dem Nutzer nahtlos verfügbar gemacht werden, ohne dabei effektiv sichtbar zu sein.

Weiser zufolge sind die besten Technologien diejenigen, die scheinbar verschwinden, tatsächlich jedoch nur in den Hintergrund geraten und unsichtbar werden. Der Mensch soll nicht in der Welt des Computers leben, sondern der Rechner soll sich in die Welt des Menschen integrieren. In Lichtschaltern, Thermostaten, Stereoanlagen und Backöfen werden bereits heute kleine Rechner verbaut, die helfen sollen, den Alltag zu erleichtern und die Idee des 'Internet of

Things' weiter zu verfolgen.

Da Ubiquitäres Rechnen zuverlässig und unsichtbar funktionieren soll, ist die Technologie der unsichtbaren Rechenmodule von großer Bedeutung. Voraussetzungen sind z.B. leistungsstarke Prozessoren, ausreichend Speicherplatz, drahtlose Kommunikation, Sensoren und Aktoren (die z.B. mit der Umwelt und dem Menschen interagieren). Der Mensch muss nicht für alle Anwendungsfälle von ubiquitärem Rechnen direkt eingebunden werden, da die Systeme auch autonom arbeiten können [WB12].

3.1.2 Motivation von Sensornetzen

Sensornetze sind sehr flexibel und können unter anderem dafür eingesetzt werden, um

- Umwelteinflüsse wahrzunehmen ('sensing')
- Umwelteinflüsse zu verarbeiten und zu analysieren ('computing')
- Daten zu übertragen ('transport')
- Netzwerke für verteile Systeme aufzubauen ('networking')
- Die Umwelt zu beeinflussen und zu verändern ('actuation')

Für viele Anwendungsfälle und Szenarien, in denen mit der Umwelt interagiert wird, soll die Benutzung von drahtlosen Sensornetzen zukünftig ausgebaut und etabliert werden. Der Einsatz von Sensornetzen kann dabei verschiedene Motivationen und Anforderungen haben:

- Direkte Interaktion mit Menschen ist nicht möglich oder nicht erforderlich (z.B. bei Überwachung einer Maschine in der Industrie)
- Der Mensch soll nur im Notfall alarmiert werden (z.B. in Notfällen oder wenn die Sensoren bestimmte Schwellenwerte erreichen)
- Es handelt sich um ein autonomes System, welches nur Selten das Handeln eines Menschen erfordert

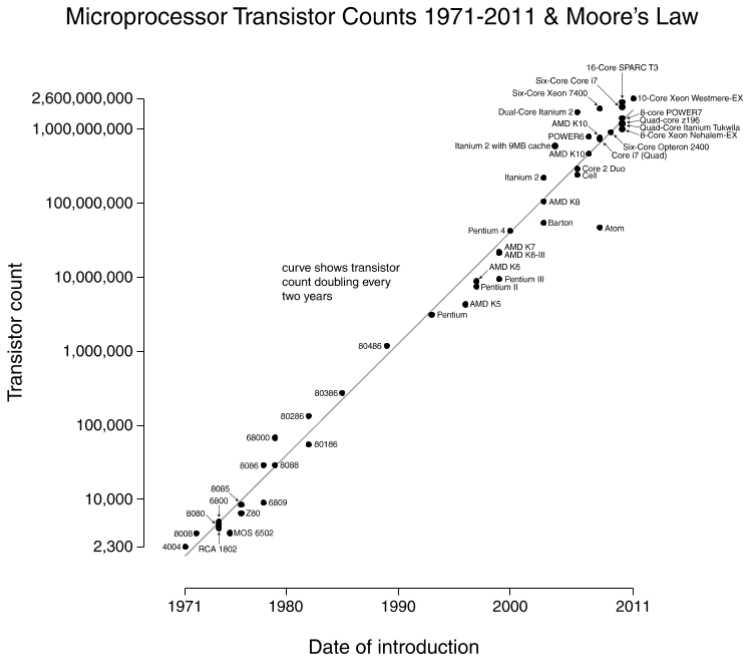

Abbildung 3.1: Mikroprozessoren-Transistoren im Laufe der Zeit [Wgs]

Eine weitere Motivation der Verwirklichung von drahtlosen Sensornetzen ist der Technologiefortschritt, der es möglich macht, immer kleinere Rechengeräte mit mehr Leistung herzustellen und miteinander zu vernetzen. Um diesen Fortschritt zu verdeutlichen, formulierte Gordon Moore 1965 ein Gesetz, welches besagt, dass sich die Anzahl der integrierten Schaltkreise auf einem Mikroprozessor alle 18-24 Monate verdoppelt. Im Gegensatz zu den Anzahl der Schaltkreise steigt die Rechenleistung der Prozessoren allerdings nicht linear an, da immer mehr Schaltkreise für den Cache des Prozessors verwendet werden, was nur geringfügig der Leistungssteigerung dient. Das Ende der Gültigkeit des Mooreschen Gesetzes wurde schon des öfteren wegen unüberwindbarer technischer Grenzen vorausgesagt, diese wurden jedoch bisher alle mit dem Einsatz neuer technischen Mittel und Materialien überwunden. Momentan schätzt der Halbleiterhersteller Intel, welcher 1968 von Moore mitbegründet wurde, dass das Mooresche Gesetz noch mindestens bis 2023 seine Gültigkeit behält. Mittlerweile existieren beim Konzern sogar explizite Pläne, die das Einhalten des Mooreschen Gesetzes sicherstellen sollen. [WB12] [Kah12] [Tuo02].

3.1.3 Bestandteile

Um die Kommunikation der einzelnen Knoten untereinander zu koordinieren, gibt es neben den normalen Sensoren weitere Bestandteile eines 'Wireless Sensor Network'.

Aktoren werden dazu benötigt, um das Sensornetz Einfluss auf die Umwelt nehmen zu lassen.

'Aggregating Nodes' werden dazu gebraucht, um die Daten von verschiedenen Sensoren zu sammlen, zu verarbeiten und zu kombinieren.

'Sink Nodes' sammeln die Daten von verschiedenen Sensoren und geben diese an die Basisstation bzw. das Backend-System weiter.

Backend-Systeme dienen zu weiteren Verarbeitung und Analyse der Daten. Diese sind von Vorteil, wenn z.B. komplexere Berechnungen durchgeführt werden sollen oder die Daten langfristig gespeichert werden sollen [WB12].

