

CanSat 2015 Team Gamma Dokumentation

Alexander Brennecke Till Schlechtweg Marc Huisinga Robin Bley
Steffen Wißmann Alexander Feldmann Kevin Neumeyer

12. Mai 2015

Inhaltsverzeichnis

1	Einleitung	3
1.1	Das Team	3
1.2	Stärken des Teams	3
1.3	Verbesserungsbereiche des Teams	3
1.4	Das Missionsziel	3
1.5	Praktischer Nutzen für den Auftragsgeber	4
2	Beschreibung des CANSAT	5
2.1	Missionsüberblick	5
2.2	Mechanisches und Strukturdesign	5
2.3	Elektrische Konstruktion	5
2.4	Softwareentwurf	5
2.5	Mechanisches und Strukturdesign	5
2.6	Bergungssystem	5
2.6.1	Elektronische Grundlagen	5
2.6.2	Embedded System	6
2.6.3	Transistor-Transistor-Logik	6
2.6.4	Analog-to-Digital-Converter	6
2.6.5	Universal-Asynchronous-Receiver-Transmitter	6
2.7	Hülle und Platzmanagement - Fachliche Grundlage	7
2.7.1	Inter-Integrated-Circuit	7
2.7.2	Python	7
2.8	Hülle und Platzmanagement - Dokumentation	7
3	Beschreibung der Bodenstation	9
3.1	Einleitung	9
3.2	Verwendete Komponenten	9
3.3	Funktionen	10
3.3.1	Nutzerfreundlichkeit	10
3.3.2	Erweiterbarkeit	10
3.3.3	Features	10
3.4	Architektur	10
4	Softwaretests	10
4.1	Nutzeranleitung	10
4.1.1	Livestream starten(Daten von einem Satelliten empfangen)	10
4.1.2	Daten importieren	10
4.1.3	Graphen anpassen	10
4.1.4	Daten exportieren	10
4.1.5	Daten mittels WLAN an Clients weiterleiten	10
4.1.6	Graphische Oberfläche personalisieren	10
4.2	Kosten-/Nutzenanalyse	10
5	Projektplanung	11
5.1	Zeitplan der CanSat Vorbereitung	11
5.1.1	Einschätzung der Mittel	11
5.1.1.1	Budget	11
5.1.1.2	Externe Unterstützung	11
5.1.2	Testkonzept	12

6	Öffentlichkeitsarbeit	13
6.1	Website	13
6.2	Schülerzeitung	13
6.3	Präsentationen	13
6.4	Ausstellung am MINT-Projekttag unserer Schule	13
6.5	Logo	13
7	Anforderungen	14
8	Reflexion des Projektverlaufes	15
8.1	Reflexion der Hardwaregruppe	15
8.2	Reflexion der Softwaregruppe	15
8.3	Reflexion der Zusammenarbeit zwischen den Teams	15
9	Anhang	16
9.1	GANTT- Diagramme	16
9.2	Hardware - GANTT	17
9.3	Hardware - GANTT	18

1 Einleitung

1.1 Das Team

Das gesamte Team besteht aus sieben Schülern und zwei betreuenden Lehrern. Die sieben Schüler sind jedoch intern in mehrere kleinere Teams aufgeteilt. Innerhalb der Teams ist jedoch kein Teammitglied vollkommen an seine Aufgaben gebunden, da uns ein guter Austausch und eine hervorragende Zusammenarbeit zwischen den einzelnen Teammitgliedern und Teams wichtig ist. Die Arbeit der Gruppen und der einzelnen Personen werden im folgenden erläutert:

Das Hardware Team besteht aus drei Personen, welche sich um den Bau des Satelliten selber, dem Design und dem Bau der Dose sowie der Programmierung des Mikrocontrollers kümmern. Zu diesem Team zählen folgende Personen:

Alexander Brennecke ist verantwortlich für das Design der Dose. Dazu zählt die Konstruktion der eigentlichen Dose und die Anordnung der Sensoren im Inneren der Dose.

Till Schlechtweg ist verantwortlich für die Funktionalität des Mikrocontrollers und den ausgewählten Sensoren.

Steffen Wißmann ist verantwortlich für die Übertragung der Daten zur Bodenstation und dem Programmcode des Mikrocontrollers.

