

## **Diplomarbeit**

# **Das Infranetviewer-System**

Lauber Michael

Giger Pascal

Zürich, 26.09.2009

# Danksagung

Wir möchten an dieser Stelle unseren internen Betreuern Alois Kaufmann und Roger Tinembart für ihre Bereitschaft danken, uns zu jedem Zeitpunkt bei Verständigungsproblemen zu helfen. Diese Unterstützung trug massgeblich zum Erfolg unseres Projektes bei. Das Interesse von Alois Kaufmann und Roger Tinembart an unserem Projekt, motivierte uns sehr.

Ein grosses Dankeschön richten wir auch an die Mitarbeiter der Firma aizo Jan Zakrzewski, Miguel Rodriguez und Johannes Winkelmann, sowie an die Mitarbeiterin Andrea Heubi. Die Realisierung unseres Projekts wäre ohne die Unterstützung der Firma aizo nicht möglich gewesen.

Wir möchten ausserdem unseren Familien, die während der Erarbeitung des Projektes unsere Launen ertragen mussten und uns dennoch stets unterstützten, unseren Dank aussprechen.

Schliesslich bedanken wir uns bei Isabel Bachmann für das Gegenlesen dieser Arbeit.

Vielen Dank Euch allen.

# Inhaltsverzeichnis

1	Zusammenfassung.....	3
2	Einleitung .....	4
2.1	Beschreibung der Voraussetzung .....	4
2.2	Ziel der Arbeit.....	4
2.3	Ist-Situation.....	5
2.4	Grundidee für das Infranetviewer-System .....	5
2.4.1	Der Sender.....	5
2.4.2	Der Empfänger.....	5
2.4.3	Der Server.....	6
2.4.4	Die Datenbank .....	6
2.4.5	Alternative zur RFID.....	6
3	Infranetviewer-System.....	8
3.1	Auswahl der optimalen Technologie.....	8
3.1.1	Die Auswahl der optimalen Netzwerktechnik.....	8
3.1.1.1	Die Variante W-LAN .....	8
3.1.1.2	Die Variante LAN.....	10
3.1.1.3	Die Variante P-LAN .....	10
3.1.2	Die Auswahl der Hardwarebausteine .....	13
3.1.2.1	Die Devolo Bausteine .....	13
3.1.2.2	Die aizo Technologie .....	14
3.1.2.3	Vergleich mit anderen Technologien.....	15
3.1.2.4	Testreihen mit dem aizo Demokoffer .....	16

3.1.3	Technologie Beschaffung.....	26
3.1.3.1	Vorstellung bei der Firma aizo .....	26
3.1.3.2	Anmeldung bei der digitalSTROM.org .....	27
3.1.3.3	Vorstellung bei der Geschäftsleitung der Firma aizo .....	31
3.1.4	Linux-Server.....	33
3.1.4.1	Die Systemübersicht des Servers (dSS).....	33
3.1.4.2	Probleme bei der Serverbildung .....	35
3.1.4.3	Fehler Bereinigt .....	39
3.1.5	Verbindung zwischen Server - Komponenten.....	40
3.1.5.1	Erster Verbindungsversuch .....	40
3.1.5.2	Zweiter Verbindungsversuch .....	41
3.1.5.3	Dritter Verbindungsversuch .....	42
3.1.5.4	Zu wenige Ressourcen.....	42
3.1.5.5	Willkommen zum digitalSTROM-Server.....	44
3.1.6	Auswertung der Daten .....	45
3.1.7	Datenverknüpfung.....	46
3.1.7.1	Änderungen an der Plattform.....	47
3.1.7.2	Das Suchportal .....	49
3.1.7.3	Aufbau der Demoanlage.....	50
4	Schlussfolgerung.....	52
5	Verzeichnisse.....	54
5.1	Abbildungsverzeichnis.....	54
5.2	Literaturverzeichnis .....	55

# 1 Zusammenfassung

Im Rahmen dieser Diplomarbeit sollte ein Geräteortungssystem geschaffen werden, das die Auffindung mobiler Geräte in einem Gebäude weitgehend garantiert. Durch die momentane Ist-Situation, unsere eigene Erfahrung und in diversen Gesprächen mit Technikern hat sich herausgestellt, dass ein Bedürfnis besteht, das Orten von Geräten einfacher zu gestalten.

Damit unsere Idee von diesem Ortungssystem in die Tat umgesetzt werden konnte, machten wir uns Gedanken, welche Grundeigenschaften ein solches System haben muss. Wir kamen zum Schluss, dass unser System mindestens aus einem Sender, der an dem zu suchenden mobilen Gerät angebaut ist; einem Empfänger, der das mobile Gerät orten kann; einer Datenbank, die den Standort des Geräts zuweisen kann; und einem Server, der für die Kommunikation zuständig ist, bestehen soll. Dadurch ergaben sich drei Hauptaufgaben, die erfüllt werden mussten, um einen Prototypen eines funktionierenden Ortungssystems am Ende dieser Arbeit präsentieren zu können.

Die erste Aufgabe beinhaltete die Auswahl der geeigneten Technologie. Wir wählten ein für unser Projekt optimales Netzwerk aus das sogenannte Power-LAN. Nach mehreren Komponententests fanden wir dann auch die ideale Hardware, deren Beschaffung sich als zeitraubend erwies, da es sich um einen Prototypen von der Firma aizo handelt. Die zweite Aufgabe bezog sich auf die Kommunikation der Hardwarekomponenten. Es handelte sich erneut um ein zeitaufwendiges und anspruchvolles Unterfangen, besonders da wir auf dem Gebiet der Server Bildung auf Linux-Basis keine Experten sind. Die Server Bildung gelang dank professioneller Unterstützung. Die dritte Aufgabe bezog sich auf die Realisierung unseres Systems mittels Erarbeitung einer Datenbank und Erstellung einer Demoanlage.

## **2 Einleitung**

### **2.1 Beschreibung der Voraussetzung**

Das Auffinden mobiler Geräte in grösseren Krankenhäusern ist ein zeit- wie auch kostenaufwendiges Unterfangen, da das Volumen der mobilen Geräte immer unübersichtlicher wird. Dieses Problem des Auffindens besteht trotz vorhandener Logiksystemen wie „Pick n Go“. Es ist ausserdem nicht nur ein Problem der Techniker in einem Krankenhaus, sondern kann durchaus auch ein Problem der Hersteller sein, die ihre vermieteten mobilen Medizintechnischen Geräte laut Bundesamt für Gesundheit (BAG) mindestens einmal pro Jahr testen müssen und diese im ganzen Krankenhaus verteilt sind. Ein Chaos ist die Folge.

### **2.2 Ziel der Arbeit**

Wir wollen in unserer Diplomarbeit ein Ortungssystem erarbeiten, das eine mögliche Lösung zur problemlosen Auffindung mobiler Geräte in einem Gebäude bieten kann. Dieses Ortungssystem wird von uns Infranetviewer-System (INVS) genannt, im nachfolgenden Text werden wir die Abkürzung (INVS) benutzen. Das Orten von mobilen Geräten soll mit dem INVS vereinfacht werden. Nicht nur das Orten wird einfacher, der aktuelle Standort der Geräte sollte ebenfalls stets ersichtlich sein, da wir das INVS mit einer Datenbank verknüpfen wollen. Das INVS soll eine flexible Lösung mit breitem Feld für unkomplizierte Erweiterung eines IT-Systems in Unabhängigkeit eines Herstellers werden.

## **2.3 Ist-Situation**

In den Krankenhäusern gibt es noch keine solchen Systeme. Sie sind aber weitgehend mit LAN und W-LAN erschlossen und diese werden auch weiter ausgebaut. Durch unsere eigene Erfahrung und in diversen Gesprächen mit Technikern hat sich herausgestellt, dass ein Bedürfnis besteht, das Orten von Geräten einfacher zu gestalten.

## **2.4 Grundidee für das Infranetviewer-System**

In unserer Diplomarbeit möchten wir ein INVS entwickeln. Dieses Geräteortungssystem besteht voraussichtlich aus einem Sender, einem Empfänger und einem Server, der die Daten des Senders an eine Datenbank weitergibt. Von dieser Datenbank kann die Suche nach dem mobilen Medizintechnischen Gerät durch Eingabe gewisser Kriterien gestartet werden. Danach soll die Datenbank anzeigen, in welchem Stock und in welchem Zimmer sich das gesuchte Gerät befindet.

### **2.4.1 Der Sender**

Der Sender sollte eine weltweit einzigartige Kennnummer, oder anders gesagt, eine einzigartige MAC-Adresse an ein Netzwerk senden. Zudem sollte er möglichst einfach an/in jedes Gerät einzubauen sein, ohne dass er die Gerätefunktionen beeinträchtigt, oder sogar nicht ausführen lässt. Wenn er aber an- beziehungsweise eingebaut ist, sollte er nicht exponiert oder einfach trennbar vom mobilen Medizintechnischen Gerät sein.

### **2.4.2 Der Empfänger**

Der Empfänger sollte die einzigartige Kennnummer/Adresse vom Sender beziehen (empfangen) und so als Knotenpunkt zwischen Sender und Server fungieren. Sender und Empfänger könnten auch aus einem Bauteil bestehen.

### **2.4.3 Der Server**

Der Server sollte Kommunikation und Schnittstelle zum Sender und Empfänger sein und mit einer Datenbank verknüpft werden.

### **2.4.4 Die Datenbank**

In der Datenbank sollten Grundinformationen ersichtlich werden:

1. Um was für ein Gerät es sich handelt
2. In welchem Raum das gesuchte mobile Medizintechnische Gerät steht
3. In welchem Stock das Gerät sich befindet

Natürlich soll es möglich sein, diese Datenbank nach Belieben mit individuell wichtigen Informationen über das mobile Medizintechnische Gerät zu ergänzen. In dieser Datenbank sollte dann auch kein Patienten/innen Name oder eine Beziehung zu einem solchen gemacht werden können. Die Privatsphäre des/der Patienten/In wird also geschützt, indem seine/ihre Daten nicht einsehbar sind. So ist schon eine wichtige Voraussetzung für die Datensicherheit erfüllt. Auch sollte die Möglichkeit bestehen, für Mietprodukte eine genaue Abrechnung zu erstellen, natürlich unter der Voraussetzung, dass die mobilen Medizinischen Mietprodukte mit einem Sender ausgestattet sind.

### **2.4.5 Alternative zur RFID**

Nach Bekanntgabe unserer Idee, dachten die meisten Leute sofort an das bestehende System RFID. Aus diesem Grund erklären wir kurz, wodurch sich unsere Idee vom bestehenden RFID unterscheidet. Unser Geräteortungssystem das Infranetviewer-System (INVS) sollte eine Alternative zur Suche von einem beliebigen Produkt, einem mobilen Gerät, Buch oder eines Medikamentes das mit Hilfe von elektromagnetischen Wellen (RFID, englisch Radio Frequency Identification) identifiziert wird, darstellen. Denn bei RFID müssten zusätzlich sogenannte Transponder mit einer Kennung an den mobilen Geräten angebracht sein. Diese Kennung wird dann über hochfrequenzielle, elektromagnetische Lesegeräte erkannt. Da sich die Reichweite der nicht aktiv mit Stromversorgten Transponder auf 1-10 Meter beläuft und man sobald diese Reichweite überschritten wird das Lesegerät nichts mehr erkennen kann, wäre unser INVS je nach Produkt, abhängig von Eigenpreis sogar eine Alternative zur RFID.

Da unser INVS über ein physikalisches Netzwerk funktionieren soll und man dadurch direkt über das Netzwerk sehen kann ob ein Produkt im Netz angeschlossen ist oder nicht hätte man das Problem der Reichweite nicht mehr. Das heisst, man müsste nicht 1-10 Meter von dem Produkt entfernt sein, also Sichtkontakt haben und dadurch ja schon wissen wo sich das Objekt befindet, sondern könnte von einem Punkt, der sich eventuell auch in einem anderen Gebäude befindet das Produkt identifizieren. Darum denken wir das unser INVS wohl eher eine Alternative zur RFID sein wird. Wie das folgende Zitat noch verdeutlicht.

RFID ermöglicht die automatische Identifizierung und Lokalisierung von Gegenständen und Lebewesen und erleichtert damit erheblich die Erfassung und Speicherung von Daten. Ein RFID-System besteht aus einem Transponder, der sich am oder im Gegenstand bzw. Lebewesen befindet und diese kennzeichnet sowie einem Lesegerät zum Auslesen der Transponder-Kennung.



**Abb. 1: Verbuchung von Büchern mit RFID-Chips**

Das Lesegerät enthält eine Software (ein Mikroprogramm), das den eigentlichen Leseprozess steuert und eine RFID-Middleware mit Schnittstellen zu weiteren EDV-Systemen und Datenbanken.



**Abb. 2: Universelles RFID Handlesegerät**

In der Regel erzeugt das Lesegerät ein elektromagnetisches Hochfrequenzfeld geringer Reichweite, vorzugsweise mit Induktionsspulen. Damit werden nicht nur Daten übertragen, sondern auch der Transponder mit Energie versorgt. Nur wenn grössere Reichweiten erzielt werden sollen und die Kosten der Transponder nicht sehr kritisch sind, werden aktive Transponder mit eigener Stromversorgung eingesetzt. Meist wird die Frequenz 13,56MHz benutzt, auf die auch Warnsysteme vor RFID-Einsatz ansprechen.