3.1.4 Topologien

Bei einem Aufbau eines Sensornetzes stellt sich grundsätzlich die Frage, wie die einzelnen Sensoren miteinander Verbindungen aufbauen und kommunizieren sollen. Ein solche Verbindungsstruktur nennt sich in der Informatik 'Topologie'. Da das Sensornetz insgesamt zuverlässig arbeiten soll, Kosten und Komplexität jedoch gering gehalten werden sollen, wurden speziell für die drahtlosen Sensornetze neue Ansätze im Bereich der Topologie erforscht. Im folgenden sollen 4 Topologie-Alternativen näher erläutert werden.

Peer-to-Peer Netzwerke erlauben es, das jeder Knoten im Netz (in unserem Fall der Sensor) mit jedem anderen Knoten direkt Kontakt aufnehmen kann. Jedes 'Peer-Gerät' ist gleichzeitig Client und Server gegenüber anderen Knoten im Netzwerk.

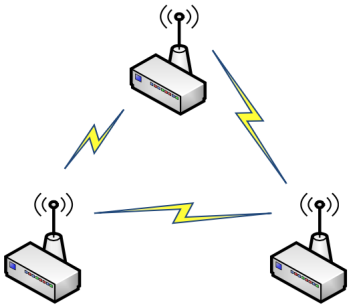

Abbildung 3.2: Peer-To-Peer Netzwerk [Kos09]

Bei der Stern-Topologie sind die Sensoren an ein zentrales Kommunikationsgerät angebunden. In diesem Fall kommunizieren die einzelnen Knoten nicht direkt miteinander. Jegliche Art von Kommunikation wird über das zentrale Gerät (auch Hub genannt) geroutet. Der Hub wird hier als Server betrachtet, wohingegen die Knoten (Sensoren) die Clients darstellen.

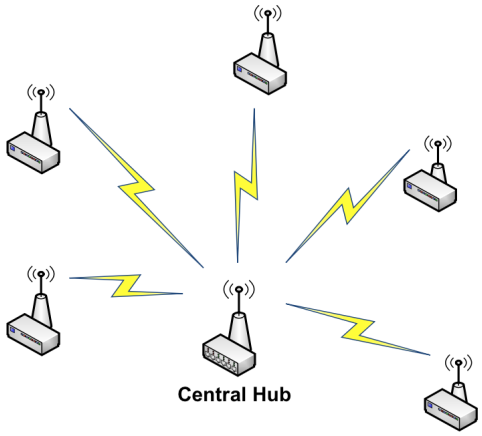

Abbildung 3.3: Stern Netzwerk [Kos09]

Die Baum-Topologie stellt eine Hybridvariante aus Peer-to-Peer und Stern dar. Sie nutzt einen sogenannten 'Root-Knoten' als zentraler Router. Eine Ebene darunter liegen die Hubs, an denen wie in der Stern-Topologie die Sensoren angebunden sind.

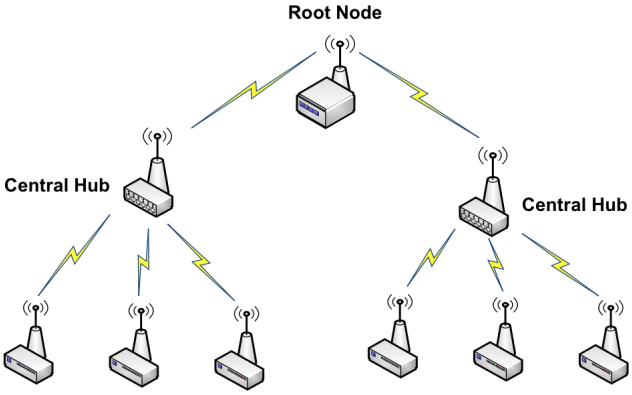

Abbildung 3.4: Baum Netzwerk [Kos09]

Eine weitere Mögliche Variante ist ein vermaschtes Netz. Die Knoten sind untereinander ohne zentralen Hub verbunden und die Daten werden einfach von Knoten zu Knoten weitergesendet, bis sie ihr gewünschtes Ziel erreicht haben [Kos09].

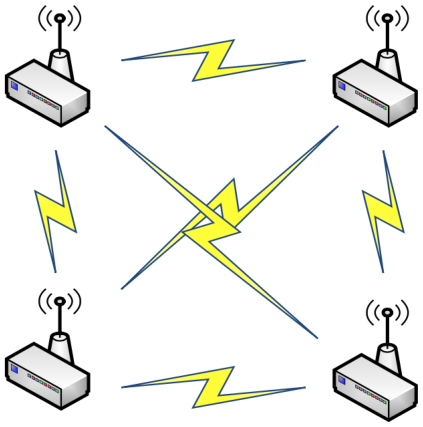

Abbildung 3.5: Vermaschtes Netzwerk [Kos09]

3.1.5 Schwierigkeiten

Beim Planen eines drahtlosen Sensornetzwerkes stellen sich vermehrt eine Schwierigkeiten heraus, die zu berücksichtigen sind. Viele dieser Schwierigkeiten sind auch untereinander abhängig und beeinflussen gleichzeitig die verwendete Elektronik, physikalische Aspekte, Rechenleistung oder auch Lebensdauer eines Sensors.

Die erste Schwierigkeit, die es bei einem Sensor zu betrachten gilt, ist die Versorgung des Sensors mit Energie. Dazu gibt es unterschiedliche Ansätze, wie z.B.

die Versorgung des Sensors mit Batterien. Dies hat allerdings den Nachteil, dass deren Lebenszyklus beschränkt ist und Batterien früher oder später ausgetauscht werden müssen. Ein Akkumulator eignet sich hier schon besser, allerdings bleibt die Frage, wie der Akkumulator mit neuem Strom versorgt werden soll.

In Zusammenhang mit dieser Fragestellung ist die Betrachtung der möglichen Gewinnung oder Rückgewinnung von Energie durch den Sensor von Vorteil. Wie kann ein Sensor Energie gewinnen, ohne dass er damit direkt versorgt werden muss? Möglichkeiten wäre die Gewinnung von Solarenergie durch den Sensor, das Nutzen von Temperaturunterschieden in der Umwelt, die Rückgewinnung der Bewegungsenergie (z.B. durch Wind o.ä.) oder das Ausnutzen von Erschütterungen und Vibrationen. Diese Möglichkeiten sollten nur in Betracht gezogen werden, wenn der Sensor eine lange Lebensdauer mit sich bringen soll. Ist der Einsatz der Sensoren in absehbarer Zeit vorüber, lohnt es sich aus Produktions- und Kostengründen die begrenzte Lebenszeit des Sensors hinzunehmen, um ihn danach z.B. durch neue Sensoren auszutauschen.

