



Faculteit Wetenschappen  
Departement Fysica

## EXPERIMENTELE BACHELORPROEF

---

# Meting van de stralingsintensiteit in de stratosfeer met behulp van weerballonnen

---

Auteur:  
Andy Martinez

Promotor:  
Prof. Dr. N. Van Remortel

# Contents

<b>1</b>	<b>Inleiding: Kosmische stralen</b>	<b>4</b>
1.1	Oorsprong en Opbouw . . . . .	4
1.2	Hoogte afhankelijkheid . . . . .	5
1.3	Belang van onderzoek . . . . .	6
<b>2</b>	<b>HACORD</b>	<b>6</b>
2.1	Belangrijkste componenten . . . . .	7
2.1.1	Deeltjes detectie en puls modificatie . . . . .	7
2.1.2	Voeding . . . . .	9
2.1.3	Sensors . . . . .	9
2.1.4	Data-acquisitie . . . . .	9
2.2	Vorige iteraties . . . . .	11
2.3	Vluchten . . . . .	11
<b>3</b>	<b>Data analyse</b>	<b>12</b>
3.1	KMI vlucht . . . . .	12
3.1.1	Hoogte en Temperatuur profielen . . . . .	12
3.1.2	Hoogte afhankelijkheid van de flux . . . . .	13
3.1.3	Hoekafhankelijkheid van de flux . . . . .	16
3.2	KNMI vlucht . . . . .	18
3.2.1	Hoogte en Temperatuur profielen . . . . .	18
3.2.2	Hoogte afhankelijkheid van de flux . . . . .	19
3.2.3	Hoekafhankelijkheid van de flux . . . . .	21
3.2.4	Detector falen tijdens vlucht . . . . .	22
3.3	Vergelijking resultaten met andere vluchten . . . . .	26
3.4	Besluit . . . . .	26
<b>4</b>	<b>ESA REXUS/BEXUS Programma</b>	<b>26</b>
4.1	Inleiding . . . . .	26
4.1.1	Organisatie . . . . .	26
4.1.2	Waarom interessant voor ons ? . . . . .	27
4.1.3	Selectie proces en verder verloop van het programma . . . . .	27
4.2	Software . . . . .	28
4.2.1	E-Link: Draadloze verbinding met experiment via ethernet . . . . .	28
4.2.2	Druksensor backup systeem . . . . .	30
4.2.3	Grondstation . . . . .	35
4.2.4	Vlucht Simulatie . . . . .	37
4.3	Elektronica . . . . .	43
4.3.1	Nieuwe PCB . . . . .	43
4.3.2	Aanpassingen t.o.v. HACORD-III . . . . .	43
<b>5</b>	<b>Algemeen Besluit</b>	<b>44</b>

# Voorwoord

Allereerst wil ik Ir. Wim Beaumont bedanken omdat hij ongelooflijk veel geduld heeft getoond bij het aanleren van elektronica zaken aan een fysica student. Hij was altijd bereid om mij te helpen ondanks zijn eigen zeer drukke schema. Uiteraard moet Dr. Nick Van Remortel ook bedankt worden voor de kans die hij me heeft gegeven om in samenwerking met de ESA een experiment te doen welke mij zeer veel heeft geleerd over praktische zaken die buiten het curriculum van de fysica richting vallen (Van embedded systems en pcb design tot organisatie van een team). Dit project zal me zeker en vast een duwtje in de rug geven voor het leven na mijn studies.

## Abstract

In this thesis we continue the work on the HACORD which stands for High Altitude Cosmic Ray Detector. This is a detector that has been created and improved by the University of Antwerp over several years.

The detector uses Geiger Muller tubes to detect cosmic rays. This detector gets flown on weather balloons of meteorological institutes in Belgium (KMI) and the Netherlands (KNMI) where it gathers data about the cosmic ray flux in function of the height of the detector. With this data we can measure the influence of the solar activity cycle on the cosmic ray flux. What we find is that the cosmic ray flux hasn't increased significantly over the years.

Since the HACORD-III we were able to discriminate between several angles where cosmic rays could come from. These results seem to agree with the expected results. We see that the more distance the cosmic rays have to traverse inside the atmosphere the less we see of them. We can see this in effect in the results because there are more particles colliding with the detector vertically than diagonally and there are more particles colliding diagonally than horizontally. This changes with the changing density of the atmosphere. When the density of the atmosphere is practically 0 it doesn't matter where the particles are coming from because there isn't much to collide with at high altitudes. The result at maximum altitude is not accurate because there aren't many measurements at that height because weather balloons pop almost instantly at the maximum height. This makes it so that the error bars on the measurements at maximum heights are big which makes it unreliable to say anything definitive about these results.

During one of the flights something went wrong with the detector. This was the perfect opportunity to do failure analysis which is not something that gets done often since most experiments that are done in the classroom are perfect setups where not much can go wrong. The failure was the detector that kept restarting during one of the colder parts of the flight.

This detector was also used to involve middle school students into the scientific process. Every year a group of interested students is invited to the university to participate in experiments with the detector. This year we worked with a group of students from the Technasium in Goess, The Netherlands. Although this part isn't explained in detail in the thesis it is still worth mentioning since getting interested students involved in an actual experiment is beneficial to sparking their interest even more.

Another major part of this thesis is the REXUS/BEXUS programme. This is a programme organised by the European Space Agency and offers the possibility for university students to create and fly an experiment on either a rocket or a balloon. For this programme the detector got redesigned from scratch so that new components could be added. There were also major additions to the software that were demanded by ESA. The ESA part of this project is still ongoing and will not be finished until January 2016 when the results from the flight in October 2015 will be presented at ESTEC.

