

Endbericht CanSat 2014

Teamname: spaceclub_berlin

Bundesland: Berlin

spaceclub_berlin

The background of the lower half of the page is a photograph of the Earth's horizon from space, showing the blue atmosphere and white clouds. A large, white, stylized arc graphic, resembling a partial circle or a satellite's path, is superimposed over the image, passing behind the 'spaceclub_berlin' text.

Inhaltsverzeichnis	Seite
1. Einleitung	2
1.1 Teamorganisation und Aufgabenverteilung	3
1.2 Missionsziele	4
2. Beschreibung des CanSat	5
2.1 Missionsablauf	5
2.2 Mechanisches- und Strukturdesign	6
2.2.1 Außenkörper	6
2.2.2 Innerer Aufbau	7
2.3 Elektrische Konstruktion	11
2.4 Technische Anforderungen	14
3. Ausgangsbedingungen für die Durchführung des Projektes	17
3.1 Physikalische Eigenschaften der Baukomponenten des CanSat	17
3.2 Kosten der verwendeten Baukomponenten	18
3.3 Externe Unterstützung	19
4. Primäre Mission	20
4.1 Missionsziele	20
4.2 Dafür benutzte Sensoren bzw. Geräte	20
4.3 Ergebnisse	21
4.3.1 Temperatursensor	21
4.3.2 Drucksensor	23
5. Sekundäre Mission	25
5.1 Missionsziele	25
5.2 Dafür benutzte Sensoren bzw. Geräte	25
5.3 Ergebnisse	26
5.3.1 Staubsensor	26
5.3.2 Kamera mit JPEG-Trigger	27
6. Bergungssystem	28
6.1 Fallschirm	28
6.2 Pieper	32
6.3 GPS	32
6.4 Bodenstation	34
6.4.1 Yagi-Antenne	34
7. Fertige Software	35
8. Problembehandlung	45
9. Öffentlichkeitsarbeit	47
10. Reflexion des Wettbewerbes	49

1. Einleitung

Unser Team setzt sich aus 3 Mädchen und 2 Jungen zusammen, die sich durch das orbitall im FEZ Berlin und den spaceclub_berlin kennen. Durch gemeinsame Projekte im spaceclub_berlin, aber auch in den Spacecamps 2012 und 2013 oder auch in Jugend forscht, sind wir ein eingespieltes Team geworden. Betreut und unterstützt werden wir von Mitarbeitern des orbitall und erfahrenen Programmierern.

Das Projekt hat uns durch einen Vortrag von Frau Remeikis besonders angezogen. Sie hat uns im spaceclub_berlin detailliert über die Anforderungen des Wettbewerbes informiert und uns dazu animiert, selber daran teilzunehmen. Motiviert uns zu präsentieren und uns neuen Herausforderungen zu stellen, haben wir uns in einem Team von 5 Mitgliedern zusammengefunden. Bei unseren spaceclub-Treffen haben wir uns erstmals an das Thema herangeschlichen und uns beraten, wie wir herangehen wollen. Als nächstes haben wir Mitarbeiter gefragt, unsere Betreuer und Lehrkräfte zu sein und uns über die Technologie des CanSat zu unterrichten. Einige unserer jetzigen Betreuer haben ihre Hilfe zudem selbst angeboten, zum Beispiel auf dem Gebiet des Programmierens. Zu den weiteren Treffen haben wir uns Ziele gesetzt, die wir erreichen wollen, haben uns weitergebildet und gelernt, letztendlich die Komponenten programmiert, konstruiert und zusammengeführt.

Natürlich haben sich während der Zeit viele Probleme aufgetan, die uns viel Zeit gekostet haben. Nichtsdestotrotz haben wir weiter gearbeitet und sind mit Hilfe unserer Betreuer weiter gekommen. Problemszenarien haben wir durchgespielt und sicher gestellt, dass die Startvorbereitung und der Start erfolgreich von staten gehen.

Während der Erarbeitung haben wir viel Material gesammelt und uns mit Experten ausgetauscht. Außerdem wurden wir mit dem Bereich des Programmierens bekannt gemacht und haben die Konstruktion geplant und ausgeführt. Alles mit dem Hintergedanken den Flug erfolgreich durchführen zu können und möglichst wenig Materialschäden davon zu tragen.

Der Wettbewerb startete für uns im Februar 2014 und endet im Oktober 2014 mit dem Start der Rakete und dem Fall des CanSats.

Die Besonderheit unseres CanSats ist die Komplexität der Systeme, die Sekundärmission und unsere Konstruktion. Wir haben nur wenige Komponenten unverändert gelassen und haben somit an viel Platz und Gewicht gespart.

1.1 Teamorganisation und Aufgabenverteilung

Das Team setzt sich aus uns fünf Teammitgliedern (T1-T5) mit entsprechender Aufgabenverteilung, sowie den Betreuern (B1-B4) und den Helfern (H1- H3) zusammen.

Tabelle 1: Teamorganisation und Aufgabenverteilung

NO	Name	Position
T1	Christina Nadolsky	Teamleitung; Bau des CanSat; Datengewinnung, Softwareentwicklung
T2	Yuna Reinhardt	Dokumentation; Kontrolle der Einhaltung der Wettbewerbsbedingungen; Leitung der Testverfahren, Softwareentwicklung
T3	Adrian Melinat	Fallschirm; Abpufferung des CanSats bei der Landung
T4	Nana Reinhardt	Bau des CanSat; Datengewinnung, Erstellung der Grafiken, Softwareentwicklung
T5	Christian Staudigl	Entwicklung der Software, Innere Anordnung der Komponenten
B1	Sylvia Reinhardt	Betreuerin
B2	Werner Bachmann	Betreuer, Unterstützung bei der Fallschirmkonstruktion
B3	Steffen Janke	Berater für die Entwicklung der Software und innere Anordnung der Komponenten
B4	Jan Klug	Unser Ausbilder für Softwareentwicklung
H1	Uwe Lathan	Technischer Berater
H2	Jörg Briesemeister	Beratung bei der Konstruktion
H3	Jürgen Kittel	Herstellung der Platinen

1.2 Missionsziele

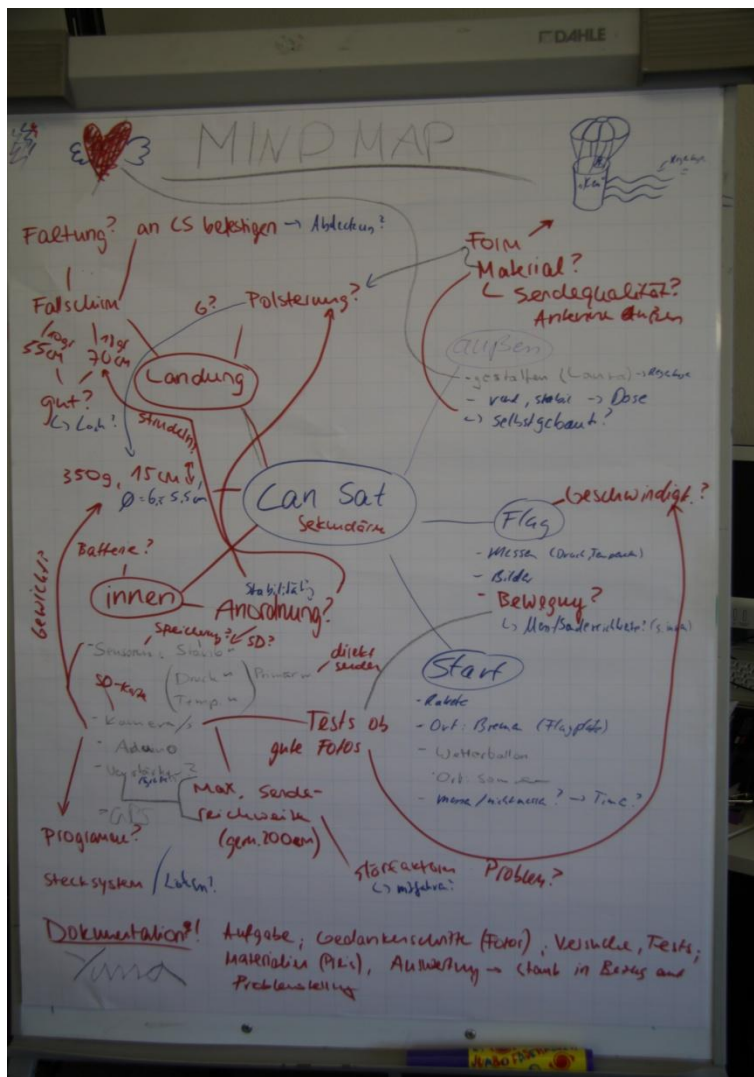


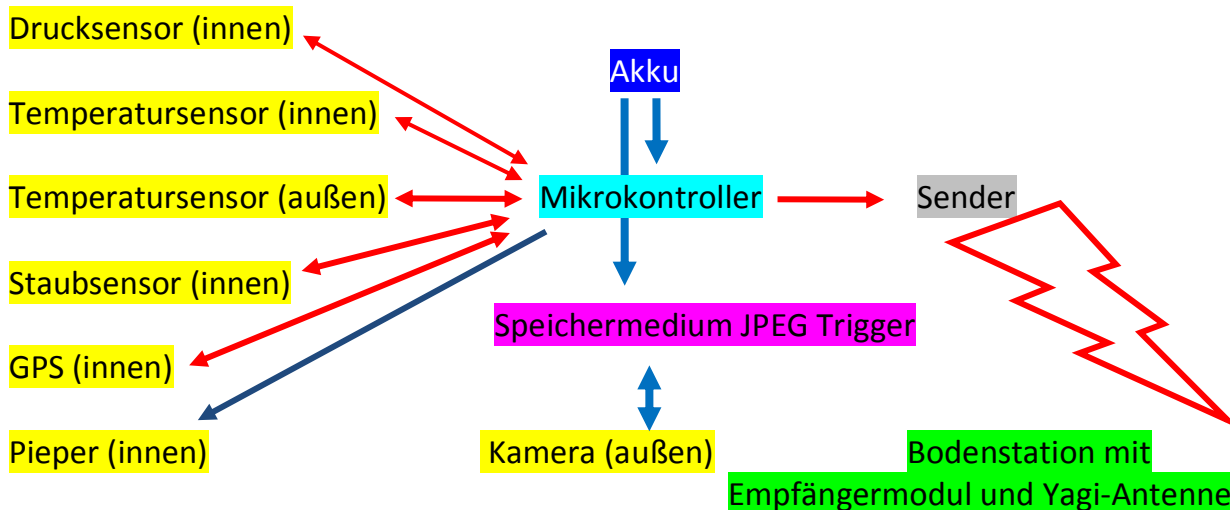
Bild 1: unser erster Missionsplan für den CanSat

In unserer Sekundären Mission werden wir die Staubkonzentration in der Luft messen. Wir bauen einen Sensor für Ermittlung der Staubkonzentration in der Atmosphäre ein, um diese in Abhängigkeit von der Flugrichtung und Flughöhe zu ermitteln. Gleichzeitig werden wir mit einer Kamera Fotos von dem überflogenen Ort machen, so dass wir die gemessene Staubkonzentration dem Ort (Fabrik, Wald, Autobahn...) zuordnen können. Um die aktuellen Messpunkte noch weiter zu belegen, werden auch die Lagedaten des Satelliten über GPS erfasst. Dies hilft uns auch, den Satelliten zu bergen.

Der CanSat-Wettbewerb stellt zur Aufgabe, einen Satelliten zu entwickeln, diesen zu bauen und zu programmieren. Er soll zwei Missionen erfüllen, zum einen die vorgegebene Primärmission und zum anderen die selbst wählbare Sekundärmission. Der Satellit wird mit Hilfe einer Rakete auf eine Höhe von bis zu 1000m gebracht und von der Rakete ausgeworfen. Dann fällt der CanSat zurück auf den Boden, wobei er dabei Messungen durchführt. Bei der Primären Mission ist es unsere Aufgabe, die Temperatur und den Druck zu messen. Das werden wir mit zwei unterschiedlichen Sensoren tun (siehe dazu 4.). Wenn wir die Daten der Temperatur in Höhe von einem Kilometer erhalten haben, werden wir daraus ein Temperaturprofil erstellen. Daraus können wir dann ablesen, wie sehr die Temperaturen in dieser Höhe sich mit denen auf dem Boden unterscheidet und wie sehr sie abfällt / schwankt.

2. Beschreibung des CanSat

Unser Cansat beinhaltet neben dem CanSat Kit, in dem zwei Temperatursensoren und ein Drucksensor vorhanden sind, verschiedene weitere Bauteile, wie ein Staubsensor, GPS, ein Pieper, und eine Kamera mit Speichermöglichkeit der Bilder auf einem JPEG Trigger (siehe Grafik 1). Die gesendeten Daten der beiden Temperatursensoren, des Drucksensors, des Staubsensors und des GPS werden am Empfängermodul in der Bodenstation mit einer Yagi-Antenne verstärkt. Die Antenne ermöglicht es, Signale aus weiteren Entfernungen besser empfangen zu können.



Grafik 1: schematische Beschreibung des Aufbaus und der Funktionsweise des CanSat

2.1 Missionsablauf

Die Mission wird folgendermaßen ablaufen:

Vor dem Start überprüfen wir die Sensoren, ob sie funktionstüchtig sind. Wir werden die entsprechende Sendefrequenz mit Hilfe des vorgegebenen Programms einstellen. Dann werden wir den Fallschirm überprüfen, ob er beim Transport oder ähnlichem beschädigt wurde und ob er nicht richtig in Position liegt. Sollte dies der Fall sein, wird er ausgetauscht oder richtig positioniert.

Die Notwendigkeit dieser Vorkehrungen beruht auf der Komplexität des CanSat. Da alle Komponenten miteinander in Verbindung stehen, kann der Ausfall einer Komponente zu einem Ausfall fast aller führen. Zum Beispiel wenn die Akkus ausfallen oder auf dem Mikrokontroller ein Kurzschluss entsteht. Wir haben aber sichergestellt, dass sich die Komponenten nicht während des Fluges verändern, sie sind also gepolstert und gesichert. Wichtig ist, dass wir zuerst den Empfänger starten und dann den CanSat, da wir am Anfang der Datenübertragung eine relevante Kopfzeile bekommen.

Der Start des CanSat wird mit einer vorgegebenen Rakete ermöglicht. Der Satellit wird in die Rakete angebracht, der Schalter an der Unterseite des CanSat betätigt, wodurch er zu senden beginnt. Nach ca. 1000 m wird der CanSat von der Rakete gelöst und der Fallschirm geht auf. Nun sinkt er mit einer Fallgeschwindigkeit von 15 Meter pro Sekunde zu Boden. Währenddessen werden die Daten an die Bodenstation gesendet und dort mit einer Yagi-

Antenne, die zur Verstärkung des Empfängermoduls angebracht wurde, empfangen. Die Kamera speichert über einen JPEG-Trigger die Bilder auf einer SD-Karte im CanSat. Die empfangenen GPS-Daten können dann zur Ermittlung der Landestelle genutzt werden, wenn sie sich also über einen längeren Zeitraum nicht mehr ändern und die Höhe des Drucksensors nicht variiert, können wir uns auf die Suche machen. In der Bodenstation müssen nun die gemessenen Daten gespeichert werden. Zur Lokalisation hilft uns ein zweites GPS-Gerät, das mit der Bodenstation verbunden ist. Erkennbar ist der Fallschirm und das glitzernde Wimpel durch grelle Farben und zusätzlich gibt der eingebaute Pieper einen Ton von sich. Die SD-Karte wird nun entnommen und die darauf gespeicherten Textdateien in Bilder umformatiert. Nun können wir zu den Daten der Sensoren auch Bilder der Umgebung zufügen. Schlussendlich müssen die gemessenen Ergebnisse ausgewertet werden. Diese Auswertung wird dann in unsere Präsentation am Tag nach dem Start mit einfließen.

2.2 Mechanisches- und Strukturdesign

Das mechanische- und Strukturdesign setzt sich aus dem Außenkörper und dem inneren Aufbau zusammen.

2.2.1 Außenkörper

Anfangs hatten wir uns für einen Außenkörper aus Carbonfaser entschieden. Die Röhre aus diesem Material ist sehr stabil, hatte jedoch den Nachteil, dass ihr Durchmesser nur 60mm war und wir somit auf die Maße (Höhe x Durchmesser) von 115mm x 60 mm kamen. Bei dem Bau des CanSat und der Unterbringung der verschiedenen Bauteile konnten wir nicht auf die fehlenden 6mm verzichten. Die Unterbringung des JPEG-Triggers und der Staubsensors war in dieser engen Röhre nicht möglich, so dass wir uns für ein anderes Material für die Außenhülle des CanSat entschieden haben.