Das Software Team besteht aus vier Personen, welche sich um das Programmieren des Analysetools und der Android Applikation kümmern. Dieses Team besteht aus folgenden Personen:

Robin Bley

Alexander Feldmann

Marc Huisinga

Kevin Neumeyer

Zudem gibt es ein Team, bestehend aus Alexander Brennecke und Till Schlechtweg, zur Organisation, Kommunikation mit Sponsoren und Öffentlichkeitsarbeit.

1.2 Stärken des Teams

1.3 Verbesserungsbereiche des Teams

1.4 Das Missionsziel

Die Idee hinter dem gesamten Projekt bezieht sich auf die extremen Umweltbelastung und ihre Folgen für den menschlichen Körper. Ausschlaggebend für diese Idee ist ein Zeitungsartikel der Zeit, welcher über eine drohende Klage der EU-Kommission in Brüssel berichtet. (vgl. Die Zeit, 24.10.2014). Die EU-Kommission droht mit einer Klage gegen Deutschland, da die deutsche Bundesregierung bisher zu wenig Aufwand betreibt, um die Feinstaubkonzentration in der Luft zu reduzieren. Wir möchten diesen Aspekt aufgreifen und Messungen durchführen um die tatsächlichen Werte zu bestimmen. Der CanSat Wettbewerb eignet sich optimal dazu, da er uns die Möglichkeit bietet die Messungen nicht nur auf dem Boden sondern in verschiedenen Schichten der Atmosphäre durchzuführen. Feinstäube stehen in Verdacht, Krankheiten wie Asthma, Herz-Kreislauf Beschwerden und Krebs zu begünstigen.

Da der menschliche Körper nicht nur durch Feinstaub belastet wird haben wir uns entschlossen auch die Intensität der UV-Strahlung, welche die Hauptursache für Hautkrebskrankungen ist, zu messen. Zusätzlich soll auch der Ozonwert bestimmt werden, da Ozon bereits in geringen Konzentrationen gesundheitsschädlich ist und zu Reizungen der Atemwege führen kann.

Für sich genommen ist jede dieser drei Größen schädlich für den Menschen. Im Zuge des Projektes wollen wir jedoch versuchen herauszufinden, ob es einen Zusammenhang zwischen ihnen gibt. Beispielsweise ist herauszufinden, ob ein höherer Ozon Gehalt gleichzeitig einen niedrigeren Feinstaubgehalt mit sich bringt.

Zusätzlich zum Bau des Messsystems im CanSat ist es unser Ziel eine einwandfreie Verarbeitung, Analyse und Präsentation der gemessenen Werte zu erzielen. Um dies zu garantieren programmieren wir ein eigenes Analysetool. Dieses Tool ermöglicht es uns die gemessenen Werte, während

des Fluges des Sateliten, auszuwerten. Die Werte sollen dabei anschaulich und in Abhängigkeit zueinander dargestellt werden.

Um die Daten auch mobil verfügbar zu haben wollen wir eine Android Applikation bereitstellen. Diese Applikation soll vorerst nur für unser Projekt optimiert sein, bei Erfolg jedoch auch die Werte andere Teams anzeigen können.

1.5 Praktischer Nutzen für den Auftragsgeber

2 Beschreibung des CANSAT

2.1 Missionsüberblick

Wir haben uns für den Satelliten überlegt, dass dieser so weit wie möglich individuell sein sollte. Daher greifen wir nicht auf das, vom Wettbewerb bereitgestellte T-Minus CanSat Kit zurück. Stattdessen haben wir uns im Detail überlegt, welche Sensoren unseren Erwartungen entsprechen und wie wir diese bestmöglich innerhalb der Dose platzieren können. Zusätzlich möchten wir nicht auf eine Cola-Dose als Hülle zurück greifen, sondern möchten auch hier unser eigenes Design erschaffen.

2.2 Mechanisches und Strukturdesign

2.3 Elektrische Konstruktion

2.4 Softwareentwurf

2.5 Mechanisches und Strukturdesign

2.6 Bergungssystem

2.6.1 Elektronische Grundlagen

Die Berechnung von Strom im geschlossenen System wird für unser Projekt benötigt, allerdings wegen der häufigen Anwendung von Digitaltechnik eher weniger. Außerdem müssen wir die Berechnung von Widerständen in Parallel- und Reihenschaltungen kennen und selber anwenden können.