Ein weiterer wichtiger Aspekt im Bereich Energie ist die Energieeffizienz. Dabei sollen alle Teile in einem Sensorknoten möglichst effizient nutzen, um die Energie sinnvoll und langsam aufzubrauchen. Nicht nur der Sensor selbst spielt dabei eine wichtige Rolle, sondern auch wie das Netzwerk um ihn herum aufgebaut und genutzt wird. Folgende Aspekte spielen bei der Energieeffizienz eine erhebliche Rolle:

- Wahrnehmen der Daten durch die Sensoren
- Verarbeitung der Daten
- Sicherung der Daten
- Übertragung der Daten
- Empfang von Daten

Damit der Energieverbrauch weiter eingeschränkt werden kann, sollten Sensoren nur aktiv sein, wenn sie wirklich benötigt werden. Ansonsten sollten sie sich in einen Sleep- bzw. Energiesparmodus begeben, um Energie zu sparen. Des Weiteren wäre zur ausschließlichen Wahrnehmung der Umgebung ein "Controller- bzw. Sensormodus und zum Senden und Empfangen von Daten ein "Radio bzw.

Übertragungsmodus von Vorteil.

Eine weitere Schwierigkeit, die sich stellt, ist der Einsatz bzw. die Verteilung der Sensoren in der Umwelt und ihre Selbstverwaltung. Die Knoten könnten entweder zufällig in der Umgebung platziert oder systematisch angeordnet werden. Hier entscheidet der jeweilige Anwendungszweck, wobei das systematische Platzieren der Sensoren meistens sinnvoller und effizienter ist. Des Weiteren sollte unterschieden werden, ob aktive oder passive Sensoren eingesetzt werden sollen. Auch hier muss je nach Anwendungsfall unterschieden werden. Passive Sensoren eignen sich besser, wenn Daten nur erfasst und übermittelt werden sollen. Aktive Sensoren sollten eingesetzt werden, wenn auf die Erfassung der Daten eine eventuelle Aktion bzw. Reaktion mit der Umwelt erforderlich ist.

Sensoren sollten bestimmte Informationen über sich selbst und ihre Nachbarn wissen bzw. ermitteln können. Dazu gehören unter anderem ihre eigene Position, die Ortung der Nachbarknoten und ihre Identifikation, ihre eigene Knotenkonfiguration und ihre kürzeste Route zu einer Basisstation. Denn sobald ein Sensornetz einmal in Betrieb genommen wurde, muss es in der Lage sein, sich autonom betreiben und verwalten zu können. Dazu zählen die Anpassung an veränderte Umweltbedingung und das Kompensieren von Fehlern. Beim Ausfall eines Sensors soll das Sensornetz weiterhin aktiv und funktionsfähig bleiben.

Auch in Hinsicht auf die Sicherheit gibt es einige Aspekte zu betrachten. Manche Sensornetzwerke übertragen empfindliche und kritische Informationen, was sie zu einem beliebten Angriffsziel macht. Sie können sowohl von innen, von außen als auch direkt an den Knoten angegriffen werden. Es stellt sich als schwierig heraus, solche Netzwerke vor Angriffen zu schützen, da sie entfernt und selbstständig arbeiten, drahtlos kommunizieren und meistens keine speziellen Sicherheitsfeatures besitzen. Dies ist aus Energie-, Kostengründen und Gründen der Form und Größe der Sensoren meist nicht realisierbar. Übliche Sicherheitstechniken sind meist nicht durchführbar, da den Knoten üblicherweise die Rechen-, Kommunikations- und Speicherressourcen fehlen. Man braucht neu entwickelte Sicherheitsmechanismen für Sensornetze, die spezielle Lösungen für die Erkennung von Eindringlingen, Verschlüsselung, Schlüsselverwaltung und Verteilung und Registrierung von neuen Knoten besitzen, sodass die Ressourcen der Sensoren ausreichen und das Sicherheitskonzept realisierbar ist [WB12].

3.2 Adhoc-Netzwerke

Adhoc-Netze sind in sich geschlossene Netzwerke, organisieren sich selbst und haben keine bestimmte Hierarchie. Sie bauen sich nur für die Dauer einer Datenübertragung auf, besitzen keine festgelegte Kommunikationsstruktur und verwalten und organiseren sich selbst.

Adhoc-Netze sind leistungsfähige und zählen als so genanntes 'Self Organized Network' (SON), welche gute Lastverteilung betreiben und ohne zentrales Management auskommen. Die Endgeräte übernehmen in diesem Fall das Routing und speichern die Routingtabellen selbst ab. Geräte, die sich dem Netzwerk anschließen, werden dynamisch in das Netz eingefügt. Bei Netzwerken nach IEEE 802.11 (WLANs) und IEEE 802.15 (WPANs - hier im Speziellen IEEE 802.15.1 - Bluetooth) werden alle Geräte selbstständig erkannt und werden dem Netz hinzugefügt. Sie sind fortan Bestandteil des Gesamtnetzes.

Bei Adhoc-Netzen mit vielen Geräten (das kann z.B. ein Sensornetz sein) wird zumeist eine Multihop-Verbindung bevorzugt. Das bedeutet, dass die Daten von einem Netzknoten, z.B. einem Sensor oder Rechner, zu dem nächsten Netzknoten weitergeleitet werden, bis es sein Ziel erreicht hat. Fällt ein Knoten aus, wird wenn möglich ein anderer Weg für die Übertragung genutzt, um Ausfälle zu vermeiden.

Ad-hoc-Netzwerke bestehen virtuell für einen begrenzten Zeitrahmen. Ad-hoc bedeutet etwa "für den Augenblick gemacht". Sie werden in WLANs, WPANs, in Sensornetzen (WPANs mit geringer Datenrate - siehe IEEE 802.15.4) und in Funknetzen von Rettungsdiensten, Polizei und Militär benutzt [LLK12].