Die Außenhülle ist ein zylindrischer Körper aus Plastik mit den Maßen (Höhe x Durchmesser): 115 mm x 66 mm und zwei Deckeln aus Plexiglas. Sie hat eine milchige Farbe und ist leicht durchlässig (siehe Bild 3).

Der untere Deckel besitzt insgesamt 14 Löcher: darunter 2 Löcher zur Befestigung der inneren Stangen, 3 für den Hauptschalter und seine Befestigung, einen für die Kamera und 5 zur Luftzufuhr für den Staubsensor, welche in einem Kreis angeordnet sind, und eins neben dem Hauptschalter, durch das der NTC (äußerer Temperatursensor) geführt wird.

Der obere Deckel hat 19 Löcher (siehe Bild2). Hierbei handelt es sich um 16, in einem Kreis angeordneten Löcher zur Befestigung des Fallschirms, welcher durch das eine Loch geführt wird und bei dem entsprechenden nebenliegenden Partner hinausgeführt wird. Desweiteren gibt es nur ein Loch für die Luftabgabe, die während des Fluges den CanSat durchströmt. Wir wollen damit einen bestimmten Luftdruck am oberen Deckel entstehen lassen, wodurch die Luft mit höherer Geschwindigkeit hinaus bläst und der Fallschirm sich möglichst schnell öffnet. Die letzten beiden Löcher sind für zur Befestigung des Deckels mit den inneren Stangen, also für Schrauben vorgesehen.



Bild 2: Oberseite des CanSat mit Durchblick zur Unterseite



Bild 3: Seitenansicht des CanSat

2.2.2 Innerer Aufbau

Der Aufbau des CanSat (siehe Bild 4) basiert auf einem horizontalen Schichtsystem der Komponenten, welche an zwei vertikalen Stangen befestigt sind. Am unteren Teil des CanSat befindet sich unsere Kombination aus Kamera und JPEG-Trigger (siehe Bild5 und Bild6), mit einem großen Loch im unteren Deckel für die Linse der Kamera. Mit einem geringen Abstand von dem JPEG-Trigger wird horizontal darüber der Staubsensor platziert. Die Luft strömt durch die unteren Löcher in den CanSat hinein, durch den Staubsensor und durch die oberen Löcher wieder hinaus. Ebenfalls mit etwas Abstand zur Gewährleistung der Luftzufuhr befindet sich die Dreierkombination aus Sender, Mikrokontroller und Platine mit den Sensoren für Druck, Temperatur (innen), GPS, Pieper und den Anschlüssen zur Stromversorgung der Kamera und des Staubsensors. Der Temperatursensor (außen) ist ebenfalls an der Platine durch lange Kabel befestigt, die nach unten geführt den Sensor neben der Kamera aus einem kleinen Loch am Deckel ragen lassen.

Um Platz zu sparen, werden die Akkus, die ebenfalls mit dem Mikrokontroller verbunden sind, in die Hohlräume zwischen der Dreierkombination gelegt und mit Schaumstoff abgepolstert. Die Elemente sind mit Abstandsschrauben voneinander getrennt. Weitere Polsterungen sind kaum möglich, da der Staubsensor eine ausreichende Luftzufuhr benötigt.

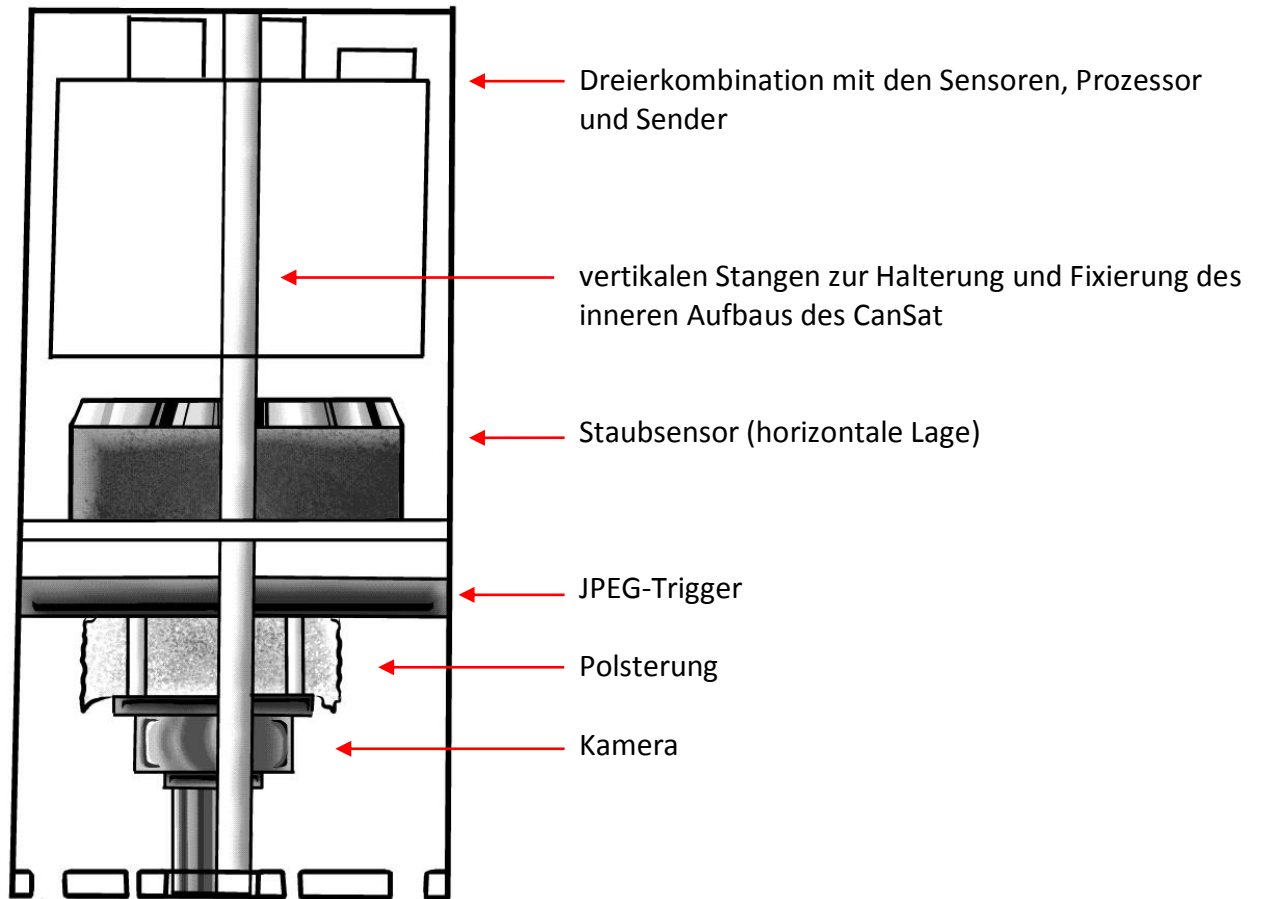


Bild 4: Überblick über den inneren Aufbau des CanSat

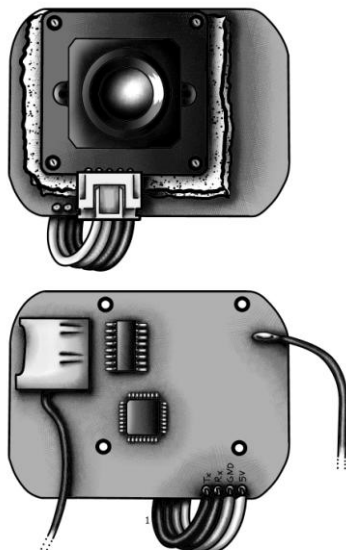


Bild 5 und Bild 6: Kamera auf abgerundeten JPEG-Trigger mit Polsterung dazwischen

Wir haben die Verbindungskabel zwischen Kamera und Trigger aus Platzgründen letztendlich über die Kamera-Platine verlegt, da sonst der Trigger nicht in die CanSat-Hülle gepasst hätte. Die ursprünglich angebrachten Kabel (siehe Bild 5) waren zu sperrig. Außerdem haben wir den JPEG-Trigger nach einer Überprüfung des Leiterplattenschemas an den Ecken abgerundet. Dadurch passt er horizontal in den CanSat-Außenkörper. Durch das Abrunden ist der Schalter, der sich am Trigger befand, nicht mehr vorhanden. Dieser war entbehrlich.

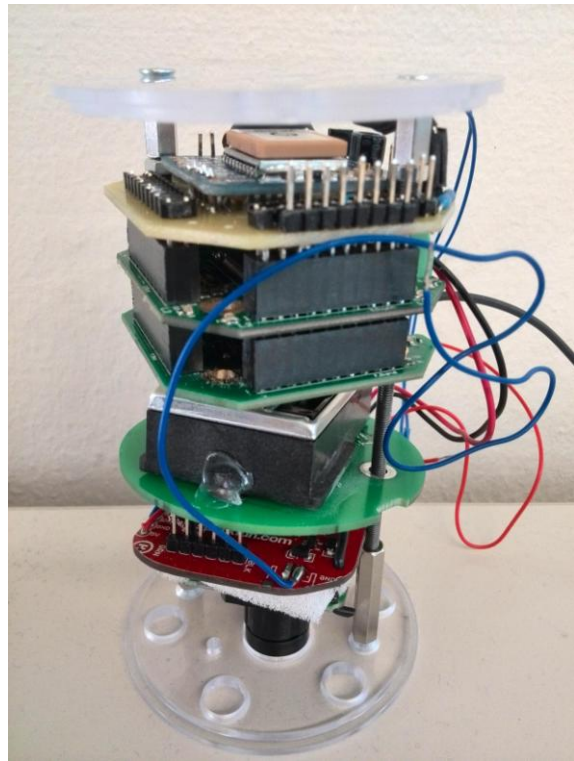
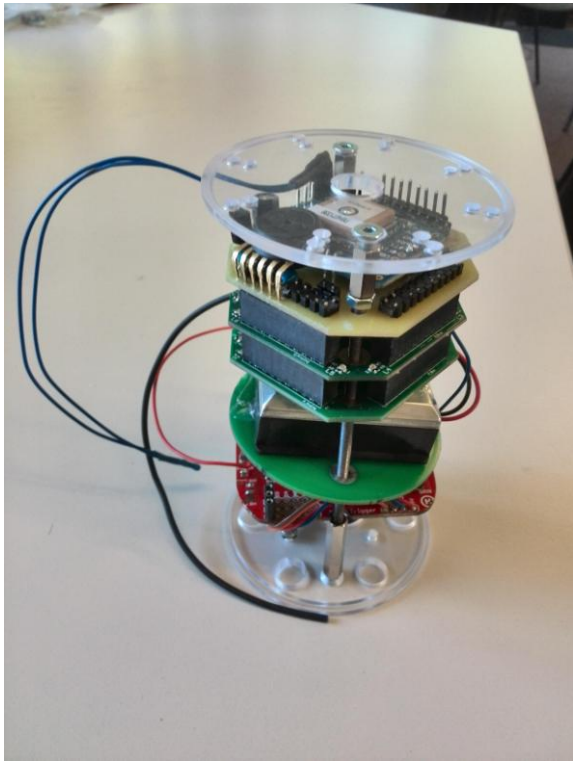


Bild 7 und Bild 8: CanSat seitlich gesehen

Das Einfügen des Staubsensors (Bild9) bereitete uns viel Kopfzerbrechen. Er ist in seinen Abmessungen sehr groß (Länge 46.0mm, Breite 30.0mm und Höhe 17.6mm) und nimmt sehr viel Platz im CanSat ein.

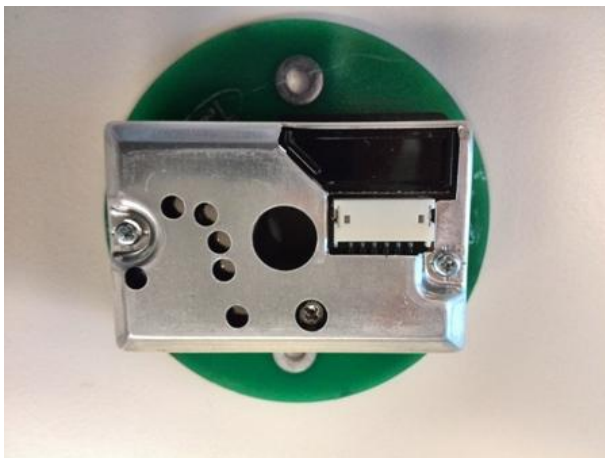


Bild 9: Staubsensor mit Anschlussbuchse für die Kabel und aufliegend auf Platte aus dem original CanSat Kit

Die Dreierkombination aus dem CanSat Kit ist nicht mehr im originalen Zustand. Wir haben die obere Platine mit den Temperatursensoren und dem Drucksensor mehrfach entsprechend den aktuellen Notwendigkeiten, nach unseren Vorgaben neu von einem Mitarbeiter des orbital (H3) anfertigen lassen.

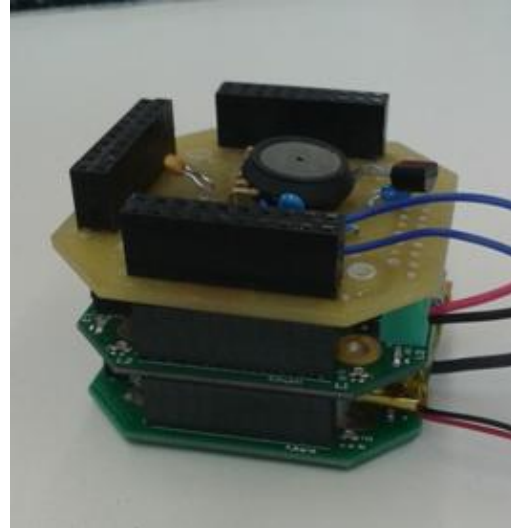
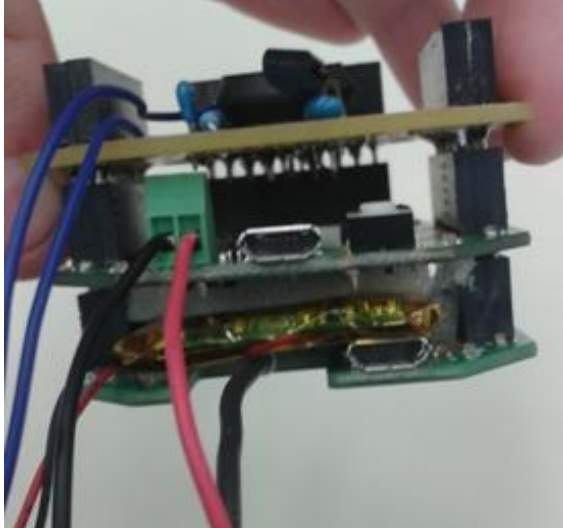


Bild 10 und Bild 11: erste neue Platine als Bestandteil der Dreierkombination (Platine/Mikrokontroller/Sender) links mit gepolsterten Akku zwischen Mikrokontroller und Sender

Auf der ersten neuen Platine befanden sich die beiden Temperatursensoren und der Drucksensor. Nach Bekanntgabe von Seiten der Wettbewerbsleitung, dass es notwendig ist, einen Pieper und GPS mit in den CanSat zu integrieren, haben wir uns entschlossen, nach unseren Vorgaben eine weitere Platine anfertigen zu lassen und die erste mit dieser auszutauschen. Diese Entscheidung wurde auch auf Grund des geringen Platzverbrauchs einer weiteren Platine gefällt. So wurden auf eine neue Platine alle weiteren Bauteile (zwei Temperatursensoren, Drucksensor, Pieper und GPS) integriert.

