



1. Inhaltsverzeichnis

1.	INHALTSVERZEICHNIS.....	2
2.	EINLEITENDE BEMERKUNGEN.....	3
3.	PROJEKTDOKUMENTATION	3
3.1.	PROJEKTENTSTEHUNG UND –PLANUNG	3
3.1.1.	DIE IDEE	3
3.1.2.	BEEINFLUSSUNGEN & ZIELE	3
3.1.3.	ZEITPLANUNG.....	5
3.1.4.	AUFGABENVERTEILUNG.....	5
3.2.	DIE REALISIERUNG	6
3.2.1.	AUSWAHL DER HARDWAREKOMPONENTEN	6
3.2.1.1.	DER XMEDIA PLAYER.....	6
3.2.1.2.	DIE XREMOTE	8
3.2.2.	AUSWAHL DER SOFTWARE	9
3.2.3.	DER HARDWAREAUFBAU	10
3.2.3.1.	XMEDIA PLAYER	10
3.2.3.2.	XREMOTE	11
3.2.4.	DIE FIRMWARE	11
3.2.4.1.	GEMEINSAME MODULE.....	11
3.2.4.2.	DER BOARDCONTROLLER.....	11
3.2.4.3.	DIE XREMOTE	12
3.2.5.	DIE SYSTEMTREIBER.....	12
3.2.6.	DIE SOFTWARE.....	12
3.2.7.	DAS GEHÄUSE.....	13
3.3.	STAND DER ENTWICKLUNG	14
3.4.	INNOVATION	16
3.4.1.	XDIMENSION - DIE INNOVATION	16
3.4.2.	DER BESCHLEUNIGUNGSSENSOR – DAS NEUE EINGABEGERÄT.....	17
3.5.	ZUSAMMENARBEIT IM TEAM	18
4.	ANHANG.....	19
4.1.	STROMLAUFPLÄNE.....	19
4.2.	BILDER	19
4.3.	VIDEOS	19
4.4.	PROJEKTLOGOS.....	19
4.5.	ZUSATZINFORMATIONEN.....	19
5.	ABBILDUNGSVERZEICHNIS	20

2. Einleitende Bemerkungen

Dieser Projektbericht soll nur einen oberflächlichen Einblick in das xMedia Projekt geben, denn die Komplexität des Projektes würde den Rahmen des Berichtes, der sich auf 20 Seiten begrenzen soll, deutlich sprengen. Sollten Sie zu einem der im folgenden Bericht genannten Punkte nähere Informationen wünschen, stehen wir Ihnen jederzeit unter der E-Mail Adresse bwa@berniwa.com zur Verfügung.

Nähere Informationen zum xDimension –der kleinen Innovation unseres Projektes – finden Sie im Kapitel „xDimension – Die Innovation“

Weitere Neuigkeiten, Bilder und Videos zu unserem Projekt finden Sie unter <http://www.project-xmedia.com>

3. Projektdokumentation

3.1. Projektentstehung und –planung

3.1.1. Die Idee

Die Projektidee, zumindest für den Kern des Projektes, wurde durch den Teamleiter Bernhard Wörndl-Aichriedler fast ein Jahr vor dem Projektstart ins Auge gefasst:

„Bei meinem Praktikum als Entwickler in einem größeren Betrieb, kamen meinen damaligen Kollegen und ich immer wieder 32bit Prozessoren zu sprechen, welche für mich damals noch in ungreifbarer Ferne schienen. Kurz vor dem Ende meines Praktikums wagte ich jedoch dennoch den Schritt, bestellte mir zwei Referenzboards bestückt mit dem, vor wenigen Wochen zuvor erschienen AP7000 von Atmel. Der Grund, warum ich 2 bestellte, lag darin, dass ich erwartete eines nach kurzer Zeit zu zerstören. Doch es kam alles anders. Die Leistung und vor allem die Komplexität des Systems übertrafen meine Erwartungen bei weitem, wodurch das Board damals für mich an Interesse verlor und ich es für ein Jahr beiseite legte um es im vierten Jahrgang wieder neu zu entdecken. Die eigentliche Idee kam dann von meinem Vater, der ein Gerät suchte, das er in sein bestehendes Heimkinosystem integrieren konnte, welches es ihm ermöglicht, Musik aus dem Netzwerk abzuspielen. Da wir zu Hause jedoch nicht über kabelgebundene Netzwerkinfrastruktur verfügten, musste das Gerät W-Lan fähig sein. Nachdem ich mit Alexander und Michael die idealen Projektpartner fand, um mich bei meinem Vorhaben zu unterstützen. Gegen Ende des vierten Jahrganges begann die Grobplanung des Projektes. Da wir jedoch mehr wollten beschlossen wir neben dem selbst Media Player, noch eine Fernsteuerung zu entwickeln. Ab diesem Zeitpunkt stand der Name fest.-xMedia

Dies sollte das nächste 3/4 Jahr unseres Lebens bestimmen...“

3.1.2. Beeinflussungen & Ziele

Ziel der Projektarbeit ist es eine Multimedia Station zu entwickeln, die es ermöglicht Musikdateien und gegebenenfalls auch Streams wiederzugeben. Im Mittelpunkt des Projektes steht eindeutig die Benutzerinteraktion, bei welcher neue Maßstäbe gesetzt werden sollten, die sich deutlich vom „old school“ Taster entfernen.

Die Projektziele wurden zunächst eher etwas lose formuliert, da keiner von uns Erfahrung mit solchen Systemen hatte. Sogar der wichtige Punkt der Implementation von Wireless Lan wurde nicht offiziell als Projektziel definiert. Wir setzten die Erwartungen mit unserer Zieldefinition bewusst nicht zu hoch an, da wir einfach nicht wussten ob unsere Ziele

bzw. mehr unsere Träumen technisch realisierbar wären. Jedoch genau dies gab uns im Laufe des Projektes die Freiheit Ideen und aktuelle Technologien einfließen zu lassen.

Da das Projekt zunächst nicht für den kommerziellen Einsatz gedacht war, sahen wir den Markt im Bereich der Unterhaltungselektronik eigentlich nicht als Konkurrenz, sondern vielmehr als Ideenquelle. So warfen wir etwas genauere Blicke auf die Playstation Portable von Sony, auf die Nintendo Wii und natürlich auch auf den Apple iPod und das neue iPhone.



Abbildung 1: Apple I-Pod

Der iPod, für viele Menschen ein Lifestylesymbol und fast sogar schon ein Modeaccessoire. Die einzigartige Steuerung über das "iPod Wheel" kombiniert mit einfachster Menüführung vermittelt dem Benutzer eine optimale User Experience und ermöglicht eine optimale Steuerung des Gerätes. Genau dieses Konzept wollten wir aufgreifen und in unsere Hardware implementieren.

Die Playstation Portable ist ein Muss für jeden Spieler. Sie besticht vor allem durch ihre W-Lan Fähigkeiten und ihrem riesigen, brillanten Display. Auch diese Fähigkeiten kamen auf die Wunschliste unseres Projekts.