$$U * R = I$$

Reihenschaltung

$$R_{ges} = R_1 + R_2 + \dots + R_n = \sum_{i=1}^n R_i$$

$$U_{ges} = U_1 + U_2 + \dots + U_n$$

$$I_{ges} = I_1 = I_2 = \dots = I_n$$

Parrallelschaltung

$$\frac{1}{R_{ges}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n} = \sum_{i=1}^n \frac{1}{R_i}$$

$$U_{ges} = U_1 = U_2 = \dots = U_n$$

$$I_{ges} = I_1 + I_2 + \dots + I_n = \sum_{i=1}^n I_i$$

Reihen und Parrallelschaltungen für Stromquellen haben wir nicht in Erwägung gezogen und gehört deshalb auch nicht zu den fachlichen Grundlagen. Weiterhin gehörte zu den elektrischen Grundlagen, dass Verständnis über einfache Dioden und über mehrere digitaltechnische Technologien welches in den nächsten Abschnitten erklärt wird.

2.6.2 Embedded System

Ein Embedded System ist in unserem Fall der BeagleBone mithilfe vom ARM Cortex-A6 mit 1Ghz, ist ein leicht modifizierter Linux Kernel mit Frontend installiert, das liebevoll Angstrom genannt wurde. Um ein paar andere Beispiele für ein eingebettetes System sind etwa ein Smart TV oder ein Router, beide haben eine Art eingebettetes System, das immer öfter auf dem Linux Kernel basiert und je nach Anwendung angepasst wurde. In unserem Fall unterstützt das BeagleBone verschiedene Technologien zum Empfangen von Daten verschiedener Bauteile, wie etwa UART, I2C, SPI, Analog, Digital, PWM, Timer und PRU. Viele dieser Technologien sind in unserem Projekt nicht in Verwendung, alle anderen Grundlagen sind unten beschrieben.

2.6.3 Transistor-Transistor-Logik

5V werden immer als logische 1 bezeichnet, damit ist gemeint, wenn der Sensor den höchsten Messwert erreicht, gibt er eine Spannung von 5V. Ist dies nicht der Fall, hat der Sensor eine andere Kennkurve, die zum Beispiel bei 3.3V aufhört. Allgemein wird aber Transistor-Transistor-Logik genutzt, welche 5V als logische 1 und geerdet als logische 0 ansieht, es gibt natürlich Toleranzen, diese sind aber bei verschiedenen integrierten Schaltkreisen und Mikrokontrollern unterschiedlich.

2.6.4 Analog-to-Digital-Converter

Andere Sensoren wie der UV-Sensor, die nur über einen internen Widerstand verfügen, der sich je nach Konzentration, an einer mathematischen Kurve orientierend, im Wert leicht verändert und dadurch die ankommende Spannung am jeweiligen Analog Pin ändert. Mithilfe eines Analog-to-Digital-Converter konvertieren wir das analoge Signal, zum Beispiel 5V, in das äquivalente digitale Signal mit der Auflösung von 12 Bits.

$$2^{12} = 4096$$

$$\frac{5V}{4096} = 0.001220703125V$$

Das bedeutet, jeder 0.001220703125V kann dargestellt werden, wobei der Arduino Mega 2560 nur 10 Bits zur Verfügung stellt.

$$2^{10} = 1024$$

$$\frac{5V}{1024} = 0.0048828125V$$

Der Arduino kann den Wert, der am analogen Pin ankommt, viel ungenauer darstellen, als das BeagleBone.

2.6.5 Universal-Asynchronous-Receiver-Transmitter

UART ist eine digitale serielle Schnittstelle zum Realisieren von einfachen Kommunikationen zwischen zwei Endpunkten, die Funktionsweise ist denkbar einfach. Wir nutzen in unserem Satelliten meist eine Baudrate von 9600bps, Baud ist die Schrittgeschwindigkeit oder Symbolrate, also 9600 bits per second. Für UART gibt es wie beim RJ45 Stecker TX und RX, die beim Aufbau einer Kommunikation gekreuzt werden. Transceiver und Receiver. Nun wird zwischen vielen verschiedenen Arten von UART unterschieden, in unserem Fall die TTL-UART Variante, welche die beim

Analog-to-Digital-Converter genannten 5V als logische 1 bezeichnen.

2.7 Hülle und Platzmanagement - Fachliche Grundlage

Um die 3D gedruckte Wand zu erzeugen wurde die 3D Modellierungssoftware **Sketchup** von Google verwendet. Sketchup bietet die Möglichkeit vergleichsweise einfach 3D Modelle zu zeichnen. Um dies zu tun muss klar sein, welche Objekte gezeichnet werden sollen. Diese Objekte müssen vermessen und innerhalb von Sketchup gezeichnet werden. Dies erfordert die Kenntnisse über gewisse mathematische Methoden zur Berechnung von Kreisen, Flächen und Körpern. Die meisten 3D-Drucker benötigen Dateien des Typs .stl, welche in Sketchup mit einem Plugin erzeugt werden können. Zum fertigen von GFK Komponenten wird ein Körper benötigt, auf welchen das GFK laminiert werden kann.

