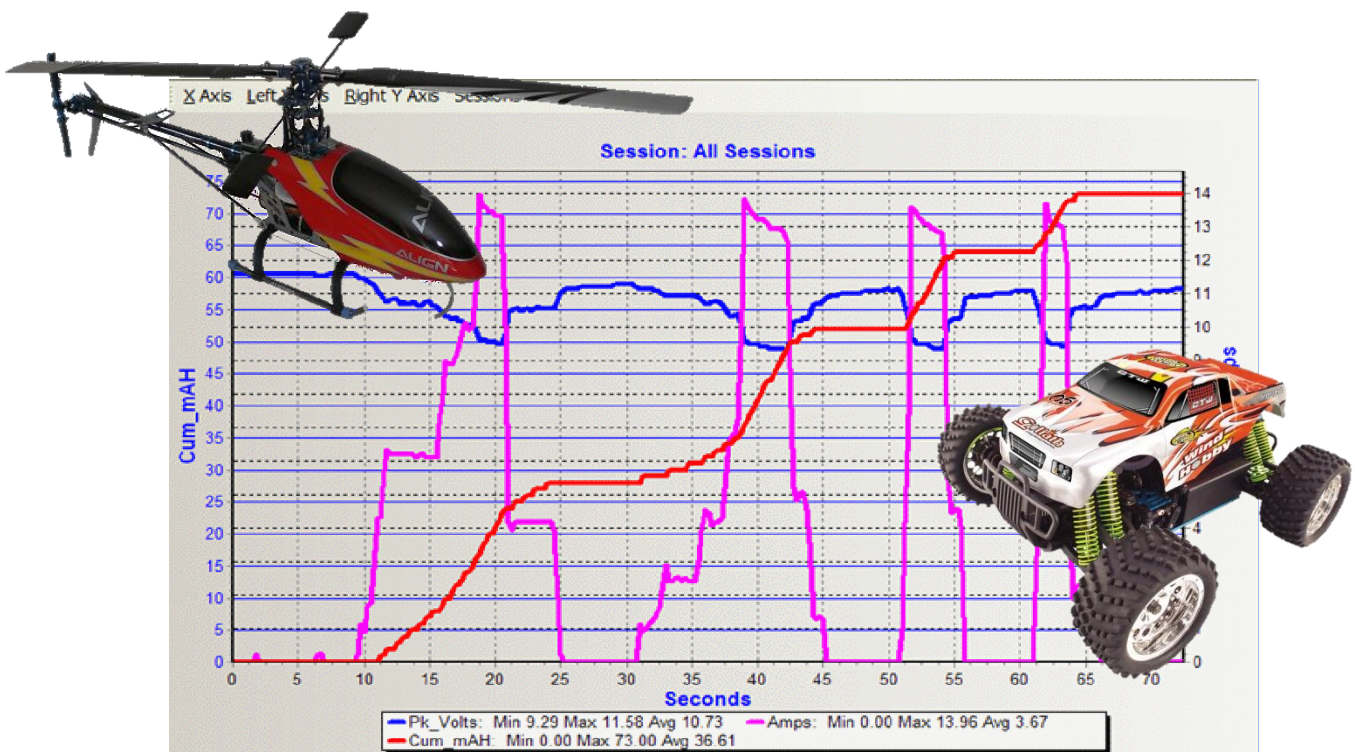


Modellbau Datenlogger mit WLAN-Kopplung Dokumentation



Technikerarbeit 2010/2011

von

Uwe Schmerbach & Benjamin Hirth



Einleitung

Das vorliegende Dokument dient der Dokumentation des Techniker-Abschlussprojekts *Modellbau-Datenlogger mit WLAN-Schnittstelle*. Wir möchten hiermit einen Überblick über das Projekt geben und die Zusammenhänge zwischen den verschiedenen Projektteilen näher aufzeigen. Außerdem werden wir auf die Handhabung und Bedienung des Datenloggers eingehen.

Am Schluss sind aufgetretene Probleme zu finden, weil ihre Bewältigung den Ablauf des Projekts wesentlich beeinflusst haben.

Wir bitten um Verständnis, dass Datenblätter, Quellcode- und Projektdateien (wegen ihrer Größe, > 1000 bedruckte Seiten) zu dieser gedruckten Dokumentation auf einer DVD im hinteren Umschlag zu finden sind und nicht in ausgedruckter Form vorliegen.

Das Projektteam

Uwe Schmerbach und Benjamin Hirth

Formatdefinition

Formatbeispiel

Beschreibung

<u>Überschrift</u> , <i>Überschrift</i> , <i>Überschrift</i>	Bezeichnet den Beginn eines Kapitels verschiedener Gliederungsebenen
Text _{Zahl}	Fußnote: Erklärungen, Quellen oder Verweise; zu finden am unteren Seitenrand; fortlaufend durchnummeriert
<i>Text</i> , Text _{Zahl}	Text, der sich etwa auf ein Programm, einen Menüpunkt oder ein Eingabefeld bezieht. Oder: Ein spezieller/technischer Begriff, den hervorzuheben den Autoren wichtig erschien
Abbildung x , Tabelle x	Beschreibung einer Abbildung oder einer Tabelle, fortlaufend durchnummeriert, eine komplette Übersicht ist am Ende des Dokumentes zu finden

Tabelle 1: Formatdefinition



INHALTSVERZEICHNIS

1	<u>VORSTELLUNG</u>	6
1.1	<i>PROJEKTTEAM</i>	6
1.2	<i>BETREUENDER LEHRER</i>	6
2	<u>PROJEKTINFORMATIONEN</u>	7
2.1	<i>PROJEKTIDEE</i>	7
2.2	<i>PROJEKTZIEL</i>	8
2.3	<i>AUFBAU</i>	9
2.3.1	<i>Blockschaltbild</i>	9
2.3.2	<i>Schaltplan Loggerplatine</i>	10
2.3.3	<i>Layout Loggerplatine</i>	11
2.3.4	<i>Web-Server</i>	11
2.4	<i>ABLAUF</i>	12
3	<u>HARDWARE</u>	12
3.1	<i>CONTROLLER AT32UC3A1512</i>	13
3.2	<i>WLAN-PLATINE SPB105</i>	13
3.3	<i>TEMPERATUR-SENSOR MCP9701A</i>	13
3.4	<i>HALL-SENSOR A1301</i>	14
3.5	<i>GPS-EMPFÄNGER SKYTRAQ VENUS 6 ST22</i>	14
3.6	<i>STEP UP REGLER TPS61200DRCT</i>	15
3.7	<i>SPANNUNGSWANDLER ADP122</i>	15
3.8	<i>FLASH-SPEICHER AT45DB642D</i>	16
3.9	<i>LIPO-AKKU-MONITOR BQ77PL900</i>	16



4	<u>TECHNISCHE REALISIERUNG</u>	16
4.1	SOFTWARE	17
4.1.1	Loggersoftware	17
4.1.2	WebServer Software	17
4.1.3	Netzwerkfunktion	18
4.2	HARDWARE	19
4.2.1	Mikrocontrollerschaltung	19
4.2.2	Beschaltung Zellenspannungsmonitor	19
4.2.3	Messschaltung Motorstrom	19
4.2.4	JTAG Schnittstelle	20
4.2.5	Spannungsversorgung	20
4.2.6	Flash-Speicher	20
4.2.7	SPI-Schnittstelle für SPB105	20
5	<u>HANDHABUNG SOFTWARE</u>	21
5.1	KONFIGURATION CLIENT	21
5.1.1	Neues Ad-hoc-Netzwerk erstellen	21
5.2	KONFIGURATION LOGGER	25
5.2.1	Sensorik einstellen	25
5.2.2	Warngrenzen einstellen	26
5.3	KOMMUNIKATION	27
5.3.1	WLAN-Einstellungen	27
5.3.2	GPS-Einstellungen	27
5.4	AUFZEICHNUNG	28
5.5	PROGRAMMIERUNG	29
5.5.1	Werkseinstellung herstellen	29
5.5.2	Firmware aufspielen	29
6	<u>HANDHABUNG HARDWARE</u>	30
6.1	STROMSENSOR	30
6.4	SCHNITTSTELLEN	32



7	<u>ERGEBNISSE</u>	33
7.1	GPS.....	33
7.2	MESSWERTE.....	33
7.3	DATEN SICHERN	34
8	<u>PROBLEME</u>	35
8.1	HARDWARE.....	35
8.1.1	Mikrocontroller.....	35
8.1.2	WLAN Platine SPB105	35
8.1.3	LDO LP2981	36
8.2	SOFTWARE.....	37
8.2.1	WLAN-API.....	37
8.2.2	Checksumme StarTraq.....	37
8.2.3	AVR32 Studio.....	38
8.2.4	WebServer	38
9	<u>ANHANG A: AUFWENDUNGEN</u>	39
9.1	VERWENDETE SOFTWARE	39
9.2	MATERIAL- / KOSTENAUFSTELLUNG	39
9.3	ZEITPLANUNG	40
9.3.1	Stundenplan.....	40
9.3.2	Wochenplan	41
10	<u>ANHANG B: QUELLENVERZEICHNIS</u>	42
11	<u>ANHANG D: TABELLENVERZEICHNIS</u>	42
12	<u>ANHANG E: DATENTRÄGERINHALT</u>	42
13	<u>ANHANG C: ABBILDUNGSVERZEICHNIS</u>	43
14	<u>DANKSAGUNG</u>	44



1 Vorstellung

1.1 Projektteam



Uwe Schmerbach
Oberes Tor 3
71126 Gäufelden

uschmer@gmx.de

Daimler AG

Software (Mikrokontroller), Hardware
Dokumentation, Pflichtenheft, Flyer



Benjamin Hirth
Kinzigstraße 41
71069 Sindelfingen

beni@driving-unlimited.de

Hörburger AG

Software (WebServer), Pflichtenheft
Präsentationsaufbau

1.2 Betreuender Lehrer



Herr Kling
Gottlieb-Daimler-Schule II
Böblinger Straße 73
71065 Sindelfingen

kling@gds2-verw.de



2 Projektinformationen

2.1 Projektidee

Da der Fortschritt der Technik auch vor dem Modellbau nicht Halt macht und auch in diesem Bereich Regelungselektronik den Einzug gehalten hat wird es für die Modellbauer immer schwieriger das jeweilige Modell zu kalibrieren und Betriebsdaten zu ermitteln.

- Die heutigen LiPo-Akkus¹ zum Beispiel müssen unbedingt vor Überhitzung und Tiefentladung geschützt werden. Dies gestaltet sich schwierig, weil der Akku bei spürbar nachlassender Leistung bereits Schaden genommen hat. Durch die sehr hohen Ströme, bis zu 450A, die diese Akkus liefern können kann sich der Akku stark erhitzen.
- Die Antriebsstränge bestehen heutzutage aus einem oder mehreren DASM² und Frequenzumrichtern³ die sehr genau eingestellt werden müssen, was nur möglich ist wenn gewisse Betriebsdaten bekannt sind, welche zuvor, teilweise sehr abenteuerlich gemessen werden müssen.
- Für den Modellbauer ist auch immer interessant wie hoch und schnell sein Modell war oder wie weit es gekommen ist. Auch die momentane oder letzte bekannte Position ist sehr interessant, vor allem wenn das Modell einmal verloren geht.

Hier kommt unser „Logger“ zum Einsatz der die Betriebsdaten während eines Einsatzes erfasst, in Echtzeit via WLAN⁴ an ein Handy oder einen Laptop übermittelt, dort speichert und anzeigt. Bei Erreichen kritischer Werte, die zuvor vom Benutzer eingestellt werden können, werden Warntöne über das jeweilige Gerät ausgegeben um sofort reagieren zu können. Nach dem Turn können die gespeicherten Werte zur Kontrolle oder Justage des Modells ausgewertet werden.