Abbildung 2: Sony PSP



Abbildung 3: Nintendo Wii

Nicht nur aus der Sicht eines Gelegenheitsspielers, sondern auch aus technischer Sicht setzt die Nintendo Wii neue Maßstäbe. Die größte technische Leistung ist jedoch nicht die Konsole selbst (der es deutlich an Leistung mangelt), sondern der Controller – die Wiimote. Auch einige Eigenschaften der Wiimote kamen auf unsere Wunschliste.

Im November 2007 erblickte das iPhone das Licht der Welt. Wie viele Entwicklungen von Apple setzte auch das iPhone einen neuen Trend mit dem Namen „Multitouch“. So wurde dieses Konzept auch auf unsere Wunschliste gesetzt.



Abbildung 4: Apple iPhone

3.1.3. Zeitplanung

Da das Projekt vom Arbeitsaufwand das Stundenpensum eines „normalen“ Maturaprojektes bei weitem überstiegen hätte, hat Bernhard bereits im Juni 2008 mit einer Analyse über die Realisierbarkeit der Punkte unserer Wunschliste begonnen.

Juni 2008	...inoffizieller Projektstart (Machbarkeitsanalyse)
August 2008	...Fertigstellung des ersten Prototypen
September 2008	...offizieller Projektstart
Dezember 2008	...Vollständiger Test der Hardware ...Funktionsfähige Systemtreiber
Januar 2009	...Fertigstellung der finalen Hardware
März 2009	...Fertigstellen des Gehäuses
April 2009	...Fertigstellen der ersten Version der Software
Juni 2009	...Projektabschluss

3.1.4. Aufgabenverteilung

Da der Projektleiter Bernhard bereits fundierte Programmierkenntnisse, sowie einige Erfahrung bei Hardwareentwicklung hatte, ist seine Hauptaufgabe der Hardwaresektor sowie die Low-Level Programmierung auf Firmware bzw. Treiberebene.

Die Entwicklung des Gehäuses wird aufgrund dementsprechender Erfahrungen von Alexander erledigt. Da Bernhard durch die Entwicklung der Hardware sehr weit ausgelastet ist, wurden Alexander bereits beim Projektstart diverse Verwaltungstätigkeiten anvertraut.

Michaels Aufgabenbereich fand sich in der Entwicklung bzw. Weiterentwicklung der Hardware, der Firmware für die Fernsteuerung, sowie die Fertigung des Gehäuses.

3.2. Die Realisierung

3.2.1. Auswahl der Hardwarekomponenten

Nun ging es an die Umsetzung des Projektes und vor allem an die Machbarkeitsanalyse unserer Ideen.

Zunächst teilten wir unser Projekt grob in zwei Gruppen ein, den xMedia Player und die xRemote. Der xMedia Player ist die Multimediastation selbst, während die xRemote die Fernsteuerung des Projektes ist. Im Laufe des Projektes kam noch ein dritter Teil, hinzu mit dem Namen xDimension, welcher von unserem Projektleiter als kleine Überraschung gedacht war.

3.2.1.1. Der xMedia Player

Nun zur technischen Umsetzung des Hauptgerätes, dem xMedia Player.

Der erste Schritt, die Wahl eines richtigen Prozessors fiel uns nicht schwer, da Bernhard bereits zwei Referenzboards, mit dem Atmel AP7000 bestückten NGW100, besaß, wählten wir diesen Prozessor. Dieser Prozessor gilt als ideal geeignet für multimediale Applikationen mit wenig Stromverbrauch und spielt in der Leistungsklasse eines ARM9 oder sogar darüber. Er besitzt einen 32bit RISC Kern und viele Hardwaremodule, wie zwei SD/SDHC/SDIO Interfaces, ein LCD Interface, eine AC97 Audio Interface als auch einen Bitstream Audio DAC und zwei Netzwerkschnittstellen Interfaces. Es stellte sich nur die Frage nach der Fertigung bzw. nach der Entwicklung eines Prozessorboards. Hier kamen wir jedoch zum Entschluss, dass es für Schüler ohne finanzielle Unterstützung unmöglich sei, den im 256-ball BGA Gehäuse ausgelieferten Prozessor zu verbauen, da hier mit Sicherheit mit 4 bis 5 Prototypen zu rechnen wäre, welche bei zwingend nötigen 8 Layer Leiterplatten, geschätzte 5000€ kosten würden – für Schüler unfinanzierbar.

Aus diesem Grund beschlossen wir das Referenzboard, auf welchem alle wichtigen Anschlüsse nach außen geführt waren, in unser Projekt zu integrieren.

Hier hatten wir zunächst eine AP7000 CPU mit 32MB SDRam, 8MB parallelen Flash Speicher sowie 8MB seriellen Flash Speicher. Außerdem hatte das NGW100 bereits zwei Netzwerkschnittstellen, einen USB Anschluss, einen RS232 Anschluss und einen SD-Karten Einschub integriert.



Abbildung 5: AP7000

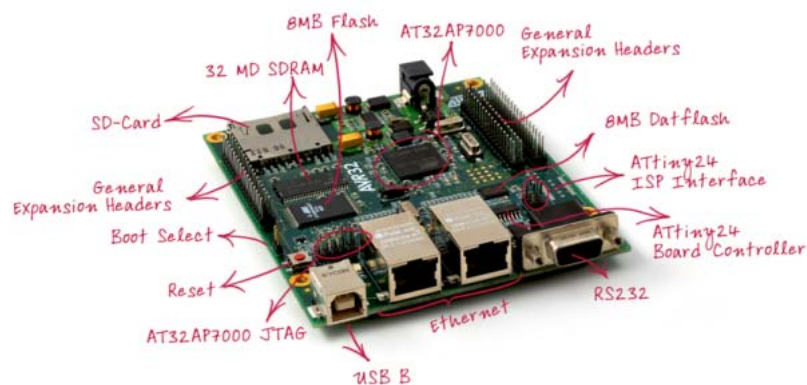


Abbildung 6: Atmel NGW100 Referenzdesign

Der zweite Schritt galt nun der Auswahl der richtigen Audio Hardware. Hier gab es grundsätzlich drei Möglichkeiten, die Benutzung des internen Bitstream DAC, einen einfachen Audio Codec oder eines höherwertigen AC97 Codec. Der interne Bitstream DAC schied nach einer kurzen Suche im Internet schnell aus, da er als sehr schwer handhabbar und instabil galt. Da wir vor allem auf die technische Realisierbarkeit achten mussten, konnten wir nur Hardware verwenden, die bereits von anderen getestet wurde. Darum hatten wir noch die Entscheidung zwischen dem STK1000 (einem Demoboard von Atmel) verbauten Atmel AT73C213 und dem Cirrus Logic CS4202, wo bereits einige sehr positiv ausgefallenen Erfahrungsberichte auf www.avrfreaks.net zu finden waren. Wir entschieden uns schließlich für den CS4202, da er als hochwertige „Soundkarte“ galt und uns mehrere Ausgänge (unverstärkt, mit Kopfhörerverstärker, optisch digital) sowie auch mehrere Eingänge bot. Dieser Chip wird auch in PC Soundkarten verbaut. Der AT73C213 hingegen war dem CS4202 aufgrund von nur 3 Audioausgängen deutlich unterlegen.