3.2.1 MANETs

MANET bezeichnet die Abkürzung für den Begriff 'Mobile Ad-hoc Network'. Jochen Schiller beschreibt in seiner Literatur 'Mobile Communications' den Begriff des MANETs wie folgt:

> „Ad-hoc-Netze kommen ohne jegliche Infrastruktur aus, insbesondere ohne eine ausgezeichnete Basisstation, welche den Medienzugriff zentral steuert. Diese Netzvariante erlaubt die spontane, nicht vorab geplante Kommunikation zwischen mobilen Endgeräten, wobei einige oder alle Endgeräte auch Daten von anderen Endgeräten weiterleiten können.“

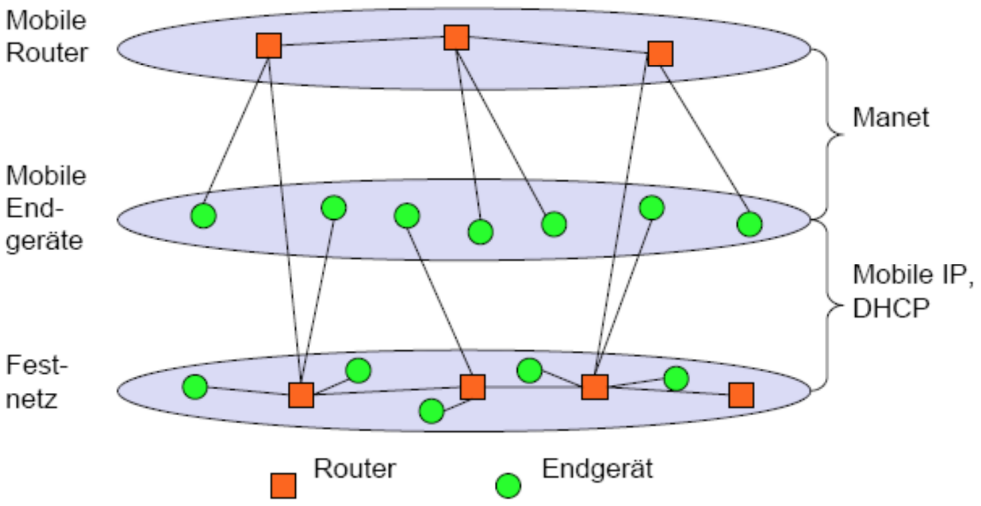

Abbildung 3.6: Einordnung von MANETs [TS11]

MANETs zeichnen sich vor allem dadurch aus, dass sie keine feste Infrastruktur besitzen. Es herrscht innerhalb des Netzes eine dynamische Topologie, was bedeutet, dass zu jedem Zeitpunkt bestimmte Knoten wegfallen und neue dazukommen können. Dies führt dazu, dass bislang bekannte und zulässige Routen wegfallen, dafür aber auch neue Routen entstehen können. Unter den Geräten besteht eine spontane Vernetzung. Das bedeutet, dass jedes Gerät sowohl Endpunkt einer Übertragung sein kann, jedoch auch in der Lage sein muss, Daten weiterleiten zu können. Durch dieses Weiterleiten von Daten entsteht eine so genannte 'Multihop-Umgebung', da die Daten nicht vom Sender direkt zum Empfänger gelangen, sondern den Empfänger über mehrere Zwischenstationen erreichen.

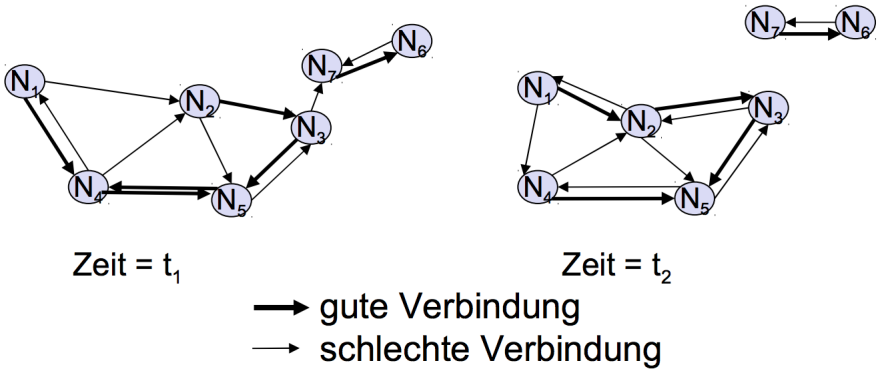

Abbildung 3.7: Mögliche Verbindungen eines Beispiel MANETs [TS11]

Mobile Ad-hoc Netzwerke besitzen eine stark begrenzte Bandbreite. Da sie meistens auf stromsparende Übertragungsprotokolle setzen, können dementsprechend auch nur geringe Mengen an Daten übertragen werden. Des Weiteren bestehen zumeist durch die unterschiedlichen Übertragungsleistungen der Geräte asymmetrische Verbindungen, welche sich für die Übertragung von Daten innerhalb des Netzes besser eignen.

Ein Problem bei MANETs stellt die beschränkte Möglichkeit der Energieversorgung dar, da die Endgeräte meistens abhängig von Batterien oder Akkus sind. Durch die hohe Menge an Geräten in einem Netz steigt dazu die Summe des potenziellen Energiebedarfes pro Gerät weiter an. Auch die Sicherung des Netzes vor physischen Angriffen ist stark beschränkt. So sind z.B. Denial-of-Service-Attacken (DoS), Überwachungsangriffe oder auch das Verfälschen von Nachrichten oft leicht zu realisieren [TS11].

3.2.2 Routingprotokolle

Klassische Routingprotokolle versagen bei dem Versuch, sie innerhalb von mobilen Ad-hoc Netzwerken einzusetzen. Gründe dafür sind folgende:

MANETs zeichnen sich durch eine hohe Dynamik aus, das Netz verändert sich spontan, schnell und ständig. Klassische Protokolle besitzen eine zu langsame Konvergenz, um dem Anspruch der sich ständig verändernden Ad-hoc Netze gerecht zu werden und sind somit nicht geeignet.