RFID-Transponder können so klein wie ein Reiskorn sein und implantiert werden, etwa bei Haustieren. Vorteile dieser Technik ergeben sich aus der Kombination von Kleinheit der Transponder, unauffälligen Auslesemöglichkeiten (z.B. neuer Pass) und geringem Preis der Transponder (teilweise im Cent-Bereich). Diese neue Technik verdrängt zunehmend den heute noch weit verbreiteten Barcode.

(Wiki, RFID: [http://de.wikipedia.org/wiki/Radio\\_Frequency\\_Identification](http://de.wikipedia.org/wiki/Radio_Frequency_Identification) [20.08.2009])

## **3 Infranetviewer-System**

### **3.1 Auswahl der optimalen Technologie**

#### **3.1.1 Die Auswahl der optimalen Netzwerktechnik**

Damit wir eine optimale Lösung für das Problem der mobilen Geräteortung präsentieren können, mussten wir die richtigen technischen Voraussetzungen finden. Zuerst widmeten wir uns der Problematik der verschiedenen Netzwerktechniken.

##### **3.1.1.1 Die Variante W-LAN**

Durch die steigende Zahl verschiedener Netzwerke, gewinnt die Minimierung nicht gewollter Strahlung vermehrt an Bedeutung. Hartmut Bärwolff et al. halten Folgendes in ihrem Handbuch fest:

Drahtlostechniken (Wireless LANs) haben in den letzten Jahren eine grosse Verbreitung durch den zunehmenden Einsatz von Laptops, Mobiltelefonen, PDAs und Handhelds erfahren. Das hat viele Gründe; einerseits unser Drang nach Mobilität, andererseits kann man durch den Einsatz dieser Technologien Kosten sparen. Der Bedarf an transparenten Informationen steigt ständig. Das Datenverkehrsvolumen verdoppelt sich alle 6 bis 12 Monate. Wir werden in den nächsten Jahren Datenraten im Bereich von einigen Terra Bit/s haben. Es wird heute eine Durchgängigkeit der Kommunikation vom Entstehungsort der Informationen bis hin zum Management angestrebt (vgl. Langheinrich, 2005). Bei der Umsetzung sind aber zahlreiche Details zu beachten. Die Wireless-Technologie ist mittlerweile so ausgereift, dass ihr Einsatz in Unternehmen, in öffentlichen Verwaltungen wie auch im privaten Bereich problemlos möglich ist. Insbesondere im medizinischen Bereich mit seinem hohen Informations- und Datenvolumen, verbunden mit der Forderung nach Mobilität von medizinischem Personal und Patienten können die Drahtlostechniken Wesentliches bewirken.

(Bärwolff et al 2006)

Zusätzlich stützen wir uns auf eine Studie von der Eidgenössischen Technischen Hochschule (ETH–Zürich), die uns Oliver Lauer (2008) zur Verfügung gestellt hat. Da werden folgende Fragen geklärt:

- Welche Auswirkungen haben die verschiedenen LANs auf Menschen und Geräte?
- Sind diese gefährlich für Patienten und Personal, in dem sie den Heilungsprozess oder sogar die Konzentration bei der Arbeit beeinflussen?
- Treten bei Geräten mit empfindlichen Sensoren und heikler Überwachung von Patienten falsche Messresultate auf oder ist sogar ein gänzliches Aussetzen möglich?

Um diese Elektromagnetischen Felder nicht noch mehr auszubauen und das Personal wie die Patienten noch mehr ungewollter Strahlung auszusetzen, kam für uns diese Variante mit dem Drahtlosen lokalen Netzwerk (W-LAN) nicht in Frage. Zu dieser Entscheidung trug natürlich auch folgende Ist-Situation bei: In der Schweiz werden in der Regel keine neuen Krankenhäuser gebaut, sondern bestehende Krankenhäuser umgebaut. Daher soll eine Variante für das INFVS gewählt werden, die auf eine vorhandene Installation basiert und ergänzt werden kann. So hält sich nicht nur der bauliche Aufwand im Rahmen sondern auch der finanzielle.

Zwar muss etwas nachgerüstet werden, allerdings in Form eines einmaligen Aufwandes. Natürlich streiten sich die Kritiker, da auch bei der Wireless Installation der Aufwand in baulicher und finanzieller Hinsicht je nach Alter eines Baus auf ein Minimum beschränkt werden kann. Trotzdem entschieden wir uns gegen das W-LAN Netzwerk, da in unseren Augen das lokale Netzwerk und das lokale Strom Netzwerk mit weniger Aufwand nachgerüstet werden kann. Wir halten es für möglich, in Zukunft zwei dieser Netzwerktechniken zu kombinieren oder sogar mit GPRS Ortung auszurüsten. Zu diesem Zeitpunkt wäre es allerdings verfrüht, dieses Projekt zu präsentieren.

Somit bleiben noch folgende Varianten:

- Lokales Netzwerk (local area network, LAN)
- Lokales Strom Netzwerk (power local area network, Power-LAN)

Um unserer Lösung des Geräteortungssystems näher zu kommen.

### **3.1.1.2 Die Variante LAN**

Bei dem lokalen Netzwerk (LAN) ist man immer von der Position des Routers in dem Gebäude abhängig und muss entweder lange Kabel (Patch kabel) bis zum PC oder dem zu Ortenden mobilen Gerät verlegen. Dafür ist die Sicherheit des kabelgebundenen Netzwerkes gegeben und als positiv zu werten. Da sich somit ein zu grosser Nachrüstaufwand abzeichnet kam diese Variante für unser Projekt auch nicht in Frage.

### **3.1.1.3 Die Variante P-LAN**

Sie ist sicherlich am einfachsten nachrüstbar, da ein einfaches Stromnetz reicht, um die notwendigen Hardwarekomponenten einzubauen. Zudem hat es in fast jedem Raum eines Krankenhauses Steckdosen in meist mehrfacher Ausführung. Da die meisten (95%) der mobilen Geräte in den Krankenhäusern an einer festinstallierten Steckdose eingesteckt sind, um ihren eingebauten Akku zu laden, ist ein weiterer Vorteil dieser Variante. So entschlossen wir unser Projekt mit einem P-LAN zu verwirklichen.

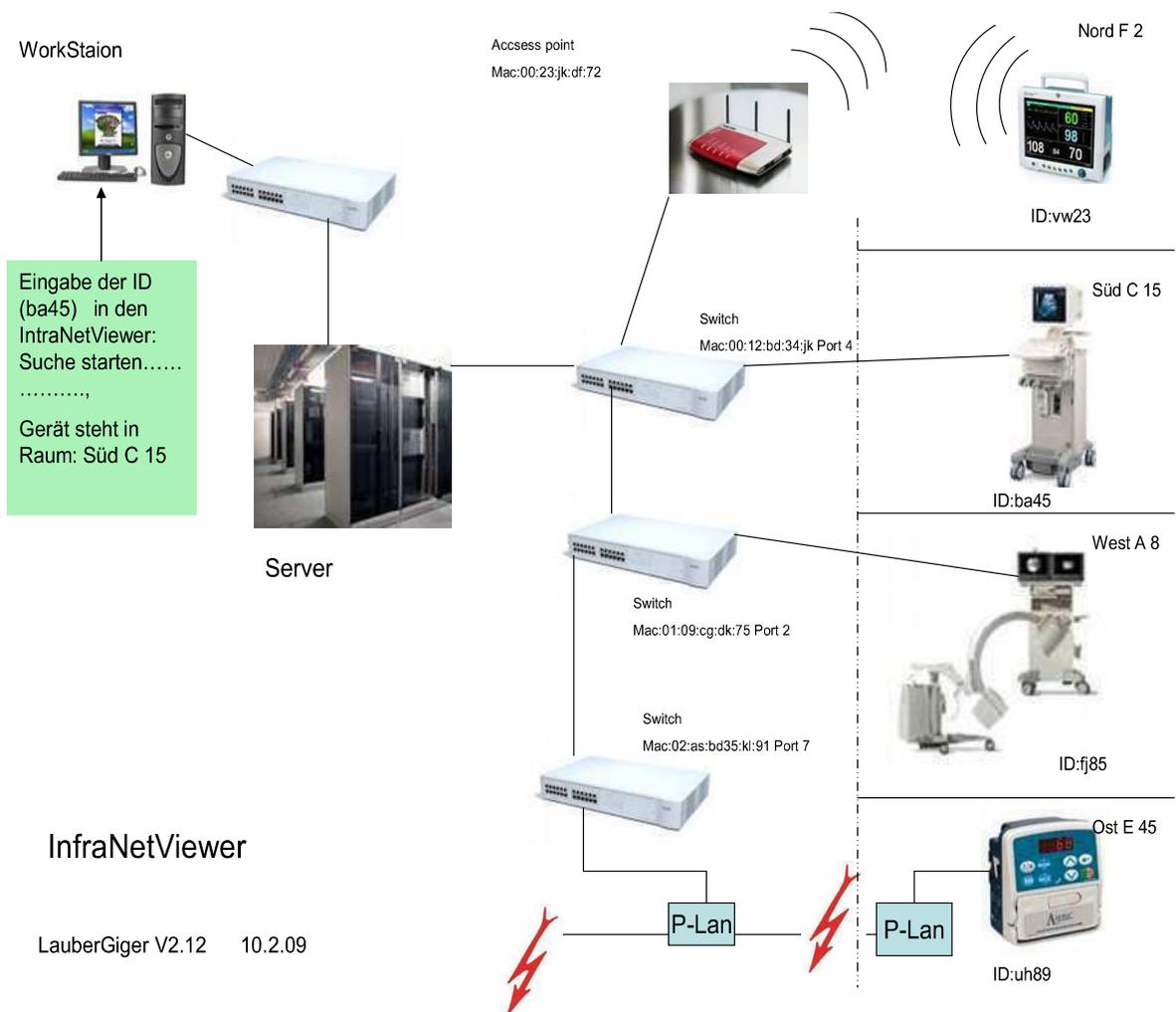
Damit unser INVS realisiert werden konnte, hatten wir folgende Idee von uns in die Tat umzusetzen.

Um über das lokale Strom Netzwerk (P-LAN) kommunizieren zu können, werden spezielle Adapter benötigt. Deren Aufgabe ist es, die ein- oder ausgehenden Signale in hochfrequente Signale umzuwandeln, die dann durch das Stromkabel in das Gebäude geschickt werden können. Das bedeutet, dass sowohl am Sender als auch am Empfänger ein solcher Adapter angebracht werden muss, sprich: man benötigt mindestens zwei solche Adapter.

So würde der eine P-LAN Adapter (1) nahe an einem Sicherungstableau den Stock im Krankenhaus definieren. Der zweite P-LAN Adapter (2), der am mobilen Gerät angebracht oder eingebaut ist, definiert den Raum durch die Steckdose in welche er gesteckt wird. Sobald der zweite Adapter (2) durch eine Steckdose an das Stromnetzwerk angeschlossen wird, sendet er seine einzigartige MAC-Adresse in das Stromnetzwerk.

Die verwendeten MAC-Adressen der Adapter (1)(Raum) und Adapter (2)(mobiles Gerät) werden durch eine Software, die in das Stromnetzwerk schauen kann, den verschiedenen Räumen und mobilen Geräten zugewiesen. Die beiden Adressen werden in einer Datenbank mit den nötigen Informationen hinterlegt, um so bei der Suche eines Gerätes den Raum und die gewünschten Gerätedaten ersichtlich machen zu können.

Gemäss unserer Vorstellung könnte das INVS auf diese Weise funktionieren und ist auf der nächsten Seite noch als Kommunikationsschema dargestellt.



**Abb. 3: Kommunikationsschema des INVS**

Nach abgeschlossener Suche nach dem geeignetsten Netzwerk, widmeten wir uns der Suche nach der optimalen Hardware für unser INVS.

## 3.1.2 Die Auswahl der Hardwarebausteine

### 3.1.2.1 Die Devolo Bausteine

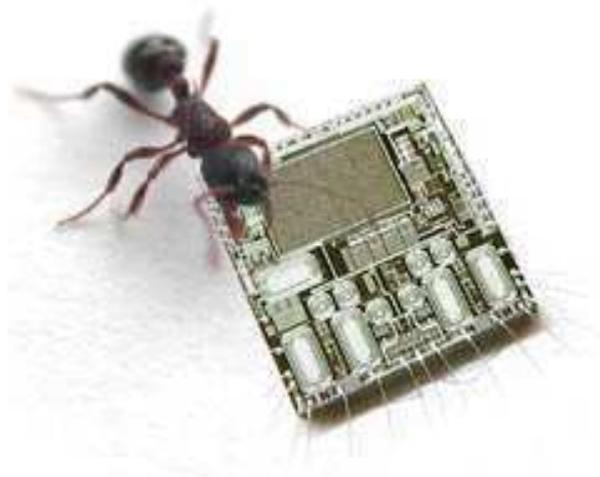
Durch das Internet wurden wir schnell auf die Firma Devolo aufmerksam, da ihre angebotenen Produkte den Grundanforderungen unseres Projektes entsprechen. Schnell mussten wir feststellen, dass sich bei diesem Produkt markante Nachteile abzeichneten. Zum einen war da die Baugröße der Adapter, die teilweise gleich oder noch größer waren als die zu ortenden Geräte, in die sie eingebaut werden sollten. Des Weiteren wirkten sich die hochfrequenten Signale, die über das Power LAN geschickt werden, gelegentlich auf nahegelegene Elektrogeräte aus. Das Piepen, das man von Handys, die zu nahe an den Stereoanlagen liegen, bereits gewohnt ist, kann auch bei Power LAN vorkommen. Auch Radios können im Empfang leicht gestört werden. So war klar, dass es unmöglich war, die Adapter an ein hochtechnologisches mobiles Medizingerät zu bauen. Auch gibt es Probleme, wenn die Räume aus getrennten Stromkreisen bestehen, wie beispielsweise in einem Operationsraum. In solch einem Fall muss ein sogenannter Phasenkoppler installiert werden, was gegen die Bauvorschriften für einen OP verstößt.