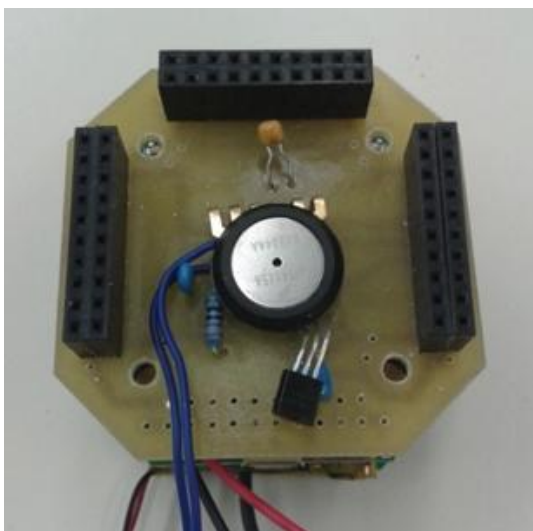


Bild 12: erste neue Platine mit beiden Temperatursensoren und Drucksensor

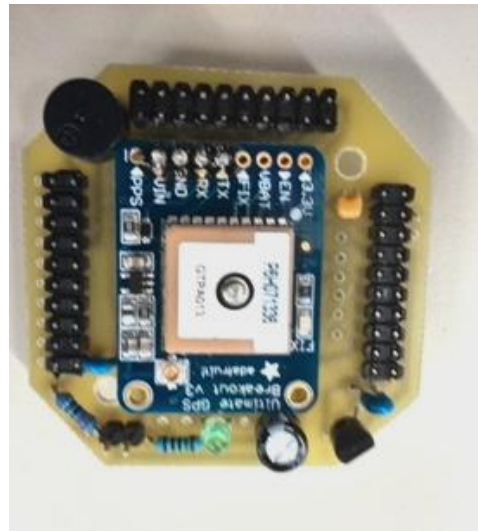


Bild 13: zweite neue Platine mit beiden Temperatursensoren, ohne Drucksensor, mit Pieper und GPS

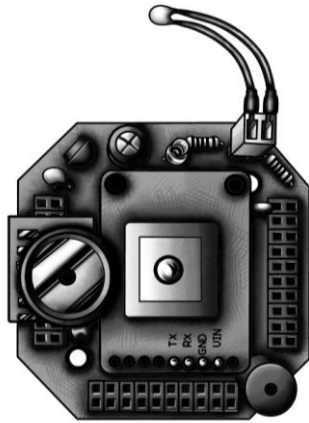


Bild14: zweite neue Platine mit mit beiden Temperatursensoren, Drucksensor, Pieper und GPS

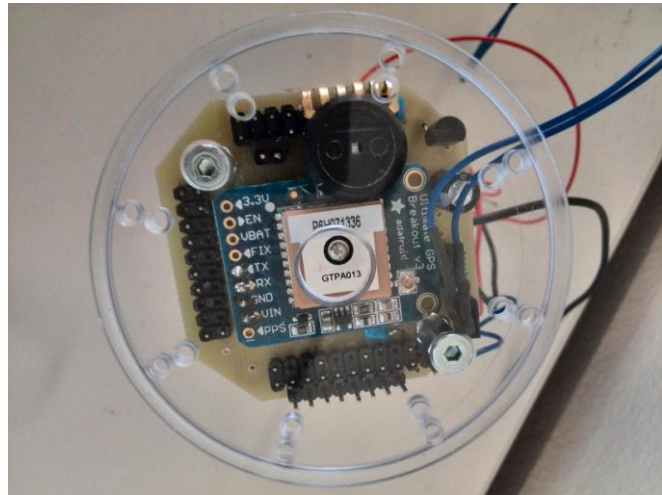


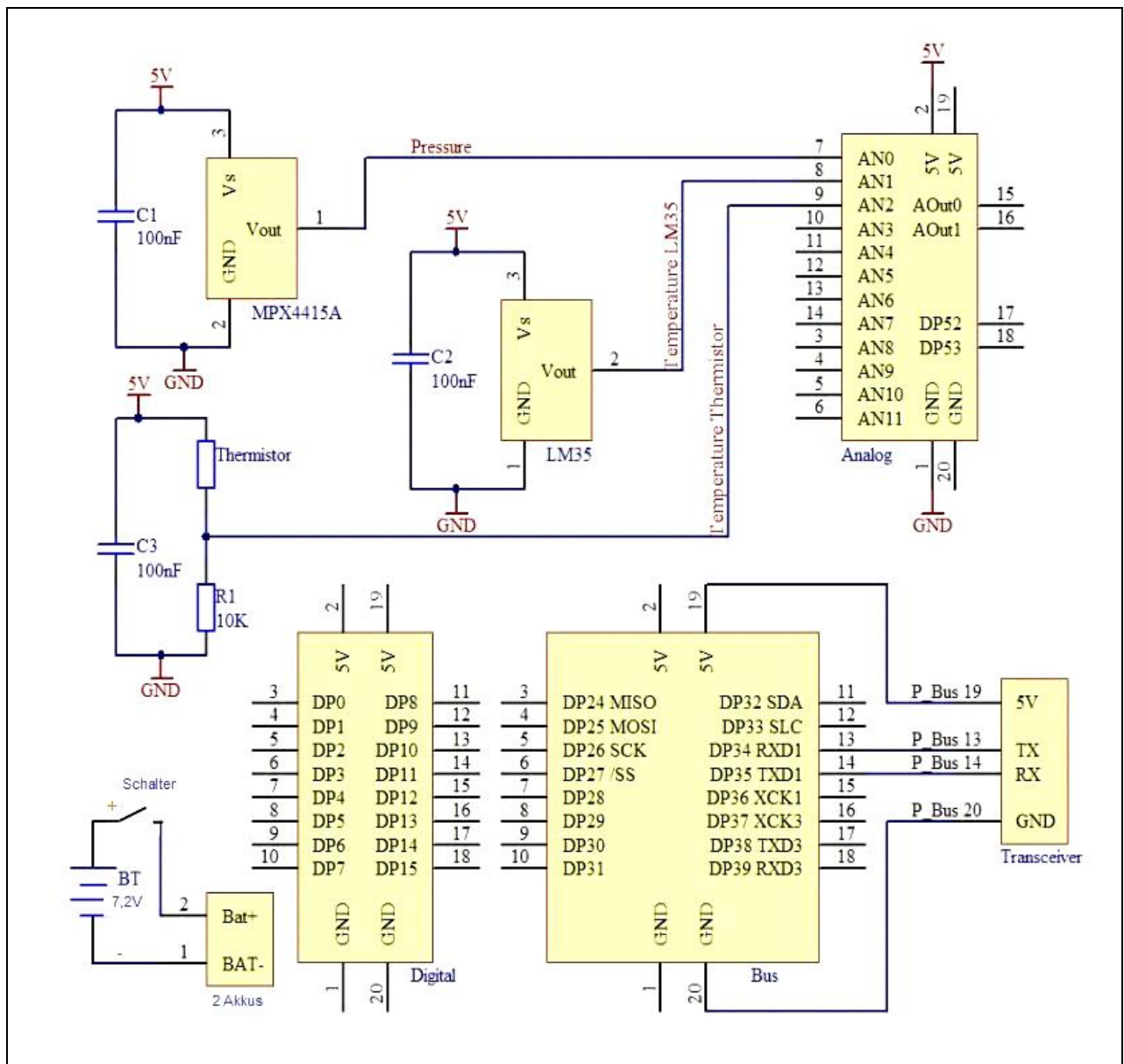
Bild 15: CanSat von oben gesehen mit der zweiten Platine

2.3 Elektrische Konstruktion

Die elektrische Konstruktion haben wir in die Bauteile für die Primärmission (pM1-pM3), die Sekundärmission (sM1-sM2a), die Bergungsmission (bM1 und bM2) und die Energieversorgung (EV) aufgeteilt (siehe Tabelle 2).

Tabelle 2: elektrische Konstruktion des CanSat (Sensoren, Bauteilbezeichnung und Energiebedarf)

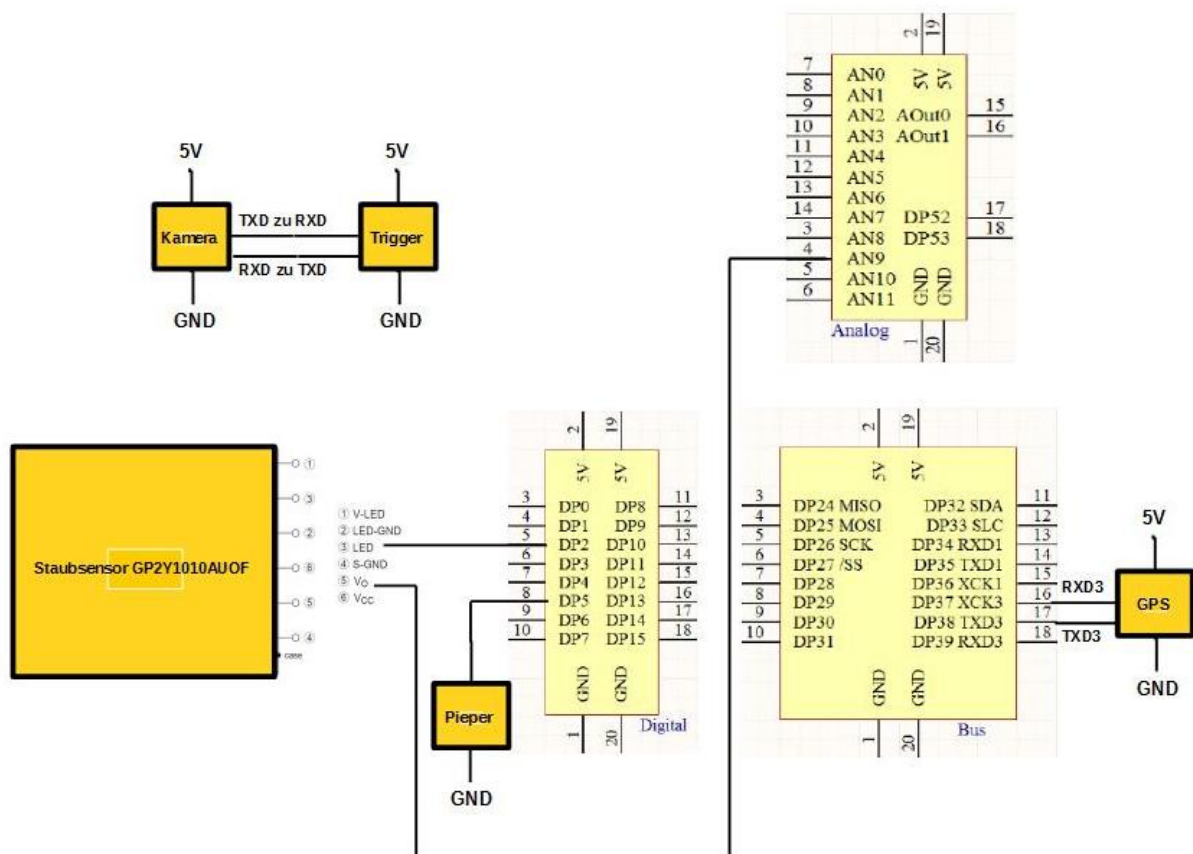
NO	Sensor	Bauteilbezeichnung	Energie
pM0	-	Prozessor und Sendeplatine	70 mA
pM1	Temperaturmessung im CanSat	LM35DZ	10 mA
pM2	Temperaturmessung außerhalb des CanSat	NTC LE100E3103JB	NTC: Variable, da Widerstand
pM3	Drucksensor	MPX4115	0.2 - 10 mA
sM1	Staubmessung, Luft kommt von außerhalb des CanSat	Sharp GP2Y1010AU0F	max. 20 mA
sM2	Kamera	LinkSprite JPEG Color Kamera (seriell)	10-100 mA
sM2a	Speichermöglichkeit der Bilder	JPEG Trigger	3,3-5V
bM1	Pieper	2 Opes 3V pie200	35 mA
bM2	GPS	GTP A013	25 mA
Gesamter Strombedarf	-	-	280 mA ohne NTC, Trigger, 1,43 h Laufzeit
EV	Akku	Lithium Polymer	400 mA*h



Grafik 2: schematische Beschreibung der elektrischen Konstruktion der **Primärmission** des CanSat (Anschlussplan der einzelnen Bauteile der Primärmission an der Platine des Mikrokontrollers) (Quelle: CanSat Kit User Manual, S.30) von uns verändert

Für die Primärmission ist die Belegung der Pins am Mikrokontroller wie folgt:

Drucksensor: A0
 Temperatursensor (innen): A1
 Temperatursensor (außen): A2



Grafik 3: schematische Beschreibung der elektrischen Konstruktion der **Sekundärmission** des CanSat (Anschlussplan der einzelnen Bauteile der Sekundärmission an der Platine des Mikrokontrollers)

Für die Sekundärmission und das Bergungssystem ist die Belegung der Pins am Mikrokontroller wie folgt:

Staubsensor: A9, DP2, 5V und GND

Trigger mit Kamera: nur 5V und GND (Kamera TXD zu RXD) am Trigger, Kamera (RXD zu TXD am Trigger)

GPS: DP37 (RXD3), DP38 (TXD3), 5V und GND

Pieper: DP5 und GND

2.4 Technische Anforderungen

In der Tabelle 3 haben wir die Anforderungen an die Konstruktion des CanSat zusammengetragen. Diese wurden in den Wettbewerbsbedingungen vorgegeben und sind einzuhalten.

Tabelle 3: anhand der Wettbewerbsbedingungen vorgegebene technische Anforderungen an die Konstruktion des CanSat

ID	Anforderung	Grund	Priorität	Verweis	<u>Überprüfungs- methoden und Lösung</u>
TA1	115 mm Höhe und 66 mm Durchmesser, Zylinder	Wettbewerbsanforderung	Sehr hoch	Grundlegender Aufbau des CanSat	Messen
TA2	Bergungssysteme maximal 45mm an oberer Seite überragen	Wettbewerbsanforderung, sicherer Abwurf	Sehr hoch	Grundlegender Aufbau des CanSat, Bergungssystem	Messen
TA3	Gewicht: 350 g	Wettbewerbsanforderung	Sehr hoch	Grundlegender Aufbau, Gewichtsorientierung	Wiegen
TA4	Keine Geschosse, Feuerwerkskörper, explosive Materialien	Wettbewerbsanforderung	Sehr hoch	Materialien	Überprüfung der Materialeigenschaften auf mögliche Gefährdung
TA5	Leicht erreichbare Batterie bzw. Akkus	Wettbewerbsanforderung	Sehr hoch falls Austausch notwendig	Grundlegender Aufbau des CanSat	Notfalltest Können durch Herausnehmen des Innenlebens ohne Probleme ausgetauscht werden
TA6	Leicht erreichbarer Hauptschalter	Wettbewerbsanforderung	Sehr hoch	Grundlegender Aufbau des CanSat	Notfalltest Von Unterseite erreichbar
TA7	Wiederverwendbares Bergungssysteme in Signalfarben, Fallgeschwindigkeit von 15 m/s	Wettbewerbsanforderung	Sehr hoch	Grundlegender Aufbau des CanSat, Bergungssysteme	Tests Fallschirm in roter bzw. gelber Farbe
TA8	Kosten von maximal 500 € (davon 200€ für das CanSat-Kit)	Wettbewerbsanforderung	Sehr hoch	Materialien	Ausführliche Dokumentation (siehe 3.2)

ID	Anforderung	Grund	Priorität	Verweis	Überprüfungs- methoden und Lösung
TA9	Luftdruck- und Temperatur- sensoren	Wettbewerbs- anforderung	Sehr hoch	Grundlegender Aufbau, Sensoren	Einbau und Überprüfung der Funktionsfähigkeit Mit Mikrokontroller sicher verbunden und Funktionsfähigkeit bestätigt
TA 10	Keine Fly-Home- Mission	Wettbewerbs- anforderung	hoch	Grundlegender Aufbau	Kein Auslegung des CanSat auf diese Missionsart
TA 11	Vor Einsatz der Startrakete müssen alle funktionsfähig sein	Wettbewerbs- anforderung	Sehr hoch		Überprüfung vor dem Start Funktionsfähigkeit vor Oktober bereits sichergestellt
TA 12	Fallschirm muss Kraft von 1000 N aushalten	Wettbewerbs- anforderung	Sehr hoch	Fallschirm	Tests
TA 13	Maximale Flugzeit 120 s	Wettbewerbs- anforderung	Sehr hoch	Voraussagungen des Flugablaufes	Tests, Berechnungen
TA 14	Glitzernder Wimpel	Wettbewerbs- anforderung	Sehr hoch	Bergungssystem	Tests Anbringung am CanSat
TA 15	Beschleunigung von 20 g aushalten	Wettbewerbs- anforderung	Sehr hoch	Voraussagung des Flugablaufs und Sicherstellung eines günstigen Ablaufs	Belastungstest mit 70 kg Masse auf die Außenhülle für einige Sekunden
TA 16	Sendequalität durch Außenmaterial nicht beeinflusst	Zur Sicherung guter Daten	Sehr hoch	Grundlegender Aufbau des CanSat	Test Sender innerhalb der Hülle und erfolgreiches Senden
TA 17	Sendekabel außen	Zum optimalen Senden der Daten	Sehr hoch	Grundlegender Aufbau des CanSat	Beachtung während des Baus
TA 18	Akustisches Signal	Leichtes wiederfinden des CanSat	Sehr hoch	Bergungssystem	Test Anbringung eines Piepers innerhalb des CanSats

<u>ID</u>	<u>Anforderung</u>	<u>Grund</u>	<u>Priorität</u>	<u>Verweis</u>	<u>Überprüfungs- methoden und Lösung</u>
TA 19	Speichermöglichkeit (SD-Karte)	Um Bilder zu Speichern	hoch	Grundlegender Aufbau des CanSat	Einbringung der SD-Karte in den CanSat für die Fotos
TA 20	Leichte Austauschbarkeit der Fallschirme	Um je nach Witterungs- bedingung zu reagieren oder bei Funktions- untauglich- keit des ersten Fallschirms	hoch	Grundlegender Aufbau des CanSat	Beobachtung während des Baus, Anbringung und Entfernen des Fallschirms ohne großen Zeitaufwand
TA 21	GPS	Leichtes Wieder- finden des CanSat	ratsam	Bergungs- system	Tests der Funktionsfähigkeit

3. Ausgangsbedingungen für die Durchführung des Projektes

Unser CanSat-Bau wird sowohl durch die physikalischen Eigenschaften der Baukomponenten (pB1-pB10), als auch durch die Kosten der verwendeten Baukomponenten beeinflusst.