Abbildung 7: CS4202



Abbildung 8: Sharp LQ043

Mit der Wahl des Displays erfüllten wir uns den ersten Wunsch auf unserer Wunschliste. Wir wählten den Sharp LQ043 – besser bekannt unter dem Namen PSP Display. Er bietet eine Auflösung von 480x272 Pixel bei einer Pixeltiefe von bis zu 24bit was 2^{24} verschiedenen Farben entspricht.

Beim VGA Ausgang setzten wir auf den erprobten und bereits im Atmel STK1000 eingesetzten ADV7125.

Für die komplette Steuerung des User Interfaces des Players wurde ein ATmega1281 eingesetzt, welcher im lötbaren TQFP64 Gehäuse ausgeliefert wird und mit 128KByte Flash, der größte Atmel 8-bit Mikroprozessor ist, der mit 3.3V betrieben werden kann.

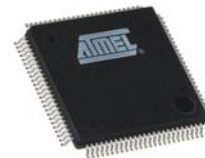


Abbildung 9: ATmega1281



Abbildung 10: QT1106

Als nächsten Punkt auf unserer Liste standen das „I-Pod Wheel“ und die kapazitiven Sensoren. Hier hatten wir eine breites Spektrum an Möglichkeiten, die PSoCs von Cypress, mTouch von Microchip und qTouch von Quantum (d.h. eigentlich von Atmel). Da wir bereits bei den Mikroprozessoren auf Atmel setzten und da Quantum einen komplette Treiber, sowie detaillierte Beschreibungen zum Design der Sensoren anboten, haben wir uns für den QT1106 von Quantum/Atmel entschieden.

Beim Bluetooth setzten wir auf bewährte Technologie. Der BlueSMiRF von Sparkfun wurde von Bernhard bereits bei anderen Projekten eingesetzt. Es ist ein sehr einfaches Bluetooth Modul, das über eine serielle Verbindung angesprochen werden kann.



Abbildung 11: BlueSMiRF

Wireless Lan ... eines der Problemthemen unseres Projektes. Der Auswahlprozess eines geeigneten Wireless Moduls dauerte beinahe zwei Monate, dass Problem vor dem wir

standen war die Kompatibilität zu unserem Prozessor und die Unterstützung des aktuellen Linux Kernel.

Wir wollten auf jeden Fall einen 802.11g Wireless mit 54Mbit Datendurchsatz. Der Linux Kernel bietet vollen Support für die Marvell 88W8385 und die 88W8686 Chips, darum wollten wir ein Modul mit diesen Chipsätzen. Jedoch gab es auch hier unterschiedliche Module, vor allem was die Ansteuerung betrifft. Einerseits ein paralleles PCI ähnliches Interface sowie das vollständig proprietäre SDIO Protokoll und eine mit GSPI, eine nicht sehr quelloffene SPI Art. Nach langer Überlegung kamen wir zum Entschluss, dass das parallele Interface für uns, aufgrund akutem Pinmangels, nicht verwendbar war. Auch das GSPI verlor an Attraktivität. Wir hatten weder Informationen ob der Linux Treiber dieses Protokoll überhaupt unterstützte, noch ob es überhaupt verwendet wird.

Wir entschieden uns schließlich für SDIO, das Protokoll, dass auf dem SD Standard aufbaut und für Außenstehende komplett unzugänglich ist, da darüber hinaus keine Datenblätter oder Spezifikationen zu finden (außer man tritt der SD Association bei, was natürlich mit einem finanziellen Aufwand verbunden ist)sind. Wir waren also zu 100% auf die Funktionstauglichkeit des Linux Treibers angewiesen. Das Finden des geeigneten Moduls gestaltete sich als äußerst schwierig. Am Ende unserer Suche gab es die Auswahl zwischen zwei Modulen, einem Modul von embeddedworks.com, einem US-Amerikanischen Hersteller und einem Modul von Zcomax einer international tätigen Firma.

Das Bestellen der Module bei Embeddedworks ist relativ einfach, nur der Preis mit 100\$ pro Modul zuzüglich rund 150\$ Versand und dem Zoll ist jedoch überzogen. So entschieden wir uns für das XG-182M von Zcomax. Das Beschaffen des Moduls stellte sich als äußerst schwierig da, die minimale Bestellmenge bei einigen 1000 Einheiten liegt und nur an Firmenkunden verkauft werden. Nach einigen Telefonaten mit der Liegenschaft in England und einem außergewöhnlich netten Verkäufer, bekamen wir zwei dieser Module zu einem wirklich sehr niedrigen Preis, jedoch ohne Dokumentation und Treiber.



Abbildung 12: XG-182M

3.2.1.2. Die xRemote

Die xRemote- Eine kompakte und moderne Fernsteuerung mit einem etwas anderen Konzept.

Als Hauptprozessor kommt der im xMedia Player als UI- & Boardcontroller verbaute ATmega1281 zum Einsatz, da er unsere Hardwareanforderungen perfekt trifft und wir viele Firmware Komponenten vom Boardcontroller auch bei der xRemote verwenden konnten.

Als Anzeige wird ein monochromer grafischer LCD von CrystalFontz verwendet, der 128x64 Pixel groß ist und bereits mit einem resistiven Touchscreen bestückt ist. Diesen wählten wir nur aus Mangel an Alternativen in dieser Größenklasse.



Abbildung 13: CFX

Um die Funktionalität der Wiimote nachzubilden, begaben wir uns auf die Suche nach einem geeigneten Beschleunigungssensor, der möglichst einfach zu handhaben war, wenig periphere Bauteile benötigte und möglichst wenig Platz auf der Leiterplatte benötigte. Unsere Wahl fiel auf LIS3LV02DQ von ST. Da er neben einem SPI Interface auch über ein I²C (oder TWI – Two Wire Interface) Interface verfügt, was für die Einbindung in unsere Hardware ideal war.

Er erfordert bis auf die Standard Stützkondensatoren keine peripheren Bauteile und wird in einem QFPN-28 Package geliefert.

Wieder auf Grund der Wiederverwendbarkeit der Bauteile und Treiber kam auch in der xRemote der QT1106 als Controller für die kapazitiven Sensoren zum Einsatz. Das gleiche galt auch bei der Wahl des Bluetooth Moduls. Hier wurde wieder das Sparkfun BlueSMiRF Modul verwendet.

Mobilität ist einer der wichtigsten Aspekte bei der Entwicklung einer Fernsteuerung, diese kann nur durch einen leistungsstarken Akku erreicht werden. Wir entschieden uns für Single-Cell Lithium Polymere Akku Packs – die mit einer Nennspannung von 3.7V und einer Kapazität von 860mAh bzw. 1100mAh genau unseren Anforderungen entsprachen. Da wir auf unsere Systemspannung von 3.3V mit einem LiPo nur eine Spannungsdifferenz von maximal 0.4V erreichen, mussten wir einen speziellen Low-Dropout Voltage Regulator verwenden. Wir entschieden uns für den Micrel MIC5205, der bis zu einem Dropout von 150mV arbeitet. Zum Laden des Akkus wurde ein perfekt geeigneter Maxim MAX1551 eingesetzt, der es ermöglicht, den Akku einerseits über USB und andererseits über eine alternative Spannungsquelle zu laden.