¹ Ein **Lithium-Polymer-Akku** ist ein wieder aufladbarer Energiespeicher und eine Weiterentwicklung des Lithium-Ionen-Akkus

² **Drehstrom-Asynchronmotor** auch Induktionsmotor genannt

³ Ein **Frequenzumrichter** ist ein Umrichter, der eine Spannung mit veränderbarer Frequenz generiert.

⁴ **Wireless Local Area Network** (dt: drahtloses lokales Netzwerk)



2.2 Projektziel

Ziel unseres Projektes war es also, einen Datenlogger zu entwickeln der folgende Referenzen aufweist:

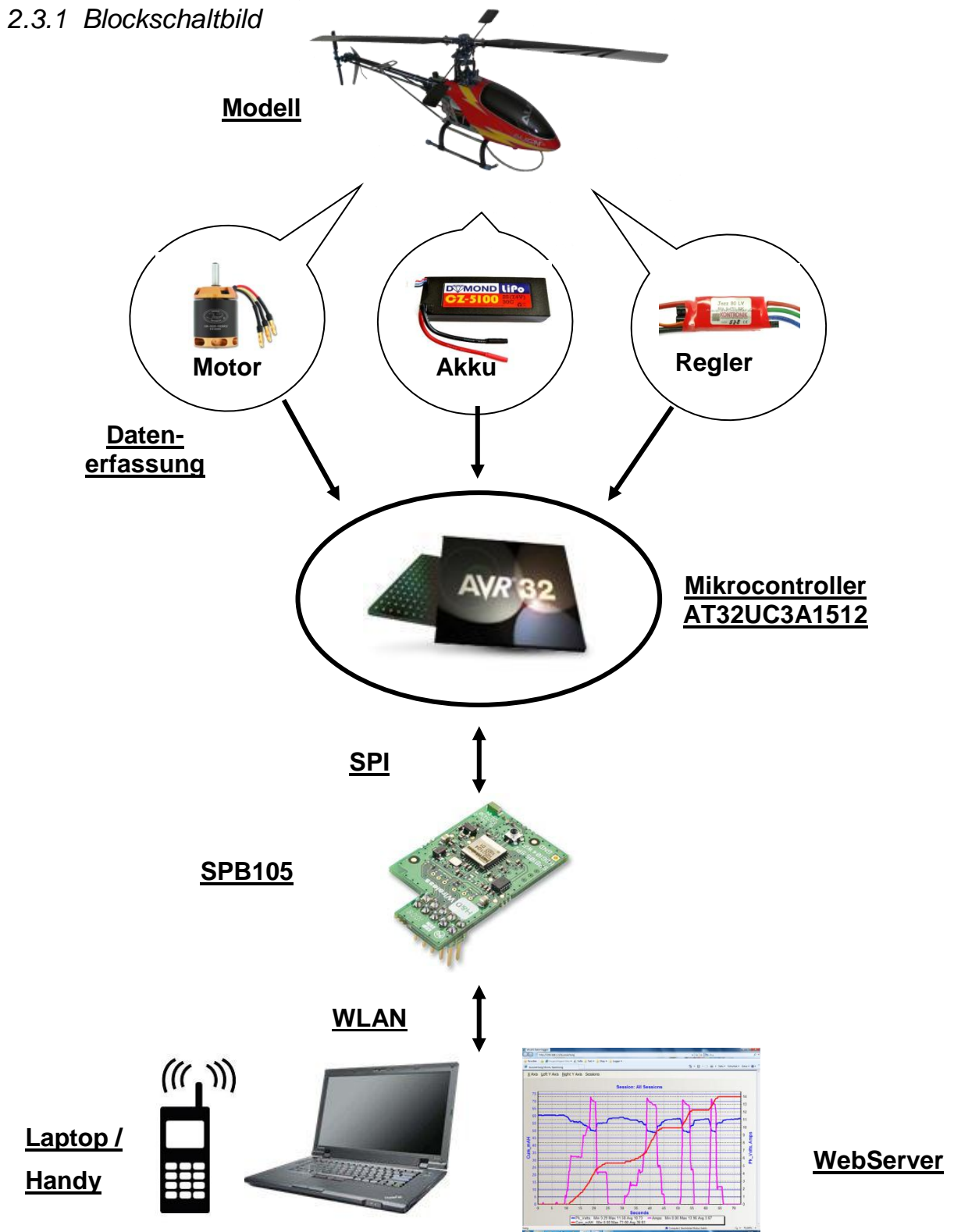
- Ausreichende Anzahl Akkuzellen messen können um auch in großen Modellen sinnvoll den oder die Akkus zu überwachen.
- Ausreichend Reserven um auch die Ströme der größeren Modelle messbar zu machen.
- Klein und leicht genug um auch in kleinen Modellen eingesetzt werden zu können.
- Die aufgenommenen Werte in Echtzeit über eine WLAN Kopplung via WebServer⁵ bereitzustellen um den Piloten rechtzeitig über Handy oder Laptop akustisch warnen zu können.
- Die erfassten Daten speichern um sie nach dem Turn für Auswertungen bereitzustellen um diese kontrollieren zu können.
- Der Preis sollte in den Regionen der Datenlogger liegen die bereits auf dem Markt erhältlich sind.
- Die Bedienung und der Einbau sollten auch von einem Laien ohne fundierte Fachkenntnisse durchgeführt werden können.
- Bei Fehlfunktion des Datenloggers sollte dem Modell kein Schaden entstehen.
- Die Visualisierung der Daten und die Konfiguration des Datenloggers sollten komplett Client unabhängig gestaltet werden, ohne etwaige Installation eines Programms oder ähnlichem um auf jedem webfähigen Gerät ausführbar zu sein.

⁵ Ein **WebServer** ist ein Computer, der Dokumente an Clients wie z. B. Webbrowser überträgt