# 1 Inleiding: Kosmische stralen

## 1.1 Oorsprong en Opbouw

Kosmische straling is de verzamelnaam van 3 soorten kosmische stralen: Galactische kosmische stralen die vanbuiten het zonnestelsel komen, Anomale kosmische stralen die afkomstig zijn van de interstellaire ruimte op de rand van de heliopauze en Energetische zonnedeeltjes die geassocieerd zijn met zonnevlammen en andere energetische zons activiteiten [2]. Kosmische straling kan opgedeeld worden in primaire en secundaire deeltjes.

Primaire deeltjes zijn diegene waar de 3 vooraf genoemde kosmische stralen soorten uit bestaan. De meeste die worden gedetecteerd zijn afkomstig van de zon, maar een kleine fractie komt van supernovae.

Secundaire deeltjes worden geproduceerd door de primaire deeltjes wanneer de primaire botsen op moleculen in onze atmosfeer. Hierdoor ontstaan er secundaire deeltjes die op hun beurt, volgens een cascade effect, met deeltjes in de atmosfeer botsen en nog meer secundaire deeltjes ontwikkelen, dit wordt een shower genoemd.

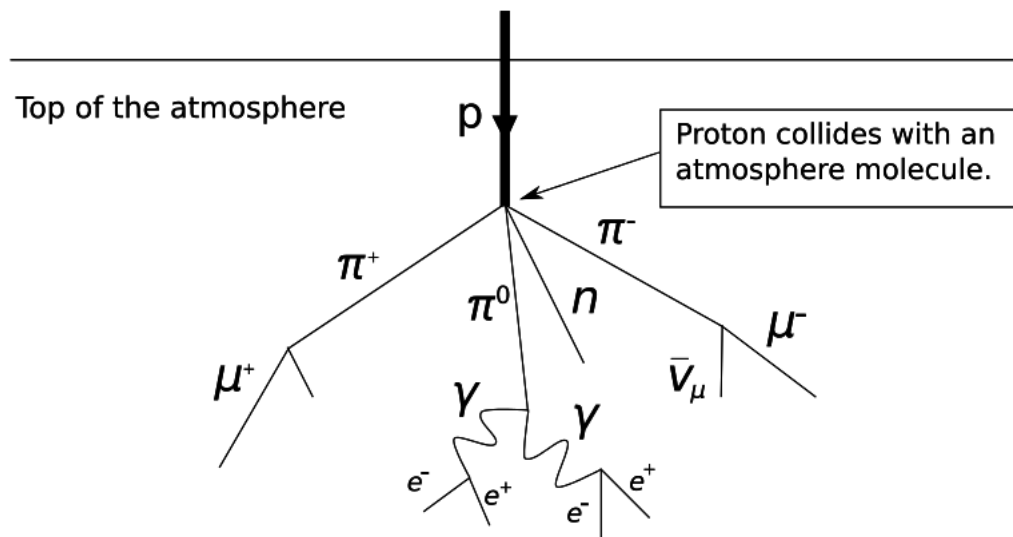


Figure 1: Ontstaan van secundaire deeltjes door het vervallen van een primair deeltje

Primaire kosmische stralen bestaan hoofdzakelijk uit protonen, fotonen, elektronen en lichte kernen tot en met ijzer. Secundaire deeltjes bestaan uit fotonen, elektronen, kaonen, pionen, positronen en muonen. Waarvan de laatste drie werden ontdekt uit kosmische stralen.

De kosmische stralen flux in functie van de energie ziet er als volgt uit:

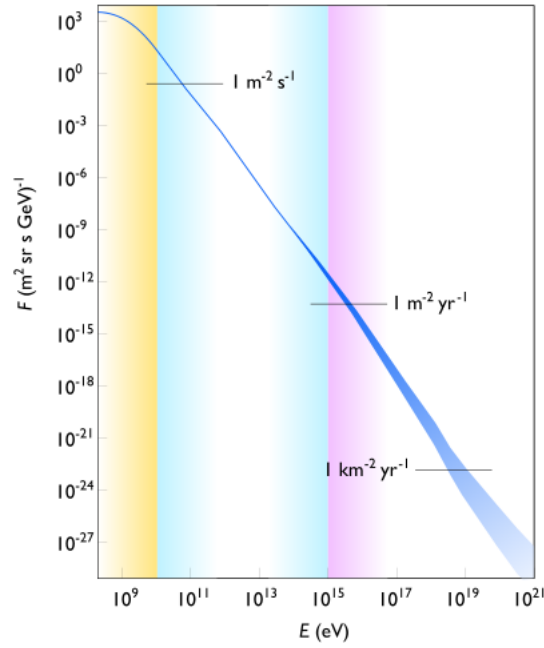


Figure 2: Kosmische stralen flux in functie van de energie

Dit maar om een idee te geven over de energieschaal en de flux van de bijbehorende kosmische stralen. De energie van kosmische stralen kunnen miljoenen keren groter zijn dan de energie van de deeltjes die worden versneld in de LHC. Hierdoor kunnen kosmische stralen ons processen laten zien die we (nog) niet kunnen zien in de LHC. Het enige probleem hierbij is dat kosmische stralen niet te controleren zijn waardoor het lastig is om gecontroleerde experimenten uit te voeren met kosmische stralen.

## 1.2 Hoogte afhankelijkheid

Als de flux van kosmische stralen wordt gemeten in functie van de hoogte dan zal deze stijgen tot op een bepaald punt. Dit punt wordt het Pfozter maximum genoemd [4]. Dit is de hoogte waarop de maximale flux wordt gemeten.