2.7.1 Inter-Integrated-Circuit

I²-C ist ein serieller Datenbus der über zwei Kabel mit einer 10-Bit-Adressierung, 1024 IC's steuern kann mit einer maximalen Geschwindigkeit 5 Mbit/s. Der Sinn des Bussystems ist es mithilfe von einer Adresse einen Datensatz oder Befehl nur an den gewünschten Empfänger zu senden, obwohl nur eine Datenleitung genutzt wird, eine Art Master/Slave System. Der Master sagt wer wann zu sprechen hat und welche Befehle von wem zu empfangen sind.

2.7.2 Python

Als Programmiersprache zur Programmierung des Beaglebone Black's, haben wir uns für die Programmiersprache Python entschieden. Es wäre zwar ebenfalls möglich gewesen den Mikrocontroller mit den Sprachen JavaScript, Java, C, C++, C# und vielen weiteren Sprachen zu programmieren, da es sich bei dem Beaglebone um ein embedded-System handelt, welches praktisch alle Programmiersprachen unterstützt, sofern entsprechende Bibliotheks existieren. Allerdings haben wir uns aufgrund der Tatsache, dass Python im Gegensatz zu Java nicht objektorientiert geschrieben werden muss, und wir auf der Hardwareseite möglichst auf objektorientierte Programmierung verzichten wollen, da sie nicht nötig ist, für Python entschieden. Ein weiteres wichtiges Argument war die gute Python-Bibliothek, welche von einer großen Community permanent gewartet und aktualisiert wird.

2.8 Hülle und Platzmanagement - Dokumentation

Wir haben uns dazu entschieden, die äußere Hülle aus GFK (Glasfaser verstärkter Kunststoff) zu fertigen. Dieses hat die Eigenschaften, dass er bei einem sehr geringen Gewicht, und bei einer geringen Wandstärke trotzdem eine gewisse Stabilität aufweist. Aus dem GFK haben wir eine Röhre mit einem Innendurchmesser von 31,5 mm und einem Außendurchmesser von 33,5 mm laminiert. Diese Röhre wurde auf eine Länge von 111 mm gekürzt und gefeilt. Um die Röhre oben und unten zu verschließen haben wir uns bei Thyssen Krupp System Engineering zwei Aluminium Deckel fräsen lassen. Diese haben uns ebenfalls durch ihr geringes Gewicht überzeugt.

Um die Elektronik innerhalb der Hülle zu platzieren und zu befestigen haben wir uns dazu entschieden eine Zwischenwand mit einem 3D-Drucker anzufertigen. Diese Wand teilt die Hülle mittig und bietet so auf beiden Seiten Platz um unser Microcontroller Board und unsere Sensorik Platine zu befestigen. Beide Bauteile werden mittels vier Gewindestangen an der Wand befestigt. Durch die Technik des 3D-Druckens ist es möglich der Wand ein sehr geringes Gewicht bei einer verhältnismäßig hohen Stabilität zu verleihen. Zusätzlich gibt es uns die Möglichkeit die Wand millimetergenau zu gestalten.

Am unteren Ende der Wand befindet sich eine Aushöhlung, sowie ein Fuß. Diese ist zum einen dafür da um den Sharp Feinstaub Sensor zu befestigen. Zum anderen gibt der Fuß der Wand und somit dem gesamten Satelliten eine gewisse Stabilität. Der Fuß besitzt auf der einen Seite der Wand

Bohrungen. Diese Bohrungen werden verwendet um die Aluminiumdeckel an der Wand zu befestigen. Da der Feinstaubsensor einen Luftzug benötigt befindet sich ein Durchlass innerhalb der Wand. Um das Microcontroller Board mit der Sensorik Platine zu verbinden existiert ein Fenster in der Mitte der Wand. Um die Sensorik Platine und das Microcontroller Board an der Wand zu befestigen existieren vier Bohrungen.

Abbildung 1: Der Satellit

3 Beschreibung der Bodenstation

3.1 Einleitung

In diesem Teil der Dokumentation werden wir die Bodenstation vorstellen, welche als Datenempfänger und als Datenverarbeitungsplattform fungiert.