2.3 Aufbau

2.3.1 Blockschaftbild





2.3.2 Schaltplan Loggerplatine

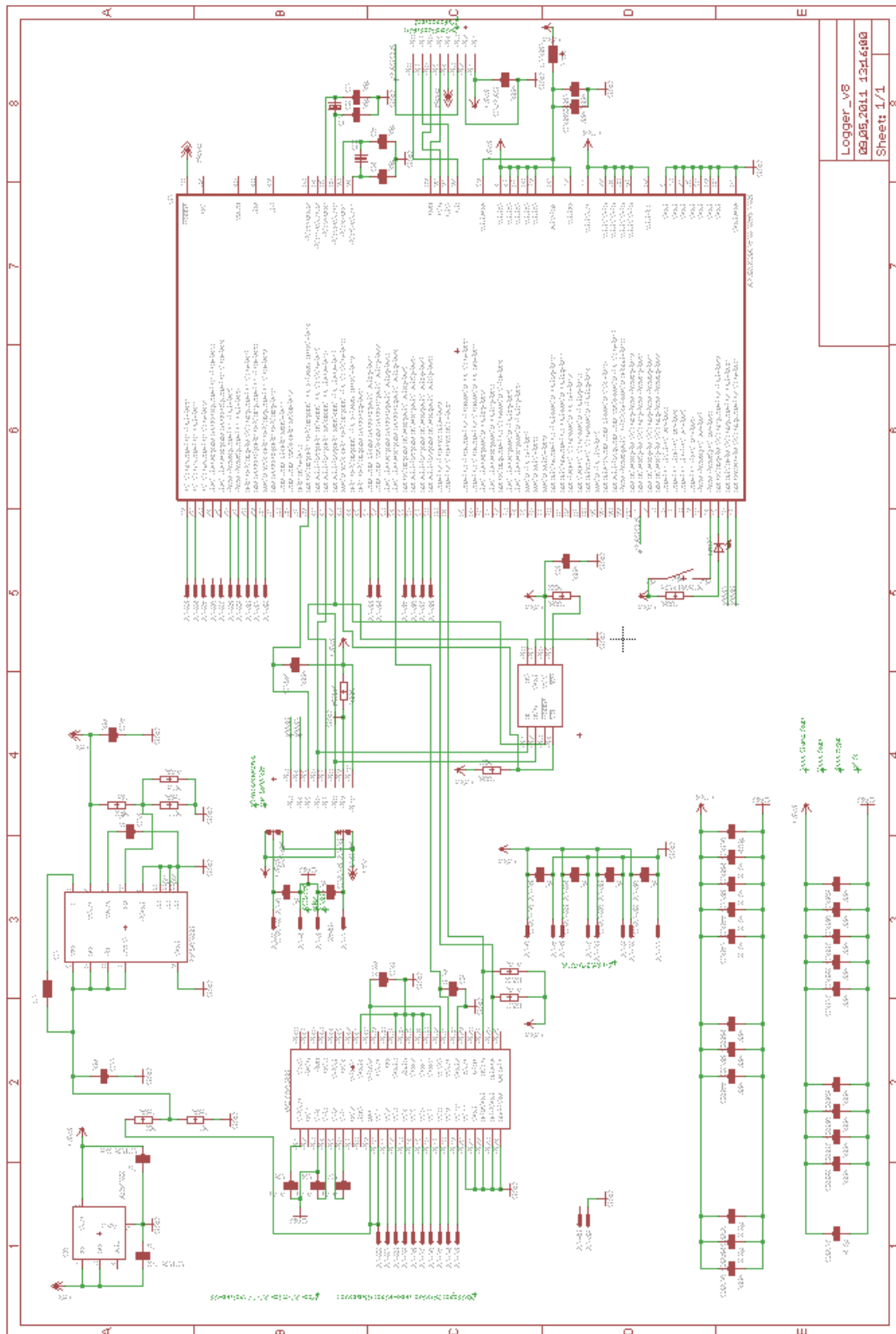


Abbildung 1: Schaltplan Loggerplatine



2.3.3 Layout Loggerplatine

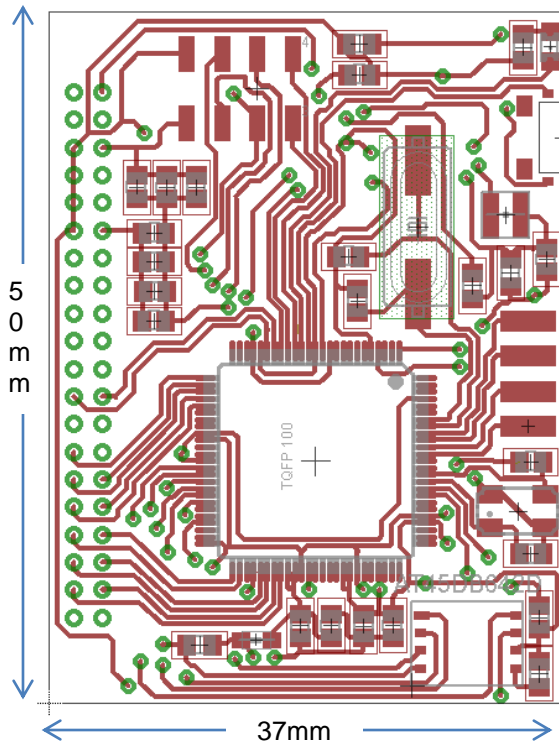


Abbildung 2: Layout Top

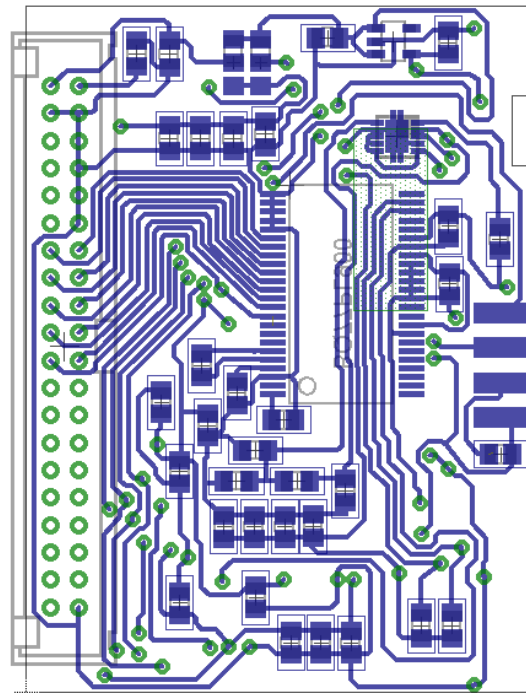


Abbildung 3: Layout Bottom

2.3.4 Web-Server

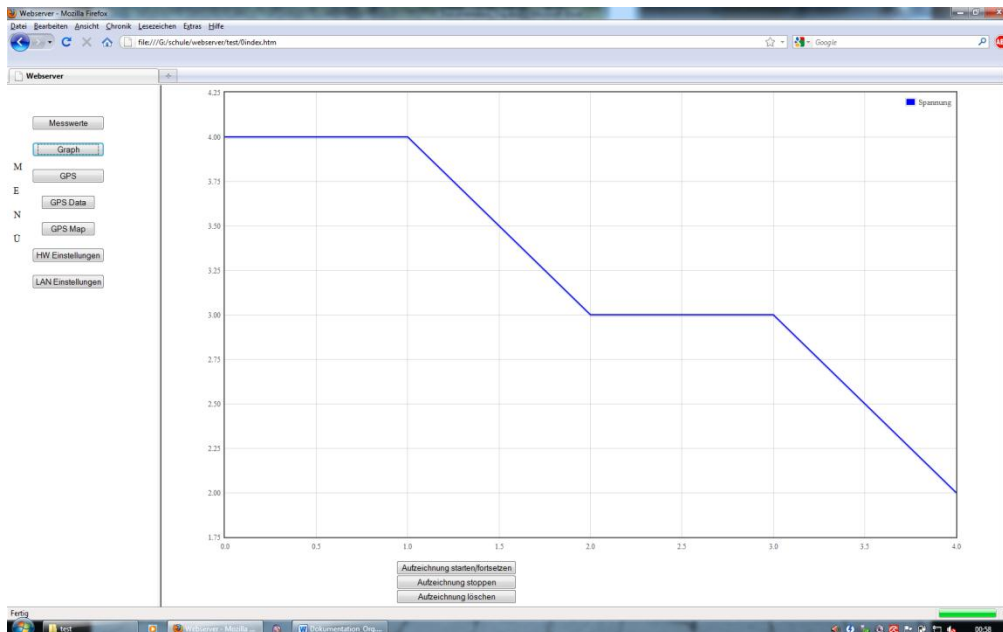


Abbildung 4: Web-Server



2.4 Ablauf

- Die Daten und Signale vom Modell werden mit Hilfe von Sensoren aufgenommen, an die Signalpegel des Mikrocontrollers „AT32UC3A1512“ der Firma „Atmel“ angepasst, dort aufbereitet und anschließend über eine WLAN- Platine „SPB105“ der Firma „H&D Wireless“ und einen WebServer, der von jedem WLAN tauglichen Client⁶ aus erreichbar ist, bereitgestellt.
- Der Datenaustausch zwischen Mikrocontroller und WLAN Platine erfolgt über den SPI-Bus⁷.
- Die Versorgungsspannung für den Datenlogger wird von dem Balanceranschluss⁸ des Akkus abgegriffen, über welchen auch die Spannung und Restkapazität - von maximal zehn Zellen - des Akkus gemessen wird.
- Während des Turns wird ein Zellenbalancing⁹ durchgeführt um Maximale Leistung und Lebensdauer des Akkus zu erzielen.
- Die Messwerte der Drehzahl sowie der Stromstärke werden an den Motorleitungen des Frequenzumrichters berührungslos über einen Hall Sensor¹⁰ abgegriffen, so muss nicht extra ein Bauteil in den Antriebsstromkreis eingelötet werden.
- Motor-, Regler- und bis zu 2 Akkutemperaturen werden über jeweils direkt angebrachte Sensoren ermittelt.
- Die Kommunikation zum GPS Empfänger wird über eine TTY-Schnittstelle empfangen und an den WebServer weiter geleitet.
- Während eines Turns werden Maximalwerte sowie die GPS-Daten in einem Flash-Speicher für spätere Auswertungen gespeichert.

3 Hardware

⁶ Ein **Client** (dt: Kunde) ist ein Computer, der Kontakt zu einem anderen Computer, dem Server, aufnimmt, um dessen Dienstleistung zu nutzen

⁷ Das **Serial Peripheral Interface** ist ein Bus-System für einen synchronen seriellen Datenaustausch

⁸ Ein **Balanceranschluss** ist ein weiterer Steckeranschluss eines Akkus, auf dem jeder Zellenkontakt einzeln herausgeführt ist

⁹ **Zellenbalancing** bedeutet ein ausgleichen der Ladung einzelner Zellen

¹⁰ Ein **Hall-Sensor** misst die Stärke von Magnetfeldern



Hier möchten wir die von uns verwendeten Bauteile vorstellen und zeigen welchem Zweck sie dienen und erklären weshalb wir uns gerade für diese Bauteile entschieden haben.

3.1 *Controller AT32UC3A1512*

Beim UC3A, wie der AT32UC3Axxxx umgangssprachlich genannt wird, handelt es sich um einen 32bit Mikrocontroller der Firma „Atmel“. Er basiert auf einer RISC¹¹ Architektur und bietet in der Ausführung wie sie bei uns als Herzstück der Platine eingesetzt wird 512k Byte Flashspeicher¹² und 64k Byte SRAM¹³. Trotz dieses großzügigen Speicherangebots und der für unsere Ansprüche mehr als ausreichenden integrierten Hardware, ist dieser Controller in einem TQFP-100¹⁴ Gehäuse untergebracht, was uns wegen der kleinen Dimensionen (14x14mm) dieses Gehäuses sehr wichtig war. Ein weiteres für uns wichtiges Merkmal des UC3 ist die geringe Stromaufnahme von nur 36mA im aktiven Modus, da wir an unseren Logger den Anspruch stellen den Akku, der eigentlich für das Modell eingesetzt ist, nicht zu sehr zu belasten.

3.2 *WLAN-Platine SPB105*

Diese Platine der Firma „HD-Wireless“, eines offiziellen Partners von „Atmel“ der auf drahtlose Kommunikation spezialisiert ist, stellt die physikalische Ebene der WLAN Kopplung dar, sie wird mit den zu sendenden Daten per SPI-Bus versorgt und wandelt diese in Funkwellen auf 2,4GHz Basis um. Wir haben uns für diese Platine entschieden weil auch sie von den Dimensionen her sehr klein gehalten, fertig kalibriert ausgeliefert wird und die API¹⁵ die die Platine handelt vollständig in das Software Framework¹⁶ von Atmel integriert ist, was die Programmierung eigentlich hätte erleichtern sollen.

3.3 *Temperatur-Sensor MCP9701A*

¹¹ **Reduced Instruction Set Computer** (dt: Rechner mit reduziertem Befehlssatz)

¹² **Flash**, ein nichtflüchtiger Speicher

¹³ **Static Random Access Memory** (dt: statisches RAM) ein flüchtiger Speicher

¹⁴ **TQFP**: Thin Quad Flat Package (dt: Flaches quadratisches Gehäuse) Gehäusebauform für Integrierte Schaltungen mit **100** Anschlüssen

¹⁵ **Application Programming Interface** (dt. Schnittstelle zur Anwendungsprogrammierung)

¹⁶ **Software Framework** (dt: Bereitgestellte Software Bibliothek)



Dieses Low-Power Linear Active Thermistor IC¹⁷ von „Microchip“, welches Temperatur in Spannung konvertiert, wurde speziell zur Erfassung von Temperaturen mit Mikrocontrollern entwickelt. Da es sich hier um einen Integrierten Schaltkreis handelt hat dieser Sensor einen Wertebereich von nur -40° bis 125°C was für unsere Anforderungen jedoch völlig ausreichend ist, dafür gibt er eine Linearisierte Kennlinie von 19,5mV pro °C aus und hat von 0° bis 70° eine sehr geringe Toleranz von nur ±2%. Mit diesem IC messen wir die Temperaturen an bis zu 2 Akkus, dem Motor und dem Regler, wobei die Unterbringung der Sensoren im Ermessen des Benutzers liegt.

3.4 Hall-Sensor A1301

Der A1301 von „Allegro“ misst die Intensität eines ihn durchdringenden Magnetfeldes, abhängig von dieser Intensität gibt er 2,5mV / Gauß¹⁸ aus. In Verbindung mit einem Ferritkern, der die erzeugten Magnetfelder eines mit stromdurchflossenen Leiters in sich bündelt, kann der A1301 den gesamten magnetischen Fluss um diesen Leiter messen und die Intensität in Spannung ausgeben. Durch diese kann nun mit Hilfe der Formel $I = \frac{B \cdot l}{\mu \cdot N}$ der Strom errechnet werden der diesen gemessenen Leiter durchfließt. Mit diesem Sensor ermitteln wir den Strom in einem der drei Motorleitern sowie die Frequenz mit der der Motor angesteuert wird. Diese Frequenz lässt sich, wenn man den Schlupf¹⁹ des Motors außen vor lässt, anhand der Formel $n = \frac{f}{p} * \left(\frac{60s}{min}\right)$ wiederum in die ungefähre Drehzahl des Motors umrechnen. Mit dieser Drehzahl wiederum lässt sich, durch das Übersetzungsverhältnis des Getriebes, die Drehzahl am Rotor oder Antriebsrad berechnen.



Abbildung 5: Hall-Sensor in Ferritring

3.5 GPS-Empfänger Skytraq Venus 6 ST22

¹⁷ **Low-Power Linear Active Thermistor IC** (dt: Niedrig Energie, linearer- aktiver-Temperatursensor)

¹⁸ **Gauß** (Magnetische Flussdichte)

¹⁹ Der **Schlupf** ist die Drehzahl-Differenz zwischen Motoransteuerung und tatsächlicher Drehzahl und beträgt normal zwischen 1-8%



Der Skytraq Venus 6 ST22 von „Perthold Engineering“ ist ein kompakter GPS²⁰ Empfänger mit einer für uns gut geeigneten TTY Schnittstelle²¹ was bedeutet, dass wir ihn ohne weiteres an die UART²² Schnittstelle unseres Controllers anschließen können. Für uns ist der Skytraq aber nicht wegen der Schnittstelle besonders gut geeignet sondern wegen seiner Fähigkeit die GPS Koordinaten besonders häufig, nämlich 10 mal pro Sekunde zu aktualisieren. Dies ist bei den schnellen Bewegungen der Modelle in denen unser Datenlogger eingesetzt werden soll unumgänglich um ein genaues Tracking²³ zu erhalten. Weiter wird der GPS-Empfänger verwendet um die Uhrzeit, die Geschwindigkeit und Höhe des Modells zu ermitteln, da diese Daten in den NMEA²⁴ Protokollen GGA²⁵ und RMC²⁶ enthalten sind.

3.6 *Step up Regler TPS61200DRCT*

Als Haupt Spannungsversorgung haben wir den Step up Regler TPS61200DRCT von „Texas Instruments“ eingesetzt der von den Akkus, mit einer variablen Eingangsspannung von minimal 2,9V bis maximal 16V, versorgt wird und die komplette Platine mit stabilen 5V betreibt. Treiben kann dieser, mit 9mm² von der Bauart sehr kleine Regler bis zu einem Ampere, was für unsere Bedürfnisse mehr als ausreichend ist.

3.7 *Spannungswandler ADP122*

Für die Versorgung des Controllers und der übrigen 3,3V Bauteile kommt der Spannungswandler ADP122 von „Analog Devices“ zum Einsatz der wiederum von den 5V des TPS61200DRCT versorgt wird. Hier war es wichtig einen Regler zu finden, der eine relativ saubere Gleichspannung liefert um den Mikrocontroller zu schützen und gleichzeitig sehr geringe Ausmaße annimmt. Dies sollte mit dem ADP122, der nur 2,9 x 1,6mm groß ist, gelungen sein.