De flux op het Pfozter maximum is afhankelijk van de zonnecyclus. Als de zon zeer actief is (veel zonnevlekken) dan zullen er magnetische velden ontstaan op de zon die kosmische stralen wegbuigen van de aarde. Dit zorgt ervoor dat de flux op Pfozter maximum lager zal liggen dan wanneer de zon minder actief is.

Op volgende figuur wordt het verloop van de flux in functie van de hoogte getoond:

Als we deze figuur doorlopen van grote hoogtes naar kleine hoogtes dan zien we dat de flux niet maximaal is op de maximale hoogte. De showers waar vooraf over werd gesproken hebben een tijd nodig om te ontwikkelen doordat deeltjes vervallen en doordat er nog botsingen kunnen veroorzaken. Op grote hoogtes hebben de showers de tijd nog niet gehad om goed te ontwikkelen waardoor er een kleinere flux wordt gemeten. Naarmate er dieper in de atmosfeer wordt gemeten dan stijgt de flux omdat de shower zich meer en meer kan ontwikkelen. De intensiteit neemt na het Pfozter maximum (rond de 20 km) weer af doordat de energie van de deeltjes in de shower zo klein zijn geworden dat de deeltjes worden geabsorbeerd door de atmosfeer.

Waar wij hoofdzakelijk in zijn geïnteresseerd is hoe de flux op het Pfozter maximum gemeten met onze detector varieert over de jaren heen. Ook zijn we geïnteresseerd in de afhankelijkheid van het verloop van de flux intensiteit met de hoogte met het seizoen, de richting en de noorderbreedte.

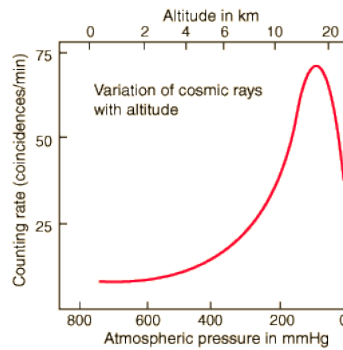


Figure 3: Verloop van de flux in functie van de hoogte [3]

### 1.3 Belang van onderzoek

Kosmische stralen hebben genoeg energie om de toestand van elektronische componenten in een geïntegreerd circuit te veranderen. Dit veroorzaakt errors zoals corruptie van data of incorrect gedrag van CPUs. Dit is een veel voorkomend probleem voor elektronica op grote hoogtes zoals satellieten, maar dit wordt ook meer en meer een probleem voor elektronica op de grond. In 1990 deed IBM een studie waaruit volgde dat computers gemiddeld gezien 1 kosmische stralings geïnduceerde error ondervinden per 256 MB RAM per maand [16]. In 2010 ondervond de Voyager 2 een storing die werd veroorzaakt door 1 geflipte bit welke waarschijnlijk door een kosmische straal werd veroorzaakt.

Een ander zeer belangrijk effect dat kosmische stralen kunnen hebben is op de gezondheid van levende organismen. Op aarde worden we beschermd van deze hoog energetische primaire kosmische stralen door onze atmosfeer. Tijdens bemande ruimtemissies hebben de astronauten deze atmosfeer niet meer als bescherming. Een actief onderzoeksgebied is het effect van kosmische stralen op het centrale zenuwstelsel. Dit onderzoek is belangrijk omdat kosmische straling in de ruimte een van de grotere problemen is voor bemande ruimtemissies naar o.a. Mars.

## 2 HACORD

HACORD staat voor High Altitude Cosmic Ray Detector. Het is de detector die werd ontworpen met de bedoeling om kosmische stralen te detecteren door deze detector met weerballonnen de lucht in te sturen. De detector vliegt al sinds 2011 met ballonvluchten mee. Elk jaar heeft een andere student aanpassingen gemaakt aan deze detector.

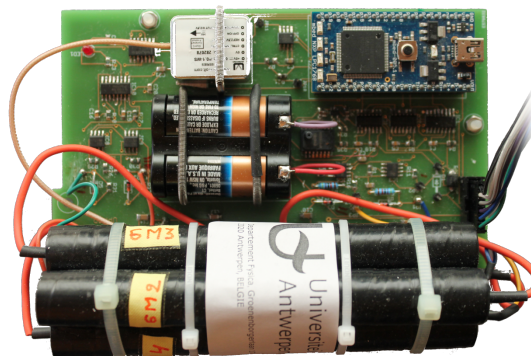


Figure 4: De HACORD-III

## 2.1 Belangrijkste componenten

De detector is in elke iteratie een beetje veranderd, maar de belangrijkste componenten bleven hetzelfde. De componenten van de vorige iteratie (en waarmee de data die in deze thesis wordt besproken werd verzameld) kan worden opgedeeld in 4 delen: Data-acquisitie, Voeding, Deeltjes detectie en puls modificatie, Sensors.

Deze verschillende onderdelen zullen kort apart worden besproken om een idee te geven hoe de detector functioneert.

### 2.1.1 Deeltjes detectie en puls modificatie

Het belangrijkste component op de detector is de Geiger Muller tube. Dit geeft ons de mogelijkheid om deeltjes te detecteren.