Abbildung 14: LiPo

USB, einer der Knackpunkte der xRemote. Hier setzten wir beim ersten Prototypen zunächst auf eine softwaremäßige USB Implementation. Diese erschien zunächst zwar funktionstüchtig, jedoch nach dem ersten Prototype, stellte sich schnell heraus, dass die Prozessorklast der USB Routinen viel zu hoch war. Daher entschieden wir uns im zweiten Schritt für den Silabs CP2102 einer USB to UART Bridge.



Da Grafiken im Flash des Prozessors sehr viel Platz verbrauchen, beschlossen wir die Fernsteuerung mit einem microSD Karten Slot auszustatten, welcher über SPI angesprochen wird.

Abbildung 15: µSD

3.2.2. Auswahl der Software

Der xMedia Player sollte sich nicht nur auf Hardware Ebene als ausgewachsener und marktreifer Player präsentieren, sondern auch auf der Software Ebene.

Als Basis für den Player wählten wir den Linux Kernel mit einer Busybox Umgebung.

Das Buildroot System (welches unter anderem von Atmel, dem Hersteller des AP7000 Prozessors angeboten wird) ermöglicht es relativ einfach eine komplett eigene Linux Distribution zu erstellen. Außerdem können ausgewählte Anwendungen sehr simpel crosskompiliert und in Distribution eingebunden werden.

Die Auswahl der richtigen Programmiersprache und der richtigen Programmierumgebung war auch ein wichtiger Punkt. Grundsätzlich kamen hier die Programmiersprachen Java und C++ in nähere Betrachtung. Die AP7000 CPU hat zwar eine spezielle Java Optimierung und Acceleration, bleibt jedoch eine halb interpretierte Programmiersprache und hat einen merklichen Performancenachteil gegenüber dem machinennahen C++. Da der xMedia Player über stark begrenzte Ressourcen verfügt, wählten wir daher C++.

Einer der wichtigsten Punkte war die Wahl eines Window Systems für Linux bzw. einer Grafikbibliothek. Zunächst tendierten wir zu, dem bei Desktopsystemen eingesetzten, XWindows bzw. zu dem normalerweise auf Embeddedsystemen eingesetzten nanoX. Dies wollten wir mit der Grafikbibliothek GTK2 kombinieren. Doch bei näherer

Betrachtung tendierten wir immer mehr zu QT. QT wurde zunächst von Nokia für die Verwendung auf ihren Mobiltelefonen entwickelt. Durch die Firma Trolltech, die später um QT entstanden ist, entwickelte sich die Bibliothek QT schnell weiter und wurde, aus unserer Sicht, zu einer der mächtigsten C++ Bibliotheken auf dem Markt. C++ mit QT ist mit C++ ohne QT nicht mehr vergleichbar, die Sprache ändert mit der Bibliothek ihren kompletten Charakter. Von einer viel zu komplexen, zwar mächtigen, aber in die Jahre gekommenen Programmiersprache zu einer modernen Sprache die ohne weiteres mit Sprachen wie Java und C# mithalten kann. Einer der weiteren Vorteile von QT ist, dass es Plattform unabhängig ist. Die Anwendungen müssen neu kompiliert werden, jedoch die Funktionalität der Applikation bleibt ohne Änderung des Codes gleich, egal ob Linux, Windows oder OS X. Des Weiteren inkludiert QT Embedded die Möglichkeit die Applikationen direkt, ohne Window System zu starten, d.h. die Anwendung bzw. QT schreibt direkt in den Linux Framebuffer.



Abbildung 16: QT Logo

Das Phonon Framework zum Abspielen multimedialer Inhalte in QT war leider nicht für die Benutzung in integrierten Systemen konzipiert und es wurde stattdessen der von Atmel spezielle optimierte MPlayer verwendet. MPlayer ist ein gut anpassbarer Multimediaplayer und ist ein echtes Leichtgewicht was Prozessor und RAM Auslastung betrifft.

3.2.3. Der Hardwareaufbau

3.2.3.1. xMedia Player

Die Entwicklung der Hardware gestaltete sich zwar als sehr fordernd und zeitlich aufwändig, jedoch gab es beim Erstellen des Stromlaufplans nur einige kleinere Knackpunkte.

Bereits bei der ersten Revision des Stromlaufplans galt der DC-DC Wandler zum Erzeugen der Hintergrundspannung für das LC-Display als einer dieser Knackpunkte. Der Boostkonverter sollte Spannung von den 3.3V (bzw. 7-12V) auf die benötigten ~24V bei 15mA bringen. Boostkonverter dieser Art haben mehrere Nachteile, sie sind Schaltwandler und schalten mit meist relativ hohen Frequenzen den Strom durch eine Spule an und aus, das wiederum sehr hohe elektromagnetische Störungen und auch hörbares Summen der Spule verursachen kann. Wir entschieden uns bei der ersten Hardwarerevision, nach langem Überlegen, für den LM2735 von National, da dieser gute Leistung und einfache Handhabung versprach, jedoch war dem leider nicht so. Gleich der erste Versuch brachte die Ernüchterung, der Boostkonverter ging in Rauch auf, und schmorte einen großen Krater in die Platine unseres ersten Prototype. Der Grund dafür war schnell geklärt. Der Ausgangstromkreis war aufgrund einer schlechten Verbindung am Display unterbrochen was zum Tod des Chips führte, außerdem summte der Chip doch deutlich hörbar, bevor er sich in die Ewigen Jagdgründe verabschiedete. Die nächsten Wochen suchten wir verzweifelt nach einem Ersatz, jedoch zeigte jeder Chip dasselbe Verhalten. Wir waren am Ende und Bernhard begann auf Basis eines seiner anderen Projekte selber einen Boostkonverter zu entwerfen, der unseren Anforderungen gerecht wurde. Der Boostkonverter auf Basis eines ATtiny85 Prozessors wurde eine Meisterleistung, mit einer Schaltfrequenz im Megahertzbereich und einer Spannungssowie einer Strombegrenzung war er sowohl zu 100% stabil, als auch für das menschliche Ohr unhörbar. Als kleines Highlight verfügt der Konverter noch über ein I²C Interface, das es ermöglicht Spannung, Strom und Schaltfrequenz zur Laufzeit zu ändern.

Außerdem schlichen sich einige kleinere Schlampigkeitsfehler ein, welche mit der zweiten Version (Revision 3) behoben wurden.

Da unser Budget sehr, sehr knapp bemessen ist, wussten wir, dass wir nur maximal zwei Boards fertigen konnten, d.h. genau einen Prototypen und eine finale Leiterplatte.