²⁰ **Global Positioning System** (dt: globales Navigationssatellitensystem zur Positionsbestimmung)

²¹ Die **TTY-Schnittstelle** ist der Vorgänger der Seriellen Schnittstelle oder auch RS232

²² **Universal Asynchronous Receiver Transmitter** (dt. Universeller nicht-synchroner Empfänger/Überträger)

²³ **Tracking** (dt: Positionsmitschrift)

²⁴ **National Marine Electronics Association** (Nationale Vereinigung für Marineelektronik)

²⁵ **GGA** Global Positioning System Fix (Positionskoordinaten)

²⁶ **Recommended Minimum Sentence C** (dt: empfohlener Minimumdatensatz)



3.8 Flash-Speicher AT45DB642D

Das 4MB große Daten Flash²⁷ von „ATMEL“ wird bei uns, wegen der, nur in dieser Ausführung enthaltenen, SPI-Schnittstelle, in der Ausführung Cason eingesetzt. Der SPI-Bus war sowieso wegen der WLAN Platine vorgesehen und da dieses Flash in einem sehr kleinen Cason Gehäuse mit 6x8mm sehr wenig Platz verbraucht, bot sich dieses Flash an, um die Möglichkeiten zu verbessern, Aufzeichnungen des Turns zu speichern um sie nach dem Turn abrufbar zu machen. Die HTML Seiten des WebServers werden ebenfalls auf dieses Flash gespeichert und können so, ohne auf den Speicherbedarf achten zu müssen, ausführlicher gestaltet werden.

3.9 LiPo-Akku-Monitor bq77PL900

Dieser Lithium-Polymer Akku Monitor mit Analog Front End ist ein IC von „Texas Instruments“, welches unter anderem zur Akku Überwachung eingesetzt werden kann. Es kann bis zu 10 Einzelzellenspannungen über einen integrierten Multiplexer an einen Ausgang weiterleiten. Diese weitergeleiteten Spannungen können über integrierte Level Shifter²⁸ reduziert werden, so dass diese Spannung an einem Mikrocontroller über einen Analog-Digital-Wandler eingelesen werden können. Die Ansteuerung des Multiplexers und des Level-Shifters wird über eine IIC²⁹ Schnittstelle zum Mikrocontroller realisiert. Gleichzeitig hat dieses IC einen Zellenbalancer integriert der Ausgleichsströme von bis zu 10mA zulässt, dies ist von großem Vorteil da die Akkus dadurch langlebiger und leistungsstärker werden.

3.10 Strom-Sensor ACS756

Der ACS756 von „Allegro“ misst den ihn durchfließenden Strom und gibt die Intensität als Spannung aus, 40mV entsprechen hier 1A. Leider ist der Strom der hier gemessen werden kann durch die Verlustleistung des Bauteils auf 100A begrenzt was den Einsatz nur in kleineren Modellen erlaubt.

4 Technische Realisierung

²⁷ Auch **Flash-EEPROM**, digitale Speicher, nicht flüchtig, wiederbeschreibbar

²⁸ Als **Level Shifter** (dt: Pegelwandler) bezeichnet man eine Schaltung die Spannungssignale einer Informationsquelle an die Eingangssignalpegel einer Informationsempfänger anpasst

²⁹ Ein **Inter-Integrated Circuit** ist ein serieller Datenbus, auch TWI (Two Wire Interface) genannt



4.1 Software

Hier möchten wir zeigen wie die Software des Loggers entstanden ist und wie wir dabei vorgegangen sind.

4.1.1 Loggersoftware

Vor dem eigentlichen Entwickeln der Software stand die Informationsbeschaffung. Hier wurden bereits die Grundsteine des Datenloggers gelegt, wir entschieden uns für einen Atmel Controller. Hier schien das EVK1105 in Verbindung mit dem SPB105 eine gute Basis darzustellen um Erfahrungen zu sammeln und Tests mit der WLAN-Technik durchzuführen. Als das EVK1105 beschafft war, hieß es sich in die neue Programmiersprache und die Entwicklungsumgebung AVR32 Studio einzuarbeiten. Hierzu wurde in erster Linie auf das „µTasker“ Basisprojekt von „M.J.Butcher Consulting“ zurückgegriffen. Die erste Herausforderung war das Umdenken auf die neue Programmiersprache und die neue Programmierumgebung, der bereits bekannte C-Syntax musste stark erweitert werden, da die 32bit Programmierung mehr Möglichkeiten eröffnete und diese in dem Basisprojekt auch umgesetzt wurden. Nachdem die grundlegenden Kenntnisse aus dem Projekt über die Arbeitsweise der Programmiersprache GNU GCC und AVR Studio 32 durch das Experimentieren mit dem Projekt vertieft waren, konnte man sich einem eigenen kleinen Projekt zuwenden, wie zum Beispiel die Integration der TTY-Schnittstelle für den GPS Empfänger. Hier konnten auch gleich die Kenntnisse des WebServers vertieft werden da die gelesenen Daten auf diesem angezeigt werden sollten. Jedoch war es hier nötig einen Debugger, den „AVR JTAG MKII“ von Atmel anzuschaffen, da das doch komplizierte Basisprojekt durch das debuggen einfacher zu verstehen und nun genauer nachzuverfolgen war, was und wie denn genau in dem Programm abgearbeitet wird. Als dieses Projekt von Erfolg gekrönt war, widmete man sich der API welche in einem Beispielprojekt des Atmel veranschaulicht wird. Nun wurde versucht die API in das Basisprojekt zu integrieren was leider wegen der fehlenden Standard Bibliothek im µTasker Projekt scheiterte. Dann wurden Teile des µTasker Projekts in das Beispielprojekt von Atmel eingebunden, nach und nach entstand so das jetzige Projekt mit Teilen beider Projekte die als Basis dienten, worauf dann aufgesetzt wurde und die eigentliche Applikation realisiert wurde.

4.1.2 WebServer Software



Da das Abrufen der Werte von jedem Gerät ohne Installation eines zusätzlichen Programms möglich sein sollte, haben wir uns entschieden dies durch einen Webserver zu realisieren. Dazu musste zuerst einmal herausgefunden werden, welche Programmiersprachen und Dateien von dem Webserver der im μ Tasker Projekt integriert ist, unterstützt werden. Dabei wurde festgestellt, dass der Webserver kein PHP unterstützt, weshalb alles mit Javascript programmiert werden musste. Das erschwerte vor allem das Speichern und Anzeigen der Messdaten, da Javascript nur wenig Möglichkeiten für den Dateizugriff bietet. Die ersten Tests mit den Javascript Dateien zeigten, dass die Größe der Dateien des Webserver begrenzt ist. Deshalb mussten Funktionen neu geschrieben und auf mehrere Dateien aufgeteilt werden.

4.1.3 *Netzwerkfunktion*

Da im Atmel Beispiel-Projekt bereits eine Applikation vorhanden war, um eine Netzwerkverbindung - über das Display gesteuert - auszuwählen, konnten hier mit dem EVK1105 einige Tests durchgeführt werden. Leider stellte sich schnell heraus, dass eine Ad-Hoc-Verbindung noch nicht implementiert war. Nach einiger Recherche und Kontaktaufnahme mit HD-Wireless, dem Entwickler der API für das SPB105, stellte sich heraus, dass eine Ad-Hoc-Verbindung erst in der nächsten Version implementiert werden würde, die eigentlich für unser Projekt rechtzeitig auf den Markt kommen sollte. Eine Verbindung zu einem Router war jedoch möglich und so konnte die Applikation für unser Projekt geschrieben werden. Als dann die neue Version der API erhältlich war, musste zu unserem Nachteil erst auf eine neue Version der Entwicklungsumgebung umgestiegen werden, weil die Einstellungen für die neue API in der alten Umgebung nicht bekannt waren und auch nicht herausgefunden werden konnten. Als dann aber endlich auch die neue Umgebung lief und das Projekt auf die neuen Einstellungen umgestellt war, stellte sich heraus, dass eine Ad-Hoc-Verbindung immer noch nicht möglich war und bis heute nicht möglich ist. Nach weiteren Anfragen und intensiver Kommunikation mit HD-Wireless konnten Fehler in der API gefunden werden die in der nächsten Version der API beseitigt werden sollen. Deshalb ist es uns leider, bis zum Erscheinen der Version 2.8, nicht möglich mit dem Logger eine Ad-Hoc-Verbindung herzustellen.



4.2 Hardware

Hier zeigen wir auf, wie die Hardware entstanden ist, wie aus einzelnen Komponenten ein Datenlogger auf einer Platine entstanden ist.

4.2.1 Mikrocontrollerschaltung

Die Beschaltung des Controllers ist in dem dazugehörigen über 800 Seiten umfassenden Datenblatt sehr gut beschrieben, was die Erstellung des Schaltplans recht einfach machte. Die hohe Anzahl an Kondensatoren, die laut Datenblatt für saubere Spannungen und Entkopplung³⁰ sorgen sollen, jedoch, erschwerten das Layout doch sehr da die Kondensatoren auch unterschiedliche Prioritäten haben was die örtliche Nähe zum Controller angeht. Deshalb, und weil der Platz auf der Platine sehr knapp bemessen ist wurden einige Kondensatoren die überflüssig erschienen weg gelassen.

4.2.2 Beschaltung Zellenspannungsmonitor

Da nur die Grundfunktionen des Monitors genutzt werden war die Beschaltung sehr einfach und konnte aus dem Datenblatt kopiert werden. Dies beschränkte sich auf einige Kondensatoren zur Entstörung. Der IIC-Bus der als Datenschnittstelle dient wurde direkt verbunden und mit zwei Pull-up-Widerständen auf das richtige Grundpotential gezogen.

4.2.3 Messschaltung Motorstrom

Es stehen zwei Stromsensoren zur Auswahl, der A1301 der kontaktlos misst und der ACS756 der in den Stromkreis eingebaut werden muss. Diese Alternative machte es nötig 3,3V und 5V für den Sensor bereitzustellen, dies wurde über Brücken auf der Platine realisiert. Beide Sensoren haben Spannungsausgaben die die Grenze des Analog-Digital-Wandlers nicht überschreiten, was eine besondere Schaltung überflüssig macht. Lediglich Kondensatoren zur Entstörung wurden eingeplant.

³⁰ Ein **Koppelkondensator** wird eingesetzt um ohne Beeinflussung der Wechselstromverhältnisse einer Schaltung eine Gleichspannung von ihr fern zu halten.



4.2.4 JTAG Schnittstelle

Die Beschaltung der JTAG Schnittstelle konnte 1 zu 1 aus dem Schaltplan des EVK1105 übernommen werden da diese nach Standard ausgeführt wurde. Für die Steckkontakte allerdings musste eine andere Lösung gefunden werden, weil es uns unnötig erschien einen Stecker auf die Platine zu löten der nur ein einziges Mal Verwendung findet um das Programm auf den Logger zu spielen. Hier kamen wir dann auf die Gleitkontakte, da sie wenig Platz benötigen, sehr leicht sind und vor allem nichts kosten.

4.2.5 Spannungsversorgung

Hier fiel die Entscheidung auf zwei unterschiedliche Spannungsregler. Der Step up, der die 5V Versorgung sicherstellt, wird von den Akkus variabel versorgt und machte daher eine etwas kompliziertere Beschaltung notwendig. Diese konnte aber nach Angaben im Datenblatt berechnet und umgesetzt werden. Der zweite Regler wird vom ersten mit 5V versorgt und gibt sehr saubere 3,3V aus, dieser Regler benötigt, da es ein Festspannungsregler ist, nur zwei Kondensatoren zur Entkopplung.

4.2.6 Flash-Speicher

Das Speichermodul wird direkt mit den zwei SPI-Bus Signalen SDA und SCL des Controllers verbunden und benötigt nur einen Koppelkondensator. Wichtig war hier den ChipSelect und WriteProtect Eingang richtig zu beschalten um auf das Flash auch zugreifen zu können.