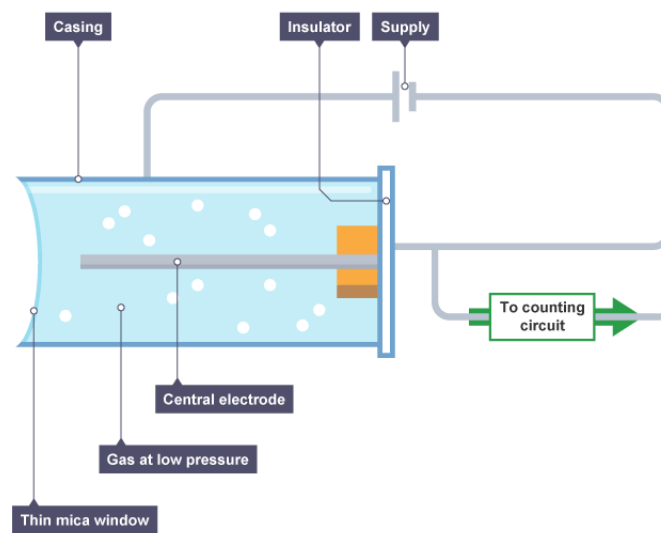


Figure 5: Schematische representatie van een Geiger Muller tube

In de Geiger Muller tube zit een gas (in ons geval een Argon-Neon mengsel) op lage druk gehouden. Tussen de centrale electrode en de wanden van de tube bestaat een groot potentiaalverschil (in ons geval 500V). Wanneer er een deeltje met genoeg energie om het gas te ioniseren in de Geiger Muller tube terecht komt dan zal het gas ioniseren waarna de geïoniseerde deeltjes door het potentiaalverschil versneld zullen worden.

Doordat de geïoniseerde deeltjes door de versnelling genoeg energie krijgen om andere gasdeeltjes te ioniseren onderweg zal er een cascade effect veroorzaakt worden in de tube. Door dit cascade effect zal er een grote elektrische puls verschijnen aan de uitgang van de tube. Deze puls zorgt ervoor dat we te weten komen dat er een hoog energetisch deeltje onze detector heeft geraakt.

Op de versie van de detector waarvan de data wordt besproken stonden er 4 Geiger Muller tubes in de volgende configuratie:



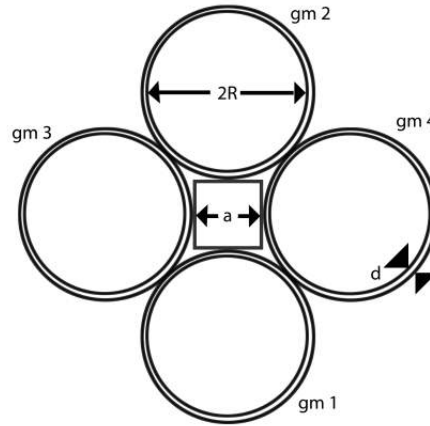


Figure 6: Configuratie van de Geiger Muller tubes op HACORD-III

Deze configuratie werd gekozen zodat er een angulaire afhankelijkheid van de kosmische stralen kan gemeten worden. De deeltjes kunnen verticaal, horizontaal of schuin invallen. Deze mogelijkheden hebben verschillende ruimtehoeken. Deze ruimtehoeken werden berekend door mijn voorganger Jeroen Van Houtte in zijn bachelorthesis [5] en waren:  $0.4373 \cdot 4\pi$  sr voor de ruimtehoek bepaald door gm1 en gm2 en equivalente combinaties en  $0.7009 \cdot 4\pi$  sr voor de ruimtehoek bepaald door gm1 en gm3 en equivalente combinaties. Dezelfde ruimtehoeken zullen gebruikt worden tijdens de analyse van de nieuwe data zodat de resultaten goed kunnen worden vergeleken.

De pulsen die uit de Geiger Muller tubes komen kunnen niet rechtstreeks in onze microcontroller worden gevoerd omdat de pulsen te scherp en te kort van duur zijn. Hiervoor worden er nog een aantal componenten tussen de Geiger Muller tube en de microcontroller gebruikt.

Het eerste component waar de puls door gaat is een comparator. Deze vergelijkt de puls met een referentiespanning. Als de puls groter is dan de referentiespanning dan wordt de puls doorgelaten naar de volgende component. Als het niet groter is dan wordt de output naar ground gekoppelt. Dit zorgt ervoor dat ruis geen counts kan veroorzaken.

Als de puls groter is dan de referentiespanning dan moet de puls nog eerst verkleint en verbreed (in de tijd) worden vooraleer de microcontroller deze puls juist kan behandelen. Dit wordt gedaan met een multivibrator. Deze zorgt ervoor dat een scherpe piek wordt omgevormd tot een blokgolf die de microcontroller wel kan gebruiken. Dit wordt duidelijk gemaakt op volgend beeld van een oscilloscoop:

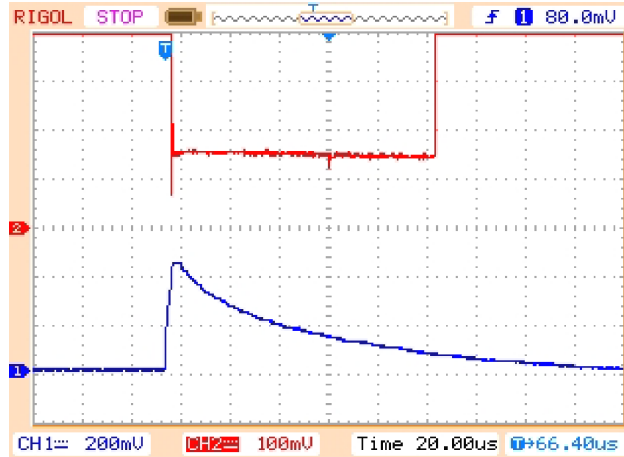


Figure 7: Effect van de multivibrators gebruikt op de detector

Op dit beeld is de scherpe korte puls te zien onderaan het oscilloscoop venster. Als deze door een multivibrator gaat dan krijgen we de blokpuls te zien bovenaan het oscilloscoop venster.