Ad-hoc Netze besitzen wie bereits erwähnt meist eine geringe Bandbreite. Hinzu kommt, dass den angeschlossenen Knoten und Geräten nur eine gewisse Rechenleistung zur Verfügung steht. Sie sind somit mit klassischen Routingprotokollen überfordert, da diese meist viel Rechenleistung und Übertragungsleistung erfordern, welche mit den mobilen Geräten nicht realisierbar sind. Der Overhead der klassischen Protokolle ist also für diese Art von Netzwerken zu groß.

Es gibt in mobilen Ad-hoc Netzen gewisse Metriken, die beim Betrieb berücksichtigt werden müssen. Die klassischen Protokolle interessieren sich nicht für diese Metriken und lassen sie außen vor. Bei der Auswahl von Routen müssen unter anderem folgende Aspekte betrachtet werden:

- Batterielaufzeit der Geräte

- Zeit der Verbindung zwischen zwei Geräten
- Energiebedarf
- Zuverlässigkeit der Verbindung

Zusammengefasst lässt sich sagen, dass Routingprotokolle für MANETs vor allem eine große Skalierbarkeit im Hinsicht auf eine große Anzahl von Geräten, Flexibilität und Effizienz im Hinblick auf Komplexität, Energieverbrauch und Speicherverbrauch erfordern. Diese werden mittlerweile intensiv erforscht, da Adhoc Netze im Zusammenhang mit dem 'Internet of Things' (IoT) oder auch dem 'Internet Of Everything' (IoE) zunehmend an Bedeutung gewinnen [TS11].

Im Folgenden sollen die Protokolle OLSR, AODV und CGSR kurz aufgezeigt und erklärt werden.

OLSR

Die Abkürzung OLSR steht für 'Optimized Link State Routing' und bezeichnet ein flaches, proaktives Linkstate-Protokoll. Proaktiv bedeutet in diesem Fall, dass in gewissen Zeitabständen automatisch Kontrollnachrichten ausgetauscht werden. Bei einem Linkstate-Protokoll sendet z.B. jeder Router den anderen Knoten im Netz den Zustand der Verbindung zu seinen Nachbarn.
Das OLSR-Routingprotokoll ist als RFC 3626 spezifiziert und erweitert normale Link-State-Protokolle, in dem die Komplexität jener Protokolle vereinfacht wird. Bei OLSR hat jeder Knoten einen Überblick über alle andere Knoten im Netzwerk und den Routen dort hin, somit kann jeder Knoten seine Routen mit dem 'Shortest Path Algorithm' eigenständig errechnen. Allerdings haben nicht alle Knoten die gleichen Bedeutungen und Aufgaben.
Um Nachbarknoten zu finden, werden sie mit einer 'Hello-Message' gesucht. Die Kontrollpakete werden in entsprechenden Zeitabständen erstellt und enthalten die Knotenadresse, eine Sequenznummer und die Nachbarknoten mit Distanzinformationen. Folglich enthält jeder Knoten die Distanzinformationen von allen anderen Knoten, so kann die Topologie erstellt und die Routen mit o.g. 'Shortest Path Algorithm' berechnet werden.

Das OLSR unterscheidet sich von anderen Link-State-Algorithmen, da es ein optimiertest Link-State-Routing verwendet. Es werden so genannte 'Multi-Point-Relays' ausgewählt und verteilt, was für ein effizienteres Routing sorgt. Die Relays

sind im Stande, die Kontrollnachrichten an die angeschlossenen Knoten weiterzuleiten.

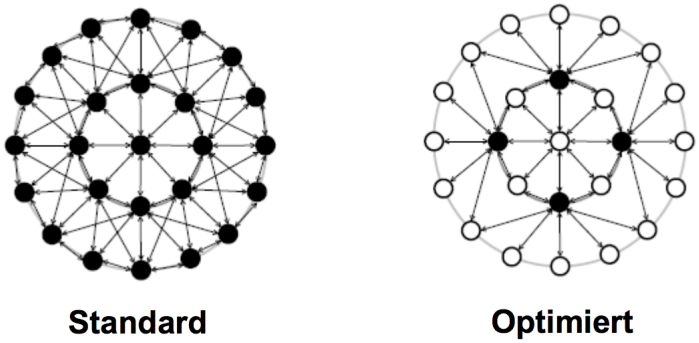

Abbildung 3.8: optimiertes LSR durch Multi-Point-Relays im OLSR [TS11]

AODV

Das Ad-hoc On-Demand Distance Vector Routingprotokoll ist ein flaches, reaktives Distanzvektorprotokoll. Reaktive Protokolle tauschen im Gegensatz zu proaktiven Protokollen nur Kontrollnachrichten aus, wenn neue Routen benutzt werden sollen. Sie benötigen weniger Bandbreite und sind dynamischer, allerdings besteht nur eine Teilkenntnis des Netzwerks, somit müssen die Routen anders berechnet werden.
AODV ist in RFC 3561 spezifiziert und ist für IPv4-Netze vorgesehen. Es erlaubt eine theoretisch unbegrenzte Anzahl von Geräten im Netzwerk. Bei AODV kennt jeder Knoten nur den 'Next Hop', also den nächsten Knoten und die Länge der Gesamtroute. Das Protokoll definiert insgesamt zwei Routingalgorithmen ‚Route Discovery' und ‚Route Maintenance'.

Bei der 'Route Discovery' wird die Route aufgebaut, wozu zwei Nachrichtentypen benötigt werden. Es wird zunächst eine 'Route-Request-Nachricht' (RREQ) per Broadcast zum Ziel gesendet. Das Ziel antwortet widerum mit einer 'Route-Reply-Nachricht' (RREP) per Unicast zurück an den Absender. Sollte eine bidirektionale Verbindung zwischen zwei Geräten herrschen, wird per Route Discovery auch eine bidirektionale Route aufgebaut.