4.2.7 SPI-Schnittstelle für SPB105

Die SPI Schnittstelle zum WLAN Modul wurde nach Vorgabe des Schaltplans zu dem Modul ausgeführt. Hier war noch Klärungsbedarf, da die Schnittstelle zum Modul auf dem EVK1105 anders ausgeführt war als im Schaltplan beschrieben. Allerdings konnte diese Differenz schnell geklärt werden, das EVK1105 hat zwei weitere Signale auf der Schnittstelle aufliegen um auch mit anderen Modulen kommunizieren zu können, weiter war die Belegung des EVK1105 nicht für das SPB105 ausgelegt, was die Verwirrung natürlich perfekt machte.



5 Handhabung Software

5.1 *Konfiguration Client*

Das Gerät mit dem sich der Datenlogger verbinden soll muss erstmals auf die Werkseinstellungen des Loggers konfiguriert werden. Bei erfolgreicher Verbindung können die Verbindungsdaten des Loggers geändert und gespeichert werden.

Hier möchten wir zeigen wie der Client, also der Laptop oder das Handy das sich mit dem Logger verbinden soll konfiguriert werden muss.

5.1.1 *Neues Ad-hoc-Netzwerk³¹ erstellen*

Folgende Einstellungen sind für eine Verbindung zu tätigen wenn der Logger mit den Werkseinstellungen betankt ist:

SSID des Netzes ³² :	Datenlogger
Verschlüsselung:	Keine
IP-Adresse des Gateways ³³ :	192.168.178.001
Subnetmask des Gateways:	255.255.255.000

Bei Windows 7 wird eine WLAN Verbindung wie folgt eingerichtet.

- Mit rechter Maustaste auf das WLAN-Symbol in der Taskleiste klicken.



Abbildung 6: WLAN-Symbol

- „Netzwerk- und Freigabecenter öffnen“

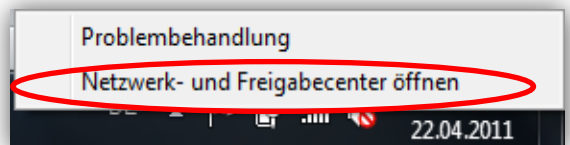


Abbildung 7: WLAN Eigenschaften

- „Drahtlosnetzwerk verwalten“

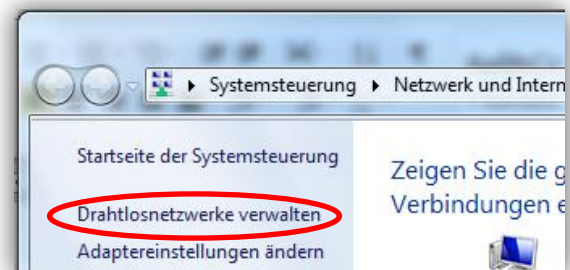


Abbildung 8: Drahtlosnetzwerk verwalten

³¹ **Ad-Hoc**: Punkt-zu-Punkt Kopplung mit WLAN

³² **Service Set Identifier**, Bezeichnung eines Drahtlosnetzwerkes

³³ Ein **Gateway** ist der Zugangspunkt zum Netzwerk, in den meisten Fällen ein Router



- „Hinzufügen“

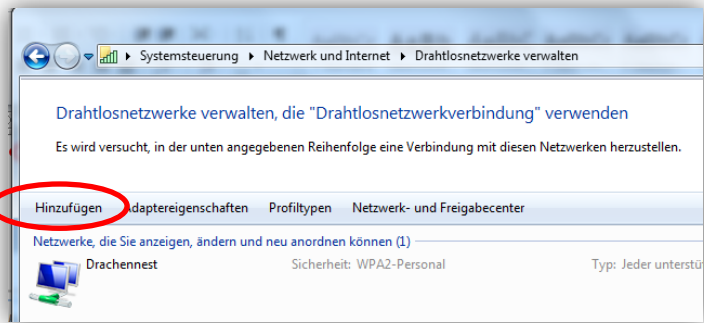


Abbildung 9: Drahtlosnetzwerk hinzufügen

- „Ad-hoc-Netzwerk erstellen“

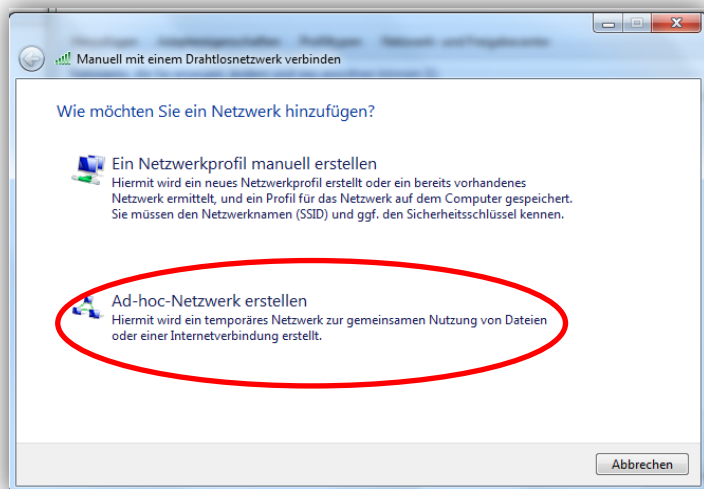


Abbildung 10: Ad-hoc-Netzwerk erstellen

- Die nächste Information mit „Weiter“ bestätigen

- Netzwerkname:
„Drachennest“
- Sicherheitstyp:
„Keine Authentifizierung“
- Mit „Weiter“ die Eingabe bestätigen

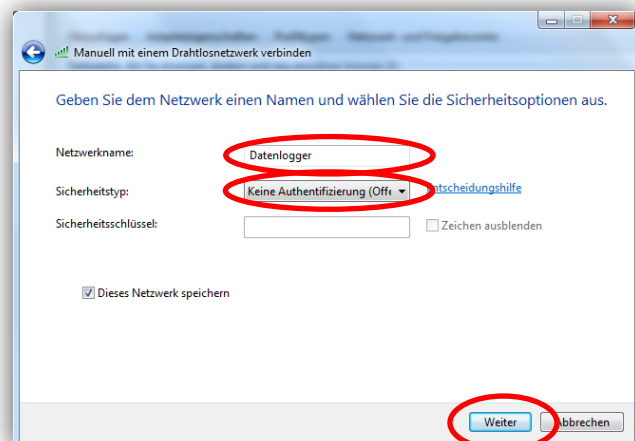


Abbildung 11: WLAN Einstellungen



5.1.2 Netzwerkeinstellungen für den WLAN-Adapter

- „Adaptoreinstellungen ändern“

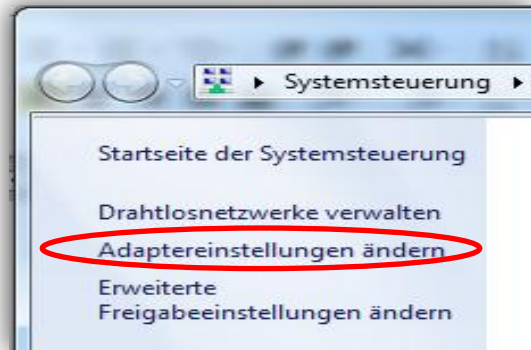


Abbildung 12: Adaptoreinstellungen ändern

- Doppelklick auf „Drahtlosnetzwerkverbindung“

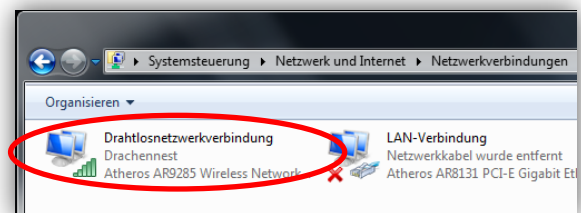


Abbildung 13: Drahtlosnetzwerkverbindung

- „Eigenschaften“

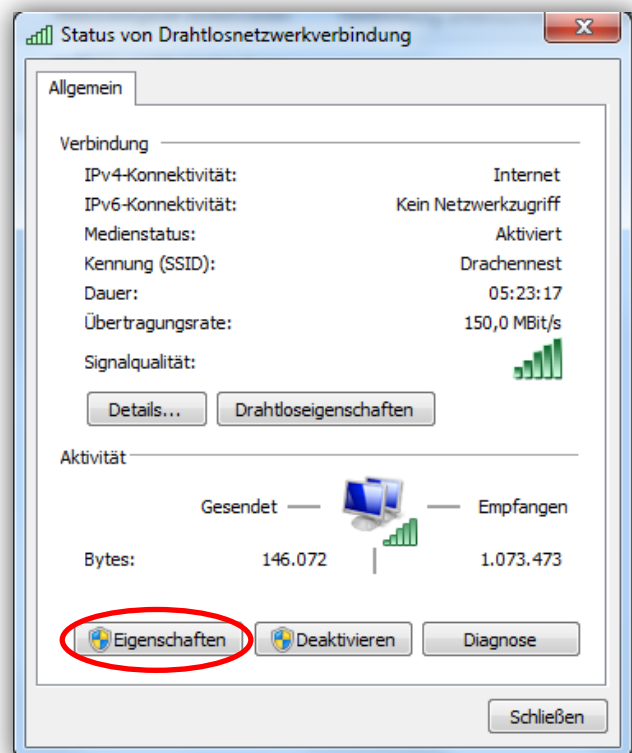


Abbildung 14: Status Drahtlosnetzwerkverbindung



- Doppelklick auf „Internetprotokoll Version 4 (TCP/IPv4)“

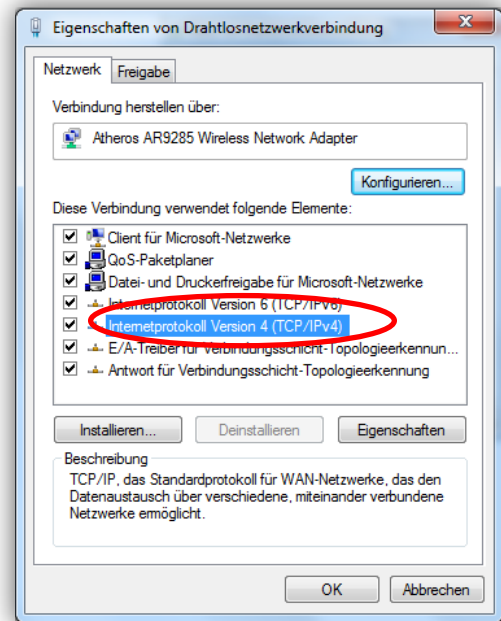


Abbildung 15: Netzwerkeigenschaften

- IP Adresse:
„192.168.178.1“

Subnetzmaske:
„255.255.255.0“

Eingeben und mit „OK“
bestätigen

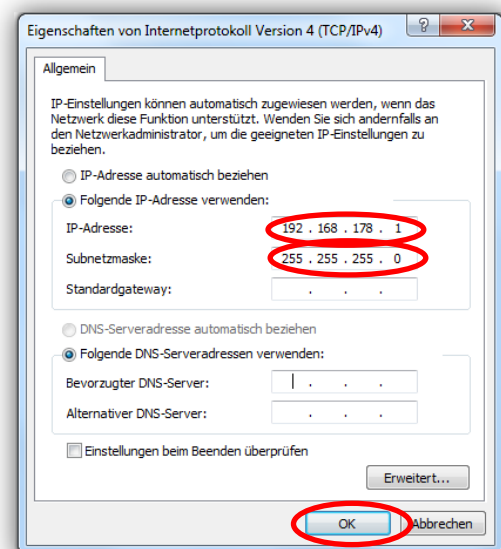


Abbildung 16: IPv4 Eigenschaften



5.2 Konfiguration Logger

Hier möchten wir zeigen wie der Logger auf eigene Wünsche hin konfiguriert werden kann.

5.2.1 Sensorik einstellen

Die an den Logger angebrachte Sensorik muss im Logger eingestellt werden. Im Bild ist zu sehen wie dem Logger die Sensorik bekannt gemacht wird.