### 2.1.2 Voeding

De detector wordt aangedreven door een batterij van 6V. Doordat de spanning van een batterij vrij snel kan veranderen, door uitwendige factoren zoals temperatuur en druk en doordat de batterij simpelweg wordt opgebruikt, heeft dit als effect dat de spanning die de detector voed niet stabiel is. Om dit te verhelpen wordt de batterij aangesloten aan een lineaire regulator. Deze moet ervoor zorgen dat al de componenten in het experiment altijd ongeveer 5V voeding krijgen.

Het tweede belangrijke component dat te maken heeft met de voeding is de DC/DC converter die de 5V voeding omvormt naar de benodigde 500V om de Geiger Muller tubes te kunnen gebruiken. Onze tubes kunnen al werken vanaf 400V, maar hebben een optimale werkspanning van 500V.

### 2.1.3 Sensors

Naast de Geiger Muller tubes staan er nog een aantal andere sensors op de detector. Er staan 2 temperatuur sensors op de detector: Een NTC en een digitale temperatuur sensor. Deze sensors zijn niet noodzakelijk voor het succes van het experiment, maar geven ons extra informatie voor volgende iteraties van detector (heel lage temperaturen en elektronica gaan soms moeilijk samen).

Naast de temperatuur sensors staat er ook nog een gecalibreerde druksensor op de detector. Dit is een zeer belangrijke component, omdat hiermee de hoogte kan bepaald worden. De hoogte wordt bepaald door een barometrische formule [17]:

$$h = -83.52x^3 + 2205x^2 - 2.564 \cdot 10^4x + 1.306 \cdot 10^5 \quad (2.1.1)$$

met  $x = \log(p)$  waar p de druk is in Pascal. Deze barometrische formule komt uit het U.S. Standard Atmosphere Model.

Met de druksensor kunnen het aantal counts gelinked worden met de hoogte waar de detector zich bevind.

### 2.1.4 Data-acquisitie

Het hart van de detector is een microcontroller genaamd de mbed LPC1768 [6].



## 2.2 Vorige iteraties

De detector die gebruikt werd voor de data die in deze thesis wordt besproken is de HACORD-III. De andere iteraties hadden telkens een ander aantal Geiger Muller tubes. Deze iteraties zagen er als volgt uit:

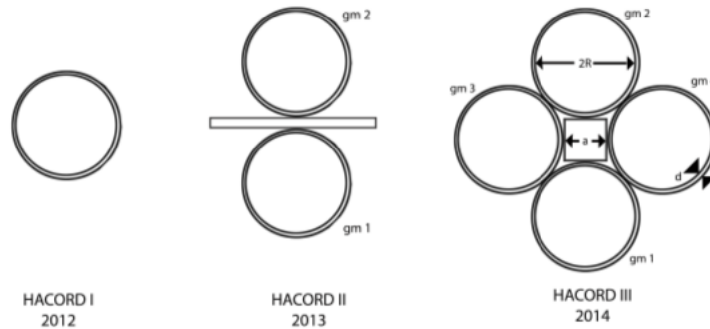


Figure 9: Configuraties van Geiger Muller tubes van oudere iteraties

Ik heb voor deze thesis data verwerkt van de bestaande detector en hierna werd de detector van 0 terug opgebouwd. Dit betekent dat ik een nieuwe PCB heb ontworpen en gesoldeerd. Het aantal Geiger Muller tubes bleef voor deze nieuwste versie hetzelfde, maar de configuratie veranderde naar de volgende:

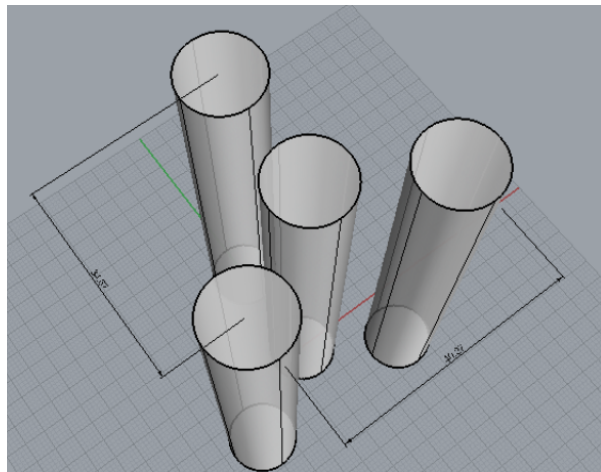


Figure 10: De nieuwe Geiger Muller tube configuratie voor het REXUS/BEXUS project van ESA

Dit nieuwe design werd uitgevoerd doordat de detector een wedstrijd van de European Space Agency (ESA) had gewonnen, het REXUS/BEXUS project. Dit zal in meer detail besproken worden in de tweede helft van deze thesis.

## 2.3 Vluchten

In deze thesis zullen er 2 vluchten worden besproken.

De eerste vlucht gebeurde in Oktober op het KMI. Dit was een project waarbij middelbare scholen hun experiment op een weerballon vna het KMI konden zetten. Doordat de eerste detector in samenwerking met het Sint-Pieterscollege in Jette werd gemaakt kwam de detector hiervoor in aanmerking.

De tweede vlucht gebeurde in April en dit in samenwerking met het Technasium van Goes in Nederland.

Er werd een dagje georganiseerd op de Universiteit van Antwerpen waarbij de leerlingen van Goes experimenten konden doen met de detector en konden bijleren over deeltjesfysica met de nadruk op kosmische straling.

Deze school heeft contact gezocht met het KNMI die met alle plezier onze detector op een van hun weerballonnen wilden zetten. Bij deze vlucht werden ook al voorlopige aanpassingen aan de software getest voor het REXUS/BEXUS project.