Abb. 4: Zwei Devoloadapter einer angeschlossen

Schliesslich senden die Adapter von Devolo einfach ihre MAC-Adresse in das Stromnetzwerk, ohne dass ihre Herkunft geortet werden kann, was für uns der entscheidende Nachteil dieses Produkts ist und noch gewichtiger wiegt als die Grösse der Adapter oder die Problematik der Bauvorschriften. Dies bedeutet, dass sich in einem Stromnetz zahlreiche unidentifizierbare MAC-Adressen befinden. Wir mussten deshalb weiter suchen, um das Gelingen unseres Vorhabens zu gewährleisten.

### 3.1.2.2 Die aizo Technologie



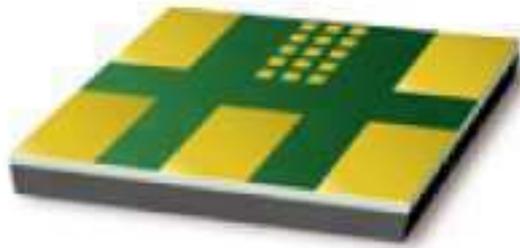
**Abb. 5: Ameise mit aizo Technologie**

Nach weiterem Recherchieren stiessen wir auf die Firma aizo AG, die eine Non-Profit-Organisation namens digitalSTROM.org ins Leben gerufen hat, mit dem Zweck Digital-Strom zum weltweiten Standard in Raumsteuerung, Gebäudeinstallation und Energiemanagement zu erheben. Die aizo AG ist in der Entwicklungsphase eines Hochvoltchips (dSID), der direkt in Geräte eingebaut oder angebaut werden kann und direkt über 230V ins Stromnetzwerk kommunizieren kann. Strom ist nicht mehr dumm! Dieser Hochvoltchip (dSID) sendet eine Identität (ID-Nummer), eine MAC-Adresse an einen Empfänger/Sender den sogenannten Meter (dsM), wo sich der dSID anmeldet. So können nach wenigen Handgriffen, Gebäude «intelligent» gemacht werden verspricht die Firma.

### 3.1.2.3 Vergleich mit anderen Technologien

	Kabel	Kabelfunk (z.B. Powerline)	Funk	digitalSTROM
Investitionskosten	-	-	o	+
Übertragungssicherheit	+	o	-	+
Baugrösse	-	-	-	+
Belastung durch Elektromog	+	o	-	+
Umbau ohne neue Kabel	-	o	+	+
Umbau ohne Ingenieur	-	-	o	+
Nutzung ohne Anleitung	-	-	o	+
Programmierung ohne PC	-	-	+	+
Kommunikation durch die Steckdose	-	+	+	+
Abhörsicherheit	+	-	-	+
Handelsübliche Bauformen	-	-	-	+
Einbaubar in üblichen Leuchtmitteln	-	-	-	+
Einbaubar in üblichen Steckern	-	-	-	+
Universal Serial Nummer (vgl. RFID)	o	o	-	+
Personalisierte Profile	-	-	-	+
Integrierte Energiemessung	-	-	-	+
Integrierter Überspannung reg.	+	+	-	+
Schutz vor Netzüberlastung	-	-	-	+
Gestaffelter Gerätestart nach Netzausfall	-	-	-	+
Kontrollierter Standby-Verbrauch unter 0.3W	-	-	-	+

aizo: Technologien Vergleich, [http://www.aizo.com/de/technologie/tech\\_vergleich.php](http://www.aizo.com/de/technologie/tech_vergleich.php)  
[10.03.2009]



15 x 18 x 6 mm

**Abb. 6: dSID-digital Strom Identität**

Von diesem Hochvoltchip und dem Gedanken, dass Strom nicht mehr dumm sein soll, begeistert, fragten wir aizo um Testmaterial an. Die Firma willigte schnell ein und liess uns durch Andrea Heubi einen Demokoffer zukommen, der uns für einen Monat zur Verfügung gestellt wurde.

#### **3.1.2.4 Testreihen mit dem aizo Demokoffer**

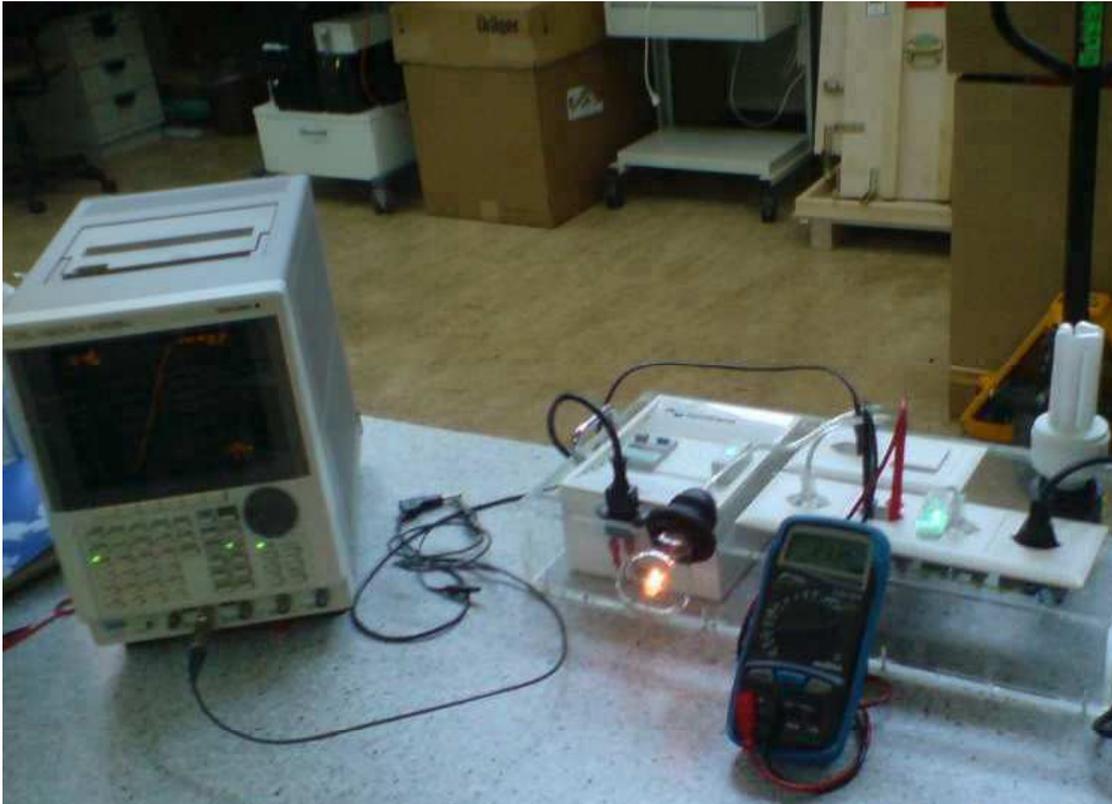
Am Sonntag 17.05.2009 begannen wir dann mit den ersten Kommunikations-/Verbindungs- Tests. In diesen Tests wollten wir verschiedene Räume eines Krankenhauses elektrisch simulieren. Da die Räume verschieden isoliert sind, war unsere Befürchtung, dass die Kommunikation des aizo dSID mit dem dsM nicht oder nur teilweise zu Stande kommen könnte. Zunächst wollten wir aber nachweisen, dass eine Kommunikation zwischen den einzelnen Komponenten der Firma aizo zu Stande kommt. Die Kommunikationsfähigkeit sollte über das gesamte Stromnetz nachgewiesen werden. Mit den in der nachfolgenden Tabelle ersichtlichen Bauteilen erstellten wir die verschiedenen Tests.

<u>Bauteile</u>				
Gerät	Typ	Seriennum.	Hersteller	Eigenschaften
Trenntrafo	ILLM-350	47080	Wagner& Grimm	350 VA Prim 230V 50-60Hz 2A Sec 230V 50-60Hz 1.52A
Oszilloskop	Suffix-20-5-F	23XV5104	YOKOGAWA	DL 1300A 4 Channel 100 Hz
Multimeter	MX	153727XFR	Metrix	Autoranging TRMS
Steckdosenleiste	FL 2-6-CH	9742		mit Filter 1 250V 6.3A 50/60Hz
Netzfilter 2	FN 284-6-06		Schaffner	110/250VAC/50-60Hz 0,22uF, 2x0.45mH
Netzfilter 3	FN 372-2/01		Schaffner	110/240V 50Hz 2x2A 0,015uF, 2x2200pF, 2x2mH
Dokingstation	Asena DS	8008-00597	Asena	U-230V
Demoanlage	Hauptmodul		Aizo	4 Steckdosen Feller Mini-Tableau Hager FI CDA 225D 25A/30mA dSM M10 230V/50Hz 10A
	Steckermodul			3xdSID10 OABO Labormuster

(Lauber/Giger 2009, Test aizo Demokoffer)

## 1. Testaufbau

Beim ersten Test wollten wir, wie schon erwähnt, die Kommunikation unter den aizo Bauteilen, konkret zwischen dem Hochvoltchip dSID und dem Meter dsM, irgendwie messbar oder erkennbar machen.



**Abb. 7: Kommunikationstest zwischen dSID und dSM**

- Oszilloskop, Multimeter, aizo Demoanlage

Die gemessenen Werte der Netzspannung waren:

Wechselspannung  $U = 232V$

Frequenz  $f = 49.954Hz$

Die gemessenen Werte auf den Steckdosen des Demokoffers:

Wechselspannung  $U = 233V$  (ohne Zusatzmodul dSD510)

Frequenz  $f = 49.958Hz$

Als wir den Hochvoltchip an den Stromkreis schlossen und ihn aktivierten, konnten wir einen Pik auf der 50Hz-Sinuskurve sehen. Durch näheres Untersuchen konnten wir feststellen, dass der Pik sowohl im steigenden wie auch im sinkenden Teil der Sinuskurve vorkommt.

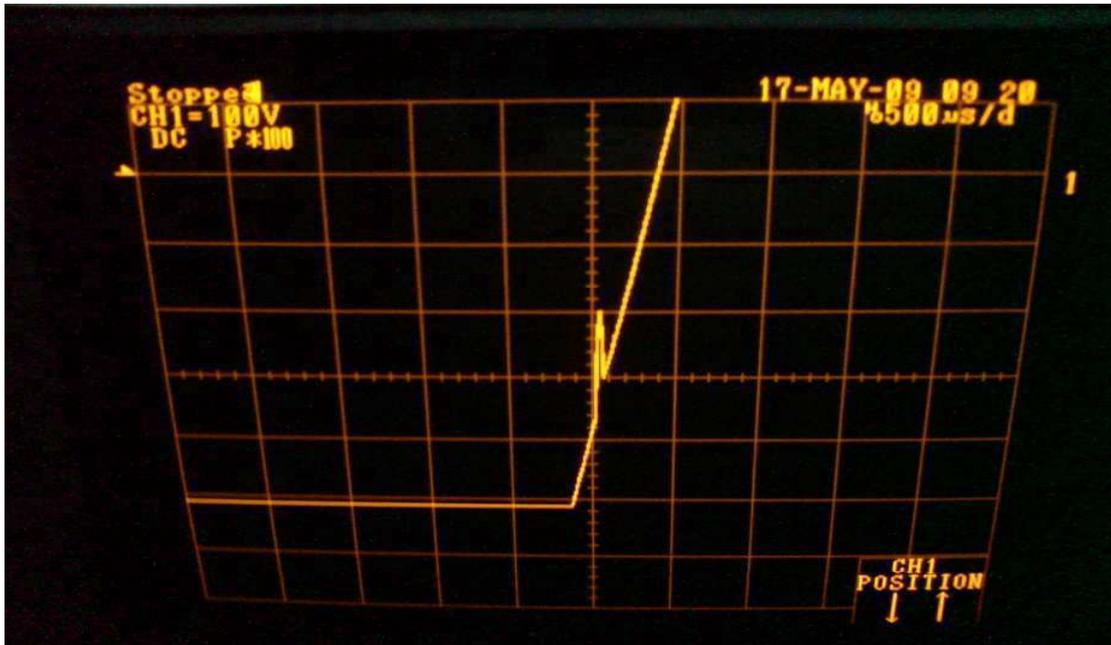
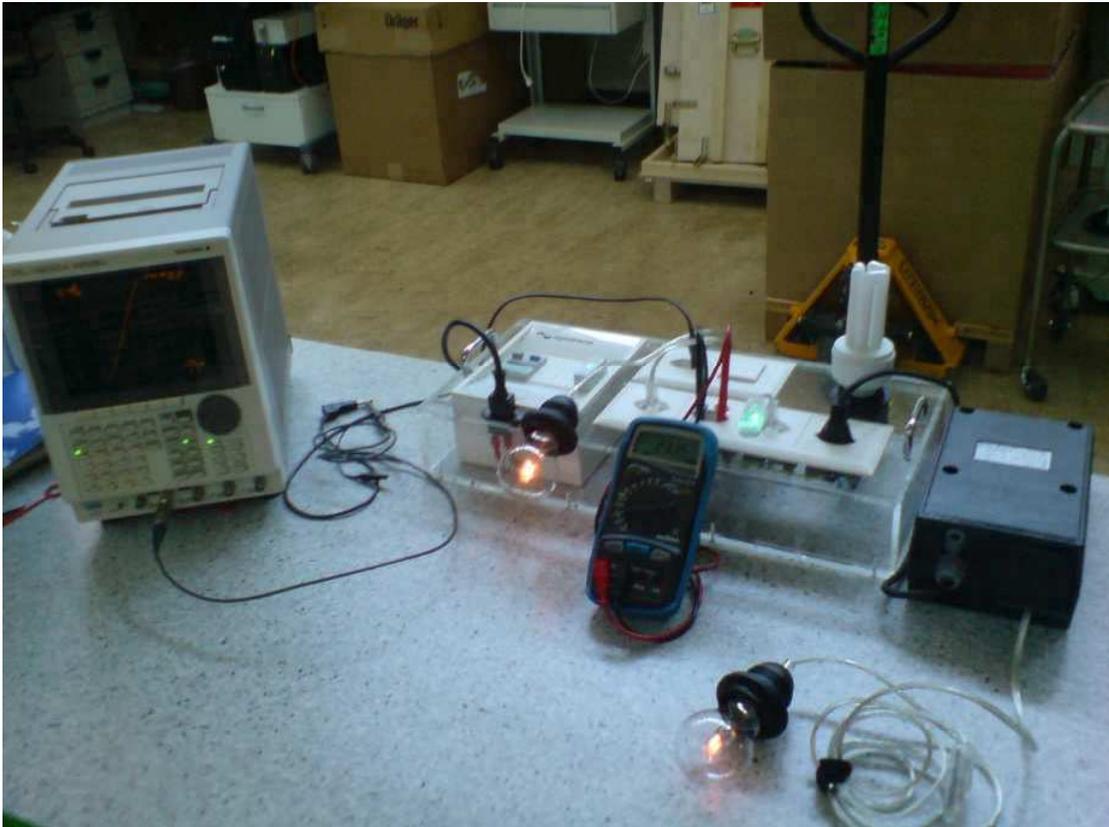