- Im Menü auf HW Einstellungen klicken

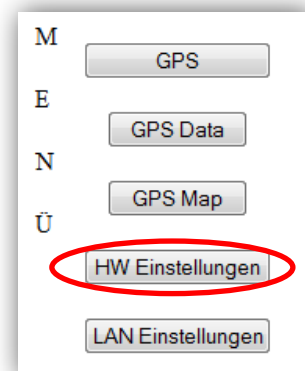


Abbildung 17: HW Einstellungen

- Den am Eingang angeschlossenen Sensor auswählen

Mit „Einstellungen übernehmen“ bestätigen

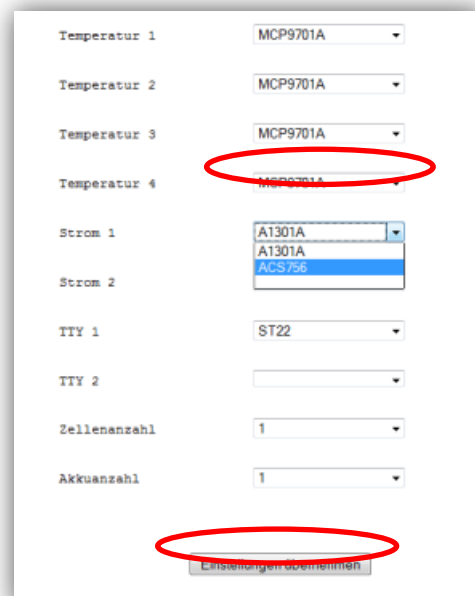


Abbildung 18: Sensor auswählen



5.2.2 Warngrenzen einstellen

Um eine Akustische Warnung bei Erreichen der Warngrenzen zu erhalten müssen die Grenzen erst definiert werden. Im Bild ist die Vorgehensweise beschrieben wie diese Grenzen eingestellt werden.

- Im Menü auf „HW Einstellungen“ klicken

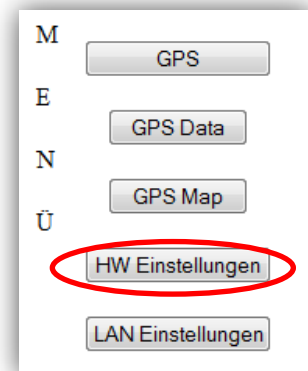


Abbildung 19: HW Einstellungen

- Wert im Textfeld für die entsprechende Warngrenze eingeben

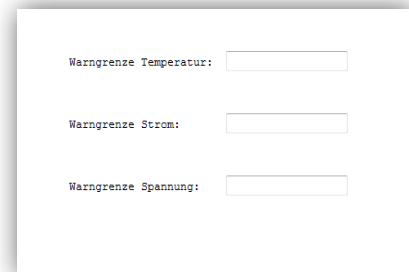


Abbildung 20: Warngrenzen



5.3 Kommunikation

Ziel dieses Kapitels ist es aufzuzeigen welche Einstellungen vorgenommen werden müssen um die Schnittstelle zu parametrieren.

5.3.1 WLAN-Einstellungen

Im Folgenden wird erklärt wie die Netzwerkeinstellungen des Loggers verändert werden können.

- Die Einstellungen werden auf der Seite „LAN Einstellungen“ geändert.
- Dazu gibt man die Daten in die entsprechenden Felder ein und wählt die Einstellungen aus die man haben möchte
- Danach drückt man den Button „Save changes“
- Anschliessend „Modify/validate settings“ drücken, um die Änderungen zu bestätigen.

WLAN - Einstellungen

Setting	Value		Modified
IP address	<input type="text" value="?vli"/>	<input type="text" value="?vid"/>	<input type="checkbox"/>
Subnet mask	<input type="text" value="?vls"/>	<input type="text" value="?vlm"/>	<input type="checkbox"/>
Gateway IP address	<input type="text" value="?vlg"/>	<input type="text" value="?vlg"/>	<input type="checkbox"/>
IEEE 802.11 speed	b (11MBit) <input type="radio"/>	g (54MBit) <input type="radio"/>	<input type="checkbox"/>
Configure using DHCP server	<input type="checkbox"/> (set IP to 0.0.0.0 if no preferred setting)		<input type="checkbox"/>
Encryption	Wep <input type="radio"/>	WPA <input type="radio"/>	<input type="checkbox"/>
Key	<input type="text" value="?vM0"/>		<input type="checkbox"/>
SSID	<input type="text" value="?vM0"/>		<input type="checkbox"/>
Settings validated	<input type="checkbox"/>		<input type="checkbox"/>

When not set, the device is waiting for validation after a network setting change

Abbildung 21: LAN Einstellungen

5.3.2 GPS-Einstellungen

Um angeschlossene Module, die über eine Schnittstelle kommunizieren, ansprechen zu können müssen diese dem Logger erst bekannt gemacht werden. Im Bild ist die Vorgehensweise zu sehen wie dies durchzuführen ist.

- Mit dem Menübutton „GPS“ kommt man zu der Seite mit den GPS Einstellungen.
- Dort wählt man die passenden Einstellungen für das angeschlossene Modul und drückt den Button „Modify settings“.

GPS - Einstellungen

Setting	Value		
Serial Speed	9600 <input type="radio"/>	38400 <input type="radio"/>	115400 <input type="radio"/>
Poll Rate	1 Hz <input type="radio"/>	4 Hz <input type="radio"/>	10 Hz <input type="radio"/>
Messages	GGA <input type="checkbox"/>	GSA <input type="checkbox"/>	GSV <input type="checkbox"/>
	GLL <input type="checkbox"/>	RMC <input type="checkbox"/>	VTG <input type="checkbox"/>
	ZDA <input type="checkbox"/>		

Abbildung 22: GPS Einstellungen



5.4 Aufzeichnung

Wenn ein Gerät mit dem Datenlogger verbunden ist, können die Daten auf dem Gerät gespeichert werden. Die Daten bleiben auch nach dem Beenden der Verbindung gespeichert und die Aufzeichnung kann beim nächsten Verbinden fortgesetzt, gespeichert oder gelöscht werden.

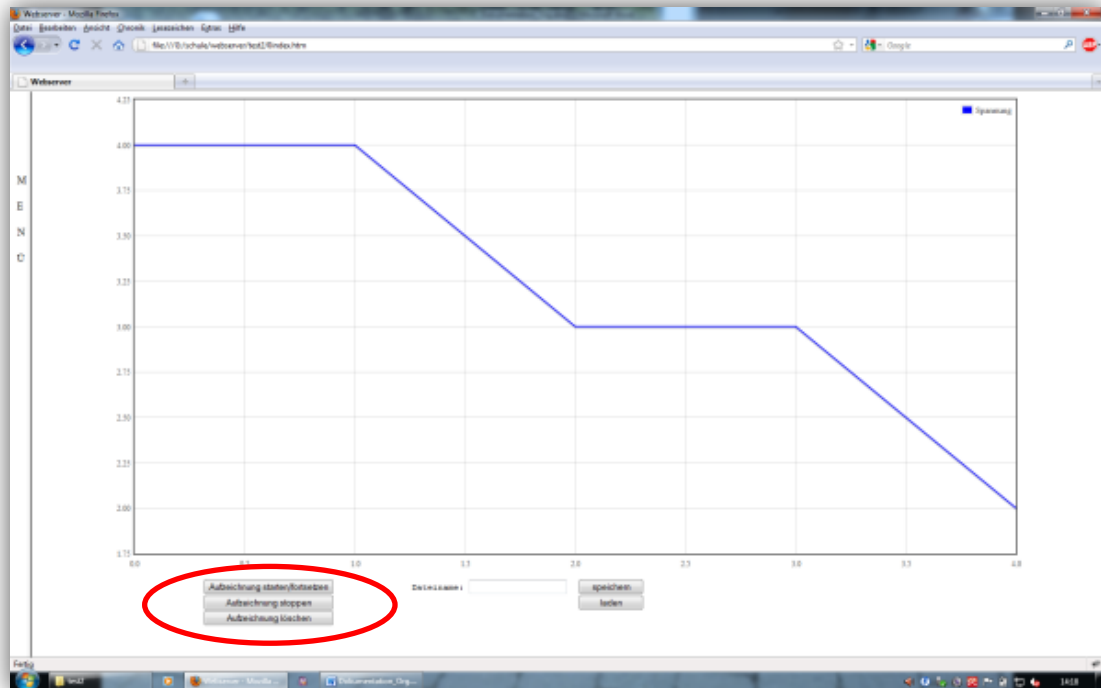


Abbildung 23: Aufzeichnung

- „Aufzeichnung starten/fortsetzen“ startet die Aufzeichnung der Daten
- „Aufzeichnung stoppen“ hält die Aufzeichnung der Daten an. Die Aufzeichnung kann mit „Aufzeichnung starten/fortsetzen“ fortgesetzt werden
- „Aufzeichnung löschen“ löscht die angezeigten Daten
- Mit dem Button „speichern“ können die Daten unter dem bei Dateiname angegebenen Namen gespeichert werden
- Mit dem Button „laden“ können zuvor gespeicherte Daten geladen werden, dazu muss bei Dateiname der Name eingegeben werden, unter dem die Daten gespeichert wurden



5.5 Programmierung

Hier zeigen wir wie das Programm auf den Logger gespielt wird und wie, wenn es nötig werden sollte, die Werkseinstellungen gesetzt werden.

5.5.1 Werkseinstellung herstellen

Sollte einmal die Situation auftreten dass nach einer Änderung der Netzwerkeinstellungen keine Verbindung mit dem Logger hergestellt werden kann, empfiehlt es sich die Werkseinstellungen zu aktivieren um eine definierte Konfiguration des Loggers herzustellen. Hierzu muss der Taster seitlich am Logger länger als 3s gedrückt werden. Im Bild ist der Taster zu sehen.

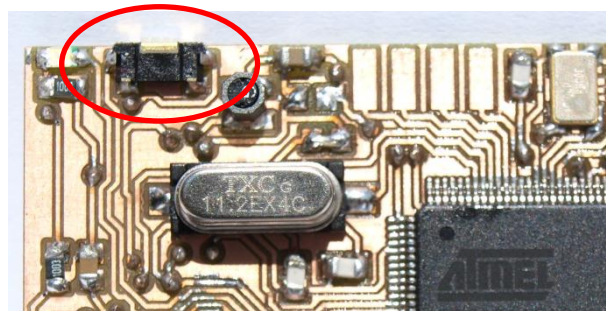


Abbildung 24: Reset Taster

5.5.2 Firmware aufspielen

Um die eigentliche Software auf den Datenlogger aufzuspielen haben wir uns etwas spezielles für die JTAG³⁴ Schnittstelle einfallen lassen um das Gewicht der Platine möglichst gering zu halten. Da das Aufspielen der Software nur einmal stattfindet, haben wir auf Steckkontakte verzichtet und nur die im Bild gezeigten Schleifkontakte auf die Platine gesetzt. Mit einem speziellen Stecker kann nun der Kontakt zwischen Logger und Debugger hergestellt werden.

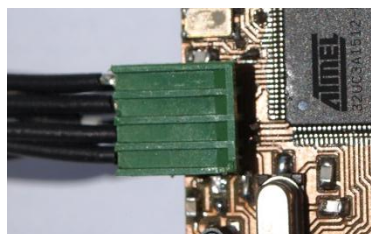


Abbildung 25: JTAG Schnittstelle

³⁴ **Joint Test Action Group**, bezeichnet den IEEE-Standard 1149.1, der eine Ansammlung von Verfahren zum Testen und Debuggen elektronischer Hardware direkt in der Schaltung beschreibt



6 Handhabung Hardware

6.1 *Stromsensor*

Es können bis zu 2 Stromsensoren des Typs A1301 oder ACS756 an den Logger angeschlossen werden, hierzu müssen zur Auswahl, je nach Sensor, wie im Bild zu sehen Brücken auf dem Logger gesetzt werden um die Spannungsversorgung sicherzustellen.

Um 5V (A1301) für den Stromsensor 1 auf den Anschluss zu legen muss die linke obere Brücke gesetzt werden. Um 5V für den Stromsensor 2 auf den Anschluss zu legen muss die rechte obere Brücke gesetzt werden.