3.2.3.2. xRemote

Da bereits einige Erfahrungen bei Designs von 8 Bit Systemen bestanden, war der Entwurf des Stromlaufplans hier denkbar einfach. Das Layout hingegen gestaltete sich, aufgrund des kleinen Formfaktors, doch recht fordernd. Bei der xRemote galt vor allem das Power Management als Herausforderung, welche jedoch mit soliden Überlegungen und guter Durchführung gemeistert wurde und anstandslos funktionierte.

Aufgrund einiger lö- und layouttechnischer Probleme musste auch von diesem Layout eine zweite Version gefertigt werden.

3.2.4. Die Firmware

Mit dem Testen und Debuggen der Hardware ging auch die Entwicklung der Firmware für den Boardcontroller und die xRemote her.

3.2.4.1. Gemeinsame Module

Da wir bei der Fernsteuerung und beim Boardcontroller mit dem ATmega1281 denselben Controller verwendeten, konnten wir alle Treiber und einige Softwareteile wiederverwenden.

Als Basis diente das von Bernhard in den letzten Jahren entwickelte blib. Eine Treiber Bibliothek die bereits diverse Systemtreiber für Timer, Analog/ Digital Konverter, Serielle Schnittstellen, I²C (TWI) SPI, Software USB und viele andere Softwarekomponenten bereitstellte.

Da wir auch den QT1106, den Controller für die kapazitiven Sensoren, auf beiden Boards verwendeten, konnten wir auch diesen Treiber, bei dem der Großteil bereits von Atmel vorgefertigt war, doppelt nutzen.

Als wichtigster Firmware Teil galt die xSie. Die xSie regelt jeden Austausch von Daten zwischen dem Boardcontroller und dem Hauptprozessor und dem Boardcontroller und der xRemote. Die xSie stellt einen Abstraktionslayer zwischen dem seriellen UART Treiber und der eigentlichen Applikation dar. xSie ist zum Packen der Daten in einem speziell für xMedia entworfenen Protokoll verantwortlich, dass es ermöglicht, eine sichere Übertragung zu gewährleisten und abgebrochene Datenübertragungen wieder aufzunehmen. Grundsätzlich sind die gesendeten Pakete in SLIP (Serial Line Internet Protocol) Frames verpackt. Diese Pakete bestehen nun aus mehreren Komponenten, einer Packet ID, einem Commandcode, der Länge der Nutzdaten, und der eigentlichen Nutzdaten. Zur Fehlererkennung wird hier ein CRC (cyclic redundancy check) eingesetzt.

3.2.4.2. Der Boardcontroller

Der Boardcontroller hat die Aufgabe, einerseits, das kapazitive Panel, mit der entsprechenden Hintergrundbeleuchtung (RGB – LEDs, die über High Speed IO Extender gemultiplext werden) anzusteuern, den Boost Konverter um den LCD zu steuern, das Power Management für einige Komponenten sowie den VGA Ausgang und die von der xRemote ankommenden Daten an die CPU weiterzuleiten.

Das Auslesen der Daten des kapazitiven Controllers gestaltet sich durch den von Atmel vorgefertigten Treiber relativ einfach und problemlos, das Multiplexen der Hintergrund RGB LED über die IO-Extender, stellte hier bedeutend mehr Softwareaufwand da, da das Timing hier kritisch ist, und die ganzen Routinen durch die Anbindung der Extender über I²C relativ zeitaufwändig sind.

Das Weiterleiten der Daten von der xRemote zur AP7000 CPU ist hier ein bedeutend leichteres Unterfangen gewesen, da durch die bereits voll ausgereifte xSie das Commandrouting hier innerhalb weniger Minuten erledigt wurde.

3.2.4.3. Die xRemote

Die xRemote hatte sehr viele Aufgaben, jedoch drehte sich alles um Usereingaben, d.h. um meist 4 bis 8 Kommandos (rechts, links, oben, unten oder weiter, zurück, Taste 1, Taste 2, Taste 3, Taste 4). Es musste der Touchscreen ausgelesen werden, die Werte des kapazitiven Sensors, sowie die Werte des Beschleunigungssensors verarbeitet werden.

Zur Abstraktion der ganzen Benutzereingaben wurde hier ein UI Layer eingeführt, der die Daten aller Benutzereingaben an zwei Ziele weiterleiten kann, einerseits an die Applikation die auf der xRemote selbst läuft und andererseits über das Bluetooth Modul direkt an den Boardcontroller und somit an die CPU. Mit einem sehr effizienten Event Routing ist es von der Applikation der xRemote sehr einfach möglich die Ziele der Eingabedaten zu wählen.

Außerdem bietet die xRemote die Möglichkeit SD / MMC und µSD Karten jeweils bis zu 2GB Kapazität zu lesen und auch das FAT16/12 Dateisystem zu verarbeiten, um Bilder, die im Flash – Speicher sehr viel Platz benötigen würden von einem Massenspeicher lesen zu können.

3.2.5. Die Systemtreiber

Die Treiber für alle an den AP7000 angebundene Baugruppen sind einer der Knackpunkte des Systems. Der Prozessorhersteller Atmel, gibt vollen Linux Support vor, doch die Treiber für die doch sehr, sehr junge AP-Prozessorserie sind noch nicht ausgereift und weisen teilweise noch schwerwiegende Fehler auf. Darum gestaltete es sich als eines der schwierigsten Unterfangen, alle Treiber lauffähig zu bekommen.

3.2.6. Die Software

Wie bereits erwähnt verwendeten wir auf dem xMedia Player ein Linux Distribution mit QT Bibliotheken.

Durch den QT Creator, einem wirklich effizienten IDE, das vom QT Entwickler Trolltech angeboten wird, war es ein Leichtes grafische Anwendungen auch für den Embedded Bereich zu erstellen. Mit einem, zwar in der verwendeten Beta Version noch etwas eingeschränkten GUI Editor, konnte mit wenigen Handgriffen und sehr geringem Programmieraufwand eine professionelle Oberfläche erstellt werden.

Im Hintergrund operierte wieder die bereits im Boardcontroller und in der xRemote eingesetzte xSie, die alle Usereingaben an die Applikation weiterleitete. Hier wurde nur das Konzept vom funktionsbasierten C Konzept auf ein objektorientiertes C++ Konzept geändert.

3.2.7. Das Gehäuse

Der **erste Eindruck** zählt!“ – Diesen Satz hört man immer wieder und bemerkt ebenso oft, dass diese Feststellung nicht aus der Luft gegriffen ist. Warum sollte diese, schon seit vielen Jahren bekannte, Feststellung nicht auch auf ein Gehäuse übertragbar sein. Inspiriert durch diese Feststellung legten wir unser Augenmerk besonders auf das Gehäusedesign und auf die verwendeten Materialien. Um dem Gehäuse eine gewisse Dynamik zu verleihen wurde es mit speziellen Rundungen ausgestattet. Auch auf exklusive Materialien wurde Wert gelegt. Hierfür beschränkten wir uns auf edles Nussholz mit eingelegten Plexiglaselmenten. Die edle Optik, hervorgerufen durch das dunkle Nussholz, verleiht unserem Projekt einen eigenen Charakter. Das Design selbst unterstreicht die Funktionalität, Moderne und Innovation unseres Projekts. Auch bei der Gehäusekonstruktion beruhten unsere Entwicklungen auf möglichst viele Schrauben zu verzichten. Dies geschah aus 2 Gründen: 1. um die mechanische Beanspruchung zu minimieren und 2. Um das Gesamtkunstwerk nicht mit störenden Schrauben zu verunstalten.