Die Bodenstation wurde von Robin Bley, Marc Huisinga und Kevin Neumeyer entwickelt.

Die zentrale Aufgabe der Bodenstation ist es, die Daten, welche vom Satelliten gesammelt werden, zusätzlich sicher am Boden zu speichern, sollte der Satellit und damit auch die lokal gespeicherten Daten verloren gehen.

Zusätzlich zur Datensicherung erfüllt die Bodenstation die Aufgabe, die empfangenen Daten auf verschiedene Arten zu visualisieren und somit dem Nutzer direkt während der Datenübertragung die Möglichkeit zu verschaffen, die Daten zu beobachten und diese zu analysieren.

Die Bodenstation ermöglicht es außerdem, dass gesicherte Daten auch nach der Datenübertragung noch betrachtet und analysiert werden können.

Unser Ziel bei der Entwicklung der Bodenstation war es, eine modulare und anpassbare Plattform zu entwickeln, welche nicht nur mit unserem Satelliten, sondern mit vielen verschiedenen Satelliten genutzt werden kann, ohne dass ein großer Konfigurationsaufwand besteht.

Um dies zu ermöglichen, haben wir die Bodenstation in mehrere Dimensionen skalierbar entwickelt, was es im Endeffekt sehr einfach macht, neue Satelliten und verschiedene Übertragungsprotokolle zur Bodenstation hinzuzufügen.

3.2 Verwendete Komponenten

Zum Erreichen unserer Ziele haben wir verschiedene Komponenten verwendet, welche einerseits der Datenvisualisierung und -analyse dienen, andererseits aber auch der Entkopplung und skalierbaren Entwicklung dienen.

Für die Bodenstation haben wir folgende Komponenten verwendet:

Java 8 als Programmiersprache, da jedes unserer Gruppenmitglieder mit Java vertraut ist, wir aber trotzdem die mächtigen funktionalen Features von Java 8 nutzen wollten

Netbeans Platform das die Möglichkeit bietet, einfach eine integrierte, modulare und entkoppelte GUI-Applikation auf Basis von Java Swing zu entwickeln

JUnit zum Testen von bestimmten, komplizierten Teilen der Applikation

JSerialComm (zum Start des Projektes noch serial-comm) zum Lesen von Daten aus seriellen Ports

NASA World Wind ist eine Software, welche Satelliten- und Luftbilder auf einem virtuellen Erdball darstellt. Daten der Bodenstation werden mittels dieser Software in Relation zur Höhe in Echtzeit visualisiert.

JChart2D zum Anzeigen von übertragenen Daten in einem zweidimensionalen Graphen in Echtzeit

JSON (JavaScript Object Notation) ist ein Datenformat, welches zum austausch von Daten zwischen Anwendungen angewandt wird. JSON ermöglicht es Daten in verschiedenen Format in Textform zu speichern und sie wieder zurück in ihre ursprüngliche Form zu interpretieren. Dieses Datenformat wird in der Bodenstationsoftware genutzt Daten mit dem Satelliten auszutauschen, zu loggen, zu exportieren und zu importieren.

3.3 Funktionen

3.3.1 Nutzerfreundlichkeit

3.3.2 Erweiterbarkeit

3.3.3 Features

Da die Software der Bodenstation auf dem Framework Netbeans Plattform basiert, lassen sich einzelne graphische Module kombinieren, welche sich per drag and drop verschieben lassen. Die Größe und Position dieser Module und des gesamten Frames lassen sich beliebig verändern. Des weiteren bietet die Software die Möglichkeit Daten von verschiedenen Satelliten zu empfangen. Empfangene Daten lassen sich mittels der graphischen Oberfläche in Graphen anzeigen, welche sich verschieden kombinieren lassen. Außerdem lassen sich die empfangenen Daten zwischenspeichern und anschließend in verschiedene Dateiformate Exportieren oder live in einer Datei loggen. Diese exportierten oder geloggten Daten lassen sich anschließend wieder einlesen und Anzeigen. CSV, Txt, JSON, KML und png sind Dateiformate, welche exportierbar sind. Davon lassen sich exportierte CSV und JSON Dateien wieder einlesen und Visualisieren. Txt Dateien werden Formatiert und somit gut leserlich für den Leser exportiert, während exportierte KML-Dateien per Google Earth geöffnet und graphisch Visualisiert werden können. Ein weiteres Feature der Bodenstationsoftware ist die Datenvisualisierung per Nasa World Wind als Modul in der graphischen Oberfläche. Diese Visualisierung zeigt den Flug des Satelliten auf einem virtuellen Globus, mittels Satelliten- und Luftbilder, und zeigt auf jeder gemessenen GPS Koordinate die gemessenen Werte der Sensoren des Satelliten. Unter anderem bietet dieses Modul der Software die Möglichkeit an der Virtuellen Erdkugel heran zu zoomen und einzelne Elemente dreidimensional darzustellen. Darstellungen mittels dieses virtuellen Erdballs sind sowohl in Echtzeit mittels eines Streams vom Satelliten als auch als import aus einer Datei möglich.