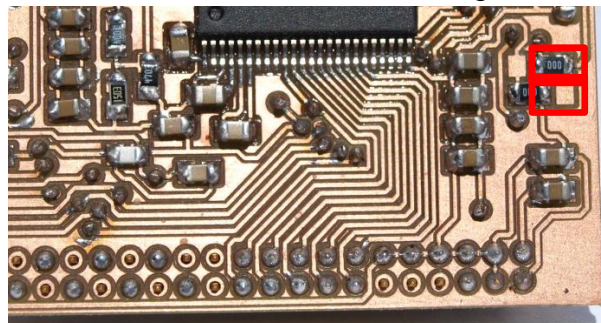


Abbildung 26: Brücken Stromsensoren 5V

Um 3,3V (ACS756) für die Stromsensoren auf den Anschluss zu legen muss jeweils die zugehörige Brücke unten gesetzt werden.

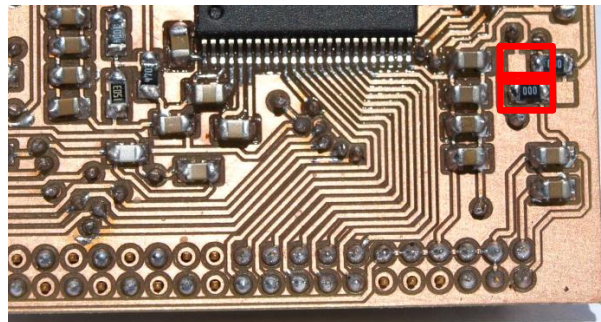


Abbildung 27: Brücken Stromsensoren 3,3V

Im Bild ist zu sehen wie die Stromsensoren mit dem Logger verbunden werden müssen.

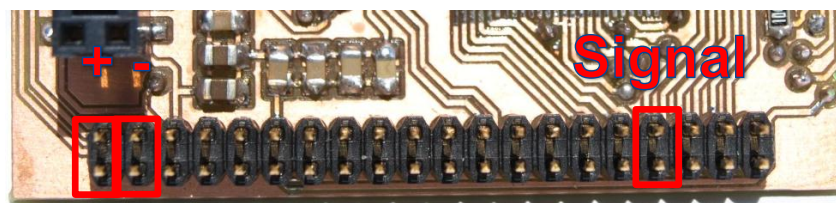


Abbildung 28: Anschluss Strom Sensoren



6.2 Akku Anschluss

Es kann entweder ein Akku mit bis zu 10 Zellen oder zwei Akkus mit je bis zu fünf Zellen an den Logger angeschlossen werden. Im Bild werden beide Szenarios dargestellt.

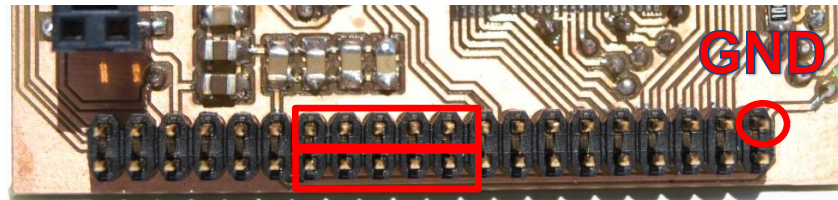


Abbildung 29: Anschluss 2 Akkus

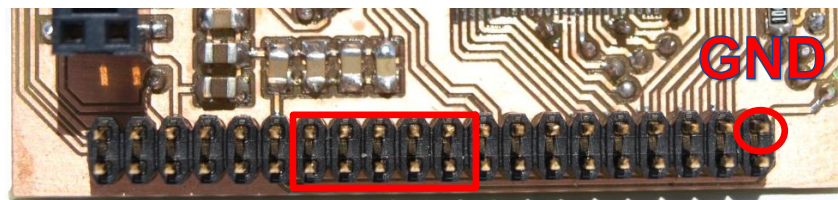


Abbildung 30: Anschluss 1 Akku

6.3 Temperatur-Sensoren

Es können bis zu 4 Temperatur-Sensoren des Typs MCP9701A an den Logger angeschlossen werden, diese werden wie im Bild zu sehen mit dem Logger verbunden.

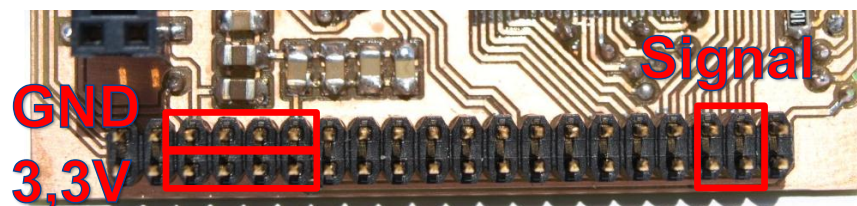


Abbildung 31: Anschluss Temperatur Sensoren



6.4 Schnittstellen

Der Logger ist mit zwei Seriellen Schnittstellen mit TTY Pegel ausgestattet. Die erste Schnittstelle ist mit Signalen für den GPS-Empfänger ausgestattet, die zweite ist komplett aufgelegt um variabel auf zukünftige Kommunikationspartner vorbereitet zu sein. In den nachfolgenden Bildern sind die Anschlüsse der beiden Schnittstellen dargestellt.

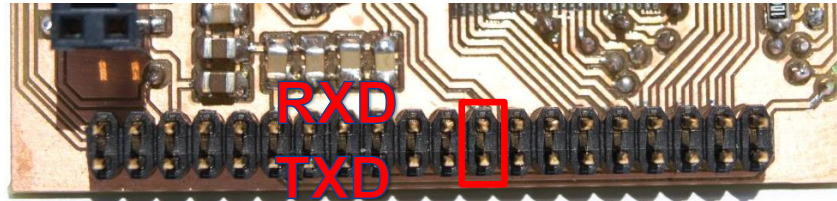


Abbildung 32: Anschluss erste TTY-Schnittstelle

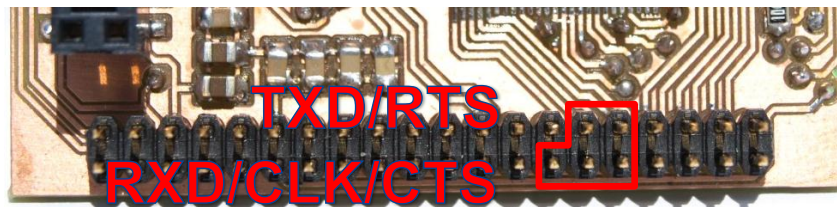


Abbildung 33: Anschluss zweite TTY-Schnittstelle



7 ERGEBNISSE

7.1 GPS

- Auf der Seite „GPS Data“ werden die empfangenen Daten des GPS Empfängers angezeigt.



Abbildung 34: GPS Data

- Außerdem kann die Position mit Google Maps angezeigt werden.

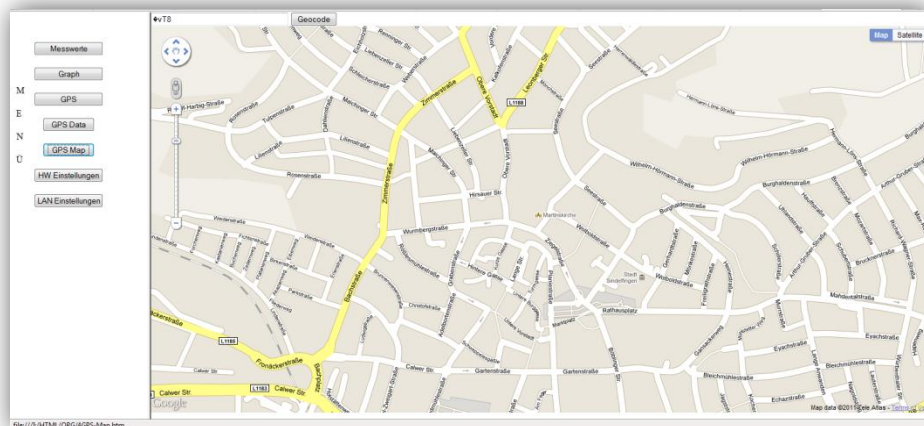


Abbildung 35: GPS Karte

7.2 Messwerte

- Auf dieser Seite werden die aktuellen Werte aller angeschlossenen Fühler angezeigt.

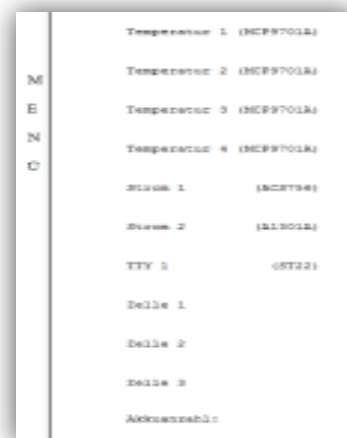


Abbildung 36: Messwerte



7.3 Daten sichern

Die Daten, die im Laufe eines Turns gespeichert wurden, können entweder gespeichert oder gelöscht werden. Wenn die Daten nicht gelöscht werden, wird die Aufzeichnung fortgesetzt.



Abbildung 37: Daten sichern

- Durch drücken des Buttons „Aufzeichnung starten/fortsetzen“ wird die Speicherung der Daten gestartet oder fortgesetzt wenn schon Daten vorhanden sind.
- Durch drücken des Buttons „Aufzeichnung stoppen“ wird die Speicherung angehalten.
- Durch drücken des Buttons „Aufzeichnung löschen“ werden die gespeicherten Daten gelöscht.
- Durch drücken des Buttons „speichern“ werden die Daten unter dem bei „Dateiname“ eingegebenen Namen gespeichert.
- Durch drücken des Buttons „löschen“ können gespeicherte Daten gelöscht werden. Dazu muss bei „Dateiname“ der Name eingegeben werden unter dem die Daten gespeichert wurden.



8 Probleme

8.1 *Hardware*

8.1.1 *Mikrocontroller*

Als das Material für die Platine bestellt wurde war auf einmal in ganz Europa der Mikrocontroller vergriffen, bei allen bekannten Distributoren ein übereinstimmender Liefertermin in KW28. Bei der Wahl dieses Kontrollers zu Beginn des Projekts waren die Bestände bei Farnell D kontrolliert worden, zu diesem Zeitpunkt waren noch ca. 50 Stück vorrätig. Beschafft wurde der Controller dann in den USA, über die Auktionsplattform EBay.

8.1.2 *WLAN Platine SPB105*

Bei Farnell D war eine Platine von HD-Wireless verfügbar, allerdings war auf der Homepage keine Angabe zu finden ob es sich um die SPB104, die über eine SDIO³⁵ Schnittstelle mit Daten versorgt wird, oder um die SPB105 handelt. Erst nach telefonischem Kontakt mit Farnell D konnte herausgefunden werden dass es sich hier um die SPB104 handelt und die SPB105 nicht beschaffbar sei. Nach weiteren Recherchen konnte die SPB105 bei Farnell GB aufgefunden werden, was nach weiterem telefonischem Kontakt mit Farnell D zur Lieferung und Aufnahme in den Produktkatalog der SPB105 führte.

Ein weiteres Problem mit der SPB105 ergab sich durch einen Fehler seitens des Herstellers. Die Platine konnte nicht direkt auf das EVK1105³⁶ gesteckt werden, es wurde ein Adapter benötigt der die Steckerbelegung ändert. Allerdings wurde auf dieses Problem nur in einer Release³⁷ hingewiesen die erst wegen Dysfunktion der Platine von uns studiert wurde. Die Lieferung dieses kostenlosen Adapters aus Schweden nahm leider 6 Wochen in Anspruch.