„Der erste Eindruck zählt - Der letzte Eindruck bleibt“



Abbildung 17: Gehäusekonstruktion



Abbildung 18: Explosionszeichnung

3.3. Stand der Entwicklung

Mit Mitte Februar 2009, dem Zeitpunkt der Erstellung dieses Berichtes, ist das Projekt zu geschätzten 70% bis 75% abgeschlossen.

Hardware:

- xMedia Player
 - o Audio (voll funktionsfähig)
 - o Wireless (voll funktionsfähig)
 - o 2. SD Card Slot (voll funktionsfähig)
 - o Netzwerk (voll funktionsfähig)
 - o LCD (voll funktionsfähig)
 - o Hintergrundbeleuchtung - Boost Converter (voll funktionsfähig)
 - o Board Controller (voll funktionsfähig)
 - o Kapazitives Panel (voll funktionsfähig)
 - o Bluetooth (voll funktionsfähig)
 - o VGA (funktionsfähig, jedoch leichte EMV Probleme bzw. Clock Jitter, Installation des bereits vorgesehenen EMV-Shield notwendig)
- xRemote
 - o Prozessor (voll funktionsfähig)
 - o Power Management (voll funktionsfähig)
 - o Kapazitives Interface (voll funktionsfähig)
 - o USB (voll funktionsfähig)
 - o µSD (voll funktionsfähig)
 - o Beschleunigungssensor(voll funktionsfähig)
 - o Bluetooth(voll funktionsfähig)

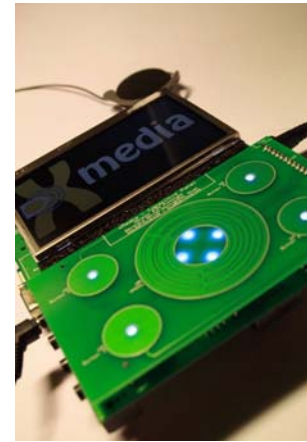


Abbildung 19: xMedia finale Hardware

Firmware:

- Board Controller
 - o xSie - Serial Interface Engine (voll funktionsfähig)
 - o Kapazitives Interface (voll funktionsfähig)
 - o xRemote Unterstützung (voll funktionsfähig)
 - o Power Management (noch nicht implementiert)
- xRemote
 - o xSie (voll funktionsfähig)
 - o LCD (voll funktionsfähig)
 - o Touchscreen (voll funktionsfähig)
 - o Kapazitives Interface (voll funktionsfähig)
 - o SD/MMC mit FAT Dateisystem - lesen und schreiben (voll funktionsfähig)
 - o USB (voll funktionsfähig)
 - o Beschleunigungssensor (voll funktionsfähig)
 - o Power Management (noch nicht implementiert)
 - o IR-Steuerung von Fernsehern (noch nicht implementiert)
 - o Hauptmenü (teilweise implementiert)



Abbildung 20: xRemote finale Hardware

Treiber:

- LCD Treiber (voll funktionsfähig)
- Audio driver (voll funktionsfähig)
- SPI to MMC/SD (voll funktionsfähig)
- Serial Driver (voll funktionsfähig)
- Wireless (Probleme mit aktuellem Linux Kernel, war jedoch schon funktionsfähig)

Software:

- QT Embedded (voll funktionsfähig)
- MPlayer (voll funktionsfähig)
- xMedia Software
- Hauptmenü (voll funktionsfähig)

- Dateibrowser (voll funktionsfähig)
- Media Player (funktionsfähig, jedoch mit eingeschränkter Funktionalität, fehlende Shuffle & Loop Funktionen, ...)
- Lautstärke Regelung (funktionsfähig, jedoch sehr einfach gehalten)
- Web Browser (nur als Demo Version, keine Benutzerinteraktion möglich)
- Webinterface für Einstellungen (noch nicht implementiert)



Abbildung 21: Hauptmenü

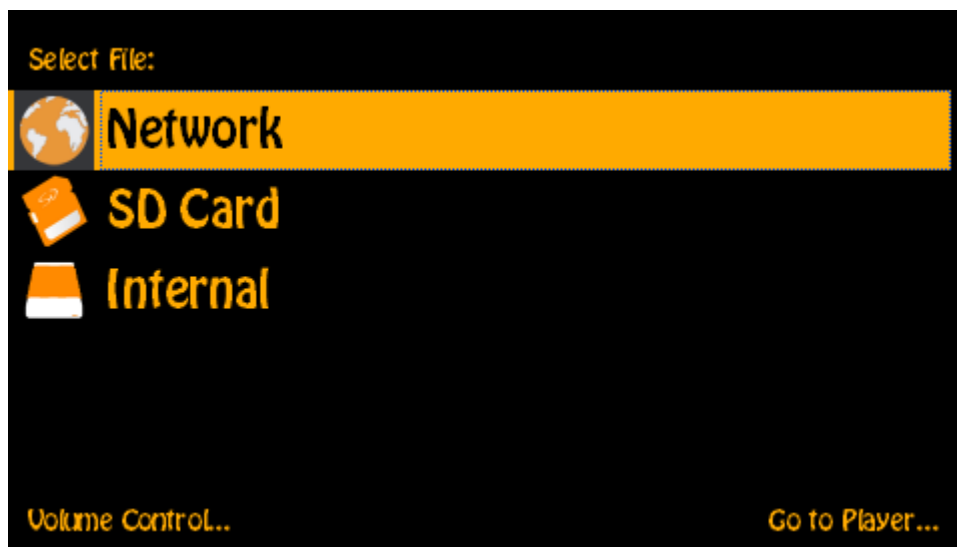


Abbildung 22: Datei Browser

Gehäuse:

- xMedia Player (vollständige 3D Simulation – kurz vor Fertigung)
- xRemote (start des Entwurfs in Kürze)

Vereinfacht könnte man sagen, dass die Hardwareentwicklung zu 100% abgeschlossen ist, es auf Firmwareebene noch Rückstände im Power Management Bereich gibt und es bei den Linux Treiber nur noch Probleme mit dem Wireless Treiber in Kombination mit dem neuen Linux Kernel gibt. Auf Software Ebene gibt es keine Probleme, jedoch noch einige offene Punkte, die relativ schnell realisierbar sind. Das Gehäuse, eines der letzten Punkte auf unserem Zeitplan steckt gerade in der Entwicklung, eine 3D Simulation ist bereits vorhanden.