Abb. 8: Pik auf steigender Sinuskurve



Abb. 9: Pik auf Sinuskurve

## 2. Testaufbau



**Abb. 10: Operationsraum Simulation**

- Oszilloskop, Multimeter, aizo Demokoffer, Trenntransformator

Zur Simulation eines Operationsraumes schalteten wir einen Trenntrafo, so wie er original auch in den Gebäuden mit eingebaut wird, dazu. Das wird gemacht, damit der Operationsraum vom übrigen Netz getrennt ist und so ein eigenes Potential hat.

Trotz des zugeschalteten Trenntrafos konnte der dSID mit dem dSM über das Stromnetz Kontakt aufnehmen und kommunizieren. Nachdem der erste Test erfolgreich verlaufen war, erstellten wir den nächsten Messaufbau.



Durch den dritten Test wollten wir die Tatsache, dass oft ein Netzfilter schon in einem mobilen Gerät eingebaut ist, simulieren. Ausserdem sollte die Situation eines Gerätes, das an einer Steckleiste mit Filter eingesteckt ist, simuliert werden. In beiden Fällen kam die Kommunikation zu Stande. So muss sich das Personal nicht kümmern, wo sie das mobile Gerät einstecken, um es ortungsfähig zu machen. Somit kamen wir der Auswahl der optimalen Hardware für das Infranetviewer-System wieder einen Schritt näher.

#### 4. Testaufbau



Abb. 12: Aufbau mit Dokingstation

- Oszilloskop, Multimeter, aizo Demoanlage, Dokingstation

Im nächsten Schritt testeten wir den Aufbau mit einer Dokingstation, wie im Bild 13 rechts ersichtlich, während dem sie mit drei Spritzenpumpen bestückt ist.

Dieser Testaufbau simuliert eine reale Situation auf einer Intensivstation (IPS), einer Notfallstation oder auf einer Pflegestation. Einfach dort wo mehrere Geräte gebraucht werden und eine Dokingstation zum Einsatz kommt.

Auch während dem die Spritzenpumpen der Dokingstation zum Teil in Betrieb oder im Ladeprozess waren, zeichnete das Oszilloskop wiederum einen Pik auf der Sinuskurve auf, wodurch auch dieser Test erfolgreich abgeschlossen werden konnte.

Nun waren da nur noch die Kombinationen der vorhergehenden Simulationen, die unsere Technikauswahl beeinflussen konnten.

## 5. Testaufbau



Abb. 13: Aufbau mit allen Komponenten

Nun kombinierten wir den Trenntrafo mit einem Netzfilter, so konnten die Komponenten ohne Probleme von beiden Richtungen her (Trafo->Netzfilter = Netzfilter->Trafo) kommunizieren. Doch dann schalteten wir noch einen Netzfilter zusätzlich in Serie vom Trenntrafo und dem ersten Netzfilter, was zu Kommunikationsproblemen führte. Zwar traten sie nur in einer Richtung auf, aber es war doch eine Bestätigung, dass dieser Pik irgendeinmal aus dem Netz gefiltert oder getrennt werden kann. Da der Pik nur vom dSID, also vom Gerät her in das Stromnetz hinein senden kann, konnten wir auch diesen Test als erfolgreich für unser Ortungssystem werten. Denn in einem Operationsraum, also einem Raum, der einen Trenntrafo eingebaut hat, werden keine Steckerleisten mit zusätzlichen Netzfiltern eingesetzt. Daher ist es sehr unwahrscheinlich, dass zwei Netzfilter in Serie nach einem Trenntrafo verwendet werden, die zu Kommunikationsstörungen führen könnten. Selbst wenn der unwahrscheinliche Fall eines zweiten Netzfilters in einem mobilen Gerät, welches in der Steckdose eingesteckt ist, also zwei Netzfilter nach Trenntrafo in Serie geschaltet wären, eintreffen würde, könnte das Gerät dennoch geortet werden. Dies weil, die Kommunikationsstörungen nur in einer Richtung (Gerät(dSID)->Empfänger(dSM)) auftraten.

Nach diesen erfolgreichen Verbindungstests des dSID im Hinblick auf die verschiedenen Raum Gegebenheiten, die in einem Krankenhaus herrschen, verabredeten wir uns erneut mit der Firma aizo. Wir wollten uns vorstellen, unsere Tests präsentieren und die Firma um eventuelle Mithilfe bei unserem Projekt bitten. Denn wir waren überzeugt, dass sich die neue Technologie der Firma aizo optimal für unser Vorhaben der Erfindung des Infranetviewer-Systems eignen würde. Der Vorteil der aizo Technologie besteht zusammenfassend aus den Ergebnissen unserer Raumtests und allgemeinen Vorteilen im Vergleich mit anderer, momentan erhältlicher Technologie. Im nächsten Zitat präsentieren die für unser Projekt relevanten Vorteile der aizo Technologie, die da sind Einfachheit der Installation und Bedienung, Nachrüstbarkeit, Kostengünstigkeit, Flexibilität, minimale Strahlung.

### 1. Einfache Installation

„In kürzester Zeit und ohne dass Schmutz entsteht, ist Ihre Wohnung digitalSTROM-fähig! Ihr Elektriker montiert ein kleines Modul hinter den Sicherungen (den digitalSTROM-Meter, sozusagen das kollektive Hirn) und eine kleine Klemme in Ihren vorhandenen Lichtschalter. Damit ist die digitalSTROM-Installation fertig und bereit für alle Arten von digitalSTROM-Geräten. Es müssen keine neuen Leitungen verlegt werden; bis zu 2000 Geräte lassen sich über eine bestehende Leitung steuern!

[...]

### 3. Sie können Geräte jederzeit nachrüsten

Ist ein Gerät noch nicht digitalSTROM-ready, lässt es sich problemlos nachrüsten. Der Chip ist so klein, dass er in einer Lampenfassung oder einem Zwischenstecker Platz hat und sich so an beliebige Geräte anschließen lässt. Für den Elektriker gibt es den digitalSTROM-Chip auch in Form und Größe von Lüsterklemmen, die sich platzsparend mit allen elektrischen Geräten verbinden lassen.

### 4. digitalSTROM ist einfach in der Bedienung

Sie müssen sich nicht um Technik kümmern, sondern können den Komfort genießen. Denken Sie zum Beispiel an die Zentralverriegelung im Auto. Wäre es nicht genauso angenehm, alle Lampen in einem Raum mit einem Schalter auszuschalten? Oder mit dem gleichen Schalter auch die Jalousien zu bedienen? digitalSTROM macht das möglich!

### 5. Bleiben Sie flexibel für die Zukunft

digitalSTROM setzt auf offene Standards. Hersteller können Geräte problemlos mit komfortablen Anwendungen ausstatten. Mit digitalSTROM setzen Sie auf eine Zukunftstechnologie: Beginnen Sie zum Beispiel mit der Lichtsteuerung und ergänzen Sie später die automatischen Jalousien. Und falls ein Klimasystem oder eine Alarmanlage auf den Markt kommt, die Ihnen gefällt – mit digitalSTROM können Sie sie problemlos integrieren.

[...]

### 7. Profitieren Sie von NIEDRIGEN Installationskosten

Da digitalSTROM das erste System mit echtem Plug & Play auf diesem Gebiet ist, lassen sich etwa 70 % der Kosten im Vergleich zu einer herkömmlichen Installation einsparen. Sie benötigen im Gegensatz zu üblichen Installationen weder einen Programmierer noch einen Techniker, wenn Sie ein neues Gerät kaufen oder ein Gerät von einem Raum in den anderen verschieben.

## 8. Wohnen ohne Elektrosmog

digitalSTROM nutzt das bestehende Stromnetz. So entsteht keine zusätzliche Strahlung wie bei drahtloser Datenübertragung. digitalSTROM übermittelt zudem nur sehr wenige Daten. Deshalb steigt auch die Strahlung der Stromleitung nicht wie bei anderen Systemen. Wohnen Sie intelligent und ohne Elektrosmog!

(aizo, Vorteile: [http://www.aizo.com/de/technologie/tech\\_vorteile.php](http://www.aizo.com/de/technologie/tech_vorteile.php)  
[02.09.2009])

### 3.1.3 Technologie Beschaffung

#### 3.1.3.1 Vorstellung bei der Firma aizo

Tatsächlich bekamen wir einen Termin beim aizo Geschäftsführer. Wir bereiteten uns mit einer Powerpointpräsentation auf den Auftritt bei aizo vor. Gespannt auf die Firma und was der Geschäftsführer von uns/ unserem Projekt halten würde, nahmen wir den Termin am 20.05.2009 wahr. Leider war der Geschäftsführer nicht in der Lage uns zu empfangen, da sein Terminkalender überfüllt war. Wir wurden stattdessen von seinem Stellvertreter Guido Pirovino empfangen, der uns in den Meetingraum begleitete. Dort angekommen begannen wir sofort mit unserer Präsentation. In einer lockeren Atmosphäre konnten wir Rede und Antwort zu unserem Projekt und den vollzogenen Tests stehen. Die Tatsache, dass wir den Pik mit dem Oszilloskop ersichtlich machen konnten, erstaunte ihn, da er das Signal, das der dSID ins Stromnetz sendet, auch noch nie schwarz auf weiss gesehen hat, da solche Tests in der Firma aizo noch nie durchgeführt wurden. Trotz Guido Pirovinos Begeisterung machte er uns klar, dass die Firma aizo kein Interesse an unserer Idee habe. Dies entkräftete unsere Befürchtungen, die Firma aizo könnte sich unsere Idee zu Nutzen machen. Die Rückgabe des getesteten Demokoffers, den wir, wie sich herausstellte, durch ein Missverständnis erhalten hatten, konnten wir um eine Woche hinauszögern. Nach dieser zusätzlichen Woche, mussten wir ihm versichern, den Demokoffer zurück zugeben. Er sicherte uns aber Hardware wie auch Software für weitere Forschungszwecke zu, sobald wir der Digitalstromorganisation beigetreten wären. Genau das war das Missverständnis gewesen: Die Möglichkeit zur Benutzung eines Demokoffers ist nämlich an eine Mitgliedschaft in der Digitalstromorganisation gekoppelt.

### **3.1.3.2 Anmeldung bei der digitalSTROM.org**

Innert Kürze reichten wir die notwendigen schriftlichen Unterlagen für einen Beitritt bei der Digitalstromorganisation gemäss den erhalten Vorgaben ein, um möglichst schnell wieder Hardware und Software von der Firma aizo zur Verfügung gestellt zu bekommen.

Dank der Unterstützung der Medizintechnikerschule, im Besonderen durch Margrit von Rickenbach-Inglin und Alois Amstutz konnten wir uns im Namen der Schule als Forschungseinheit anmelden. Dies hatte die Minimierung der Mitglieder Kosten zur Folge und begünstigte unseren Mitgliederstatus. Als Forschungsinstitut mussten wir auch, wie in Punkt 2 des Mitgliedschafts-Antrages beschrieben, einen kurzen Steckbrief einreichen, welchen Rickenbach-Inglin und Amstutz wie folgt formulierten.