3.1 Physikalische Eigenschaften der Baukomponenten des CanSat

Bei dem Bau des CanSat muss beachtet werden, dass er ein Gewicht von 350g nicht über- oder unterschreitet. Außerdem können nicht zu große Bauteile verwendet werden, weil sie sonst nicht in die Außenhülle des CanSat passen. Die Beschaffung der Baukomponenten für den CanSat war für uns dadurch eine große Herausforderung.

Tabelle 4: Masse und Größe der Baukomponenten im CanSat

ID	Bestandteil	Masse in g	Größe in mm
pB1	2 Akkus (Lithium Polymer)	19,0	40,0x30,0x3,0
pB2	selbst gefertigte Platine mit Bauelementen (2Temperatursensoren, 1 Drucksensor, Pieper, GPS)	23,0	50,0x50,0x10,0
pB3	Zweierkombination aus CanSat Kit (Mikrokontroller/Sender)	37,0	50,0x50,0x25,0
pB4	Staubsensor Sharp GP2Y1	15,0	46.0x 30.0x17.6
pB5	JPEG Trigger	11,3	11,0x40,0x60,0
pB6	Kamera (LinkSprite JPEG Color Camera TTL Interface)	11,2	32x32x30
pB7	Fallschirm	15,0	260 (Durchmesser)
pB8	Außenkörper des CanSat inklusive Befestigungen (Metallstangen)	96,0	115x66
pB9	Glitzernder Wimpel	8,0	150
pB10	Restliches Material	114,5	dem CanSat angepasst
	insgesamt	350,0	

3.2 Kosten der verwendeten Baukomponenten

Gemäß der Wettbewerbsbedingungen soll der Bau des CanSat Kosten von 500Eur nicht überschreiten.

Tabelle 5: Kosten der Baukomponenten des CanSat

ID	Bestandteil	Preis in €
pB1	Lithium Polymer (2x)	16,0
pB2	selbst gefertigte Platine mit Bauelementen (2Temperatursensoren, 1 Drucksensor)	CanSat-Kit
pB3	CanSat Kit mit Sensoren (2Temperatursensoren, 1 Drucksensor), sowie Zweierkombination bestehend aus Mikrokontroller und Sender, Metallstangen und Empfänger)	CanSat-Kit 200,00
pB4	Staubsensor Sharp GP2Y1	13,50
pB5	JPEG Trigger	27,06
pB6	Kamera (LinkSprite JPEG Color Kamera) mit 5 Pin Kabel	43,53
pB7	Fallschirm	3,0
pB8	Außenkörper des CanSat	10,0
pB9	Glitzernder Wimpel	0 (im Abfall gefunden)
pB 10	Restliches Material	0 (im Abfall gefunden)
pB 11	GPS-Sender	38,49
pB 12	Pieper	2
pB 13	verschiedene Kabel	10,0
pB 14	Yagi Antenne	0
	insgesamt	363,58

3.3 Externe Unterstützung

Die Räumlichkeiten und das Budget für unser Projekt erhalten wir durch die Kooperation mit dem orbital, dem Raumfahrtzentrum im FEZ Berlin. Das orbital führt in Kooperation mit dem DLR den spaceclub_berlin, in dem wir Mitglieder sind. Desweiteren stammen ein Großteil unserer Betreuer aus dem orbital oder kooperieren anderweitig mit Ihnen. Daher kommt auch das Mitwirken der Sternfreunde im FEZ e.v auf ideeller und motivierender Basis und die Verfügbarkeit einer Yagi-Antenne sowie der Beratung zur Montur und Einsatzfähigkeit durch die JugendTechnikSchule. Während des spacecamps 2014 im orbital, dem Raumfahrtzentrum im FEZ Berlin, knüpften wir eine Verbindung mit dem Institut für Luft- und Raumfahrt an der TU Berlin, an der ebenfalls CanSats (Nanosatelliten) gebaut werden. Mit Ihnen setzten wir uns in Verbindung bezüglich der Stromversorgung und Kameratechnik, also ebenfalls eine ideelle Kooperation. Während des Spacecamps 2014 verwanden wir „Senseboxen“, die wir anhand einer Veröffentlichung der TU Münster gekauft haben. Bis zu diesem Zeitpunkt war die „Sensebox“ nur stationär benutzbar. Wir nahmen Kontakt zur TU Münster auf und erhielten ein Programm, um diese mobil zu machen. Aus der Programmierung dieser erhielten wir viele Anregungen, z.B. deren Höhenformel war eine besserer Version im Vergleich zu unserer. Materielle Unterstützung erhielten wir auch von Thales Alenia Space Deutschland GmbH.

Partner und Sponsoren

FEZ Berlin

orbital, Raumfahrtzentrum im FEZ Berlin

Ansprechpartner: Dr. Sylvia Reinhardt

Thales Alenia Space Deutschland GmbH

Ansprechpartner: Florian Reutter

TU Berlin, Institut für Luft- und Raumfahrt

Ansprechpartner: Cem Avsar

DLR School Lab Berlin

Ansprechpartner: Dr. Christoph Pawek

Uni Münster

GI@School - Institut für Geoinformatik

Ansprechpartner: Mario Pesch

Sternfreunde im FEZ e.v.

Ansprechpartner: Steffen Janke, Jan Klug

Jugend Technik Schule

Ansprechpartner: Sieghard Scheffczyk

4. Primäre Mission

4.1 Missionsziele

Bei der Primären Mission war es unsere Aufgabe, die Temperatur und den Druck zu messen. Das führen wir mit zwei unterschiedlichen Sensoren durch: der NTC und der LM35 (siehe dazu 4.2). Wenn wir die Daten der Temperatur in Höhe von einem Kilometer erhalten haben, werden wir daraus ein Temperaturprofil erstellen. Daraus erkennen wir die Abhängigkeit der Temperatur von der Höhe. Die dazu benötigte Höhe erhalten wir durch eine Rechnung in der Programmierung. Rund alle 8 Meter verändert sich der Druck um 1 hPa, daher rechnen wir nur den Druck mit der gültigen Höhenformel zur aktuellen Höhe um und senden sie an die Bodenstation. Zusätzlich dazu werden wir die Fallgeschwindigkeit ermitteln - ebenfalls mit einer Formel. Diese kann man aus dem Programm für den Drucksensor entnehmen.

Wir erhalten also in unserem gewählten Zeitabstand die Daten zu: Temperatur außen und Temperatur innen in °C, Druck in hPa, daraus die Höhe in m und die Fallgeschwindigkeit in m/s. Der Grund zur Messung dieser Daten ist die letztendlich die Ermittlung der Flughöhe des CanSat und dessen Fallgeschwindigkeit. Wir messen nicht nur die Temperatur außen, sondern auch innen, um entweder die Isolation der Außenhülle festzustellen oder die Luftzufuhr im CanSat, die benötigt wird um mit dem Staubsensor verwertbare Ergebnisse zu erhalten. Das Zweite liegt bei uns im Vordergrund. Die Fallgeschwindigkeit von 15 m/s wurde von der Wettbewerbsleitung vorgegeben. Durch Ermittlung der Fallgeschwindigkeit können wir entnehmen, ob wir diese Bedingung mit unserem Fallschirm erfüllen konnten.

Zur Gewährleistung der Funktionstüchtigkeit der Sensoren haben wir mehrere Tests durchgeführt.

4.2 Dafür benutzte Sensoren bzw. Geräte

Wie im Abschnitt 2.2.2 schon erwähnt, verwenden wir für den Cansat nicht mehr die Originalplatine des CanSat Kits. Auf der ersten neuen Platine befanden sich die beiden Temperatursensoren und der Drucksensor. Auf der zweiten Platine wurde zusätzlich zu den Bauteilen der Primärmission noch Bauteile des Bergungssystems (einen Pieper und GPS) mit untergebracht sowie die Kondensatoren und Widerstände zur Gewährleistung einer funktionierenden Elektronik.

Tabelle 6: Baukomponenten für die Primärmission des CanSat

NO	Sensor	Bauteilbezeichnung	Energie
pM1	Temperaturmessung im CanSat	NTC LE100E3103JB	10 mA
pM2	Temperaturmessung außerhalb des CanSat	LM35DZ	NTC: Variable, da Widerstand
pM3	Drucksensor	MPX4115	0.2 to 4.8 V Output

Demzufolge befindet sich darauf der NTC, der LM35DZ, der MPX4115 und das GPS-Modul, sowie der Pieper.

Zur Messung der inneren Temperatur verwenden wir den LM35DZ und zur äußeren Messung wurde der NTC durch ein langes Kabel nach draußen verlegt. Der Drucksensor wurde umgeklappt und befindet sich demnach auch innen.

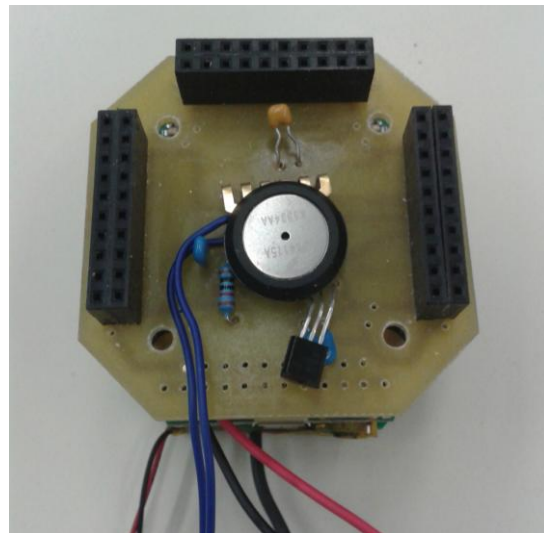
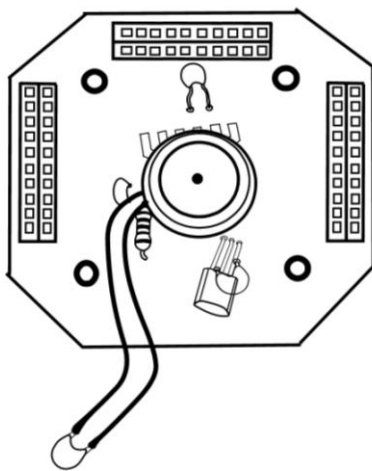


Bild 16 und Bild 17: erste neue Platine mit beiden Temperatursensoren und Drucksensor

Die folgenden Messungen bzw. Testreihen haben wir mit der ersten neuen Platine durchgeführt.

4.3 Ergebnisse

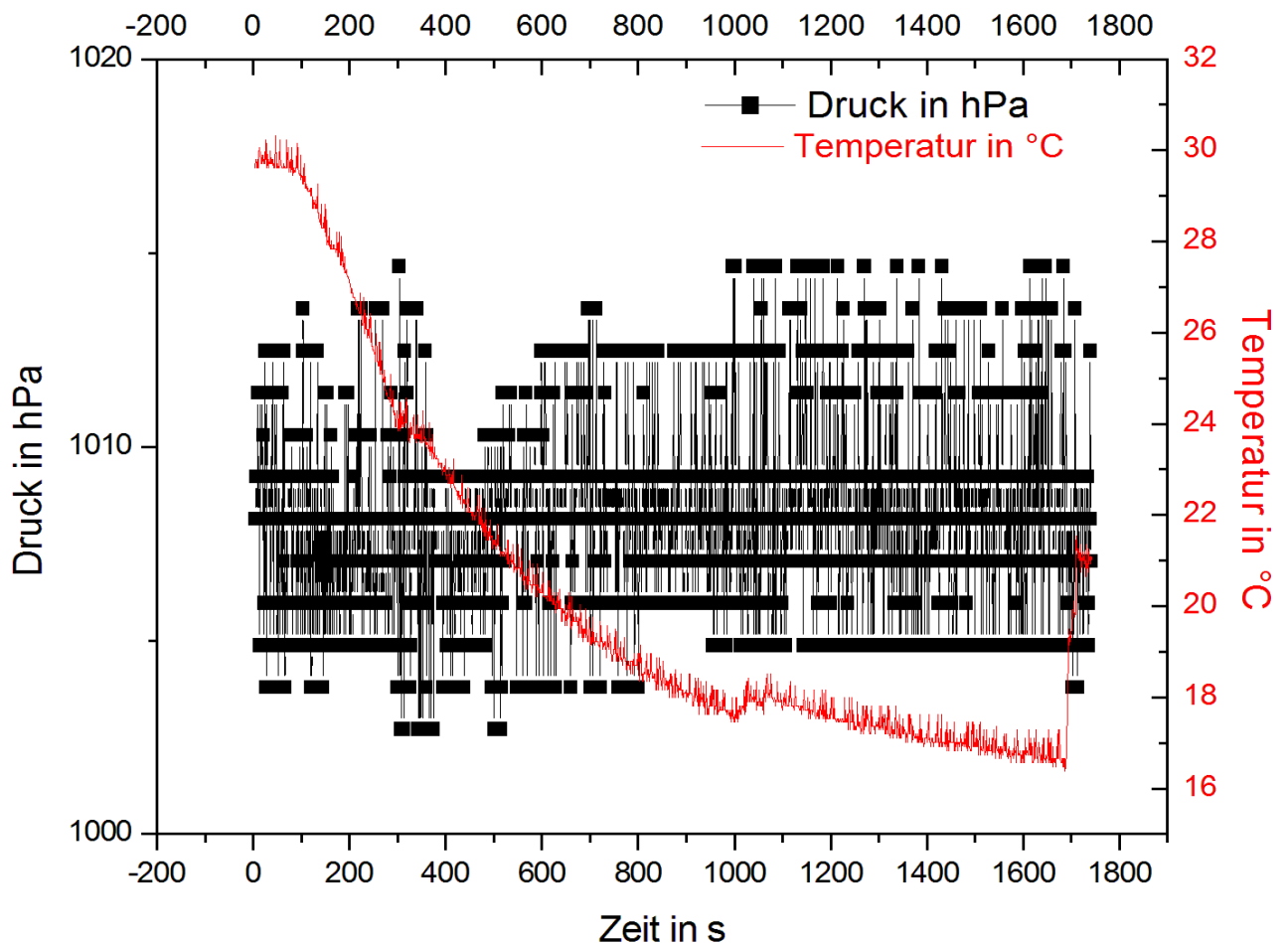
Die Komponenten für die Primärmission wurden getestet und ihre Programmierung fertig gestellt. Wir können damit sicher gehen, dass sie funktionieren und realistische Werte liefern.

Zum Einen testeten wir die Sensoren durch das Legen des CanSat in einen Kühlschrank. Zum Anderen legten wir ihn in eine Vakuumkammer, wodurch wir belegten, dass der Drucksensor funktioniert und die Komponenten selbst niedrigsten Druck aushalten.

Besonders wichtig war jedoch auch, zu wissen, ob die Programmierung auch funktioniert.

4.3.1 Temperatursensor

Wir legten den CanSat mit der ersten neuen Platine, die mit den Sensoren für die Primärmission bestückt war für ca. eine halbe Stunde (1700s) in den Kühlschrank.