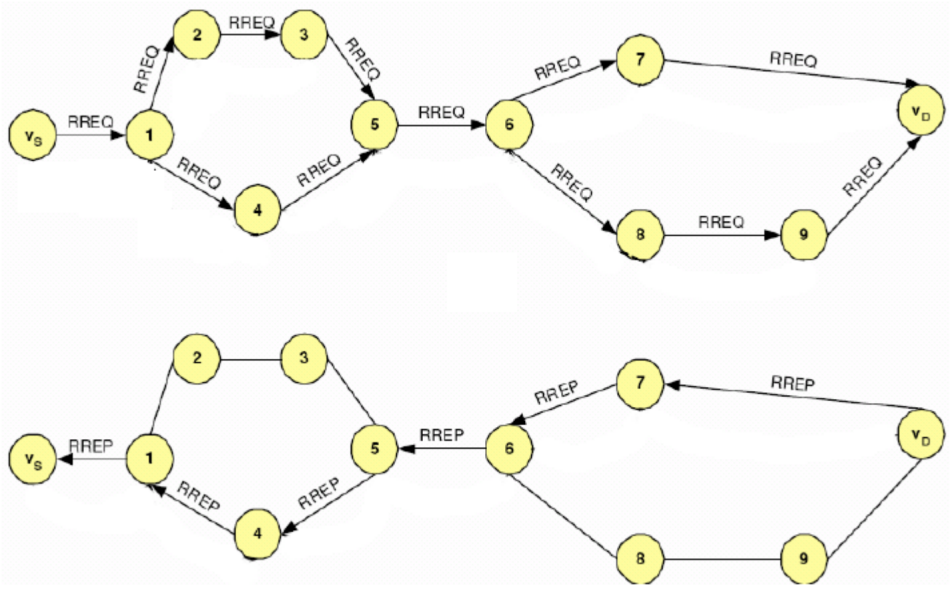

Abbildung 3.9: 'Route Discovery' im AODV-Protokoll [TS11]

Bei 'Route Maintenance' handelt es sich um die Verwaltung der Routen. Es wird eine 'Route-Error-Nachricht' (RRER) versendet, sobald eine defekte Route diagnostiziert wird. Diese Nachricht wird an alle Nachbarn versendet, so dass jeder Knoten über das Wegfallen der Route informiert wird.

CGSR

Beim Clusterhead-Gateway Switch Routing (CGSR) handelt es sich um ein hierarchisches Protokoll, was bedeutet, dass für verschiedene Knoten unterschiedliche Rollen vorgesehen sind.
Das Netzwerk wird bei CGSR in Cluster aufgeteilt, die sich allerdings teilweise überdecken müssen. Für jedes dieser Cluster wird ein 'Clusterhead' ernannt. Ein Knoten, welcher sich in zwei Clustern gleichzeitig befindet, wird als Gateway bezeichnet.

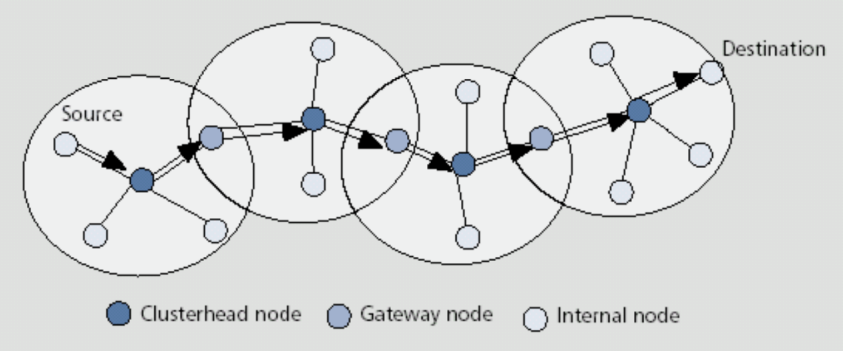

Abbildung 3.10: Clustering im CGSR-Protokoll [TS11]

Das Clusterhead-Gateway Switch Routing basiert auf einem Distanzvektor-Protokoll. Jeder Knoten im Netzwerk speichert sowohl eine Distanzvektor-Routingtabelle, als auch eine 'Cluster Member'-Tabelle. Die Distanzvektor-Tabelle enthält neben den normalen Routingeinträgen einen Routingeintrag zum Clusterhead jedes Clusters. In der 'Cluster Member'-Tabelle wird für jeden Knoten der Clusterhead gespeichert, damit klar ist, welcher Knoten zu welchem Cluster gehört.

Der große Vorteil von CGSR besteht in der signifikanten Reduzierung der Größe der Routingtabellen im Vergleich zu anderen Distanzvektorprotokollen, somit ist das CGSR für Ad-hoc Netzwerke dank seiner geringeren Komplexität gut geeignet.

3.3 IEEE 802.15.4

Und hier kommt ebenfalls noch Inhalt rein.

3.4 SunSPOT

Die Firma Oracle besitzt im Rahmen seiner Java-Technologie eine Vormachtstellung im Bereich der Smartphones. Auf der Welt sind schätzungsweise über eine Milliarde Smartphones mit der Java-Technologie lizenziert. [Hor08]
Ziel von Oracle ist es, auch in den zukunftsnahen Technologien mit ihrer Programmiersprache Java auszustatten und diese Produkte zu etablieren.

Ein erster Schritt in diese Richtung is das von Oracle entwickelte "SunSPOT"-Sensornetzwerk. SunSPOT bedeutet "Sun Small Programmable Object Technology" und ist eine Plattform für Java-basierte drahtlose Sensornetzwerke. Sie

bestätigt den Trend, dass in immer kleiner werdenden Geräten zunehmend leistungsfähigere Technologien eingesetzt werden. Dabei ist wichtig, dass jene Geräte, am Besten drahtlos, miteinander kommunizieren können und jederzeit von überall auf der Welt steuerbar bleiben. Das SunSPOT Starter Paket besteht aus einer Basisstation und 2 Sensoren.

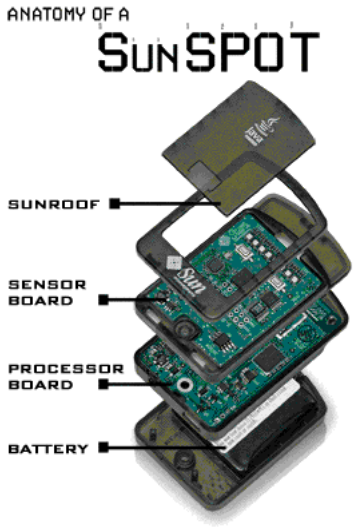

Abbildung 3.11: Anatomie eines Standard SunSPOT-Sensors [Uni]

Die Hardware der SunSPOT-Sensoren ist modular aufgebaut. Das bedeutet, dass man die verfügbaren Boards frei nach Belieben aufeinander stecken und somit verbinden kann. Dabei können maximal bis zu 3 Boards + Stromversorgung miteinander verknüpft werden. [Hor08]

3.4.1 Gerätearten

Man unterscheidet bei den SunSPOT-Geräten grundsätzlich zwischen dem Basisstation SPOT und den Free Range SPOTs. Der sogenannte 'Basestation SPOT' hat ein eSPOT Prozessor-Board ohne eigene Batterie. Er wird mit Hilfe des USB-Ports mit Strom versorgt. Die Station dient grundsätzlich als Schnittstelle zwischen Free Range SPOTs und dem PC (der Workstation) auf Basis von IEEE 802.15.4. Die Free Range SPOTs enthalten ebenfalls ein Prozessor-Board (auch Mainboard genannt), hinzu kommt ein wiederaufladbarer Li-Ion Akkumulator. Des Weiteren enthalten die SPOTs in der Standardauslieferung das eDemo Sensor Board. Alle Boards, welche man dank modularem Aufbau hinzustecken kann,

sind separat erhältlich und werden im Folgenden näher erläutert.