3.4 Architektur

3.5 Softwaretests

3.6 Nutzeranleitung

3.6.1 Livestream starten(Daten von einem Satelliten empfangen)

3.6.2 Daten importieren

3.6.3 Graphen anpassen

3.6.4 Daten exportieren

3.6.5 Daten mittels WLAN an Clients weiterleiten

3.6.6 Graphische Oberfläche personalisieren

3.7 Kosten-/Nutzenanalyse

4 Projektplanung

4.1 Zeitplan der CanSat Vorbereitung

4.1.1 Einschätzung der Mittel

4.1.1.1 Budget Um das CanSat Projekt zu finanzieren konnten wir aktuell noch keine Sponsoren finden. Jedoch konnten wir uns mit unserem Schulverein verständigen, welcher uns finanziell unterstützen wird. Da wir nicht auf das T-Minus Kitt zurückgreifen sondern stattdessen ein anderes Mikrocontroller Board verwenden können wir ungefähr 150 € sparen. Der 200 € Watterot Gutschein, welcher vom Wettbewerb gestellt wird, ist in unseren Rechnung noch nicht inbegriffen. Dies liegt daran, dass noch nichts bei Watterot bestellt wurde, bzw. die Bestellung lange vor der Annahme am Wettbewerb getätigt wurde. Im Nachfolgenden sind alle Ausgaben und Einnahmen aufgelistet.

Tabelle 1: Ausgaben

Ausgabe	Datum	Empfänger	Grund
-12,16 €	08.01.2015	Watterott	BMP180 Breakout
-28,99 €	09.01.2015	eBay - rcskymodel	Ultimate GPS
-14,32 €	10.01.2015	Spark Fun Electronics	UV-Sensor
-51,99 €	10.01.2015	Amazon	Beagle Bone Black
-17,30 €	01.12.2014	eBay - hdt-preiswert	GFK-Set 1kg Polyesterharz + 20g Härter + 2m ² Glasfasermatte
-3,54 €	23.03.2015	toom baumarkt	6 x Schleifpapier
-3,79 €	23.03.2015	toom baumarkt	Filzrolle
-4,49 €	23.03.2015	toom baumarkt	Plüschwalzen
-2,19 €	23.03.2015	toom baumarkt	Mundschutz
-1,99 €	23.03.2015	toom baumarkt	Farbwanne
-4,99 €	23.03.2015	toom baumarkt	Einmalhandschuhe
<hr/>			
- 145,75 €			

Tabelle 2: Einnahmen

Einnahmen	Datum	Absender	Grund
17,30 €	01.12.2014	Alexander Brennecke	GFK-Kauf
107,46 €	10.01.2015	Alexander Brennecke	Sensorenkauf
20,99 €	23.03.2015	Alexander Brennecke	toom Einkauf
<hr/>			
145,75 €			

4.1.1.2 Externe Unterstützung Externe Unterstützung erhielten wir von vielen Lehrern unserer Schule, welche uns Fragen zur Elektrotechnik und Softwareprogrammierung beantworten konnten. Zusätzlich haben wir finanzielle Unterstützung durch den Schulverein unserer Schule erhalten (siehe 5.1.1.1). Unterstützung außerhalb unserer Schule erhielten wir durch folgende Personen/Organisationen:

- Das **Hackerspace Bremen e.V.**, welches uns ihren 3D-Drucker zur Verfügung gestellt hat. Zusätzlich konnten wir dort unsere Platine ätzen.
- **Prof. Martin Schneider** von dem Hochfrequenzlabor der Universität Bremen, welcher uns geholfen hat unsere Antenne an die Frequenz und die Wellenimpedanz anzupassen.

- Das Umweltlabor der **Atlas Elektronik GmbH** hat uns geholfen den CanSat, hinsichtlich seiner Stabilität, zu testen und die Sensoren korrekt zu kalibrieren.