3.4. Innovation

3.4.1. xDimension - die Innovation

Als Innovation wollte sich der Projektleiter Bernhard einen langen Traum verwirklichen. Der Film „Minority Report“ inspirierte ihn dermaßen, dass er ein Gerät nur mittels Fingerbewegungen steuern wollte, nicht etwa mit Druck auf Sensoren. Mit anderen Worten, wenn der Benutzer seine Finger nach rechts zieht, selektiert er im Menü den nächsten Punkt, wenn er sein Finger danach nach unten zieht, wählt er den entsprechenden Punkt aus. Genau diese Gesture Control wurde laut unseren Informationen noch nie für Consumer Elektronik eingesetzt, geschweige denn zur Steuerung eines Multimediaplayers.



Abbildung 23: Minority Report

Da nicht feststand, wie fordernd der xMedia Player und die xRemote werden würden, wurde xDimension zunächst nicht offiziell in das Projekt aufgenommen.

Als später einige große Hürden des Projektes überwunden waren, konnten wir einige Stunden bzw. Tage in den Projektteil mit dem Namen xDimension investieren.

Ziel des Projektteils xDimension war es, Finger, die mit speziellen Markierungen versehen sind, verfolgen zu können, bzw. Gesten der Punkte/Finger zu erkennen und an den xMedia Player weiterzuleiten.

Zur Realisierung im Detail gab es viele Möglichkeiten. Das Konzept war jedoch recht schnell klar. Ein starkes Infrarotpanel strahlt auf die Finger des Benutzers, die mit Infrarotreflektoren bestückt sind – die Lichtstrahlen werden zurückgeworfen und von einer Kamera aufgenommen. Nun müssen die Daten noch verarbeitet werden und die Gesten erkannt werden – der schwierigste Teil des Projektes.

Es gab grundsätzlich zwei Ansätze dieses Projekt anzugehen, eine Möglichkeit, ist es eine Kamera per Firewire oder USB an einen kleinen PC anzuschließen, und dort mit der entsprechenden Software die Kameradaten zu verarbeiten. Als zweiter Ansatz kam das zweite NGW100 – Linux Board in Betracht, der AP7000 ist mit einem Imagesensorinterface ausgestattet, das es ermöglicht, Daten von einer angeschlossenen Kamera zu verarbeiten, um sie später zu analysieren.

Diese Methoden waren jedoch bei weitem zu kompliziert, zu unhandlich - schlicht und einfach unrealisierbar. Darauf folgend viel unser Augenmerk wiederum auf den Controller der Nintendo Wii – der Wiimote. Im großen weiten Internet fanden wir eine Beschreibung des Innenlebens einer Wiimote - ... ein Bluetooth Module, ein Beschleunigungssensor, ein EEPROM, ein Infrarot Imagesensor, ... – ein Infrarot Imagesensor? Genau dieser Infrarotsensor kam genau richtig für unser Projekt. Er ermöglicht Infrarotpunkte (sogenannte Blobs) aufzuspüren und zu verfolgen - nicht nur einen Punkt, sondern die stärksten vier Infrarotblobs. Wohl der größte Vorteil des Sensors ist die Ansteuerung, die über I²C so einfach und unkompliziert wie nur irgend möglich ist. Ein Nachteil dieses Sensors liegt darin, dass er für Normalpersonen nicht erhältlich ist, und es keinerlei offizielle Dokumentation gibt.

Da der Sensor nicht als Einzelstück erhältlich ist, war der erste Schritt, der Tod eines der innovativsten Eingabegeräte unserer Zeit, einer nagelneuen Wiimote. Da wir nicht die Zeit und das Geld hatten ein komplett neues Mikrocontrollersystem zu entwickeln, entschlossen wir uns das Ganze als Zusatzmodul für die Leistungsstarke xRemote zu realisieren. Nach wenigen Wochen Entwicklungszeit hatten wir ein fertiges Infrarotpanel und ein Trägermodul für den Infrarotsensor.

Über das Elektronikmagazin Elektor kamen wir schließlich zum Treiber für den Imagesensor und nach kürzester Zeit funktionierte der erste Prototyp von xDimension und erfasste die Position von bis zu 4 Fingern gleichzeitig.

Nach weiteren Tagen bzw. Wochen Arbeit funktionierte die erste Testversion der Gestikererkennung und der xMedia Player konnte in seinen Grundfunktionen mittels Fingerbewegungen gesteuert werden.

Der aktuelle Stand der Dinge, betreffend xDimension ist relativ gut – die erste Testversion funktioniert – zwar mit leichten Ungenauigkeiten bei Gestikererkennung – aber sie funktioniert.

Eine Demonstration von xDimension finden Sie unter:

<http://www.project-xmedia.com/main.php?page=multimedia>

3.4.2. Der Beschleunigungssensor – das neue Eingabegerät

Bei unserer Arbeit mit dem Beschleunigungssensor der xRemote, den wir zunächst nur zur Erkennung der Lage der Fernsteuerung (Landscape Mode wie beim I-Phone) einsetzen wollten, kamen wir auf eine neue Methode der Navigation. Der Benutzer muss nun nicht mehr die Tasten, seien es normale oder kapazitive suchen, sondern steuert den xMedia Player mit der Fernsteuerung selbst, genauer gesagt mit der Bewegung der Fernsteuerung. Will der Benutzer im Menü nach rechts – kippt er die Fernsteuerung nach links – will er nach rechts, kippt er sie nach rechts.

Diese Steuerung ist sehr intuitiv aufgebaut, einfach zu begreifen und funktioniert vollkommen Problemfrei - unser Tipp für die nächst Generation von Fernsteuerungen.

Eine Demonstration der xRemote mit dem Beschleunigungssensor als Eingabegerät finden Sie unter:

<http://www.project-xmedia.com/main.php?page=multimedia>

3.5. Zusammenarbeit im Team

Wie in jedem Projekt ist einer der wichtigsten Aspekte die Zusammenarbeit der einzelnen Projektmitglieder. Wir harmonierten gut miteinander und versuchten alles, wie besprochen, so gut wie möglich in die Tat umzusetzen. Die umfangreiche Arbeit am Projekt erforderte auch einen sehr großen Zeitaufwand. Darum war es wichtig, dass beim Arbeiten ein gewisser Spaßfaktor vorhanden war, der jedoch niemanden von der Arbeit abhielt.

Bei den täglichen Teambesprechungen hatten wir immer wieder unsere Ziele für den Tag besprochen und die jeweiligen anderen Projektmitglieder über die genaueren Arbeitsschritte, die durchzuführen waren, informiert.

So bekam jeder einen Überblick über den Aufgabenbereich des anderen. Wir würden das Arbeiten am Projekt auch als „Paralleles Lernen“ bezeichnen, da jeder einen Spezialbereich über hat und gleichzeitig mit den anderen über ihr jeweiliges Thema mitlernt.

Bevor Beendigung eines Arbeitstages wurden die erreichten Ziele des Tages besprochen und es wurde auf die nächsten Ziele näher eingegangen. Durch diese Besprechungen konnte sich jeder individuell auf neue Situationen vorbereiten.