3.4.2 Verfügbare Boards

Das sogenannte eSPOT Prozessor-Board besitzt in der aktuellsten Version eine 400 MHz 32-bit ARM CPU von Atmel, zusammen mit einem Flashspeicher von 8 Megabytes und einem Megabyte SRAM Hauptspeicher. Weiterhin ist es ausgestattet mit einem Radio Transceiver basierend auf IEEE 802.15.4 und einer USB 2.0 - Full Speed Schnittstelle. Der im SunSPOT integrierte Akkumulator hat eine Leistungsfähigkeit von 770mAh. Der maximale Energieverbrauch liegt zwischen 40-100 mA, abhängig von der Nutzung der integrierten LEDs, des Transceivers und anderer angeschlossener Geräte. [Hor08] [Ora10b]

Der Free Range SPOT wird dazu standardmäßig mit dem eDemo Sensor Board ausgeliefert. Es besitzt in der aktuellen Version einen 2G/4G/8G 3-Achsen-Beschleunigungssensor, einen Lichtsensor, 8 RGB 24bit LEDs, einen Infrarot-Sender & Empfänger, ein kleiner Lautsprecher, 2 Knopfschalter, 4 analoge Eingänge, 4 I/O Pins, diverse weitere I^2C- und USART-Interfaces, einen EEPROM und 4 100mA Ausgangspins, mit denen es möglich ist, den SunSPOT-Sensor z.b. an weitere Lautsprecher oder andere Geräte anzuschließen. [Hor08] [Ora10a]

Weitere Boards, welche man nach Bedarf dazustecken kann, sind das eProto-Board, ein Board welches direkte Zugriffe auf das Prozessorboard ermöglicht und einen SD-Kartenslot besitzt, damit man die Daten dauerhaft speichern kann, das eSerial Board zum Verbinden via RS232 und das eFlash SD-Kartenleser Board. [Hor08]

Kapitel 4

Praktische Arbeiten

Das ist das normale Todo, inline.
Mit dem \newline Befehl erzwingt man eine neue Zeile.

In diesem Kapitel werden wir die praktischen Arbeiten, die wir mit SunSPOTS durchführen erläutern. Dies wird in zwei Unterkapiteln durchgeführt. Man sollte damit das hier besser aussieht auch viel Text haben. Wenn nur eine Zeile unterstrichen ist sieht das nicht ganz so gut aus.

Also das hier ist ein Improvement und ist einfach in einer anderen Farbe.

Das nach dem \unsure Befehl folgende wird bis zum Zeilenende rot unterstrichen.

4.1 Erste Schritte

Um den Umgang mit den SunSPOTs zu erlernen und ihre Software und Hardware kennen zu lernen, bietet Oracle auf der Website eine Anleitung inklusive Übungen an, welche sukzessive abgearbeitet werden kann. Die Anleitung umfasst die Installation der Software sowie das Ausführen von Testprogrammen, welche die Fähigkeiten des SunSPOTs demonstrieren.

4.1.1 Installation

Zur Installation der Verwaltungsapplikation 'SunSPOT Manager' und dem SunSPOT SDK, mit welchem man die Programm für die SunSPOTs schreibt, startet man das 'SunSPOT Manager Java Webstart' Programm. Das Webstart-Programm lädt darauf hin den Manager und das SDK herunter und installiert es auf der Workstation. Zur einwandfreien Benutzung des Managers müssen folgende Tools vorhanden sein:

- ein aktuelles Java Development Kit
- Apache ANT

- optional: NetBeans IDE

Das Installationsprogramm weist auf ein Fehlen oben genannter Tools hin und installiert diese bei Bedarf nach. Sollte dies nicht automatisch durch das Programm geschehen, können diese Programme auch manuell nachinstalliert werden. Während der Installation wird nach dem SunSPOT SDK gefragt, welches installiert werden soll. Je nach Version der SPOTs ist hier die entsprechende Version auszuwählen.

Normalerweise wird beim Anschließen per USB der entsprechende Treiber für die SPOTs automatisch installiert. Bei aktuellen 64-bit Windows-Versionen (7/8/8.1) kommt es allerdings zu Schwierigkeiten. Eine spezielle Treiber-Datei wurde am 7. Mai 2010 durch Bob Alkire im Oracle-Blog veröffentlicht und kann dort heruntergeladen werden. Mit ihr ist es möglich, die SPOTs auch mit aktuellen 64-bit Windows-Plattformen zu verwenden [Alk10].

Sobald der Manager und alle zugehörigen Treiber erfolgreich installiert wurden, können Programme mit Hilfe von NetBeans oder einer anderen IDE geschrieben und auf den SPOT übertragen werden.

4.1.2 Übung 1 - Flashlight

In der ersten Übung soll das erste Beispielprogramm 'Flashlight' auf den SPOT übertragen, dort ausgeführt und die zugehörigen Programmausgaben (Konsolen- & Debugausgaben) betrachtet werden.

Nach dem erfolgreichen Übertragen des Programms blinken die LEDs des SPOTs in kurzem Abstand, vorausgesetzt, es wird ein bestimmter Helligkeitswert, ausgelesen vom Lichtsensor, nicht überschritten. Mit Hilfe des rechten Knopfschalters auf dem Sensorboard kann außerdem die Farbe der LEDs geändert werden. Die Konsolenausgabe des Programms besteht aus dem momentanen Helligkeitswert, der vom Lichtsensor erfasst wird.