#### **Steckbrief der höheren Fachschule Medizintechnik**

Die Höhere Fachschule Medizintechnik Sarnen bietet seit 1996 einen auf die Medizintechnik spezialisierten Studiengang an. Die berufsbegleitende Weiterbildung führt zum eidgenössisch anerkannten Abschluss Dipl. Medizintechniker/In HF und befähigt die Absolventen/innen Funktionsweise, Instandhaltungsprozesse und Regulative von medizintechnischen Geräten zu verstehen, Aufgaben aus dem Fachgebiet sachkompetent zu bearbeiten und mit allen Beteiligten des medizinischen Umfelds dank einer gemeinsamen Sprache zu kommunizieren. Die Studierenden erhalten das Rüstzeug, um in ihrem Bereich Projekt- und Führungsverantwortung zu übernehmen.

Medizintechnikerinnen und Medizintechniker arbeiten in Spitälern, in medizintechnischen Industriebetrieben und in Instandhaltungsfirmen. Unterdessen gibt es mehr als 120 Absolventinnen und Absolventen, die das Berufsbild der Medizintechnikerin und des Medizintechnikers in der Arbeitswelt profilieren.

An der Schule entstehen in den Diplomarbeiten immer wieder innovative Ansätze. Die Mitgliedschaft bei digitalSTROM erlaubt es unseren Studierenden das Know How der Organisation in der medizintechnischen Umgebung anzuwenden.



(Rickenbach-Inglin, Amstutz, 2009)

Dieser Steckbrief wurde dann auch veröffentlicht, wie hier auf der offiziellen Website der Digitalstrom Organisation zu sehen ist.

digitalSTROM.org

D E Home Kontakt Presse Newsletter Login

search »

Über uns Technologie Mitglieder Aktuell Stellenangebote Downloads

→ Überblick  
→ Mitglied werden

## HF für Medizintechnik Sarnen

Die Höhere Fachschule Medizintechnik Sarnen bietet seit 1996 einen auf die Medizintechnik spezialisierten Studiengang an. Die berufs begleitende Weiterbildung führt zum eidgenössisch anerkannten Abschluss Dipl. MedizintechnikerIn HF und befähigt die Absolventinnen und Absolventen Funktionsweise, Instandhaltungsprozesse und Regulative von medizintechnischen Geräten zu verstehen, Aufgaben aus dem Fachgebiet sachkompetent zu bearbeiten und mit allen Beteiligten des medizinischen Umfelds dank einer gemeinsamen Sprache zu kommunizieren. Die Studierenden erhalten das Rüstzeug, um in ihrem Bereich Projekt- und Führungsverantwortung zu übernehmen.

Medizintechnikerinnen und Medizintechniker arbeiten in Spitälern, in medizintechnischen Industriebetrieben und in Instandhaltungsfirmen. Unterdessen gibt es mehr als 120 Absolventinnen und Absolventen, die das Berufsbild der Medizintechnikerin und des Medizintechnikers in der Arbeitswelt profilieren.

An der Schule entstehen in den Diplomarbeiten immer wieder innovative Ansätze. Die Mitgliedschaft bei digitalSTROM.org erlaubt es unseren Studierenden, das Know-how der Organisation in der medizintechnischen Umgebung anzuwenden.

© digitalSTROM.org 2009 [Impressum](#)

**Abb. 14: Print Screen Steckbrief**

Da wir bei unseren Tests lediglich einen Pik auf der Sinuskurve feststellen konnten und so noch kein Code oder Daten ersichtlich waren, die im Stromnetz gesendet werden, riet uns Guido Pirovino, einen Konverter zu kaufen, der Signale von einer RS485 Schnittstelle auf eine Ethernet Schnittstelle konvertieren kann. Gemäss Pirovino entspricht dieses Vorgehen dem Vorgehen des Servermoduls von aizo. Es ist anzumerken, dass unser Demokoffer nicht mit einem Servermodul dSS ausgerüstet gewesen war.



90 x 17,5 x 70 mm

**Abb. 15: dSS-digital Strom Server**

Der digitalSTROM-Server (dSS) bildet die Hierarchieebene oberhalb der Räume für raumübergreifende Kommunikation und die Schnittstelle zu digitalSTROM-fremden Komponenten, z. B. anderen Hausbussystemen. Seine dS485-Schnittstelle dient zur Kommunikation mit anderen dS-Komponenten

(z. B. digitalSTROM-Meter). Eine zusätzliche serielle Schnittstelle COM erlaubt die Anbindung von Modems oder speziellen Komponenten, unter anderem für Wartungszwecke. Der integrierte Webserver ermöglicht den Zugriff auf das digitalSTROM-System über Standard-TCP/IP-Netze wie das Internet. Der Netzwerkanschluss dient auch zur Einbindung in andere Automatisierungssysteme.

(aizo, Server: [http://www.aizo.com/de/produkte/prod\\_dSS.php](http://www.aizo.com/de/produkte/prod_dSS.php) [02.09.2009])

## Konverter EX-6051IS

Wir suchten sofort einen geeigneten Konverter, damit wir ihn noch direkt an der Demoanlage testen konnten. Denn die Woche, die wir zusätzlich bekommen hatten, musste noch ausreichen, um den Konverter zu testen und erste Signalzeichen zu sehen. Im Distrelec-Shop fanden wir dann den EX-6051IS von dem Hersteller EXSYS.



**Abb. 16:** Zwei Konverter EX-6051IS

Wir kauften ihn und installierten die Software auf dem Notebook. Nachdem wir den Konverter mit der Demoanlage und unserem Notebook verbunden hatten, konnten wir erste Signalzeichen sehen (siehe im Anhang unter Einstellungen Konverter). Durch den Start des Programms „Test MP2“ wurde die Kommunikation mit dem dSM10 sichtbar.

Test MP2

COM 17

→ < feH >

< ffH >

< fbH >

< faH >

< eeH >

< ecH >

< f8H >

< f6H >

< f2H >

< f7H >

< f9H >

Es wurden einzelne Signaldaten übermittelt, während der dSID angemeldet war. Auch ohne Anmeldung des dSID wurden im Netz Daten versandt. Wir konnten nur annehmen dass diese Daten für Information des momentanen Stromverbrauchs benötigt werden. Da diese Frage für unser Projekt irrelevant ist, wurde sie im Rahmen dieser Arbeit nicht weiter verfolgt.

### **3.1.3.3 Vorstellung bei der Geschäftsleitung der Firma aizo**

Wiederholt versuchten wir im Folgenden mit der Firma aizo telefonischen Kontakt herzustellen, leider ohne Erfolg. Es stellte sich heraus, dass wir Guido Pirovino nicht mehr kontaktieren konnten, da er zu diesem Zeitpunkt bereits nicht mehr für die Firma arbeitete. Wir waren überrascht und enttäuscht, da die Realisierung unseres Projektes von den Versprechen Pirovino's abhängig war. Wir waren gezwungen, Alternativen in Betracht zu ziehen.

Uns war klar, dass wir mit der Devolo Technik durchaus etwas erreichen könnten, die erfolgreiche Realisierung unseres Projektes war jedoch fragwürdig. Bedeutete dies, dass unser Projekt zum Scheitern verurteilt war? Die Tatsache, dass unsere Kontaktperson aizo verlassen hatte, war nicht der einzige Faktor, der die Zusammenarbeit mit aizo erschwerte. Die Firma verlagerte ihren Standort von der Hardtumstrasse in Zürich nach Schlieren wodurch die Firma wenig Zeit für interessierte Medizintechnikstudenten aufbringen konnte oder wollte.

Unser Ehrgeiz und unsere Ausdauer führten schliesslich zum Erfolg. Andrea Heubi gewährte uns einen zweiten Vorstellungstermin, nachdem wir sie fast täglich telefonisch kontaktiert hatten. Jan Zakrzewski, der Geschäftsführer der aizo Schweiz, wollte uns mit seinem Softwareentwickler Miguel Rodriguez am Mittwoch 22.07.2009 um 11.30 Uhr für eine halbe Stunde empfangen.

Wir bereiteten uns gut vor, um unser Projekt mittels aizo Komponenten zu retten. Tatsächlich wurden wir an diesem Tag von Jan Zakrzewski und Miguel Rodriguez empfangen. Nach der Begrüssung wollten wir unser Projekt erneut vorstellen. Doch das war gar nicht mehr nötig, da die Herren unser Projekt bereits beeindruckend gut kannten.

Sie fragten uns direkt, wie sie unser Projekt unterstützen könnten, worauf wir unsere Wünsche äusserten. Sie sicherten uns im Bereich von Software und Hardware Unterstützung zu, indem sie uns wieder einen Demokoffer sowie zusätzliche Einzelkomponenten zur Verfügung stellten.

Zudem beantworteten sie unsere Fragen über die Kommunikation des dSID und des dSM und des dSS. Die Software basiere hauptsächlich auf der Programmiersprache C++. Miguel Rodriguez riet uns, selbst einen Server zu erstellen, der auf Linux Basis laufen sollte. Dies würde es uns ermöglichen, mit der Kommunikation etwas Produktives anzufangen und nicht nur die Daten sichtbar zu machen. Auf diese Weise sollte dann auch die Möglichkeit bestehen, die einzelnen IDs des Hochvoltchips einer Datenbank zuzuweisen. Sie selber würden auch mit diesem Server arbeiten und so könnte er uns am besten unterstützen, wenn wir weitere Fragen hätten.

Wir bekamen noch den Zugang zu den dSS Linux Applikationen und dem Quellcode. Die nötigen Applikationen konnten wir somit direkt online über die Digitalstrom Organisation downloaden.

Das halbstündige Meeting mit Jan Zakrzewski und Miguel Rodriguez bedeutete die Wende zur erfolgreichen Realisierung unseres Projekts. Denn wir hatten uns nur noch bis Ende Woche Zeit gegeben, unser Projekt mit Hilfe der Firma aizo zu realisieren. Da die Zeit knapp geworden war, hatten wir uns schon mit der Möglichkeit auseinandergesetzt, unser Vorhaben entgegen unserer Ergebnisse mit der Technologie von Devolo durchzuführen.

### **3.1.4 Linux-Server**

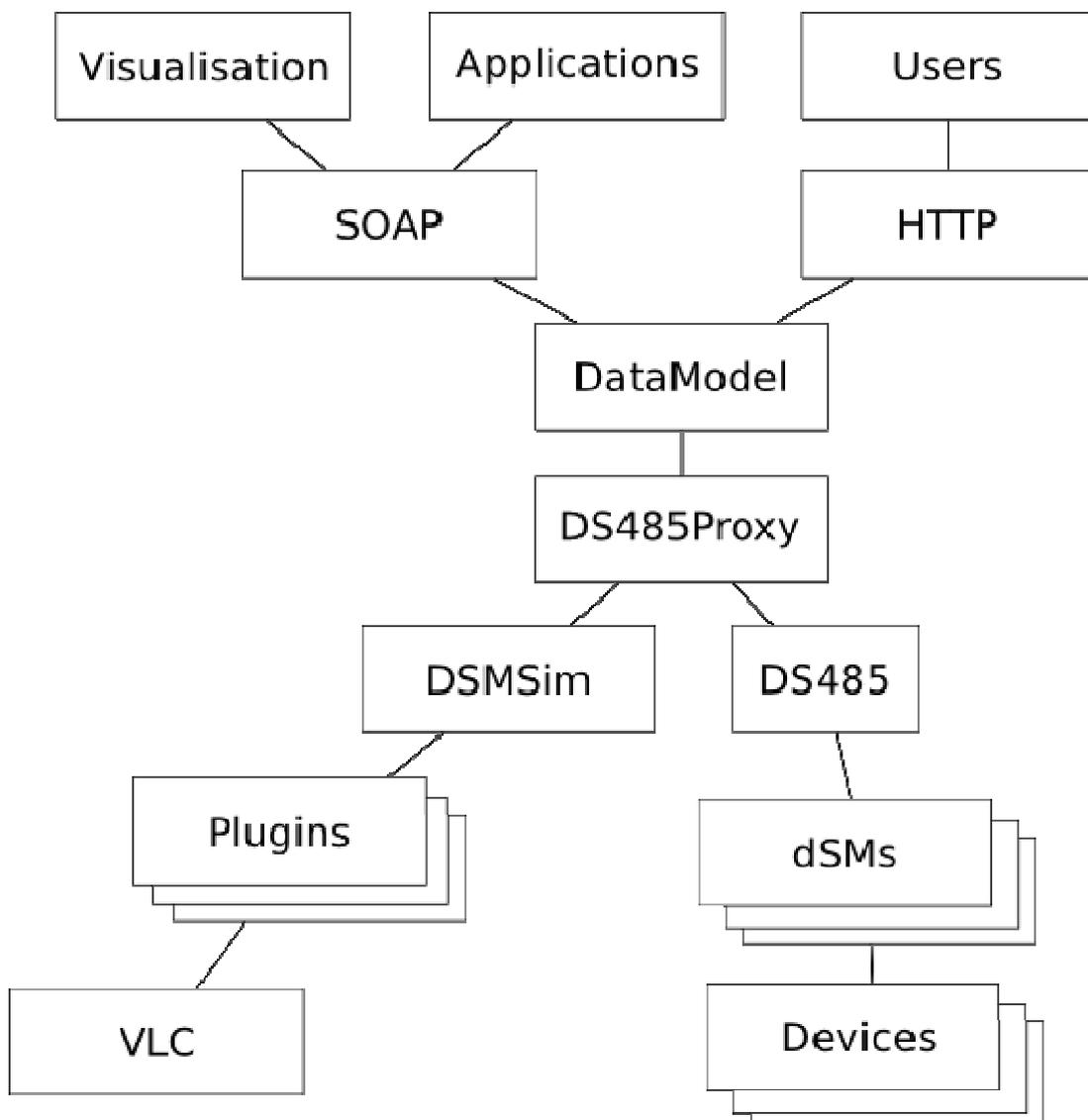
Nun war klar, dass wir den Tipps von Miguel Rodriguez Folge leisten würden und unsere zwei Rechner auf Linux umrüsten wollten. Durch Informationen von Kollegen und dem Internet entschieden wir uns für die Ubuntu Version der Linux Plattform. Am 29.07.2009 versuchten wir Ubuntu als zweite Partition auf unseren Windows Notebooks zu installieren. Dieses Unterfangen gelang uns auf Anhieb. Nun mussten wir das Linux Programm so weit bringen, dass wir einen Server bilden konnten. Der Server war wie folgt aufgebaut.