4.1.2 Testkonzept

5 Öffentlichkeitsarbeit

5.1 Website

Unsere Website **Team Gamma** wurde bereits für den europäischen CanSat Wettbewerb 2015 verwendet. Diese haben wir weiter geführt und dort in unregelmäßigen Abständen aktuelle Informationen über das Projekt veröffentlicht. Da die Website durch den europäischen Wettbewerb bei anderen europäischen Teams bekannt ist wird die Website in Englisch geführt. Man findet dort zusätzlich einige Dokumente, Fotos und Videos. Die Informationen auf der Website sind meist relativ detailliert verfasst.

5.2 Schülerzeitung

In der Schülerzeitung unserer Schule sind bereits diverse Artikel über unser Projekt erschienen und sollen auch in Zukunft erscheinen. Diese Artikel handeln zumeist von dem Wettbewerb selber und gehen weniger auf die technischen Details ein.

5.3 Präsentationen

Da wir das CanSat Projekt bereits seit einiger betreiben kommt es immer wieder vor, dass wir es vor unserer Klasse präsentieren. Dies kommt zum Beispiel dann vor, wenn wir Teile des Projektes in Schulprojekte einfließen lassen. Zusätzlich haben wir, beispielsweise am Tag der offenen Tür unserer Schule, diversen Schulbesuchern das Projekt und den Wettbewerb näher gebracht.

5.4 Ausstellung am MINT-Projekttag unserer Schule

Im Schuljahr 2015/2016 findet an unserer Schule ein Tag der MINT Projekte statt. Dieser Tag wird von einer Schülergruppe unserer Parallelklasse organisiert und wir wollen an diesem Tag natürlich unser Projekt vorstellen.

5.5 Logo

Das Logo wurde ebenfalls aus Gründen der Wiedererkennbarkeit aus dem vorherigen Jahr übernommen. Das Aussehen des Logos wurde von drei Faktoren beeinflusst:

- Das Zeichen in der Mitte soll dem Gamma Logo ähneln, welches zu unserem Teamnamen passt
- Das Zeichen soll zusätzlich, wenn man es um 180° dreht, dem Lambda Logo ähneln. Da bei dem Entwurf unserer Antenne immer wieder auf Lambda gestoßen sind, sind daraus diverse interne Späße entstanden, die wir in das Logo einfließen lassen wollen.
- Das Logo des Computerspiel Halfife, welches von einigen Teammitgliedern gespielt wird

6 Anforderungen

7 Reflexion des Projektverlaufes

7.1 Reflexion der Hardwaregruppe

Als wir angefangen haben das gesamte Projekt zu planen haben wir uns als Ziel gesetzt Ende Mai fertig zu sein. Dieses Datum haben wir Aufgrund der Abgabe unseres P5 gewählt, für welches wir das CanSat Projekt ebenfalls einreichen wollen. Uns war bewusst, dass dies ein sehr hoch gestecktes Ziel ist. Im Nachhinein haben wir relativ schnell gemerkt, dass wir dieses Ziel nicht erreichen können. Diese Verzögerung wurde durch mehrere Faktoren hervorgerufen. Dazu zählt der enorm hohe Anspruch den wir uns selber gesetzt haben. Dieser hatte immer wieder zur Folge, dass viele Dinge mehrfach oder gründlicher gemacht werden mussten, als es zu Anfang geplant war. Zum anderen haben wir verhältnismäßig lange gebraucht um uns auf eine finale Idee festzulegen und diese zu präzisieren. Da wir uns jedoch kontinuierlich zum arbeiten getroffen haben konnten wir dennoch gute Fortschritte erzielen. Wir lagen zwar die meiste Zeit über hinter unserem Zeitplan, konnten jedoch die Reihenfolge der zu bearbeitenden Aufgabenpakete größtenteils einhalten.