Grafik 4: Test der Primärmission (Druck- und Temperaturmessung im Kühlschrank)

Dieses Diagramm zeigt die Temperatur- und Druckveränderung als Funktion Zeit. Die Temperatur wird in °C, der Druck in hPa und die Zeit in s angegeben. Das Augenmerk liegt hierbei auf dem Temperatursensor (NTC), da wir den CanSat in einen Kühlschrank gelegt haben. Wir erkennen hier ein deutliches Sinken der Temperatur zwischen 0 und 1700 s von ca. 30 auf 16°C. Bei circa 1000 s ereignet sich ein positiver Ausschlag der Temperatur. Hier wurde der Kühlschrank geöffnet und es strömte warme Luft hinein. Nach 1700 s wurde der CanSat wieder aus dem Kühlschrank genommen, nun steigt die Temperatur rapide an.

Daraus schlussfolgern wir zum Einen, dass dieser Temperatursensor realistische Werte anzeigt und funktioniert, jedoch zum Anderen mehr Zeit benötigen um zu erkalten. Also werden die Änderungen ins Negative (Temperatur fällt) langsamer und mit kleinerer Änderungsrate gemessen als die Änderungen ins Positive (Temperatur steigt). In Relation zur Höhe in der sich der CanSat befindet, haben wir demnach keine realistischen Werte, da der Sensor zu lange braucht um die Temperaturunterschiede wahrzunehmen.

Mit dem LM35Dz konnten wir keine Tests durchführen, da dieser über einen längeren Zeitraum kaputt war und noch einige Fehler mit sich bringt.

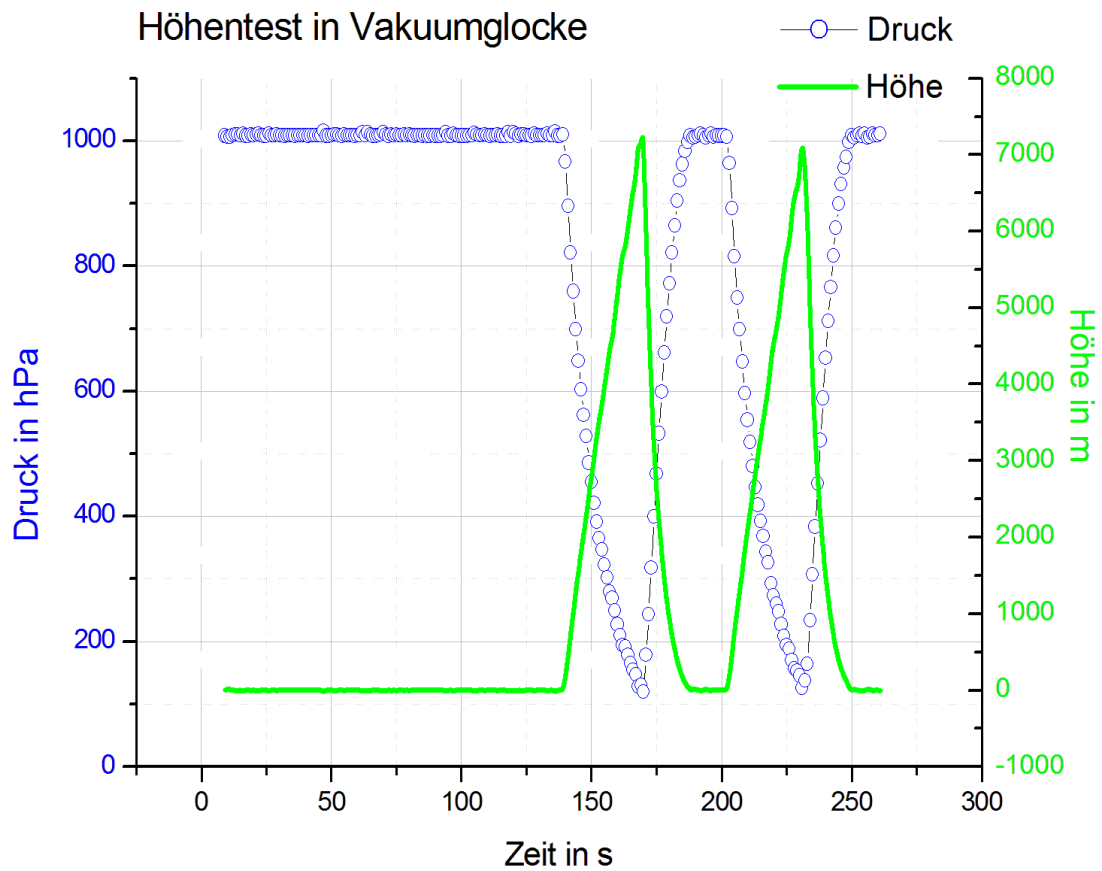
4.3.2 Drucksensor

Als weiteren Test der Primärmission legten wir den CanSat mit der ersten neuen Platine, die mit den Sensoren für die Primärmission bestückt war für ca. 5min (300s) in eine Vakuumblocke, wodurch wir belegten, dass der Drucksensor funktioniert und die Komponenten selbst niedrigsten Druck aushalten.



Bild 18: Test des Drucksensors auf der ersten neuen Platine in einer Vakuumblocke im orbitall

Bei diesem Test haben wir den Drucksensor in eine Vakuumblocke gelegt, den Druck darin verringert und währenddessen die Daten gesammelt. Der blaue Graph zeigt den Druck in hPa an. Man kann sehen, dass sich der Druck am Anfang im normalen Bereich befindet. Nach knapp zwei Minuten fällt der Druck schlagartig ab, in dieser Zeit haben wir die Luft aus der Glocke abgepumpt und dann mit einem Mal wieder rein gelassen, was man an dem plötzlichen Anstieg des Graphen erkennen kann. Das haben wir einmal wiederholt. Im Programm (siehe 4.2) haben wir die Rechnung eingebracht, so dass die Höhe ausrechnet und als Ergebnis mit angegeben wird. Die Höhe in m, in Abhängigkeit vom Druck, wird durch dem grünen Graphen dargestellt. Je niedriger der Druck, desto höher befindet sich der CanSat.



Grafik5: Test der Primärmission (Druckmessung in der Vakuumglocke)

Hierdurch haben wir festgestellt, dass der Drucksensor gut die Veränderung des Druckes wahrnehmen kann und realistische Werte liefert. Ebenfalls ist die Höhenformel richtig und lässt sich auch anwenden. In diesem Versuch haben wir auch die Fallgeschwindigkeit gemessen, die ebenfalls realistisch war. Der zweite Teil der Primärmission ist also auch funktionstüchtig.

5. Sekundäre Mission

Das Forschungsthema der Sekundären Mission konnten wir frei wählen.

5.1 Missionsziele

Die Feinstaubkonzentration in der Luft stellt heute neben der Abgas- und Stickstoffdioxidbelastung eine Bedrohung für die Bronchialsysteme dar. Besonders durch Emissionen aus Kraftfahrzeugen, Kraft- und Fernheizwerken, Öfen und Heizungen in Wohnhäusern oder auch durch Bodenerosionen gelangt Feinstaub in die Luft. Das Bundesumweltamt definierte zur Gewährleistung gesundheitsunschädlicher Luft besondere Richtlinien, die je nach Art des Staubes in Städten eingehalten werden müssen. Ist dies nicht der Fall, müssen bestimmte Maßnahmen getroffen werden. Da Großstädte wie Berlin oder Bremen durch Gewerbe und Industrie geprägte Städte sind und eine ausgebaute Infrastruktur besitzen, muss die Luftqualität regelmäßig überprüft werden. Ziel unserer Sekundärmission ist die Messung der Staubkonzentration in Abhängigkeit von der Flugrichtung und Flughöhe. Um die Daten einer bestimmten Region zuzuordnen, z.B. Wald, Industriegebiet oder auch Autobahn, werden Fotoaufnahmen gemacht. Um die aktuellen Messpunkte noch weiter zu belegen, werden auch die Lagedaten des Satelliten über GPS erfasst. Der Staubsensor wird uns Daten in $\mu\text{g}/\text{m}^3$ senden, die wir im Endeffekt in einem Höhe-Staubkonzentration-Diagramm auswerten und mit aktuellen Daten vergleichen können. Währenddessen nimmt die Kamera Bilder auf und speichert sie auf eine MicroSD-Karte im JPEG-Trigger. Wenn der CanSat auf dem Boden gelandet ist, können wir die MicroSD-Karte entnehmen und die Fotos erhalten.

5.2 Dafür benutzte Sensoren bzw. Geräte

Wie im Abschnitt 2.2.2 schon erwähnt, verwenden wir für den CanSat nicht mehr die Originalplatine des CanSat Kit. Wir entwickelten eine zweite Platine, auf der zusätzlich zu den Bauteilen der Primärmission noch Bauteile des Bergungssystems (einen Pieper und GPS) und Bauteile der Sekundärmission (sM1- sM3) mit angeschlossen wurden.

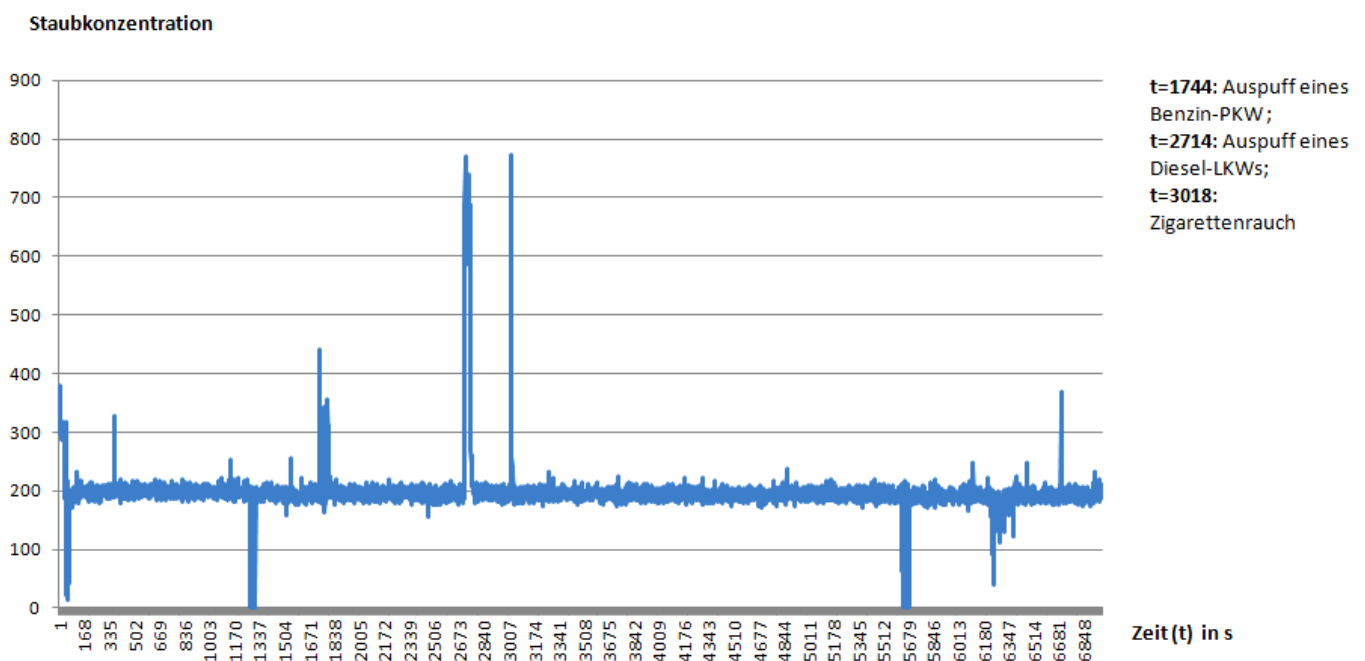
Tabelle 6: Baukomponenten für die Sekundärmission des CanSat

NO	Sensor	Bauteilbezeichnung	Energie
sM1	Staubmessung	Sharp GP2Y1010AU0F	max. 20 mA
sM2	Kamera	LinkSprite JPEG Color Kamera	10-100 mA
sM2a	Speichermöglichkeit der Bilder	JPEG Trigger mit MikroSD-Karte	3,3-5V
sM3	GPS	GTP A013	25 mA
Energieversorgung	Akku (2x)	Lithium Polymer	400 mAh

5.3 Ergebnisse

5.3.1 Staubsensor

Ein Hauptziel unserer Sekundärmission ist die Messung der Staubbelastung der Luft mit Hilfe eines Staubsensors. Dieser ist das sperrigste Bauteil in dem CanSat. Normalerweise wird der Staubsensor außen angebracht. Wir haben aber uns überlegt, dass es auch möglich ist, den Staub zu messen, indem der Sensor innen angebracht wird und durch große Löcher in dem Außengehäuse des CanSat mit Außenluft, inklusive des Staubes in der Luft, versorgt wird. Dafür mussten wir einen Weg finden, Löcher in die Hülle zu bohren, sodass Luft von außen in den CanSat hinein und wieder herausströmen konnte, damit der Staubsensor die Staubkonzentration seiner Umgebung messen kann. In beide Deckel haben wir die Löcher gebohrt. Jetzt kann die Luft hinein strömen, im Staubsensor nach Staubpartikeln untersucht werden und durch den anderen Deckel wieder entweichen.



Grafik 6: Test des Staubsensors als Teil der Sekundärmission unter verschiedenen Bedingungen

Bei diesem Diagramm haben wir den Staubsensor in verschiedene Umgebungen gebracht. Als Erstes haben wir ihn an den Auspuff eines PKWs, der mit Benzin angetrieben wird, danach an den Auspuff eines LKWs, der mit Diesel angetrieben wird und danach haben wir den rauchenden LKW-Fahrer gebeten in unseren Sensor zu pusten. Die Staubkonzentration, also die Rohdaten die der Sensor liefert, liegt in der Spannweite von **0 - 1024**. Die Zeit **t** wird in **Sekunden** angegeben. In der Zeit **t₁=1744** haben wir den Staubsensor an den Auspuff des PKWs gehalten. An dieser Stelle kann man einen eindeutigen Ausschlag sehen, der Sensor hat also die Staubkonzentration messen können. In den Zeiten **t₂=2714** und **t₃=3018** kann man etwas Ähnliches sehen, mit dem Unterschied, dass der Ausschlag größer ist. Der bei der Zeit **t₃** gemessene Zigarettenrauch hat bei dem LKW-Fahrer für großes Erstaunen gesorgt, da der Ausschlag Werte erreicht, die nah an dem des PKW-Auspuffes liegt. Der Sensor hat also auch hier eine Veränderung der Umweltbedingung gemessen. Zu den gemessenen Daten müssen wir sagen, dass der Sensor Partikel erst ab einer

bestimmten Größe messen kann. Das heißt, dass kleinere Teilchen von ihm nicht angezeigt werden und so die Daten zum Teil unvollständig sind. Die Ausschläge gegen Null sind darauf zurückzuführen, dass ein Anschlusskabel aus unserem Testaufbau herausgerutscht war. Aus diesem Test schlussfolgern wir, dass die Funktionsfähigkeit des Staubsensors gegeben ist. Wir konnten also die Programmierung und den Aufbau verwenden, um realistische Werte zu erhalten. Ebenfalls bemerken wir, dass der Sensor gut auf die sich ändernden Staubkonzentrationen reagiert wodurch wir selbst in der kurzen Flugphase Ergebnisse erhalten müssten.

5.3.2 Kamera mit JPEG-Trigger

Ein weiteres Hauptziel unserer Sekundärmission ist die Aufnahme von Bildern des überflogenen Gebietes mit Hilfe einer LinkSprite JPEG Color Kamera und die Speicherung dieser Bilder auf einem JPEG Trigger mit MikroSD-Karte.

Die Kamera ist in den Boden der Außenhülle des CanSat teilweise mit eingearbeitet. Das Objektiv kann so klarere Bilder mit dem Blick zur Erde machen. Durch die Bilder können wir auch im Nachhinein sehen, wie sehr der CanSat getaumelt ist. Außerdem können wir parallel zur ermittelten Staubkonzentration dem untersuchten Ort ein Bild zuordnen und die Umgebung betrachten.

Verbunden ist die Kamera mit Hilfe eines 5 Pin-Kabels mit dem JPEG-Trigger, welcher wiederum abgeschrägt wurde. Erklärung dazu siehe 2.2.2.