#### **3.1.4.1 Die Systemübersicht des Servers (dSS)**

Jedes Subsystem kann freigegeben oder gesperrt werden durch die commandline. Das Herzstück des dSS ist sein Daten Model. Es stellt die API zur Verfügung, die erlaubt, dass ein Entwickler die Geräte, Zonen und Gruppen wie gewünscht manipulieren kann. Das Fundament bildet die SOAP-Schnittstelle, um die API Anwenderprogrammierschnittstelle zugänglich für die Aussenwelt zu machen. Zusätzlich zur SOAP-Schnittstelle gibt es eine JSON-Schnittstelle, die mit JavaScript zusammen in einem Webbrowser läuft.

Der DS485Proxy ist dafür verantwortlich, die physischen und simulierten Geräte zu kontrollieren. Er wandelt abstrakte Befehle vom Daten Model um, konvertiert sie zu einem oder mehreren DS485-Bilder. Diese Bilder werden dann entweder zu einem simulierten dSM gesandt oder an die Leitung in Abhängigkeit des Bestimmungsortes weitergegeben.

Der simulierte dSM kann durch Plug-Ins erweitert werden. Ein Beispiel ist ein simulierter dSID, der über die vorhandene Telnet-Schnittstelle einen VLC (Mediaplayer) kontrollieren kann.



**SOAP** (ursprünglich für *Simple Object Access Protocol*) ist ein Netzwerkprotokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote durchgeführt werden können. SOAP stützt sich auf andere Standards: XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht (vgl. TCP/IP-Referenzmodell)

(Wiki, SOAP: <http://de.wikipedia.org/wiki/SOAP>[07.09.2009])

Der **VLC media player** (anfänglich *VideoLAN Client*) ist ein portabler, freier Mediaplayer sowohl für diverse Audio-, Videocodecs und Dateiformate als auch DVDs, Video-CDs und unterstützt unterschiedliche Streaming-Protokolle.

(Wiki, VLC: [http://de.wikipedia.org/wiki/VLC\\_media\\_player](http://de.wikipedia.org/wiki/VLC_media_player)[07.09.2009])

Eine **Programmierschnittstelle** ist eine Schnittstelle, die von einem Software-system anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. Oft wird dafür die Abkürzung **API** (für engl. *application programming interface*, deutsch: „Schnittstelle zur Anwendungsprogrammierung“) verwendet. Im weiteren Sinne wird die Schnittstelle jeder Bibliothek (*Library*) als API bezeichnet.

(Wiki, API: <http://de.wikipedia.org/wiki/Programmierschnittstelle>[07.09.2009])

### 3.1.4.2 Probleme bei der Serverbildung

Nach mehreren Versuchen den Server zu bilden, stockte unser Programm immer wieder an der selben Stelle. Wir sahen, dass uns zum Bilden des Servers immer noch gewisse Dokumenten Bibliotheken fehlten.

Uns war nicht bekannt, wie wir an diese Bibliotheken gelangen und diese in unserem Unbuntu einsetzen konnten. Bei unseren Nachforschungen in Foren und bei bekannten Linux Usern stiessen wir auf Unverständnis, da die meisten Linux benutzen, um im Internet zu surfen.

Allmählich unter Zeitdruck, versuchten wir wiederholt unsere Kontaktpersonen bei aizo zu erreichen. Leider blieben unserer Bemühungen auf Grund der Sommerferien vergeblich, worauf wir uns an Alois Kaufmann wendetet, der uns allerdings auch nicht weiter helfen konnte. Schliesslich kontaktierten wir den Inhaber der Firma Brain, Roger Tinembart. Wir hofften, durch Hinzuziehung eines Profis möglichst schnell an essentielle Informationen zu gelangen. Obwohl Roger Tinembart sehr beschäftigt war, erhielten wir schnell einen zweistündigen Termin, um unser Anliegen vorzubringen.

Zwar reichte die Zeit nicht aus, um alle Probleme vollständig aus dem Weg zu räumen, Roger Tinembart konnte aber durch genaues Betrachten der Doku Datei feststellen, dass es sich um ein ausgewachsenes bidirektionales Kommunikationsprotokoll handelt.

Dank Roger Tinembart wussten wir nun genauer auf was wir schauen mussten und was wir wissen mussten von dem Serverprogrammierer der Firma aizo. Nach dem Treffen teilte uns Roger Tinembart mit, dass er versucht hatte den Surfer unter SUSE (Open Suse 11) zu bilden und dass ihm noch einige Pakete fehlen würden, die im RAEDME erwähnt würden. Dadurch wurde uns nicht nur bewusst, dass es andere Freeware von Linux Versionen gab so wie Suse, Unbuntu, Mandriva, etc., sondern auch, dass wir gar nicht wussten, welche Version von der Firma aizo verwendet wird.

Also kontaktierten wir Jan Zakrzewski und Miguel Rodriguez um eine Auflistung der Pakete zu erhalten, die es zum Bilden des Servers braucht und um zu erfahren, welche Version von Linux sie verwenden. Die Antwort kam dann von Johannes Winkelmann, der uns mitteilte, dass er mit Unbuntu 9.04 arbeite und es folgende Pakete brauche, die via apt-get verfügbar sind:

cmake	2.6.2-1ubuntu1
g++	4:4.3.3-1ubuntu1
gsoap	2.7.9l-0.2
libboost-filesystem1.37-dev	1.37.0-3ubuntu3
libboost-system1.37-dev	1.37.0-3ubuntu3
libical-dev	0.43-2ubuntu1
libmozjs-dev	1.8.1.16+nobinonly-0ubuntu1
libpoco-dev	1.3.3p1-2
libxml2-dev	2.6.32.dfsg-5ubuntu4.2

„APT (Advanced Package Tool) ist ein Paketmanagement-System, das im Bereich des Betriebssystems Debian GNU/Linux entstanden ist. Mittels APT ist es sehr einfach, Programmpakete zu suchen, zu installieren oder auch das ganze System komplett auf den neuesten Stand zu bringen.“

(Wiki, apt-get: <http://wiki.ubuntuusers.de/apt-get> [08.09.2009])

Also hatten wir durch Zufall unsere Notebooks mit der richtigen Linuxversion ausgerüstet. Nach diesen Neuigkeiten trafen wir uns erneut, um die notwendigen Pakete zu installieren. Unser Ubuntu sollte so aufgerüstet werden, dass der dSS gebildet werden konnte. Nach den Downloads starteten wir dann das Bildungsprogramm mit dem CMAKE-2.6.4. Der Server begann sich zu bilden, bei 98% kam jedoch der Fehler.

```
-- WARNING: you are using the obsolete 'PKGCONFIG' macro use  
FindPkgConfig
```

```
--
```

```
Configuring done
```

```
-- Generating done
```

```
-- Build files have been written to: /home/
```

```
[ 15%] Built target tests
```

```
[ 58%] Built target core
```

```
[ 65%] Built target unix
```

```
[ 71%] Built target webservices
```

```
[ 96%] Built target shttpd
```

```
[ 98%] Building CXX object CMakeFiles/dss.dir/main.cpp.o
```

```
cclplus:
```

```
warnings being treated as errors
```

```
/home//dss/main.cpp: In function »int main(int, char**)«:
```

```
/home//dss/main.cpp:164:
```

```
Fehler: veraltete Konvertierung von Zeichenkettenkonstante in  
»char*«
```

```
Fehler: veraltete Konvertierung von Zeichenkettenkonstante in
»char*«
```

```
/home//dss/main.cpp:164:
```

```
Fehler: veraltete Konvertierung von Zeichenkettenkonstante in
»char*«
```

```
make[2]: *** [CMakeFiles/dss.dir/main.cpp.o]
```

```
Fehler 1
```

```
make[1]: *** [CMakeFiles/dss.dir/all]
```

```
Fehler 2
```

```
make: *** [all] Fehler 2
```

Wir konnten der Fehlermeldung entnehmen, dass „eine veraltete Konvertierung von der Zeichenkettenkonstante im >>char\*<<“ vorhanden war. Wir versuchten den Fehler mit anderen Daten zu eliminieren, doch das Bilden des Servers stoppte wieder an derselben Stelle.

Wir waren uns sicher, dass das Programm in ein Unterprogramm greift, das nicht im Fehler beschrieben ist. Erneut benötigten wir professionelle Hilfe. Wir fragten bei der aizo um ein neues main.cpp an, denn wir vermuteten, dass sich durch die unzähligen manuellen Änderungen etwas zusätzlich verändert hatte, da wir in unseren Tests oft Daten verschoben, kopiert oder extrahiert hatten.

Die aizo liess uns einen neuen main.cpp zukommen, wodurch das Problem aber nicht gelöst werden konnte. Es lag folglich nicht an main.cpp. Wir kontaktierten Johannes Winkelmann, der direkt mit dem Server der aizo zu tun hat. So bekamen wir einen Termin mit ihm und konnten das Problem vor Ort zeitsparend und effizient angehen, was angesichts unseres zeitlichen Drucks entscheidend war. Wir hofften, den Fehler möglichst schnell zu identifizieren und zu beheben mittels der Hilfe von Johannes Winkelmann und so den Server einsatzfähig zu machen.

```

-- WARNING: you are using the obsolete 'PKGCONFIG' macro use
FindPkgConfig
-- Configuring done
-- Generating done
-- Build files have been written to: /home//dss/build
[ 15%] Built target tests
[ 58%] Built target core
[ 65%] Built target unix
[ 71%] Built target webservices
[ 96%] Built target shhttpd
[ 98%] Building CXX object CMakeFiles/dss.dir/main.cpp.o
[100%] Building CXX object CMakeFiles/dss.dir/namespaces.cpp.o
Linking CXX executable dss
[100%] Built target dss

DSS v0.4a2 released at 20090622
[2009-09-03 12:08:54][Debug] Starting Webserver...
[2009-09-03 10:08:54][Info] DSS stating up....
[2009-09-03 10:08:54][Info] Loading config
I/O warning : failed to load external entity "data/config.xml"
Error loading properties from "data/config.xml"
[2009-09-03 10:08:54][Fatal] Could not parse config file
~/dss/build$

```

### 3.1.4.3 Fehler Bereinigt

Unser Hoffen wurde erhört und wir trennten uns von Johannes Winkelmann mit einem funktionierenden dSS. Dadurch, dass wir das `#ifdef WITH_TESTS` nicht starteten, wurde auch der Server fertig gebildet. Wir passten den folgenden Abschnitt mit `aizo` an, um allfällige versteckte Fehler auszuschliessen.

```

#ifdef WITH_TESTS
    cout << "compiled WITH_TESTS" << endl;
    if(runTests) {
        cout << "running tests" << endl;
        const char* params[3] = {"dss", "--report level=detailed", "--
log level=all"};
        ::boost::unit_test::unit_test_main( &init_unit_test, 1,
const_cast<char**>(params) );
        cout << "done running tests" << endl;
    }
}

```

Nach vollständigem Aufbau des DSS und Bereinigung der Programmierfehler, konnten wir uns folgenden Aufgaben zur Verwirklichung unseres Projektes widmen:

- 1. Eine Verbindung der aizo Komponenten zu unserem Server erstellen.
- 2. Die Daten der dSID (MAC-Adresse), dsM (Zonen) auswerten.
- 3. Die Daten mit einer Datenbank verknüpfen)

### 3.1.5 Verbindung zwischen Server - Komponenten

#### 3.1.5.1 Erster Verbindungsversuch

Den ersten Versuch einer Verbindung zwischen dSS und dsM starteten wir am Mittwoch 02.09.2009. Wir verwendeten dazu den schon vorhandenen EXSYS Konverter EX-6051IS, mit dem wir schon ganz am Anfang versuchten hatten, brauchbare Daten zu erhalten.