7.2 Reflexion der Softwaregruppe

7.3 Reflexion der Zusammenarbeit zwischen den Teams

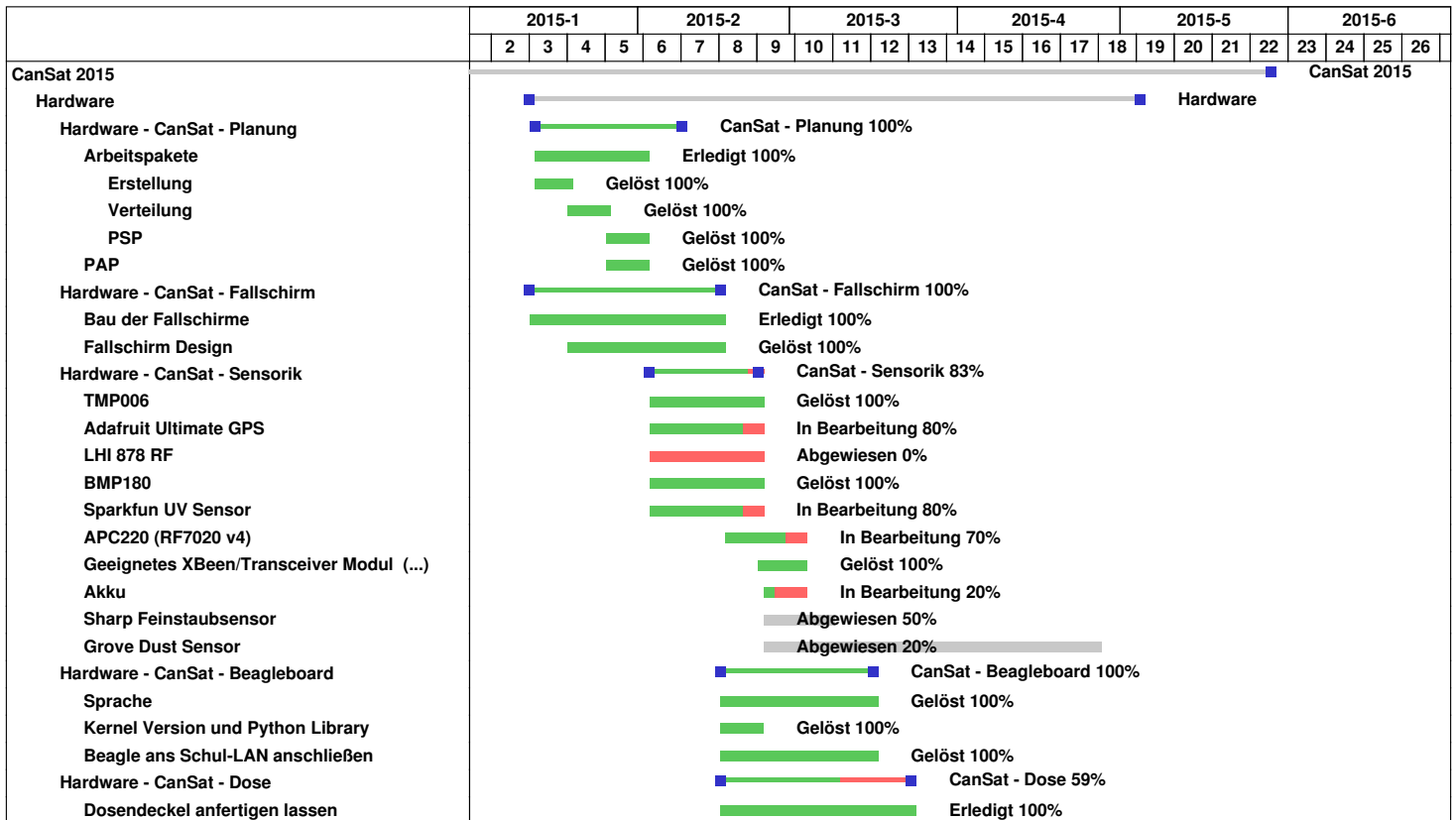
Da der inhaltliche Schwerpunkt der beiden Teams relativ wenig miteinander zu tun hat sollte es theoretisch relativ wenige Berührungspunkte geben. Dies war bei unserer Projektarbeit jedoch nicht so. Da die Arbeit der beiden Halbgruppen zur gleichen Zeit in der gleichen Räumlichkeit stattfand war es oft so, dass teamübergreifend diskutiert wurde. Dies hat den Vorteil, dass beide Teams nochmal einen anderen Blick auf eventuelle Problemstellungen bekommen und so einfache oder bessere Lösungen für Probleme finden können. Zusätzlich lief die Absprache über den Datenaustausch zwischen Bodenstation und CanSat sehr gut. Die beiden Teams haben also hervorragend kooperiert und gemeinsam versucht ein bestmögliches Gesamtprodukt zu erschaffen.

8 Anhang

8.1 GANTT- Diagramme

8.2 Hardware - GANTT

Hardware



8.3 Hardware - GANTT