Bild 19: Hier sieht man noch einmal die Kamera, verbunden mit dem Trigger

6. Bergungssystem

Unser Bergungssystem besteht aus einem knallig farbigen Fallschirm, einem GPS-Modul und einem Pieper und einem glitzernden Wimpel.

6.1 Fallschirm

Um unseren CanSat nicht für die Abwurfversuche zu benutzen, haben wir eine Testdose hergestellt. Als Testdose füllten wir eine normale 0,3l Getränkedose mit flüssigem Lötzinn, bis sie 350g wog. Mit dieser konnten wir dann alle Fallversuche durchführen. Diese fanden auf dem Dach des FEZ-Berlin statt. An der Dose befestigten wir den Fallschirm und zusätzlich noch eine Schnur, um die Dose nach dem Fall wieder nach oben ziehen zu können. Durch diese Versuche konnten wir feststellen, ob der Fallschirm aufgeht und ob wir die vorgeschriebene Fallgeschwindigkeit erreichen. Den letzten Fallversuch führten wir auf dem Aussichtsturm in Woltersdorf (Berlin) durch. Dieser ist größer als das FEZ und daher können wir das Öffnen und den Flug des Fallschirms länger und besser untersuchen.



Bild 20: Testdose für Abwurfversuche mit dem Fallschirm

Die Testdose wurde für mehrere Abwurfversuche benutzt. Sie wurde besonders bei dem Test mit der Fallgeschwindigkeit von ca. 15m/s demoliert.

Die Wahl des Stoffes für den Fallschirm und die Entscheidung der richtigen Fallschirmgröße war auch etwas schwieriger als wir vorher angenommen hatten. Nach vielen Versuchen haben wir uns für einen kreisförmigen Fallschirm entschieden, der in seiner Mitte ein Loch hat. Sein Schattendurchmesser beträgt 9,0 cm mit Loch und 7,8 cm ohne das Loch. Zum Testen der Fallgeschwindigkeit nahmen wir den Stoff von fertigen Fallschirmen, den wir zuschnitten und umsäumten, damit dieser nicht ausfranst. Dann haben wir ihn auf der Schattenseite mit Gaze (einem Netz) ausgestattet, um so für Stabilität zu sorgen und um zu verhindern, dass der CanSat sich an den Seilen verheddert und dadurch anfängt zu taumeln.

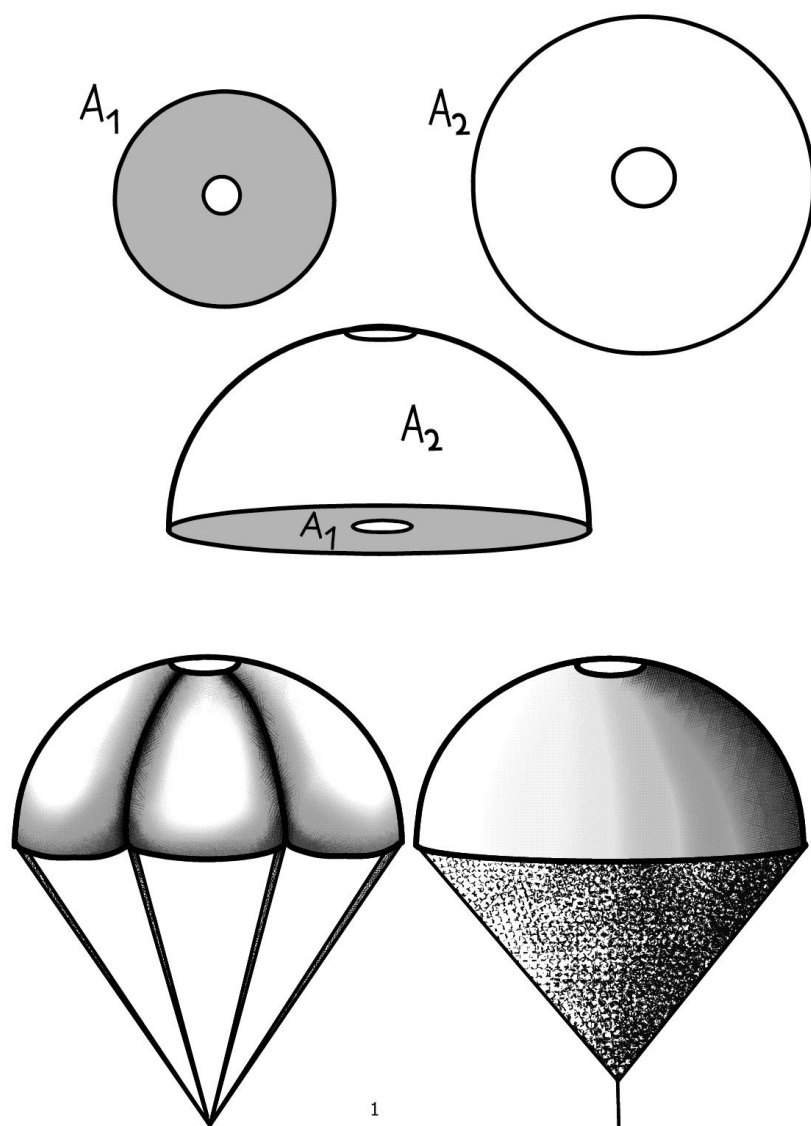


Bild 21: Schema zur Konstruktion des Fallschirms



Bild 22: von uns konstruierter Fallschirm mit Netz



Bild 23: das sind unsere zwei Fallschirme mit unterschiedlichen Farben

Der Außenhülle des CanSat hat an dem oberen Deckel 16 Löcher, von denen immer zwei dicht aneinander liegen, um die Schnüre des Fallschirms dort gut anzuknoten. Der Fallschirm hat nur eine Schnur, die mittig liegt. An diese werden aber noch acht Schnüre angebracht und diese letztendlich am CanSat befestigt. Der Fallschirm wird in einer Chaosfaltung am CanSat verpackt, gehalten von einem Ring aus feinstem Papier. Beim Abwurf des CanSat wird sich der Fallschirm aus dem Papierring befreien und entfalten. Das Papier zerreißt dabei und fällt zu Boden.

Berechnung des Fallschirmdurchmessers

Gegeben: $g = 9,81 \frac{m}{s^2}$

$$c_w = 1,33$$

$$\rho = 1,2 \frac{kg}{m^3}$$

$$v = 15 \frac{m}{s}$$

$$m = 0,35 kg$$

// Erdbeschleunigung

// Strömungswiderstand

// Luftdichte

// Wettbewerbsvorgabe, Fallgeschwindigkeit

// Masse des Lansat, Wettbewerbsvorgabe

Gesucht: $A_{\text{Schatten}} = A_s$; r_s

$A_{\text{Fallschirm}} = A_F$; r_F

// Schattenfläche; - radius

// Fallschirm- "- ; - - "-

Lösung: $m \cdot g = \frac{1}{2} \cdot A \cdot c_w \cdot \rho \cdot v^2$

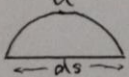
$$A_s = \frac{2 \cdot m \cdot g}{c_w \cdot \rho \cdot v^2}$$

$$= \frac{2 \cdot 0,35 kg \cdot 9,81 \frac{m}{s^2}}{1,33 \cdot 1,2 \frac{kg}{m^3} \cdot (15 \frac{m}{s})^2}$$

$$A_s \approx 0,019 m^2 = \pi \cdot r_s^2 \Leftrightarrow r_s = \sqrt{\frac{A_s}{\pi}} = \sqrt{\frac{0,019 m^2}{\pi}}$$

$$r_s \approx 0,077 m$$

$$A_s < A_F$$



$$u = d_F = \pi \cdot r_s$$

$$= \pi \cdot 0,077 m$$

$$d_F \approx 24,43 cm = 0,244 m = 2 r_F \Leftrightarrow r_F = \frac{d_F}{2} = \frac{0,2443 m}{2}$$

$$r_F = 0,122 m$$

↳ ohne Loch!

Berechnung des Lochradius

$$A_F = \pi \cdot r_F^2 = \pi \cdot 0,122^2 m^2 \approx 0,04687 m^2$$

r_F wird auf 13 cm gesetzt

$$r_{\text{Loch}} = r_F - \sqrt{\frac{A_F}{\pi}} \approx 0,13 m - \sqrt{\frac{0,04687 m^2}{\pi}}$$

$$r_{\text{Loch}} \approx 0,00785 m \approx 0,785 cm$$

↳ Loch wird auf 1 cm gesetzt

Daraus resultiert:

$$r_F = 13 cm$$

$$r_{\text{Loch}} = 1 cm$$

Bild 24: Berechnung der Fallschirmgröße

6.2 Pieper

Der Pieper soll uns bei der Bergung des CanSats behilflich sein. Es ist ein winziger Lautsprecher, aufgebracht auf der zweiten neuen Platine, der Pieptöne aussendet. Wir können ihn mit einer beliebigen Melodie ausstatten, die wir im Programm festlegen und je nach Bedarf ändern können.

Wir haben zum Test des Piepers die Titelmelodie von dem Spiel „Super Mario Bros.“ programmiert und damit die Annahme bestätigt. Der Pieper wird bei der Mission entweder ein Piepen in regelmäßigen Abständen oder ein SOS in Morsezeichen erzeugen.



Bild 25: Pieper (links oben) und GPS-Sender (in der Mitte) auf der zweiten neuen Platine

6.3 GPS

Der GPS-Sender wird im CanSat eingebaut. Er wird ebenfalls bei der Bergung nach der Landung behilflich sein. Seine Aufgabe ist es, möglichst genau die Koordinaten der Position des Satelliten zu ermitteln und an die Bodenstation zu senden. Damit können wir aber nicht nur die Position bestimmen, sondern auch, in welcher Höhe sich der CanSat befindet. Diese Information dient lediglich zum Vergleich mit der Höhe, die wir mit dem Drucksensor ausrechnen, um sicher zu gehen, dass der ausgerechnete Wert im Bereich des Möglichen liegt. Wir werden an der Bodenstation einen Computer haben. Dann können wir unsere aktuelle Position herausfinden und die gesendeten Daten eingeben und ermitteln so, in welche Richtung wir gehen müssen, um den CanSat zu bergen. Den GPS-Sender sieht man auf dem Bild 24.

Table-2: GGA Data Format			
Name	Example	Units	Description
Message ID	\$GPGGA		GGA protocol header
UTC Time	064951.000		hhmmss.sss
Latitude	2307.1256		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12016.4438		dddmm.mmmm
E/W Indicator	E		E=east or W=west
Position Indicator	Fix 1		See Table-3
Satellites Used	8		Range 0 to 14
HDOP	0.95		Horizontal Dilution of Precision
MSL Altitude	39.9	meters	Antenna Altitude above/below mean-sea-level
Units	M	meters	Units of antenna altitude
Geoidal Separation	17.8	meters	
Units	M	meters	Units of geoids separation
Age of Diff. Corr.		second	Null fields when DGPS is not used
Checksum	*65		
<CR> <LF>			End of message termination

Bild 26: Hier sieht man die gesendeten Daten des GPS-Modules

Wir benötigen jedoch nur die Weltzeit (UTC Time), Breitengrade (Latitude), Nord oder Süd (N/S Indicator), Längengrade (Longitude), Süd oder West (E/W Indicator), Satelliten (Satellites Used), Höhe über Normalnull (MSL Altitude) in m, Geoid-Höhe (Geoidal Separation) in m.

Die Daten können wir betrachten und ermitteln, wie der Cansat sich bewegt hat, z.B. den nördlichsten und südlichsten Punkt zur Betrachtung der horizontalen Bewegung. Außerdem können wir dann die Region feststellen, indem wir die Koordinaten in Google Maps eingeben und uns letztendlich der entsprechende Ort anzeigt wird.

6.4 Bodenstation

Am Boden werden wir einen Laptop dabei haben, welcher mit Windows 7 arbeitet. Er dient als Datenanzeige, also durch den seriellen Monitor werden uns die gesendeten Daten angezeigt und wir können sie direkt nutzen. Veranschaulicht werden diese in einem Diagramm, das wir mit dem Grafikprogramm „Origin“ bereits vor Ort erstellen können. Somit erhalten wir direkt einen Überblick über die Zustände des Satelliten und seiner Sensoren. Die Empfängerreichweite wird mit Hilfe der Yagi-Antenne verstärkt und wir erhalten die Daten selbst bei großer Entfernung. Zur Bergung des Satelliten wird ein weiteres GPS-Modul an die Bodenstation angeschlossen. Mit diesem können wir dann unsere GPS-Daten auslesen und uns in die Richtung des Satelliten bewegen. Es handelt sich hierbei um den gleichen GPS-Sender wie der im CanSat.

6.4.1 Yagi-Antenne

Eine Yagi-Uda-Antenne ist eine Antenne zum Empfangen und Senden elektromagnetischer Wellen. Da der Empfänger an dem Arduino nicht für die Distanz von 1000m ausgelegt ist, verwenden wir eine Yagi-Antenne, die für unsere Frequenzen von 432,99 bis 435,06 ausgelegt ist. Allerdings müssen wir darauf achten, die Antenne immer genau auf den CanSat zeigen zu lassen, sonst bekommen wir keine Daten.

Wir verbinden die Antenne mit dem Empfänger durch ein Koaxialkabel. Der Empfänger befindet sich in einer Schachtel, welche an der Antenne befestigt ist. Wir wählten dafür eine Dose für Blutdruckmessgeräte. Diese kann man aufklappen und kommt somit schnell an den Empfänger heran. Das Verbindungskabel für den PC wurde durch die Außenwand gelegt.

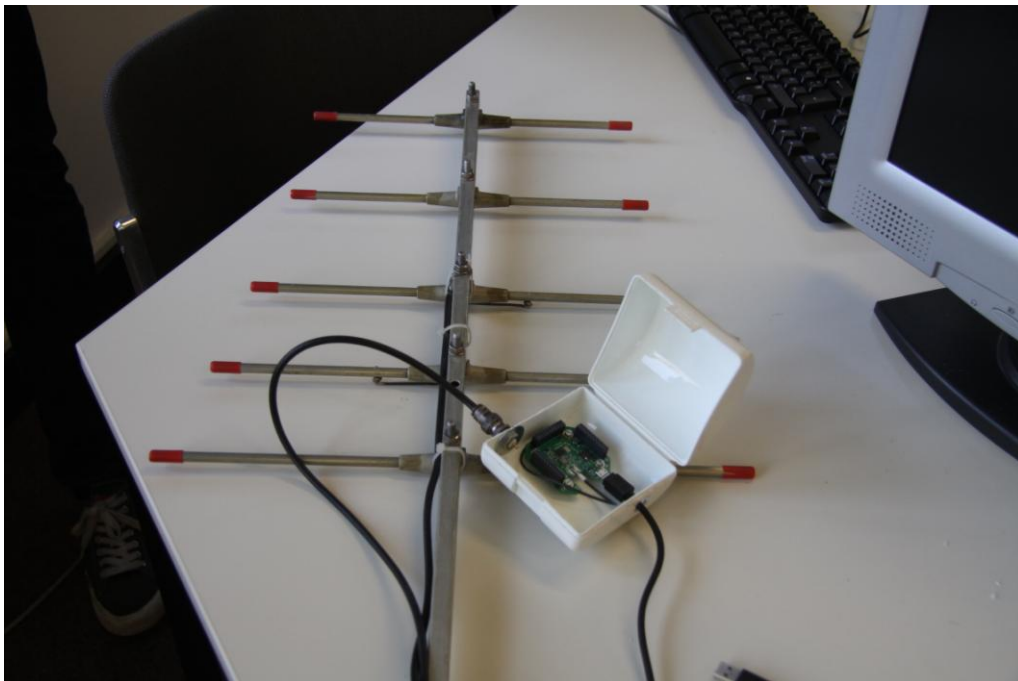


Bild 27: Yagi-Antenne mit Empfänger aus dem CanSat Kit

7. Fertige Software

```
1
/*****
*****
2 * DEFINITIONEN *
3
*****/
4 #define debug 1 // debug = 1 = Ausgabe auf Serial, = 0 nur Senden
5
6 #define Vcc 5.0 //Supply voltage
7
8 #define Bitrate 19200
9
10 #define druckPin A0
11 #define tempPin A1 // int. Sensor LM35 (3 Beine)
12 #define tempNTCPin A2 // ext. Sensor NTC (2 Beine)
13 #define beepPin 5 // digitaler D-Pin 5 für Beeper
14 #define dustPin A9 // angeschlossen an Pin A9
15 #define dustLEDPin 2 // LED an D-Pin 2 angeschlossen
16
17 // LM35 Temp sensor
18 #define TmpSens 100 // Sensitivity. Deg.Celsius /volt
19
20 // NTC-sensor variables
21 #define NTC_Grad 18.1699 // The gradient value for the NTC sensor
22 #define NTC_Off 21.0 // The offset value for the NTC sensor
23
24 #define TRENNER "," // Trennzeichen zwischen den Werten
25
26
*****/
27 * VARIABLEN *
```