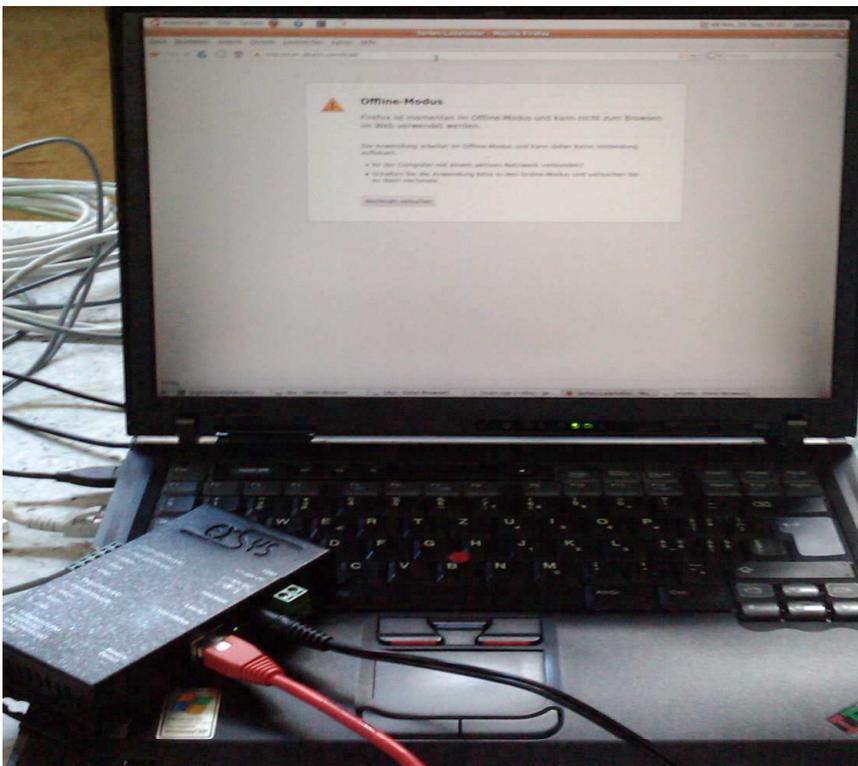
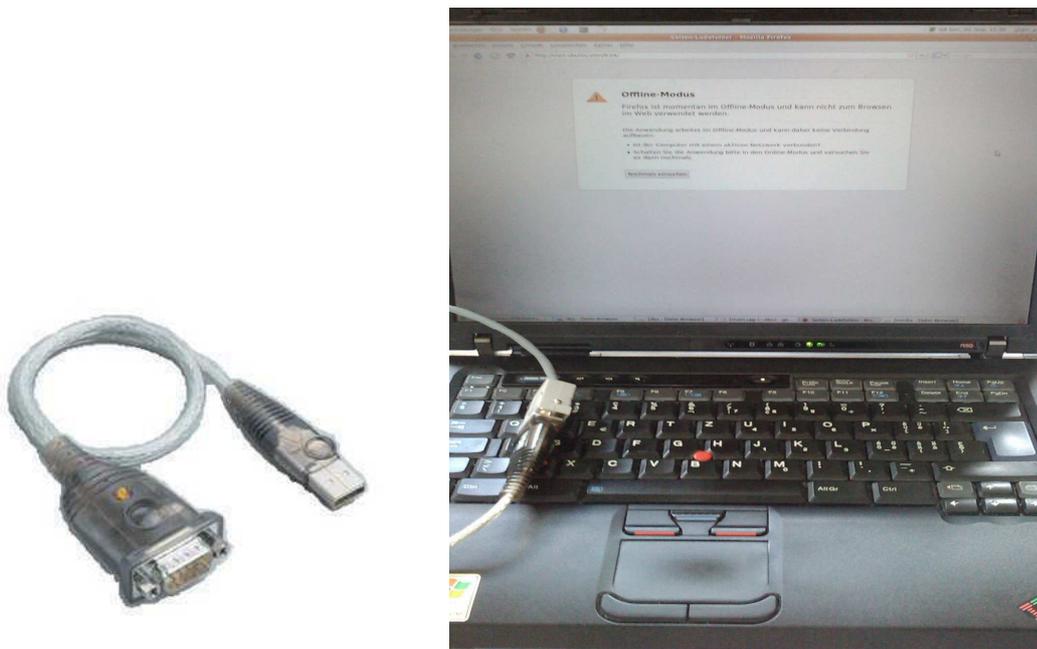


Abb. 17: Konverter EX-6051IS mit Server (Notebook)

Leider wurden wir mit dem Kompatibilitätsproblem zwischen Windows und Linux konfrontiert, da der Konverter natürlich nicht Linux tauglich war. Wir konnten den Konverter zwar über Linux (Ubuntu) verbinden, updaten und pingen, es gab trotzdem eine Fehlermeldung, die die Information enthielt, dass der Port 8080 vom Localhost nicht geöffnet werden kann.

### 3.1.5.2 Zweiter Verbindungsversuch

So starteten wir den zweiten Versuch mit einem USB zu RS232 Konverter Typ UC-232A auch vom Hersteller EXSYS.



**Abb. 18: Konverter Typ UC-232A mit Server (Notebook)**

Leider war auch dieser Konverter nicht Linux kompatibel und unserer Bemühungen einer Verbindung herzustellen blieben erfolglos.

### 3.1.5.3 Dritter Verbindungsversuch

Wir mussten einen USB/RS485 Konverter beschaffen, der das Protokoll des Digitalstromservers (dSS) akzeptiert. Wir entschieden uns für den EXSYS EX-1303, der einen funktionierenden Treiber für Linux (Ubuntu) hat.

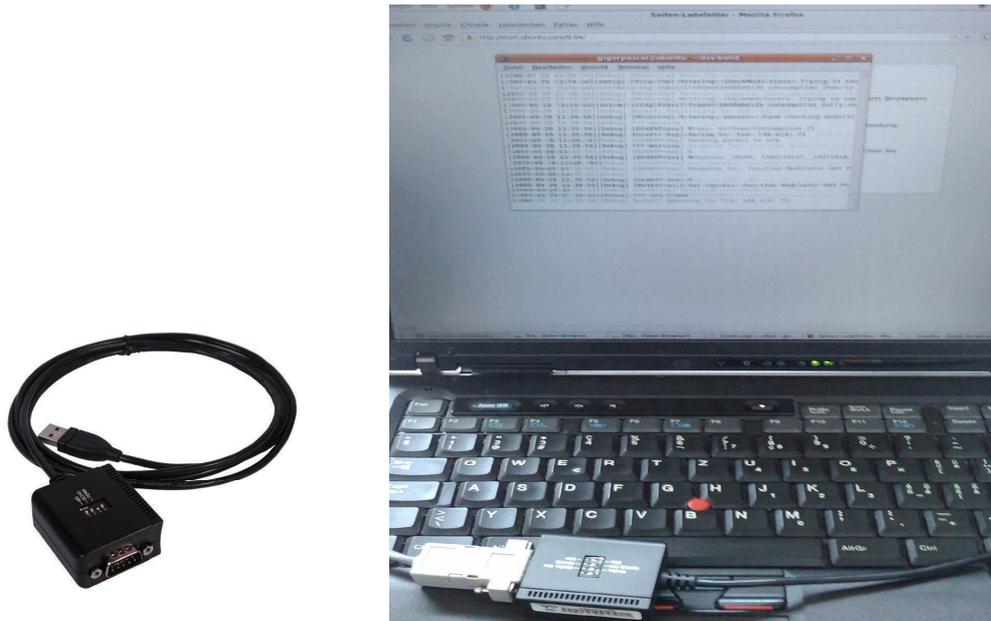


Abb. 19: USB/RS485 Konverter EXSYS EX-1303 mit Server (Notebook)

Als wir den Treiber installiert hatten, konnten wir eine erfolgreiche Kommunikation zwischen diesem Konverter EX-1303 und dem Server herstellen. Im Index des Localhost wurden dann die Devices (Bausteine) sichtbar. Wir sahen, dass beispielsweise die Zone 321 angezeigt wurde und die darunter angemeldeten Chips (dSID).

### 3.1.5.4 Zu wenige Ressourcen

Durch die beanspruchte Leistung des dSS vom PC konnten nicht alle Programme ausgeführt werden. Dem dSS wurden für die Verarbeitung der Kommunikationsdaten zu wenige Ressourcen vom PC zur Verfügung gestellt. Dadurch entstand ein Daten Over Flow, wodurch der localhost 8080 nicht angezeigt werden konnte.

Deshalb änderten wir den Befehl (./dss <dss6.log>), so dass der DSS nicht mehr im Terminal visualisiert wird, sondern in ein LOG-File angelegt wird.

```
DSS v0.4a2 released at 20090622
[2009-09-04 15:15:10][Debug] Starting Webserver...
[2009-09-04 13:15:10][Info] DSS stating up....
[2009-09-04 13:15:10][Info] Loading config
[2009-09-04 13:15:10][Debug] [DS485Proxy] Logchannel created
[2009-09-04 13:15:10][Debug] [Apartment] Logchannel created
[2009-09-04 13:15:10][Debug] [WebServer] Logchannel created
[2009-09-04 13:15:10][Debug] [WebServices] Logchannel created
[2009-09-04 13:15:10][Debug] [DSSim] Logchannel created
[2009-09-04 13:15:10][Debug] [DSSim] LoadZones: found zone (1)
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: found device
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: it's a switch
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: found device
[2009-09-04 13:15:10][Debug] [DSSim] LoadZones: found zone (1)
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: found device
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: found device
[2009-09-04 13:15:10][Debug] [DSSim] LoadGroups: Adding device 4 to
group 1 in zone 1
[2009-09-04 13:15:10][Debug] [DSSim] LoadZones: found zone (4)
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: it's a switch
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: found device
[2009-09-04 13:15:10][Debug] [DSSim] LoadDevices: found device
[2009-09-04 13:15:10][Debug] [DSSim] LoadGroups: Adding device 5 to
group 1 in zone 4
[2009-09-04 13:15:10][Debug] [EventInterpreter] Logchannel created
[2009-09-04 13:15:10][Debug] [Metering] Logchannel created
[2009-09-04 13:15:10][Debug] [FakeMeter] Logchannel created
[2009-09-04 13:15:10][Debug] creating thread for "DS485Controller"
[2009-09-04 13:15:10][Info] DS485Controller::execute: Opening
'/dev/ttyUSB0' as serial device
[2009-09-04 13:15:10][Debug] creating thread for "DS485Proxy"
[2009-09-04 13:15:10][Debug] creating thread for "Apartment"
[2009-09-04 13:15:10][Debug] creating thread for "WebServer"
[2009-09-04 13:15:10][Debug] creating thread for "WebServices"
[2009-09-04 13:15:10][Debug] creating thread for "EventInterpreter"
[2009-09-04 13:15:10][Debug] [Metering] Writing files to:
data/webroot/metering/
[2009-09-04 13:15:10][Debug] creating thread for "Metering"
[2009-09-04 13:15:10][Debug] [FakeMeter] Writing files to:
data/webroot/metering/
[2009-09-04 13:15:10][Debug] creating thread for "FakeMeter"
[2009-09-04 13:15:10][Debug] creating thread for "BonjourHandler"
[2009-09-04 13:15:10][Debug] EventInterpreter: Loading subscrip-
tions from 'data/subscriptions.xml'
[2009-09-04 13:15:10][Debug] [WebServer] Webserver: Listening on
port(s) 8080
```

```
[2009-09-04 13:15:10][Debug] [WebServer] Webserver: Configured aliases: /=data/webroot/
[2009-09-04 13:15:10][Info] [WebServer] Webserver started
[2009-09-04 13:15:10][Debug] Destroying thread: BonjourHandler
[2009-09-04 13:15:10][Info] [Apartment] Apartment::execute: Waiting for interface to get ready
[2009-09-04 13:15:10][Debug] Sensed traffic on the line, changing to csSlaveWaitingToJoin
?????????S$** Waiting for 19
```

Nun konnte der dSS gestartet werden. Da auch die nötigen Ressourcen vorhanden waren, um den Webserver, den dSS und die Konfigurationsdateien zu laden. Im nächsten Befehl starteten wir die Verknüpfung mit der seriellen Schnittstelle.

```
DS485Controller::execute: Opening '/dev/ttyUSB0' as serial device
```

### 3.1.5.5 Willkommen zum digitalSTROM-Server

Durch das Aufrufen des localhost:8080/ im Webbrowser(Firefox) öffnete sich folgendes Hauptfenster:

## Welcome to the digitalstrom-server

[Config](#) [Bell](#)

Connected devices: 9

- Zone 1 [[Turn On](#) | [Turn Off](#)]
  - (No Name) 4 Off [[Turn On](#) | [Turn Off](#)]  
3504175fe0000000ffc00011
  - (No Name) 1 Off [[Turn On](#) | [Turn Off](#)]  
3504175fe0000000ffc00004
  - (No Name 3) Off [[Turn On](#) | [Turn Off](#)]  
3504175fe0000000ffc00006
- Zone 321 [[Turn On](#) | [Turn Off](#)]
  - Schalter
  - (No Name 2) Off [[Turn On](#) | [Turn Off](#)]  
3504175fe0000000ffc00005
- Zone 2 [[Turn On](#) | [Turn Off](#)]

Zum ersten Mal sahen wir die Kommunikation zwischen dSID und dSM nicht nur als Pik auf dem Oszilloskop sondern auch als MAC-Adressen.

### 3.1.6 Auswertung der Daten

Nun mussten wir herausfinden, welche Daten wir benötigten, um die notwendigen Informationen für unser Projekt zu erhalten. Am Dienstag 08.09.2009 hatten wir erneut einen Termin mit dem Softwareentwickler bei der aizo AG in Schlieren. Zu den Hardwarebauteilen (10 dSID's & 2 dSM), die wir erhielten, konnte Johannes Winkelmann uns genau sagen, welche Daten für uns relevant sind und welche nicht. So wussten wir nun, wo die einzelnen ID's in der Software angelegt werden.

```
</device>
<device dsid="3504175fe0000000ffc00004">
<name>Sim Lampe 1</name>
<location>
<x>1.1</x>
<y>2.5</y>
<z>0</z>
</location>
</device>
<device dsid="3504175fe0000000ffc00006">
<name>Sim Lampe 3</name>
<location>
<x>0</x>
<y>0</y>
<z>0</z>
</location>
</device>
```

Wir sahen nun, wie sich die ID's (dSID) an die zugehörigen Meter (dSM) anmeldeten. So konnten wir feststellen, an welcher Steckdose der beiden Demokoffern der dSID eingesteckt ist.