Die Distanz zwischen den Wohnorten war jedoch ein großer Nachteil für die Zusammenarbeit. Dies machte es sehr kompliziert an Wochenenden oder in den Ferien gemeinsam am Projekt zu arbeiten oder die Fortschritte des Einzelnen den Anderen zu präsentieren. An solchen Tagen konnte nur online oder via Telefon kommuniziert werden. Auftretende Probleme konnten so nur auf Umwegen beseitigt werden.

Da unser Projekt eine Eigeninnovation ist und ohne eine Firma durchgeführt wurde, war es unmöglich Firmen dazu zu begeistern uns bei dem kostenaufwendigen Projekt finanziell zu unterstützen.

Interessant am Teamprojekt war, dass man immer wieder auf neue Herausforderungen stieß. Neue Problemlösungen mussten gefunden werden und so konnten wir unserer Kreativität freien Lauf lassen.

Das selbstständige Arbeiten war auch sehr positiv im Hinblick auf die Zukunft, da man auch später im Berufsleben immer wieder auf Hindernisse trifft, welche man aus eigener Kraft und mit neuen Ideen überwinden muss.

Für die Zukunft ist auch mitzunehmen, dass wir bereits ein Projekt im Team realisiert haben und so eventuell in späteren Gruppenarbeiten einen Vorteil, auf Grund der gemachten Erfahrungen, aus diesem Projekt ziehen.

Obwohl wir uns bereits seit 5 Jahren kennen und uns dachten viel über einander zu wissen, lernten wir immer wieder neue Stärken und Schwächen des Anderen kennen.

Durch die Teamarbeit und die gegenseitige Hilfestellung konnte auch die eine oder andere Schwäche ausgebessert werden.

Hinter diesem Projekt steckt ein immenser Zeitaufwand. An vielen Tagen wurde auch bis in die Nachtstunden gearbeitet, um das nächste Ziel zu erreichen.

Neben der gesamten Hard- und Softwareentwicklung gab es noch viele andere Bereiche, mit denen man sich zeitintensiv beschäftigen musste. Zu diesen Aufgaben zählten unter anderem die Gehäusekonstruktion, Bestellungen von Bauteilen, Leiterplattenfertigung, Homepage und vor allem Dokumentation.

4. Anhang

4.1. Stromlaufpläne

Im Ordner „/Stromlaufpläne/“ auf dem beiliegenden Datenträger finden Sie den Großteil der Stromlaufpläne des Projektes.

Die Stromlaufpläne des Projektes sind in mehrere Teile eingeteilt:

- Stromlaufplan der Prozessorplatine (von Atmel) – 13 Seiten
Datei: /Stromlaufpläne/at32ngw100_schematics.pdf
- Stromlaufplan des xMedia Player Hauptplatine – 8 Seiten
Datei: /Stromlaufpläne/xMedia_Rev4.pdf
- Stromlaufplan des xMedia Panels – 2 Seiten
Datei: /Stromlaufpläne/xMedia_Panel_Rev4.pdf
- Stromlaufplan der xRemote – 4 Seiten
Datei: /Stromlaufpläne/xRemote_Rev4.pdf

4.2. Bilder

Im Ordner „/Bilder/“ auf dem beiliegenden Datenträger finden Sie einige Bilder des xMedia und xRemote Prototypen.

4.3. Videos

Der Ordner „/Videos/“ beinhaltet einige kurze Demonstrationsvideos von den Prototypen / Finalversion aller Projektteile.

- xmedia.wmv – Einleitung, Hardwareerklärung, Funktionsdemonstration
- xremote.wmv – Hardwareerklärung, Funktionsdemonstration (mit xMedia Player)
- xdi – xDimension Funktionsdemonstration

4.4. Projektlogos

Die Logos unseres Projektes finden sie auf dem beiliegenden Datenträger im Verzeichnis „/Logos/“

4.5. Zusatzinformationen

Zusätzliche Informationen zum Projekt beinhaltet der Ordner „/Zusatz/“.

- Kurzbeschreibung des kompletten Projektes
„/Zusatz/xMedia_summary.pdf“
- Kurzbeschreibung des kompletten Projektes (für den Druck optimiert)
„/Zusatz/xMedia_summary_print.pdf“
- Präsentation des Projektes (Englisch)
„/Zusatz/xMedia_presentation.pdf“

5. Abbildungsverzeichnis

Titelbild	-
Vom Projektteam erstellte Grafik	
Abbildung 1: Apple I-Pod	4
http://pics.computerbase.de/1/3/7/3/6/1_m.jpg	
Abbildung 2: Sony PSP	4
http://www.portablegaming.de/attachments/sonstiges-psp/5134d1158443912-pimp-my-psp-sony-psp-front-back-kopie.jpg	
Abbildung 3: Nintendo Wii	4
http://www.modfreak.ch/store/images/nintendo_wii.jpg	
Abbildung 4: Iphone 3G	4
http://scr3.golem.de/screenshots/0806/Apple_iPhone_3G/iphone3g_pair.jpg	
Abbildung 5: Atmel AVR32 Chip	6
http://www.msc-ge.com/d/newsletter/img/atmel_nl16_avr32_ap7000.jpg	
Abbildung 6: Atmel NGW100 Referenzdesign	6
http://www.avrfreaks.net/wiki/index.php/Image:Ngw_overview.png	
Abbildung 7: CS4202	7
http://www.cirrus.com/en/images/home/stylized_chip.gif	
Abbildung 8: PSP LCD	7
http://www.sparkfun.com/commerce/images/products/PSP-LCD-01-L.jpg	
Abbildung 9: Atmega1281	7
http://media.digikey.com/photos/Atmel%20Photos/313-100-LQFP.jpg	
Abbildung 10: QT1106	7
http://www.saelig.com/miva/graphics/00000001/qt1106725.jpg	
Abbildung 11: Bluesmirf	7
http://www.sparkfun.com/commerce/images/products/00582-03-L.jpg	
Abbildung 12: XG-182M	8
http://www.zcomax.co.uk/webimg/DSC_0303-1.JPG	
Abbildung 13: CFAX	8
http://www.sparkfun.com/commerce/images/products/LCD-Breakout-1.jpg	
Abbildung 14: LiPo	9
http://www.sparkfun.com/commerce/images/products/00341-02-L.jpg	
Abbildung 15: µSD	9
http://www.freeaccess.com.au/images/images060711/SandiskMicroSD1GB.JPG	
Abbildung 16: QT Logo	10
http://www.cgal.org/images/Qt_logostrap_CMYK.png	
Abbildung 17: Gehäusekonstruktion	13
Solid-Edge Konstruktion	
Abbildung 18: Explosionszeichnung	13
Solid-Edge Konstruktion	
Abbildung 19: xMedia finale Hardware	14
Foto der Hardware	
Abbildung 20: xRemote finale Hardware	14
Foto der Hardware	
Abbildung 21: Hauptmenü	15
Screenshot des Hauptmenüs	
Abbildung 22: Datei Browser	15
Screenshot des Untermenüs	
Abbildung 23: Minority report	16
http://inspireaction.mindandmedia.com/wp-content/uploads/2007/06/minority-report-01.jpg	