28

*****/

29 unsigned long letzteSendung = 0;

30 int Intervall = 1000; // alle wie viel ms wird eine Messung gemacht und
gesendet

31 int TimeSinceStart = 0; // Zeit in Sekunden seit Start des CanSat (Start gleich
einschalten)

32

33 //Primärmission (1)

34 int sensorValue = 0; // Variable to store the value coming from the sensor

35 float druckBoden = 0;

36 float druckDurchschnitt = 0; // Variablen setzen zur späteren Verwendung

37 float hoehe = 0; // aktuelle Hoehe

38 float althoehe=0; // Hoehe bei der letzten Messung

39 float fall= 0; // aktuelle Fallgeschwindigkeit

40

41 boolean ermittleGPS = false; //Wird gerade GPS ermittelt

42

43 // Staubsensor (2)

44 int dustValue = 0; // Wert der vom Dust Sensor kommt

45 float voltage = 0; // errechnete Voltzahl

46 float dustDensity = 0; // letztendliche Staubkonzentration

47

48 // Pieper einschalten

49 boolean steige = false;

50 boolean unter150 = false;

51

52 //allgemeine Laufvariable

53 int i=0;

54

55 //Header fuer die Exceltabelle, wir nur einmal am Anfang gesendet

56 String header = "time,Druck,Temp (int),Temp

(ext),Hoehe,Fallgeschw.,Staubwert,Staubspannung,Staubdichte,GPS

Header,Uhrzeit,Latitude,N/S,Longitude,E/W,Position,Satelliten,HDOP,MSL

Hoehe,M,Geoid Hoehe,M,Age of Diff,Checksumme";

57

```

58
/*****
*****
59 * FUNKTIONEN *
60
*****
*****/
61 //Primärmission (1)
62 float getPressure (int pin) { //Methode für Druck
63     // float volt = Bit2Volt(pin); //liest pin ein
64     // float hPa = 10*(volt/(0.009*Vcc)+(0.095/0.009)); //Umrechnung von V
zu Pa und dann zu hPa
65     float hPa=0;
66     for (int i=1;i<=10;i++) { // 10 Werte mitteln
67         hPa += (float)(((analogRead(pin)/1024.0)+0.095)/0.0009);
68         // hPa = hPa+(hPa*0.02); // Sensor Eichung, zeigt immer zu
wenig an
69         hPa += 17.0; // Sensor Eichung, Fehler laut Datenblatt linear +17
70     }
71     return hPa/10.0; // zurueckgeben
72 }
73

74 float getTemperatureIntern(int pin){ //LM35 an A1
75     float volt=0;
76     float result=0;
77     for (i=1;i<=5;i++){ // 5 Werte mitteln
78         volt = Bit2Volt(pin); //liest pin ein
79         result +=volt*TmpSens; //umrechnung
80     }
81     return result/5.0; // zurueckgeben
82 }
83

84 float getTemperatureExtern(int pin){
85     float volt = Bit2Volt(pin); // liest pin ein
86     float Temp=NTC_Grad*volt-NTC_Off; // umrechnen Volt nach Grad Celcius
87     return Temp; // zurueckgeben
88 }
89

90 float Bit2Volt(int n){ // Funktion zum Lesen und umwandeln

```

```

91  int raw = analogRead(n); // Messdaten von pin 'n' lesen
92  float volt = (float)raw*5.000/1023; // Umrechnung für Volt
93  return volt; // zurückgeben der Voltzahl
94 }
95

96 float calcAltitude(float pressure){ // rechne Hoehe
97     float A = pressure/druckDurchschnitt; // Gemessener Druck
    /Druckdurchschnitt
98     float B = 1/5.25588; // B als Konstante einsetzen (aus Datenbank)
99     float C = pow(A,B); // Logarithmus von A und B
100     C = 1 - C;
101     C = C /0.0000225577;
102     return C;
103 }
104

105 String floatToString(float number){
106     String stringNumber = "";
107     char tempChar[10];
108     dtostrf(number, 4, 2, tempChar);
109     stringNumber += tempChar;
110     return stringNumber;
111 }
112

113 String GetGGA() {
114     //Serial.println("ermittle gga");
115     String gps_output = "";
116     boolean GGAready = false;
117     boolean start = false;
118     int inByte = 0;
119     char inChar;
120
121     ermittleGPS = true;
122     while (!GGAready) {
123         // Serial.println(millis());
124         if (Serial3.available()) {
125             inChar = Serial3.read();
126             inByte = int(inChar);
127             //Serial.print(inChar);    // Debug Ausgabe
128             //Serial.print(inByte);    // Debug Ausgabe
129             //Serial.print(", "); // Debug Ausgabe

```

```

130         if (start) {
131             if (inByte==13) { // Abbruch Bedingung erreicht
132                 GGAready=true;
133             } else {
134                 gps_output += inChar;
135             }
136         } else { // warten auf 10 im Stream, also Ende der Zeile
137             if (inByte==10) {
138                 start = true;
139             }
140         } // Ende if (start)
141     } // Ende if (available)
142 } // Ende while
143 ermittleGPS = false;
144 return(gps_output);
145 }
146
147
/*****
*****
148 * SETUP ROUTINE *
149
*****
*****/
150 void setup () { //Voreinstellungen
151     pinMode(16, OUTPUT); // init LED Debug beim Einschalten
152     pinMode(17, OUTPUT); // init LED
153     pinMode(18, OUTPUT); // init LED
154     pinMode(19, OUTPUT); // init LED
155     pinMode(20, OUTPUT); // init LED
156     pinMode(21, OUTPUT); // init LED
157     pinMode(22, OUTPUT); // init LED
158     pinMode(23, OUTPUT); // init LED
159     digitalWrite(16, LOW); // set LED on
160     digitalWrite(17, LOW); // set LED on
161     if (debug == 1) {
162         Serial.begin(Bitrate);
163     }
164     digitalWrite(18, LOW); // set LED on
165     Serial1.begin(Bitrate); // Initialisiere das Sendmodul (19200 bits per
second):
166

```



```

167  Serial3.begin(9600);                                // init GPS an Serial2
168  // Serial3.println("$PMTK220,100*2F"); // 10 Hz, ging gut mit Arduino, mit
T-Board nicht mehr
169  Serial3.println("$PMTK314,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0*29"); // Nur
GGA ausgeben
170  digitalWrite(19, LOW); // set LED on
171

172  pinMode (beepPin,OUTPUT);                            // Piepser zum Laerm
machen
173  pinMode (dustLEDPin,OUTPUT);                        // LED des Staubsensor
174

175  digitalWrite(20, LOW); // set LED on
176

177  //Primärmission
178  float hPa;
179  int j = 0;
180  for (int i = 0; i < 20 ; i++) { // Schleife, 20 mal
181      j++;
182      hPa = getPressure(druckPin);
183      if (800 < hPa && 1500 > hPa){
184          //zum Berechnen des 0-Druckes...sollten unwahrscheinliche Werte
gemessen werden,
185          //sollen diese ignoriert und durch neue ersetzt werden
186          druckBoden += hPa;
187      } else {
188          i--;
189      }
190      if (j > 50){
191          i = 20;
192      }
193  }
194  digitalWrite(21, LOW); // set LED on
195  druckDurchschnitt = druckBoden/20; // Berechnung des
Druckdurchschnittes
196
197  if (debug == 1) { // debugging
198      Serial.print("hPa:");
199      Serial.println(hPa);
200      Serial.print("druckDurchschnitt:");

```

```

201      Serial.println(druckDurchschnitt);
202      Serial.print("setup done : ");
203      Serial.print(hoehe);
204      Serial.println("m");
205      Serial.println(header);
206  } // Ende debug
207  digitalWrite(22, LOW); // set LED on
208  Serial1.println(header);
209  digitalWrite(23, LOW); // set LED on
210
211  // kurzer Beep beim Einschalten
212  digitalWrite (beepPin, HIGH);
213  delay(100);
214  digitalWrite(beepPin,LOW);
215
216  delay(500);      // warten fuer eine halbe Sekunde
217  digitalWrite(16, HIGH); // set LED off
218  digitalWrite(17, HIGH); // set LED off
219  digitalWrite(18, HIGH); // set LED off
220  digitalWrite(19, HIGH); // set LED off
221  digitalWrite(20, HIGH); // set LED off
222  digitalWrite(21, HIGH); // set LED off
223  digitalWrite(22, HIGH); // set LED off
224  digitalWrite(23, HIGH); // set LED off
225  delay(500);      // warten fuer eine halbe Sekunde
226  digitalWrite(16, LOW); // set LED on
227  digitalWrite(17, LOW); // set LED on
228  digitalWrite(18, LOW); // set LED on
229  digitalWrite(19, LOW); // set LED on
230  digitalWrite(20, LOW); // set LED on
231  digitalWrite(21, LOW); // set LED on
232  digitalWrite(22, LOW); // set LED on
233  digitalWrite(23, LOW); // set LED on
234  delay(500);      // warten fuer eine halbe Sekunde
235  digitalWrite(16, HIGH); // set LED off
236  digitalWrite(17, HIGH); // set LED off
237  digitalWrite(18, HIGH); // set LED off
238  digitalWrite(19, HIGH); // set LED off
239  digitalWrite(20, HIGH); // set LED off
240  digitalWrite(21, HIGH); // set LED off
241  digitalWrite(22, HIGH); // set LED off

```

```

242  digitalWrite(23, HIGH); // set LED off
243 }
244
245
246
/*****
*****
247 * LOOP ROUTINE                                     *
248
*****
*****/
249 void loop () { //Schleife <3
250   if (millis()-letzteSendung >= Intervall) { //Sendeintervall = 1s
251     letzteSendung = millis();
252     TimeSinceStart = letzteSendung / 1000;
253     String output = "";
254
255     // ***** PRIMÄRMISSION
*****
256     float hPa = getPressure(druckPin);
257     hoehe = calcAltitude(hPa);
258     fall = (altehoehe - hoehe) / (Intervall / 1000) ;
259     altehoehe = hoehe;
260
261     if (TimeSinceStart>100) { // Erst nach 100 Sekunden prüfen, so
schnell geht da nix
262         // Steigt die Rakete
263         if (steige==false && hoehe>150){
264             steige = true;
265         }
266         // Fällt der CanSat, unter 150m soll der Piepser dann
anfangen.
267         if (unter150 == false && steige==true && hoehe<150){
268             unter150 = true;
269         }
270     }
271
272     output += (String)(TimeSinceStart);                // Anzeigen des
Zeitcodes

```

```

273         output += TRENNER;
274         output += floatToString(hPa);                                // Anzeigen des
Druckwertes
275         output += TRENNER;
276         output += floatToString(getTemperatureIntern(tempPin)); //
Anzeigen des Temperaturwertes INT LM35
277         output += TRENNER;
278         output += floatToString(getTemperatureExtern(tempNTCPin)); //
Anzeigen NTC Temperaturwert EXT
279         output += TRENNER;
280         output += floatToString(hoehe);
281         output += TRENNER;
282         output += floatToString(fall);
283
284         // ***** STAUBSENSOR
*****
285         digitalWrite(dustLEDPin, LOW);                                // LED anschalten
286         delayMicroseconds (280);
287         dustValue=analogRead(dustPin);                                // Staubsensor
auslesen
288         delayMicroseconds (40);
289         digitalWrite(dustLEDPin, HIGH);                                // LED ausschalten
290
291         delayMicroseconds (9680);                                    // Abklingen,
vermutlich nicht nötig
292
293         voltage = dustValue *0.0048875;                                // umrechnen der
ausgelesenen Daten zu Volt
294         if (voltage <= 3.5) {                                        // nicht Linearität des
Sensors beachten
295             dustDensity = voltage /7.5;
296         }
297         else {
298             dustDensity = (voltage-3.5)*2 +0.4;
299         }
300
301         output += TRENNER;
302         output += (String)(dustValue);                                // Darstellen von
gemessener Voltzahl
303         output += TRENNER;
304         output += floatToString(voltage);                            // errechneter Voltzahl

```

```

305         output += TRENNER;
306         output += floatToString(dustDensity); // letztendliche
Staubkonzentration
307

308         // ***** GPS *****
309         String temp = "";
310         if (!ermittleGPS) {
311             temp = GetGGA();
312             if (!temp.equals("")) {
313                 output += TRENNER;
314                 output += temp;
315                 output += TRENNER;
316             }
317         }
318
319         if (debug == 1) {
320             Serial.println(output);
321         }
322         Serial1.println(output);
323
324     } // Ende im Sekudentakt Senden
325
326     if (unter150){ // ab 150m soll der Pieper sich bemerkbar machen.
327         // Test Beeper
328         digitalWrite (beepPin, HIGH);
329         delay(100);
330         digitalWrite(beepPin,LOW);
331     }
332 } // Ende Loop

```

8. Problembehandlung

In der Entwicklungsphase des CanSat kamen viele Probleme mit den Sensoren, der Wahl des Außenmaterials, der Platzaufteilung im Cansat und der Programmierung auf. Trotz dessen haben wir die Probleme gemeistert, die Sensoren perfektioniert und den CanSat so fertiggestellt. Im folgenden Text werden wir jedes einzelne Problem auflisten und unsere Lösungen dazu aufzählen.

Als erstes haben wir uns mit dem Temperatursensor beschäftigt. Dadurch, dass wir die Temperatur innerhalb und außerhalb des Satelliten messen, brauchen wir zwei Sensoren: der LM35 und einen NTC. Der LM35 hatte seltsame Werte angegeben, Temperaturen die 400°C überschritten, was auf einen Defekt zurückzuführen war. Wir wissen bis heute nicht woran das lag, konnten aber das Problem lösen, indem wir ihn einfach durch einen neuen Sensor ersetzt haben. Nun bekommen wir realistische Werte. Ein weiteres Problem stellte die Geschwindigkeit des Erfassens der Daten. Die Sensoren brauchen zu viel Zeit um mitzukriegen, dass sich die Temperatur um sie herum ändert. Das stellt ein großes Hindernis dar, denn der Fall dauert nur ein paar Sekunden und wir haben auch nur einen Versuch, und wenn die Sensoren nur eine geringe Änderung anzeigen, können wir kein realistisches Temperaturprofil erstellen. Um dies zu beheben, verwenden wir notfalls ein Modell über die Temperaturschwankungen in Relation zur Höhe. Auch wenn hier andere Einwirkungen wie z.B. Winde nicht berücksichtigt werden, kann man es zumindest als Vergleich verwenden.

Die Sensoren waren auf einer Platine angelötet, die wir mit dem CanSat-Kit schon fertig bekommen haben. Diese Platine war allerdings fehlerhaft gelötet und wir haben falsche Werte damit gemessen. Also haben wir eine neue Platine dafür hergestellt und sie darauf angebracht. Als erstes waren wir zuversichtlich, dass sie jetzt funktionieren würden, haben aber dann bemerkt, dass wir einen falschen Widerstand angelötet haben. Dieser wurde ausgetauscht. Als nächstes haben wir den Drucksensor in Anspruch genommen. Dieser war ebenfalls im Kit enthalten und auch schon auf der Platine aufgelötet. Ihn haben wir auch auf die selbstgemachte Platine gesetzt. Da er aufrecht steht und so unnötig Platz in die Höhe verbraucht, haben wir ihn umgeklappt und Platz für den Rest geschaffen. Auf der neuen Platine haben wir einen neuen Kondensator für den Sensor eingelötet. Der nun gemessene Luftdruck war permanent 50 hPa unter Normaldruck. Wir vermuteten, dass der Kondensator die Schuld daran trug. Auch dieser wurde ausgetauscht. Nun bekamen wir richtige Werte und konnten mit dem Drucksensor abschließen.