³⁵ **Secure Digital Input Output**, Sichere digitale Ein-Ausgabe

³⁶ Das **EVK1105** ist ein Entwicklungsboard für den AT32UC3A0512 von Atmel

³⁷ **Release Dokumentation**: Änderungsdocumentation in Bezug auf Software



8.1.3 LDO LP2981

Dieser LDO³⁸ Spannungsregler von Texas Instruments war eigentlich als 5V Spannungsversorgung vorgesehen. Leider ließen wir uns von den Angaben im Datenblatt täuschen, es stellte sich schnell heraus, dass dieser LDO mit der Eingangsspannung vom Akku nicht zurechtkam. Im Datenblatt steht zwar dass dieser Regler mit einer Versorgungsspannung auskommt die bis auf einen Minimalwert von 1,6V sinken kann, jedoch bezieht sich dieser Wert dann auf die kleinste Bauart dieses Reglers, der 1,3V am Ausgang liefert. Für unsere 5V Bauart waren die maximalen 3,7V in geladenem Zustand des Akkus schon zu wenig.

³⁸ **Low Drop Out:** Ein Spannungsregler mit einer geringen minimal erforderlichen Differenz zwischen Ein- und Ausgangsspannung



8.2 Software

8.2.1 WLAN-API

Die WLAN-API für SPB104 und SPB105 wurde von HD-Wireless unter folgenden Gesichtspunkten geschrieben.

- Einfach : Die API ist so einfach wie möglich zu halten, einfach zu bedienen.
- Minimale Größe : Die API ist einsetzbar auf sehr begrenzte Mittel-Plattformen.
- Portable: Die API ist auf jeder Plattform mit einem standardkonformen C-Compiler einsetzbar.
- OS unabhängig: Die API ist einsetzbar bei Systemen mit einem Echtzeit-Betriebssystem sowie Applikationen, die auf der "Bare Metal" von einer Hardware-Plattform (also ohne Betriebssystem).

Leider sind die Punkte Portabel und OS unabhängig nicht so erfolgreich durchgesetzt worden. Es bereitete größte Schwierigkeiten die API in ein bestehendes, vom Software Framework weitestgehend unabhängiges, Projekt einzubinden da die API auf viele vom Framework bereitgestellten Ressourcen zugreift und die Programmierung der API sehr spartanisch beschrieben ist.

8.2.2 Checksumme StarTraq

Um den GPS Empfänger zu konfigurieren müssen ihm mehrere Telegramme mit bestimmten IDs und Inhalten geschickt werden. Zu diesem Telegramm gehört auch immer eine Checksumme, die jeweils berechnet werden muss. Jedoch funktioniert diese Berechnung bei der Konfiguration der Geschwindigkeit der Seriellen Schnittstelle nicht, hier musste ausprobiert werden welche Werte der Empfänger hier akzeptiert. Bei anderen Punkten wie z.B. der Poll Rate oder der erwünschten GPS-Protokolle funktioniert die Berechnung ohne Probleme.



8.2.3 AVR32 Studio

Schon die Installation von AVR32 Studio aus dem Hause „Atmel“ brachte Probleme mit sich. So werden bei dem von uns zum Einsatz gekommenen Entwicklerboard EVK1105 zwei CDs mitgeliefert die die gesamte benötigte Software enthalten sollten. Leider wurden wir nach der Installation nach Vorgabe mit einem Fehler anstatt von einer Welcome-Site begrüßt. Dieser Fehler konnte nur, nach langer Fehlersuche und einigen neuen Installationen, durch ein Update eines Softwarepakets behoben werden. Das Paket, welches auf der CD mitgeliefert und automatisch installiert wurde, war zu alt.

Obwohl die C-Entwicklungsumgebung AVR32 Studio von „Atmel“ die neueste Software von Atmel auf dem Markt ist, musste mit mehreren Tricks gearbeitet werden um die Software unter Win7 im 64bit Modus zu fehlerfreier Funktion zu bringen. Nochmal mehr Know-how und viel Zeit brauchte es um den Debug-Modus in Betrieb zu nehmen.

Kurz vor Abschluss des Projektes musste auf AVR Studio5, eine neue Entwicklungsumgebung von Atmel, umgestiegen werden, da die neueste Version der API, die für uns sehr wichtig war weil in ihr die Verbindung mit Ad-Hoc-Netzwerken erst ermöglicht wurde, auf dieses neue Programm ausgerichtet ist.

8.2.4 WebServer

Beim Testen des Webservers haben wir festgestellt das er kein PHP unterstützt, was vor allem das Anzeigen und Speichern der Messdaten erschwerte. Das Problem wurde mit JavaScript gelöst, was allerdings deutlich aufwendiger war.

Als nächstes stellten wir fest, dass die Größe der Dateien begrenzt ist. Deshalb mussten wir einige Seiten des Webservers ändern und den Quellcode auf mehrere Dateien aufteilen.



9 Anhang A: Aufwendungen

9.1 Verwendete Software

Software	Version	Lizenz
Microsoft® Office 2010		Professional
AVR32 Studio	2.6.0	Proprietäre Software
AVR Studio5		Proprietäre Software
AVR32 Toolchain	2.4.2	GNU
AVR UC3 Software Framework	1.7.0	Proprietäre Software
FLIP	3.4.0	Proprietäre Software
NotePad++	5.6.8	Freeware
Eagle Light	5.11.0	Freeware
FireFox		Freeware
Namoroka		Freeware
Internet Explorer		Freeware
WireShark		Freeware

Tabelle 2: Verwendete Software

9.2 Material- / Kostenaufstellung

Material	Beschreibung	Kosten
AT32UC3A1512	Mikrocontroller	15 €
SPB105	WLAN Platine	40 €
Elektronikbauteile		25 €
Kabel und Stecker		5 €
Platinenmaterial		20 €
	Summe	105 €

Tabelle 3: Material- / Kostenaufstellung

Da dieses Projekt in erster Linie für den Eigenbedarf entwickelt wird, werden die gesamten Kosten vom Projektteam finanziert.



9.3 Zeitplanung

9.3.1 Stundenplan

Aufgaben	Zeitungfang in Stunden		
	Uwe Schmerbach	Benjamin Hirth	Gesamt
Projektdefinition	15	15	30
Erstellung Pflichtenheft	15	5	20
Einarbeitung CAD-Software (Eagle)	10		10
Einarbeitung Software Web Server		10	10
Einarbeitung Software AVR32 Studio	20		20
Schaltungsentwurf / Platinenlayout	30		30
Bauteilbeschaffung	20		20
Bestückung / Aufbau des Prototyps	20		20
Programmierung Web Server		40	40
Programmierung Mikrocontroller	50		50
Funktionstest	10	10	20
Optimierung Hardware	20		20
Optimierung Software	10	10	20
Fertigung und Test des Endgerätes	20	5	25
Dokumentation	20	20	40
Vorbereitung Präsentation	20	20	40
Summe der Arbeitsstunden	280	135	415



Tabelle 4: Stundenplan



9.3.2 Wochenplan

Aufgabe	Sep	Okt	Nov	Dez	Jan	Feb	Mrz	Apr	Mai	Jun
Projektdefinition	Uwe Schmerbach Benjamin Hirth									
Erstellung Pflichtenheft		Uwe Schmerbach Benjamin Hirth								
Einarbeitung Web-Software		Benjamin Hirth								
Einarbeitung CAD-Software		Uwe Schmerbach Benjamin Hirth								
Einarbeitung Software MC		Uwe Schmerbach								
Schaltungsentwurf / Platinenlayout			Uwe Schmerbach Benjamin Hirth							
Bauteilbeschaffung			Uwe Schmerbach Benjamin Hirth							
Bestückung / Aufbau Prototyp				Uwe Schmerbach Benjamin Hirth						
Programmierung Web-Server				Benjamin Hirth	Benjamin Hirth	Benjamin Hirth				
Programmierung MC				Uwe Schmerbach	Uwe Schmerbach	Uwe Schmerbach				
Funktionstest						Uwe Schmerbach Benjamin Hirth				
Optimierung Hardware							Uwe Schmerbach Benjamin Hirth			
Optimierung Software							Uwe Schmerbach	Benjamin Hirth		
Endgerät Fertigung und Test								Uwe Schmerbach Benjamin Hirth		
Dokumentation				Uwe Schmerbach	Benjamin Hirth	Uwe Schmerbach	Benjamin Hirth	Uwe Schmerbach	Benjamin Hirth	Uwe Schmerbach Benjamin Hirth
Vorbereitung Präsentation									Uwe Schmerbach Benjamin Hirth	Uwe Schmerbach Benjamin Hirth

Tabelle 5: Wochenplan

Uwe Schmerbach 
 Benjamin Hirth 



10 Anhang B: Quellenverzeichnis

www.wikipedia.de
www.atmel.com
www.utasker.com
www.hd-wireless.se
www.mikrocontroller.net
www.allegromicro.com
www.elektronik-kompodium.de
http://de.selfhtml.org
Javascript kurz & gut von David Flanagan & Lars Schulten
Javascript inklusive jQuery von Christian Wenz

11 Anhang D: Tabellenverzeichnis

Tabelle 1: Formatdefinition	2
Tabelle 2: Verwendete Software	39
Tabelle 3: Material- / Kostenaufstellung.....	39
Tabelle 4: Stundenplan	40
Tabelle 5: Wochenplan	41

12 Anhang E: Datenträgerinhalt

- C-Programm samt Entwicklungsumgebung und Treibern
- HTML Seiten für den WebServer
- Dokumentation in Dateiformat
- Pflichtenheft in Dateiformat
- Projektflyer in Dateiformat
- Einige Loggeraufzeichnungen von verschiedenen Turns
- Datenblätter der eingesetzten Bauteile



13 Anhang C: Abbildungsverzeichnis

Abbildung 1: Schaltplan Loggerplatine.....	10
Abbildung 2: Layout Top	11
Abbildung 3: Layout Bottom	11
Abbildung 4: Web-Server.....	11
Abbildung 5: Hall-Sensor in Ferritring	14
Abbildung 6: WLAN-Symbol	21
Abbildung 7: WLAN Eigenschaften	21
Abbildung 8: Drahtlosnetzwerk verwalten	21
Abbildung 9: Drahtlosnetzwerk hinzufügen.....	22
Abbildung 10: Ad-hoc-Netzwerk erstellen	22
Abbildung 11: WLAN Einstellungen	22
Abbildung 12: Adaptereinstellungen ändern	23
Abbildung 13: Drahtlosnetzverbindung	23
Abbildung 14: Status Drahtlosnetzwerkverbindung	23
Abbildung 15: Netzwerkeigenschaften.....	24
Abbildung 16: IPv4 Eigenschaften	24
Abbildung 17: HW Einstellungen	25
Abbildung 18: Sensor auswählen	25
Abbildung 19: HW Einstellungen	26
Abbildung 20: Warngrenzen	26
Abbildung 21: LAN Einstellungen	27
Abbildung 22: GPS Einstellungen.....	27
Abbildung 23: Aufzeichnung	28
Abbildung 24: Reset Taster	29
Abbildung 25: JTAG Schnittstelle	29
Abbildung 26: Brücken Stromsensoren 5V	30
Abbildung 27: Brücken Stromsensoren 3,3V	30
Abbildung 28: Anschluss Strom Sensoren.....	30
Abbildung 30: Anschluss 1 Akku.....	31
Abbildung 32: Anschluss erste TTY-Schnittstelle	32
Abbildung 33: Anschluss zweite TTY-Schnittstelle	32
Abbildung 34: GPS Data.....	33
Abbildung 35: GPS Karte.....	33
Abbildung 36: Messwerte.....	33
Abbildung 37: Daten sichern.....	34



14 Danksagung

Unser Dank geht an alle Beteiligten die mit Problemen und Fragen konfrontiert wurden, hervorheben möchten wir jedoch Mark Butcher von „M.J.Butcher Consulting“ für die zahlreichen Problemlöseanstöße und die großartige Unterstützung, Ulrich Weber von „matronic GmbH & Co. Electronic-Vertriebs KG“ für die Samples, Eugen Hörner für die Anfertigung der Platine und Åke Jernberger von “H&D-Wireless” für die Unterstützung beim Auffinden der API-Probleme.