**Abb. 20: Zwei Demokoffer mit Zone 315/321**

Anschliessend wollten wir versuchen, die einzelnen dSID's anders zu definieren, so dass wir gerätespezifische Informationen zuordnen konnten. Weil sich jeder ID (dSID), sobald man ihn an das Stromnetz anschliesst am nächsten Meter (dSM) anmeldet und das auch direkt in der Software ersichtlich ist, mussten wir ihn auch da definieren. In der Software konnten wir dann jeden zusätzlichen dSID umprogrammieren und gerätespezifische Informationen dem jeweiligen dSID zuordnen.

### **3.1.7 Datenverknüpfung**

Das Erstellen einer vollständigen Webbasierenden Plattform würde den Zeitrahmen in dieser Arbeit sprengen. Aus diesem Grund haben wir an einer vorhandenen Plattform Änderungen vorgenommen, um eine Demoversion zu präsentieren, welche auf der Programmiersprache xml aufgebaut ist. Für den Markt ist diese Variante sicherlich zu umständlich. Die Lösung dieses Problems wird ausserhalb dieser Arbeit eine wichtige Aufgabe sein und massgeblich zum Erfolg der Markteinführung beitragen.

### 3.1.7.1 Änderungen an der Plattform

Folgende Änderungen haben wir vorgenommen:

- Im nächsten Programmskriptabschnitt konnten wir den Hintergrund ändern und die Ausgabefenster der Suchergebnisse definieren.
  - Mit `border` wird die Umrandung des Feldes festgelegt
  - `width` bestimmt die Feldgrösse
  - `list-style` beschreibt die Art und Weise der Auflistung.

```
body {  
    margin:80;  
    padding:0;  
}
```

```
ul.drag {  
    border: 3px solid green;  
    width: 250px;  
    list-style: none;  
    margin:20;  
    padding:2;
```

- Wenn die angezeigte Liste leer ist (empty), wie im nächsten Programmskriptabschnitt wird das Feld auf die minimale Grösse verkleinert.

```
    /*  
    The bottom padding provides the cushion that makes the  
empty list targetable. Alternatively, we could leave the padding  
off by default, adding it when we detect that the list is  
empty.  
    */  
    padding-bottom:20px;  
}
```

- Durch folgenden Programmskriptabschnitt wird das Suchfunktionfeld erstellt. Dies ermöglicht die spezifische Suche nach bestimmten dSID's.

Mit den Befehlen:

```
<div class="container">
```

- Wird der Hintergrundkasten definiert.

```

```

- Wird die Quelle des Bildes, die Position sowie die Grösse definiert.

```
&nbsp;   <h2> Gerät: <input type="text" id="deviceName"
value="Name" /> </h2>
```

- Wird das Suchfeld mit dem Text definiert.

```
<h1><input type="button" id="findButton" value="Suchen"
/> </h1>
```

- Wird der Suchknopf mit beinhaltendem Text definiert.

```
</div>
<div class="container fl">
<h2> <div id="other_zones">(loading...) </div> </h2>
</div>
```

- Wird die gesuchte dsID in der entsprechenden Zone/Raum angezeigt.

- Benutzeroberfläche optimieren

In diesem Abschnitt wurde der Text formatiert, der Situationsplan des Gebäudes erstellt und Firmenlogos eingefügt.

```
</div>
<div class="container">
<h1>Building</h1>
<p>

<span>Sponsored:</span>


```

```





</div>

```

- Mit dem Befehl `` wird das gewünschte Bild geladen und auch in der gewünschten Grösse (`width="100"`) angezeigt.

### 3.1.7.2 Das Suchportal

Durch die Eingabe der Webadresse `http://localhost:8080/inv.html` präsentiert sich nun unser Infranetviewer-System Websuchportal. Dieses Portal sah, nach unseren Getätigten, wie im oberen Abschnitt beschriebenen, Änderungen wie folgt aus:

Das Infranetviewer-System Suchportal



Abb. 21: Hauptseite des Infranetviewer-Systems

Auf dieser Seite kann man nun durch Eingabe in das Suchfenster das gewünschte Gerät suchen. Sofern dieses Gerät am Stromnetz angeschlossen ist, wird es in dem angemeldeten Raumfenster (Nord A15) angezeigt werden. Diese Möglichkeit haben alle PCs, die sich an dem Netzwerk befinden. Das Netzwerk kann natürlich auch mit Wireless ausgestattet sein, so könnte man mit mobilen Geräten wie Handy oder Notebook auch unterwegs die Daten vom Websuchportal abrufen und Änderungen des gesuchten Gerätestandortes feststellen.

### 3.1.7.3 Aufbau der Demoanlage

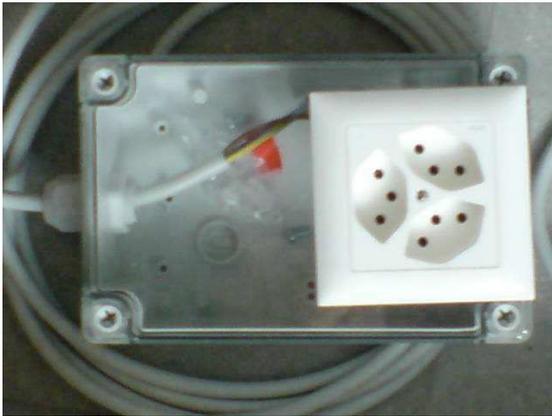
Damit wir unser Projekt präsentieren können, konstruierten wir eine Demoanlage, die aus folgenden Komponenten besteht:

Einem Sicherungstableau, wie es in jedem Gebäude vorkommt.

Im unteren Bereich des Tableaus befinden sich die Klemmen, an welchen die einzelnen Stromkabel angeschlossen werden. Von dort werden dann die aizo Meter dSM (im Beispiel von Bild 22 deren 2) und die FI's (in Bild 22 ebenfalls zwei) verdrahtet. Bei unserer Demoanlage definiert ein dSM einen Raum, so können wir also zwei verschiedene Räume simulieren, in denen sich mobile Geräte dSID's befinden können. Diese Räume sehen bei unserer Demoanlage folgendermassen aus. (Bild 23)



**Abb. 22: Tableau**



Raum eins und Raum zwei werden beide mit eine 3er Steckdosenbox simuliert. Wenn man ein mobiles Gerät (dSID) in einer Steckdose einsteckt, meldet sich der dSID am Meter (dSM) an. Der dSID wird einfach durch eine Klemme in ein mobiles Gerät eingebaut.

**Abb. 23: Raum 1 und Raum 2**



**Abb. 24: dSID in Klemme und eingebaut**

Die ganze Demoanlage ist so nun jeder Zeit einfach transportier-und demonstrierbar. Da wir den Server auf einem Notebook haben, können wir nur noch den Konverter für die Datenübertragung an den Server und den dSM anschliessen und schon kann in unserem Demo P-LAN nach mobilen Geräten gesucht werden.

## 4 Schlussfolgerung

Unser Projekt der Entwicklung des Geräteortungssystems Infranetviewer-System, stellte uns vor drei grundlegende Aufgaben. In einem ersten Schritt fanden wir in der aizo Technologie die ideale Hardware. Dies stellte uns vor verschiedene grundsätzliche Herausforderungen. Zunächst galt es unser Projekt professionell zu vertreten, um die gewünschte Technologie von der Firma zur Verfügung gestellt zu bekommen. Des Weiteren handelte es sich bei der Technologie um einen Prototypen, der sich nach wie vor in Entwicklung befand und daher nicht problemlos einsetzbar war, was uns vor die zweite grundlegende Aufgabe innerhalb des Projektes stellte. Obwohl wir auf dem Gebiet der Server Bildung auf Linux Basis keine Experten waren, gelang uns auch dank professioneller Unterstützung die Serverbildung. Damit war die Voraussetzung zur erfolgreichen Realisierung unseres Projektes mittels Erarbeitung einer Datenbank und Erstellung einer Demoanlage geschaffen. Das Infranetviewer-System wurde zum Leben erweckt.

Dieses Projekt wäre zum Scheitern verurteilt gewesen, wenn wir nicht gemeinsam an einem Strick gezogen hätten. In dieser kurzen Zeit alleine ein funktionierendes Geräteortungssystem zu erschaffen, wäre aus zeitlichen Gründen nicht möglich gewesen. Die Aufteilung der Arbeit, ermöglichte es uns, länger an einer Idee festzuhalten und nicht sofort auf eine Alternative zurückgreifen zu müssen.

Das Infranetviewer-System reagiert auf die bestehenden Bedürfnisse zur problemlosen Ortung von Geräten in Krankenhäusern, welche bis anhin von keinem System abgedeckt werden. Obwohl es sich bei unserem Projekt um einen unveröffentlichten Prototypen handelt, stiess das Infranetviewer- System bereits auf reges Interesse. Verschiedene Krankenhäuser haben bereits Interesse an einer Testanlage angemeldet.

Aus diesem Grund wird es notwendig sein, in einer zukünftigen Weiterentwicklung die Datenbank ausführlicher und dadurch bedienungsfreundlicher zu gestalten. Die Möglichkeit die dSID mit einem zusätzlichen Netzwerk wie beispielsweise GPRS zu kombi-

nieren ist auch noch nicht ausgeschöpft. Des Weiteren wäre eine Standardisierung durch die MepV denkbar, wobei alle mobilen medizinischen Geräte mit einem dSID ausgerüstet werden müssten. Es gelang uns mit dieser Arbeit den Grundstein zur Etablierung des Infranetviewer-Systems zu schaffen, einer allfälligen Markteinführung steht prinzipiell nichts mehr im Wege.

# 5 Verzeichnisse

## 5.1 Abbildungsverzeichnis

Abb. 1: Verbuchung von Büchern mit RFID-Chips.....	7
Abb. 2: Universelles RFID Handlesegerät .....	7
Abb. 3: Kommunikationsschema des INVS .....	12
Abb. 4: Zwei Devoloadapter einer angeschlossen.....	13
Abb. 5: Ameise mit aizo Technologie.....	14
Abb. 6: dSID-digital Strom Identität .....	16
Abb. 7: Kommunikationstest zwischen dSID und dSM .....	18
Abb. 8: Pik auf steigender Sinuskurve .....	19
Abb. 9: Pik auf Sinuskurve .....	19
Abb. 10: Operationsraum Simulation.....	20
Abb. 11: Aufbau mit Netzfiltern .....	21
Abb. 12: Aufbau mit Dokingstation .....	22
Abb. 13: Aufbau mit allen Komponenten .....	23
Abb. 14: Print Screen Steckbrief.....	28
Abb. 15: dSS-digital Strom Server.....	29
Abb. 16: Zwei Konverter EX-6051IS.....	30
Abb. 17: Konverter EX-6051IS mit Server (Notebook).....	40
Abb. 18: Konverter Typ UC-232A mit Server (Notebook).....	41
Abb. 19: USB/RS485 Konverter EXSYS EX-1303 mit Server (Notebook) .....	42

Abb. 20: Zwei Demokoffer mit Zone 315/321.....	46
Abb. 21: Hauptseite des Infranetviewer-Systems .....	49
Abb. 22: Tableau.....	50
Abb. 23: Raum 1 und Raum 2.....	51
Abb. 24: dSID in Klemme und eingebaut.....	51

## 5.2 Literaturverzeichnis

- aizo, Server: [http://www.aizo.com/de/produkte/prod\\_dSS.php](http://www.aizo.com/de/produkte/prod_dSS.php) [02.09.2009]
- aizo, Vorteile: [http://www.aizo.com/de/technologie/tech\\_vorteile.php](http://www.aizo.com/de/technologie/tech_vorteile.php) [02.09.2009]
- Bärwolff et al. IT-Systeme in der Medizin, Wiesbaden: Vieweg + Teubner, 2006.
- Lauer, Oliver. Wireless Patient Monitoring. Anforderungen und Perspektiven. ETH Zürich (Unveröffentlichtes Manuskript), 2008.
- Wiki, API: <http://de.wikipedia.org/wiki/Programmierschnittstelle> [07.09.2009]
- Wiki, apt-get: <http://wiki.ubuntuusers.de/apt-get> [08.09.2009]
- Wiki, RFID: [http://de.wikipedia.org/wiki/Radio\\_Frequency\\_Identification](http://de.wikipedia.org/wiki/Radio_Frequency_Identification) [20.08.2009]
- Wiki, SOAP: <http://de.wikipedia.org/wiki/SOAP>[07.09.2009]
- Wiki, VLC: [http://de.wikipedia.org/wiki/VLC\\_media\\_player](http://de.wikipedia.org/wiki/VLC_media_player)[07.09.2009]

Hiermit bestätigen wir, dass wir vertraut sind mit den von der Medizintechnikerschule Sarnen herausgegebenen, Regeln zur Sicherung wissenschaftlicher Redlichkeit und diese gewissenhaft befolgt haben.

Zürich, 26.09.2009

Michael Lauber

Pascal Giger