Dann nahmen wir den Staubsensor ins Visier. Als er bei uns angekommen ist und wir ihn ausprobieren wollten, bemerkten wir, dass ein bestimmter Stecker fehlte, damit man Kabel anschließen konnte. Das hinderte uns am Weiterarbeiten. Wir haben den Stecker dafür bei einem Elektronikfachhandel erworben und dachten, wir hätten das Problem gelöst. Natürlich wurde uns trotz Vorzeigen des Anschlusses ein falscher Stecker verkauft. Letztendlich haben wir online doch noch das richtige Modell gefunden und konnten die notwendigen Tests durchführen. Ein weiteres Problem stellte die Größe des Sensors dar. Er verbrauchte ungefähr 30% des gesamten Platzes innerhalb des Satelliten. Das konnten wir nur durch die „Verkleinerung“ der anderen Sensoren kompensieren. Der Staubsensor an sich kann nur Partikel ab einer bestimmten Größe messen. Auch wenn es nicht änderbar ist, bleibt diese Schwierigkeit.

Die Kamera erhält ihre Befehle über den JPEG-Trigger. Sie macht Bilder und speichert sie, allerdings war die Zeitspanne zwischen den Aufnahmen der Bilder zu groß. Sie hat alle 4 Sekunden ein Bild gemacht, doch bei einer Flugzeit von maximal 120 Sekunden ist dies unproduktiv. Lösen können wir das Problem bisher noch nicht und hinzu kommt, dass nicht ausgeschlossen ist, dass der CanSat viel wackelt, während er hinunter fällt. Dadurch könnten wir eine Mehrheit an verwackelten Bildern erhalten.

Beim Kauf des Triggers haben wir keine guten Erfahrungen gemacht. Wir haben insgesamt 4 Trigger bestellt, wovon 2 von Anfang an nicht funktioniert haben und einer während der Testzeit den Geist aufgegeben hat. Unser Fehler war, dass wir die Kommentare zu dem Produkt nicht beachtet haben, die alle aussagten, der Trigger weise im Laufe der Zeit einen Kurzschluss auf. Letztendlich haben wir einen funktionierenden Trigger rausgesucht. An sich verbrauchte der Trigger außerdem zu viel Platz innerhalb des CanSat. Also haben wir die überschüssigen Verbindungen der Platine abgeschliffen und die Anschlüsse für die Spannung umgelegt. Vorher passte er nur vertikal in den CanSat, jetzt passt er horizontal hinein.

An dieser Stelle haben wir uns weiter mit dem GPS beschäftigt. Erst wollten wir ein anderes Modell als das aktuelle nehmen, ein größeres und schwereres: EM-406A. Bei diesem hatten wir allerdings Probleme mit der Software. Der Sender hat sich schwierig bis fast gar nicht programmieren lassen. Trotzdem haben wir es versucht und sind daran fast verzweifelt. Das andere Modell (GTP A013) war kleiner, leichter und vor allem besser zu programmieren. So haben wir an Platz und an Gewicht des gesamten Satelliten gespart.

Dann hatten wir noch das Problem mit dem Akku. Erst wollten wir eine normale 9V Batterie nehmen. Aber diese sind sehr groß und somit unpraktisch. Sie hat ungefähr 1/6 des Platzes innerhalb des CanSat eingenommen. Letztendlich haben wir uns für zwei 3,7V Akkus entschieden. So sparen wir Platz und Gewicht, da diese zu dem leichter sind als die Batterie. Wir führten beide Stromversorgungskabel zusammen und löteteten sie an die Platine.

Durch die neue Regelung, dass die Fallgeschwindigkeit 15 m/s betragen soll, mussten wir den Fallschirm dementsprechend optimieren. Wir haben eine Formel zum Berechnen der Schattenfläche. Das Problem dabei war, dass wir Anfangs nicht wussten, dass damit die Schattenfläche berechnet wird. Wir haben also eine Weile mit einem zu kleinen Fallschirm die Test durchgeführt. Nun testen wir mit der richtigen Fläche.

Der Fallschirmstoff muss reißfest, nährbar und stabil sein und dem Luftwiderstand standhalten. Wir testeten mit Fallschirmen, die wir gekauft hatten und passten sie unseren Bedingungen an. Außerdem befestigten wir Gaze (ein Netz) daran, das dazu dient, die vorherigen Schnüre zu ersetzen. Diese haben sich zuvor von dem Fallschirmstoff gelöst und dies bereits nach wenigen Tests mit maximaler Fallhöhe von 20 Meter. Das Netz ist an dem gesamten Umfang des Fallschirm gefestigt und hat sich bei den Tests als optimal herausgestellt.

Bei der Yagi-Antenne kamen wenig Schwierigkeiten auf.

Während wir die ganzen Sensoren programmiert und getestet haben, haben wir parallel nach einem Material für die Hülle des CanSat gesucht. Die Hülle musste stabil sein, mindestens 70Kg aushalten, da während des Falls so viel Gewicht auf die Hülle einwirkt, und sie musste die inneren Geräte vor dem Aufprall so gut es möglich ist schützen. Als erstes kam uns ein Carbon-Rohr in den Sinn, da es die nötige Stabilität besaß und in unseren Augen sich als würdig erwies. Allerdings wäre

es bei dem Aufprall höchstwahrscheinlich zersplittert und die Sensoren wären auch kaputt gegangen. Also haben wir uns für eine Hülle aus Plastik entscheiden. Diese ist genauso stabil wie das Carbon-Rohr, mit dem Unterschied, dass es bei dem Aufprall höchstens Risse bekommt, die Sensoren innen aber trotzdem unbeschädigt bleiben. Da die Luftzufuhr gewährleistet sein muss, bohrten wir Löcher in die Deckel oben und unten.

Als nächstes kümmerten wir uns um die innere Polsterung. Wir vermuteten, dass die Sensoren während des Aufpralls aufeinander treffen könnten. Wir mussten demzufolge eine innere Polsterung einbauen, damit dies verhindert wird. Dafür verwenden wir Schaumstoff, der zwischen die Komponenten gelegt wird. Zum Beispiel zwischen den Akkus und den Platinen.

Das größte Problem war die Platzeinteilung. Wie bereits erwähnt, sind viele Komponenten zu groß gewesen, um sie alle einzubauen. Wir haben die Temperatursensoren und den Drucksensor von deren ursprünglichen Platinen abgelötet, zwei neue Platinen hergestellt und sie zusammen mit dem GPS-Sender darauf wieder angelötet. Außerdem haben wir den Trigger abgeschliffen, so dass wir ihn horizontal einbauen konnten. Hierbei wurden alle Kanten abgeschrägt und die Längsseiten gekürzt. Zudem ersetzten wir die große Batterie durch zwei zusammen kleinere Akkus. Diese schieben wir in die vorhandenen Lücken, z.B. zwischen dem Sender und dem Mikrokontroller. Zum Schutz vor Kurzschlüssen etc. polstern wir sie durch Schaumstoff. Dieser wird generell nur in Lücken gegeben um zu polstern und zu stabilisieren. Das Objektiv der Kamera nahm zu viel Platz in der Höhe ein, also haben wir sie in den unteren Deckel eingebaut. Demnach bohrten wir ein abgeschrägtes Loch in den Deckel, in das das Objektiv gesetzt werden kann.

9. Öffentlichkeitsarbeit und zukünftige Projekte

Um den Wettbewerb auch an andere Schüler und Schülerinnen weiterzuleiten, ist dieser nicht nur auf unserer Internetseite www.spaceclub-berlin.de vertreten sondern auch auf der Homepage des orbitall www.orbitall-berlin.de. Im spacecamp 2014 war der CanSat-Wettbewerb das Hauptthema. Die Campteilnehmer bauten selber mit einer Sensebox einen CanSat. Mit einem Wetterballon wurden diese dann auf eine Höhe von ca. 50 Meter gebracht. Währenddessen sammelten sie Daten und zeichneten diese auf.

Einen Pressebeitrag zum geplanten Bau des CanSat im spaceclub findet man unter:

http://www.jugendtechnikschiule.de/upload/pdf/KONTEXIS_48_2014.pdf

All unsere Ergebnisse haben wir am Wochenende des 20. und 21. Septembers 2014 beim Raumfahrtwochenende im FEZ Berlin vorgestellt. Dabei handelt es sich um eine Veranstaltung rund um das Thema Raumfahrt. Dort vertreten ist nicht nur unser spaceclub_berlin, sondern auch das orbitall Berlin und das DLR. Bei dieser Veranstaltung haben unser CanSat und die Satelliten aus Senseboxen aus dem spacecamps 2014 einen ganzen Stand eingenommen. Betreut wird dieser Stand von unserem Team. So haben die Besucher die Möglichkeit sich über das Thema zu informieren und direkt Fragen zu stellen.

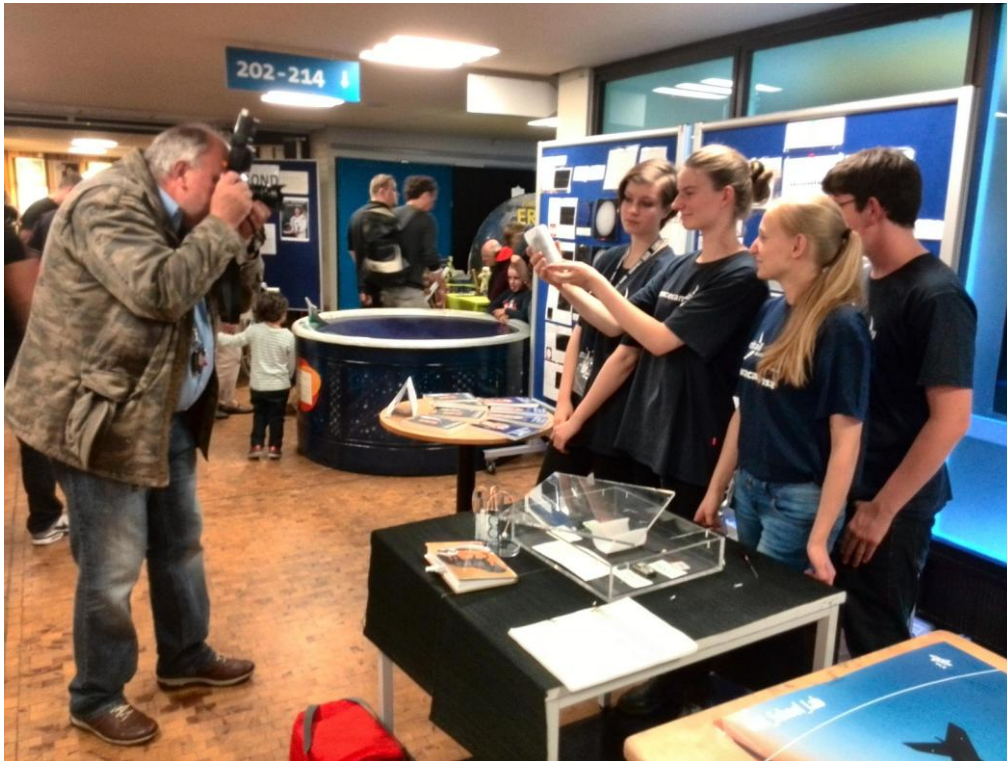


Bild 28: wir stellten den CanSat zum Raumfahrtwochenende (20.9.-21.9.2014) im orbitall aus, wobei auch Presse anwesend war und mit uns ein Interview durchführte

Im Oktober 2014 werden wir eine modifizierte Sensebox zusammen mit einer Kamera an einen Wetterballon anbringen und diesen in die Stratosphäre fliegen lassen, um auch in dieser Höhe Umweltdaten der Sekundärmission messen zu können.

Eine weitere besondere Aktion findet zur Landung des Philae-Landers am 11. November 2014 statt. Dieser Lander soll im Rahmen der Rosetta-Mission auf dem Kometen Tschurjumow-Gerasimenko landen und dort mehrere Messungen durchführen.

Zusätzlich wollen wir bei Jugend forscht 2015 teilnehmen, da sich dieses Projekt aufgrund des Aufwands und der erlernten Fähigkeiten besonders gut für eine fachliche Arbeit eignet.

Außerdem wird in Kürze ein Mitglied des spaceclub_berlin, Jasper Morse, eine Reportage über das Projekt drehen. Er wird uns darüber auf Englisch interviewen und das Video auf seinem YouTube Kanal veröffentlichen.

Die Ermittlung bzw. Kontrolle der Staubkonzentration in der Luft vor allem in Großstädten, wie z. B. in Berlin, ist von großer Bedeutung. Zu hohe Staubkonzentrationen belasten die Umwelt und damit auch die Gesundheit der Menschen. Die Messungen werden permanent durchgeführt. In der Labormission des orbitall wird dafür ein Messplatz eingerichtet, an dem Schüler die Möglichkeit haben, aktuelle Messdaten auszuwerten und bewerten zu können. Dafür soll der CanSat permanent über dem Gebäude des orbitall im FEZ Berlin platziert werden. Die gemessenen Daten werden über einen längeren Zeitraum erfasst und mit öffentlich zugänglichen Umweltdaten verglichen. Diese Daten und ihre Auswertung werden 2015 in einer oder zwei Jugend forscht Arbeiten veröffentlicht.

www.orbitall.de

www.spaceclub_berlin.de

10. Reflexion des Wettbewerbes

Der Wettbewerb war für uns alle eine große Herausforderung. Keiner aus unserem Team hat jemals an so einem großen Gemeinschaftsprojekt mitgearbeitet. So sind wir ohne jegliche Ahnung über die Planung, das Dokumentieren, das Einhalten der Zeit und den eigentlichen Verlauf des Projektes in den Wettbewerb gegangen. Wir haben zu spät angefangen, daran zu arbeiten, was zum Ende hin immer mehr Stress verursacht hat. Wir waren angespannt und hatten die Befürchtung, es nicht rechtzeitig zu schaffen. Das hätten wir wahrscheinlich auch nicht, wenn nicht so viele Leute mitgeholfen und uns betreut hätten. Viele Fachleute aus unserem jetzigem Team sind auf uns zugekommen und haben ihre Hilfe angeboten. Sie haben uns viele Sachen beigebracht und uns in jeder Hinsicht unter die Arme gegriffen. Einige von ihnen kannten wir vorher nicht mal. Durch die enge Zusammenarbeit haben wir uns alle besser oder sogar überhaupt kennengelernt. Es war eine einmalige Erfahrung das Projekt gemeinsam zu Ende zu führen. Nun möchten wir ihnen unser Feedback zum Wettbewerb mitteilen, was wir für gut befunden haben und womit wir nicht zufrieden waren.

Fangen wir mit den positiven Aspekten an. Die Zeit von beinahe einem dreiviertel Jahr war auf jeden Fall ausreichend. In dieser Zeit hat man es gut schaffen können das Projekt zu vollenden und ein vorzeigbares Ergebnis zu erhalten. Außerdem war es eine große neue Herausforderung, die wir hoffentlich gemeistert haben. Die Idee an sich zu einem Wettbewerb zum Bau eines Satelliten zu entwickeln, und damit Jugendliche für Technik und die Erforschung unserer Umweltbedingungen zu begeistern, war äußerst einfallsreich. Außerdem waren die Anforderungen durchaus zu meistern. Sie waren nicht zu hoch gestellt und die Anzahl war ebenfalls im annehmbaren Bereich.

Kommen wir nun zu eher unerfreulichen Seiten des Wettbewerbs. Die Änderungen der Wettbewerbsbedingungen o kurz vor der Abgabe des Endberichtes hat uns viele Male aus der Bahn geworfen. Dadurch, dass zum Beispiel immer wieder neue Fallgeschwindigkeiten festgelegt wurden, mussten wir unseren Fallschirm komplett neu gestalten und zusammenbauen, und das einige Male. War die Geschwindigkeit beim Fallen so hoch angelegt, dass wir zusätzliche Polsterungen einbringen mussten und da viel Zeit in Anspruch nahm. Es wurden auch komplett neue Anforderungen gestellt, wie zum Beispiel der glitzernde Wimpel. Zudem ging der Arduino, der mit dem CanSat-Kit geliefert wurde, kurz vor Schluss kaputt. Aber nicht nur der, sondern beinahe alle Platinen im Kit hatten Fehlfunktionen. Wir waren gezwungen fast jede Platine neu herzustellen und mit dem Rest zu verkabeln, was uns ebenfalls in der Zeit zurück warf. Ein weiterer Punkt stellt der Shop Watterott dar. Wir durften nur bei diesem Shop unsere Sensoren und Materialien einkaufen. Wir haben viele schlechte Ware zugeschickt bekommen. Entweder waren die meisten kaputt oder schlampig gelötet oder sie waren unvollständig. So ging es bei beinahe allen Teilen, die wir dort bestellt haben.

Nicht desto Trotz danken wir Ihnen, das sie uns die Teilnahme an dem Wettbewerb ermöglicht haben und freuen uns, Ihnen unsere Ergebnisse zu präsentieren.