## 3 Data analyse

### 3.1 KMI vlucht

#### 3.1.1 Hoogte en Temperatuur profielen

Zoals eerder vermeld wordt de hoogte berekend uit de druk met de barometrische formule:

$$h = -83.52x^3 + 2205x^2 - 2.564 \cdot 10^4x + 1.306 \cdot 10^5 \quad (3.1.1)$$

met  $x = \log(p)$  waar p de druk is in Pascal. Deze barometrische formule komt uit het U.S. Standard Atmosphere Model [17].

Als dit wordt toegepast op de drukmetingen van het KMI krijgen we volgend hoogte profiel:

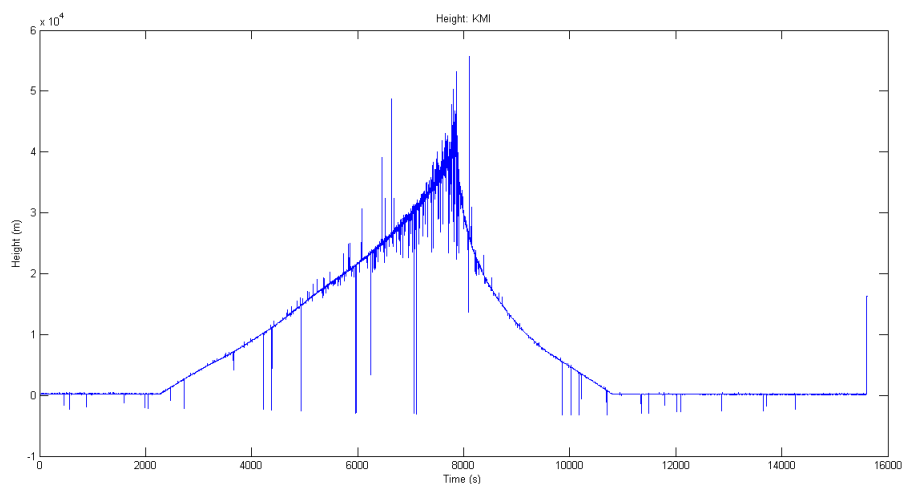


Figure 11: Hoogte profiel van KMI vlucht

Hierop is te zien dat de detector een maximum hoogte had rond de 45 km.

De temperatuur werd gemeten met een NTC weerstand en een digitale temperatuursensor en hieruit krijgen we volgende resultaten te zien:

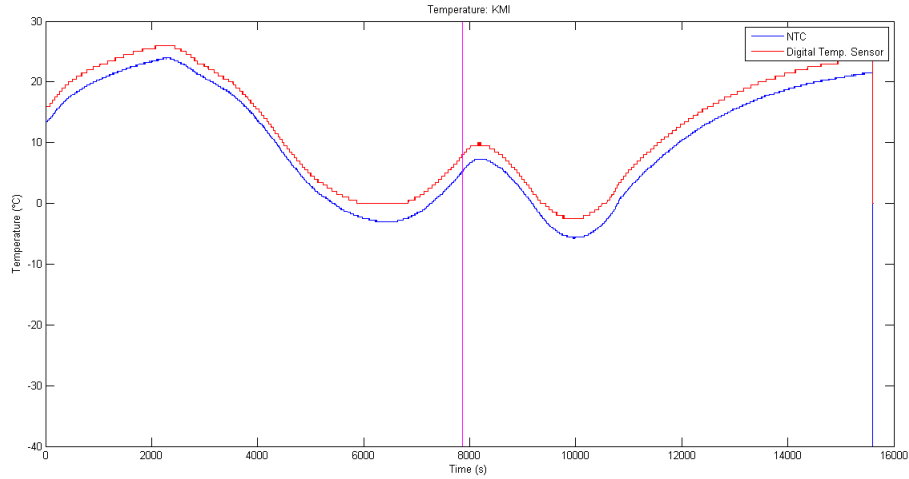


Figure 12: Temperatuursprofiel van KMI vlucht met maximum hoogte aangeduid (roze lijn)

Het absolute minimum gemeten door de NTC lag rond de  $-5^{\circ}\text{C}$ . Deze temperatuur ligt boven alle minimum temperaturen van de componenten die gebruikt werden op de detector. Merk op dat de temperatuursensors de buiten temperatuur niet meten, maar eerder de temperatuur op de PCB vlakbij componenten die warmte dissiperen.

### 3.1.2 Hoogte afhankelijkheid van de flux

Als de totale flux van elke Geiger Muller tube wordt uitgezet in functie van de tijd dan krijgen we volgende figuur te zien:

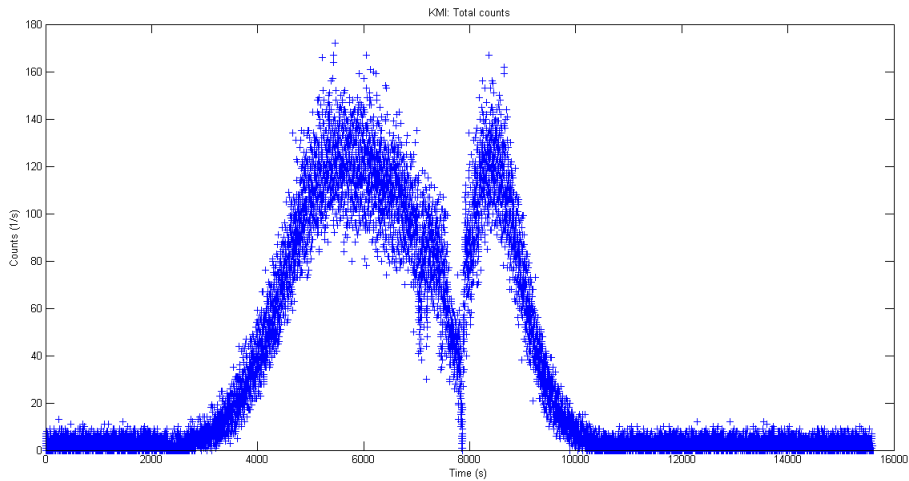


Figure 13: Totale flux in functie van de tijd voor de KMI vlucht

Merk op dat het verloop van de flux twee keer wordt doorlopen. De linkerhelft van de figuur heeft een breder verloop dan deze aan de rechterhelft. Dit is uiteraard het opstijgen en het dalen. De linkerhelft van de plot beeldt de flux uit tijdens het opstijgen, wat relatief traag gaat. De rechterhelft toont dezelfde flux maar tijdens het dalen, wat sneller gaat dan het opstijgen waardoor dit verloop smaller is.

We kiezen ervoor om enkel te werken met de stijgende helft van de vlucht waardoor de figuur er als volgt uitziet:

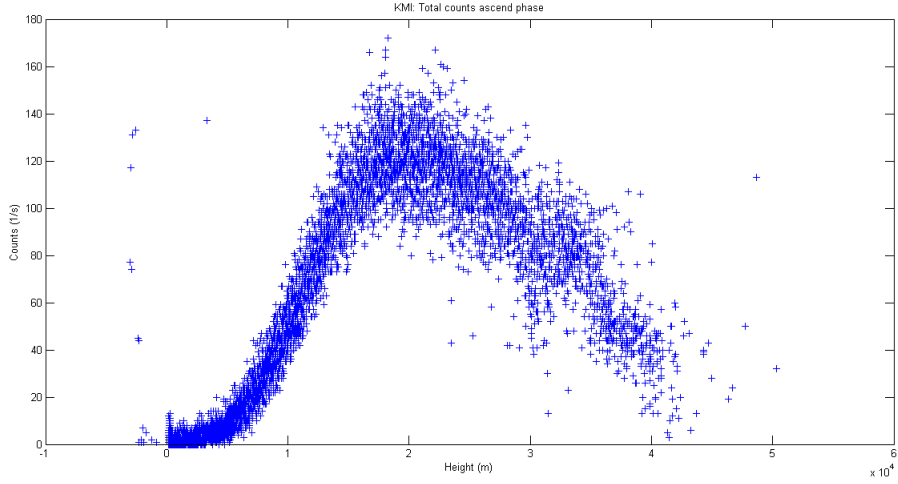


Figure 14: Flux van het stijgende deel van de vlucht

Zoals duidelijk te zien op deze figuur bevindt het maximum van de flux zich niet bij de maximum hoogte zoals uitgelegd in de inleiding.

Om op deze data een goede analyse te kunnen doen en deze te kunnen vergelijken met vluchten van andere jaren wordt er dezelfde uitmiddeling gehanteerd als andere jaren.

$$\langle x \rangle = \frac{\sum_{90\text{punten}} x_i}{90} \quad (3.1.2)$$

$$s_x = \frac{\sqrt{\sum_{90\text{punten}} x_i^2}}{90} \quad (3.1.3)$$

Deze afwijking komt omdat de flux benaderend Poisson verdeeld is.

Bij deze uitmiddeling wordt er telkens over 90 seconden uitgemiddeld. Als dit wordt gedaan voor het totale aantal counts van de enkele Geiger Muller tubes dan wordt het volgende resultaat bekomen:

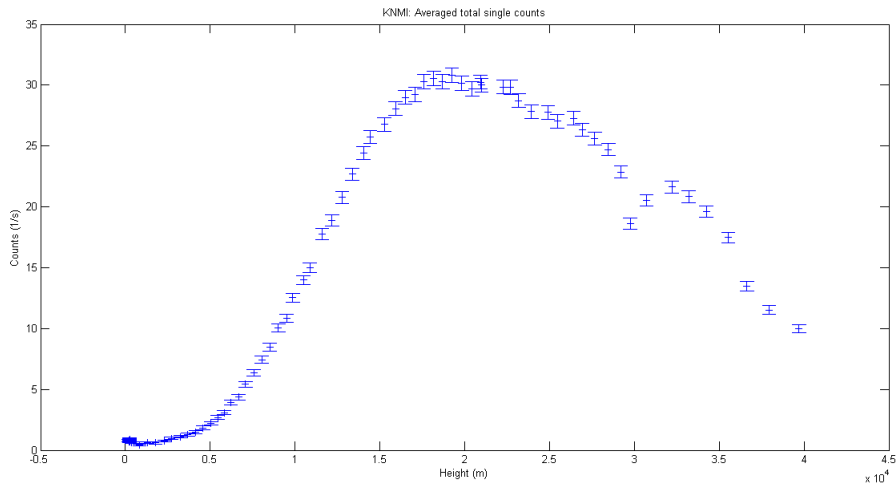


Figure 15: Uitgemiddelde counts van de enkele Geiger Muller tubes

Uit dit resultaat kan het eerder besproken Pfozter Maximum berekend worden. Dit maximum zullen we voor beide vluchten berekenen en dan vergelijken met de andere vluchten om te zien of we een verandering in activiteit van de zon kunnen waarnemen.