Abbildung 4.1: Leuchtende LEDs des 'Flashlight'-Programms

Für die geplante Raumüberwachung ist diese Übung insofern hilfreich, als das hier der Umgang mit den LEDs und deren Verhalten auf äußere Einflüsse gezeigt wurde. In der späteren Implementierungsphase der Raumüberwachung sollen LEDs anzeigen, ob eine Bewegung und somit ein illegales Eindringen in den Raum stattfand.

4.2 Implementierung einer Raumüberwachung

4.2.1 Idee

Hier kommt die Idee die wir hatten hin.

4.2.2 Umsetzung

4.3 Erweiterungsmöglichkeiten

Kapitel 5

Vergleichbare Projekte

5.1 Smart Greenhouse

5.2 Bot-So

Kapitel 6

Zusammenfassung

Hier kommt die Zusammenfassung des Projektes hin. Diese besteht aus Beschreibung, Vorgehensweise und Ergebnis. Insgesamt umfasst sie etwa eine Seite.

Anhang

Beispielergebnis Bedienungsanleitungen

Manchmal benutzt man Worte wie Hamburgefonts, Rafgenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten - man nennt diese Sätze »Pangrams«. Sehr bekannt ist dieser: The quick brown fox jumps over the lazy old dog. Oft werden in Typoblindtexte auch fremdsprachige Satzteile eingebaut (AVAIL® and Wefox™ are testing aussi la Kerning), um die Wirkung in anderen Sprachen zu testen. In Lateinisch sieht zum Beispiel fast jede Schrift gut aus. Quod erat demonstrandum. Seit 1975 fehlen in den meisten Testtexten die Zahlen, weswegen nach TypoGb. 204 § ab dem Jahr 2034 Zahlen in 86 der Texte zur Pflicht werden.

Literaturverzeichnis

[Alk10] Alkire, Bob: *SPOTs on 64-bit Windows 7*. https://blogs.oracle.com/ralkire/entry/spots_on_64_bit_windows, May 2010

[Arda] Arduino CC: *Arduino-Logo*. http://http://www.arduino.cc/,

[Ardb] Arduino CC.: *Arduino Uno*. http://www.arduino.cc/en/Main/ArduinoBoardUno,

[Ardc] Arduino CC: *Arduino Uno R3 Front*. http://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg,

[Car98] Carnegie Mellon University Computer Science Department: *The Only Coke Machine on the Internet*. https://www.cs.cmu.edu/~coke/history_long.txt, Juni 1998

[Cua13] Cuartielles, David: *Arduino FAQ – With David Cuartielles*. http://medea.mah.se/2013/04/arduino-faq/, April 2013

[Ele15] Elektor: *Klangsensitive LED-Kunst mit Arduino*. http://www.elektor.de/news/klangsensitive-led-kunst-arduino/, März 2015

[Hor08] Horan, Bernard: *Sun SPOTs*. https://www.dropbox.com/sh/l2kch3izg7lwdpl/AADjbAi2ukAjtaZY8zVzMpdHa/IoT/sunspot.pdf, 2008

[Kah12] Kahle, Christian: *Intel: Mooresches Gesetz gilt noch mind. 10 Jahre*. http://winfuture.de/news,72001.html, September 2012

[Kos09] Kosmerchock, Steven: *Wireless Sensor Network Topologies*. http://www.k5systems.com/TP0001_v1.pdf, 2009

[LLK12] Lipinski, Klaus ; Lackner, Hans ; Kafka, Gerhard: *Ad-hoc-Netz*. http://www.itwissen.info/definition/lexikon/Ad-hoc-Netzwerk-ad-hoc-network.html, March 2012

[Mul15] MULTICHERRY: *Top half of Raspberry Pi 2 Model B v1.1 viewed directly from above.* http://commons.wikimedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_top_new_(bg_cut_out).jpg, Februar 2015

[Ora10a] ORACLE CORP.: *Sun^{TM} SPOT eDEMO Technical Datasheet Rev 8.0.* http://www.sunspotworld.com/docs/Yellow/edemo8ds.pdf, Oktober 2010

[Ora10b] ORACLE CORP.: *Sun^{TM} SPOT Main Board Technical Datasheet Rev 8.0.* http://www.sunspotworld.com/docs/Yellow/eSPOT8ds.pdf, Oktober 2010

[Ras] RASPBERRY PI FOUNDATION: *Logo der Raspberry Pi Foundation.* https://www.raspberrypi.org/wp-content/uploads/2011/10/Raspi-PGB001.png,

[TS11] TIMM, Constantin ; SPINCZYK, Olaf: *Software ubiquitärer Systeme - Ad-hoc-Netzwerke.* https://ess.cs.tu-dortmund.de/Teaching/SS2011/SuS/Downloads/04.1-Adhoc-Netzwerke.pdf, 2011

[Tuo02] TUOMI, Ilkka: *The lives and deaths of moores law.* http://firstmonday.org/ojs/index.php/fm/article/view/1000/921, November 2002

[Uni] UNIVERSITY OF SOUTHERN CALIFORNIA: *Standardmäßiger Aufbau eines SunSPOT-Sensors.* http://anrg.usc.edu/ee579_2012/Group07/img/spotanatomy.jpg,

[WB12] WOLF, Lars ; BÜSCHING, Felix: *Wireless Sensor Networks - Introduction and Applications.* https://www.dropbox.com/sh/l2kch3izg7lwdpl/AABu1b-vt8FCuUD2iU905F0ca/IoT/RecentTopics_Chapter02_WSN-Introduction-and-Applications.pdf, 2012

[Wei91] WEISER, Mark: *The Computer for the 21st Century.* http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html, September 1991

[Wgs] WGSIMON: *Mooresches Gesetz.* http://commons.wikimedia.org/wiki/User:Wgsimon#mediaviewer/File:Transistor_Count_and_Moore%27s_Law_-_2011.svg,

[Wil15] Willems, Eddy: *IoT: The Internet of Things... ehm... / Ein Balance-Akt zwischen Benutzbarkeit und Sicherheit?!* https://tcadistribution.wordpress.com/tag/sicherheitsexperten/, März 2015

[ZEI15] ZEIT ONLINE, dpa, AFP, rav: *Hacker konnten BMW-Türen jahrelang per Handy öffnen.* http://www.zeit.de/mobilitaet/2015-01/bmw-hacker-sicherheit, Januar 2015

Notes