Om het Pfozter Maximum te bepalen moet er een functie gezocht worden die het maximum van het count verloop goed fit. Deze functie wordt gehaald bij de zwarte straler en ziet er als volgt uit:

$$F = \frac{a \left(\frac{h}{b}\right)^3}{\exp\left(\frac{h}{b}\right) - c} \quad (3.1.4)$$

Het bovenste deel van de uitgemiddelde counts in functie van de hoogte wordt gefit met deze functie. Er wordt een minimalisatie procedure gebruikt om de kleinste kwadraten som zo klein mogelijk te maken. Deze procedure zal dus volgende uitdrukking minimaliseren:

$$LSQ = \sum_i (\text{Count}_i - F(h_i, a, b, c))^2 \quad (3.1.5)$$

De minimalisatie procedure zal de fitparameters a, b en c zoeken waarbij deze som zo klein mogelijk is.

Het resultaat van deze minimalisatie is de volgende fit:

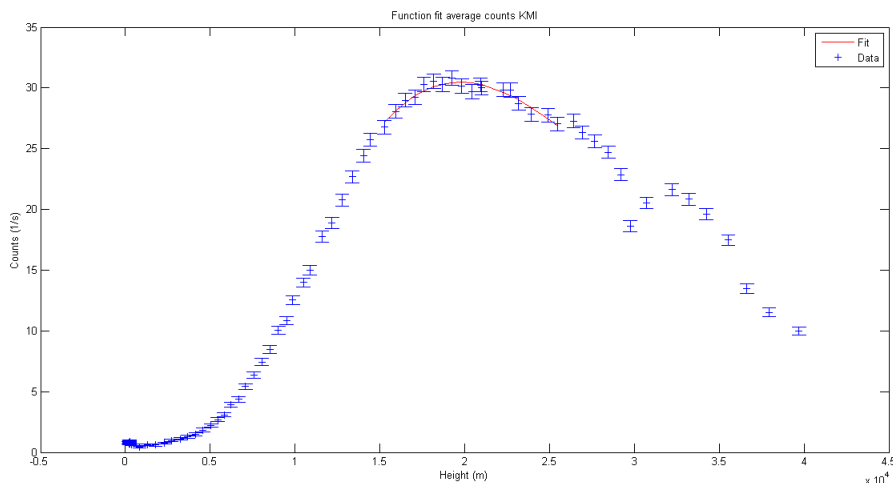


Figure 16: Functie gefit door het maximum van de gemiddelde counts van de KMI vlucht

Het maximum van de functie kunnen we vinden door de functie af te leiden en deze uitdrukking gelijk te stellen aan 0 en de vergelijking op te lossen naar de hoogte. Hierbij vinden we dat het maximum van de functie zich bevindt op  $(19.759 \pm 0.010)$  km. Als deze waarde wordt ingevuld in de fitfunctie krijgen we de maximale flux. De afwijking wordt berekend met volgende propagatieformule:



$$s_F = \sqrt{\left(\frac{\partial F}{\partial a}\right)^2 s_a^2 + \left(\frac{\partial F}{\partial b}\right)^2 s_b^2 + \left(\frac{\partial F}{\partial c}\right)^2 s_c^2 + \left(\frac{\partial F}{\partial h}\right)^2 s_h^2} \quad (3.1.6)$$

$$\frac{\partial F}{\partial a} = \frac{\left(\frac{h}{b}\right)^3}{\exp\left(\frac{h}{b}\right) - c} \quad (3.1.7)$$

$$\frac{\partial F}{\partial b} = \frac{ah^3 (3cb + \exp\left(\frac{h}{b}\right)(h - 3b))}{b^5 (c - \exp\left(\frac{h}{b}\right))^2} \quad (3.1.8)$$

$$\frac{\partial F}{\partial c} = \frac{a\left(\frac{h}{b}\right)^3}{(\exp\left(\frac{h}{b}\right) - c)^2} \quad (3.1.9)$$

$$\frac{\partial F}{\partial h} = \frac{ah^2 (\exp\left(\frac{h}{b}\right)(3b - h) - 3bc)}{b^4 (c - \exp\left(\frac{h}{b}\right))^2} \quad (3.1.10)$$

Uit deze formules kan het Pfozter maximum berekend worden welke voor de KMI vlucht  $(30.48 \pm 0.31) \frac{1}{s}$  is. Deze waarde zal vergeleken worden met de waarden van andere jaren. De hoogte waarop het Pfozter Maximum zich bevindt zou ongeveer hetzelfde moeten blijven maar de flux op dit maximum zou hetzelfde patroon moeten volgen als de zonnecyclus.

### 3.1.3 Hoekafhankelijkheid van de flux

Door de configuratie van de Geiger Muller tubes kunnen er verschillende invalshoeken worden bestudeerd. Deeltjes kunnen verticaal, schuin of horizontaal door de detector gaan. De verticale of horizontale situatie heeft dezelfde ruimtehoek en hebben evenveel mogelijkheden terwijl de schuine situatie een andere ruimtehoek heeft en meerdere mogelijkheden.

Deze ruimtehoeken werden in vorige hoofdstukken vermeld. Deze zijn belangrijk zodat we de fluxen van al de combinaties kunnen delen door de ruimtehoek zodat we de verschillende combinaties op gelijke voet kunnen behandelen en vergelijken.

Omdat de ruimtehoeken enkel zijn berekend voor dubbele coincidenties (dit wil zeggen coincidenties waar het deeltje door maximaal 2 Geiger Muller tubes is gegaan) zal ook enkel met deze resultaten gewerkt worden voor de hoekafhankelijkheid.

Allereerst wordt er op elke combinatie dezelfde uitmiddeling als bij de hoogte afhankelijkheid uitgevoerd waarna er werd gedeeld door de bijhorende ruimtehoek en het aantal combinaties van Geiger Muller tubes er per hoek mogelijk waren. Het resultaat is de volgende:

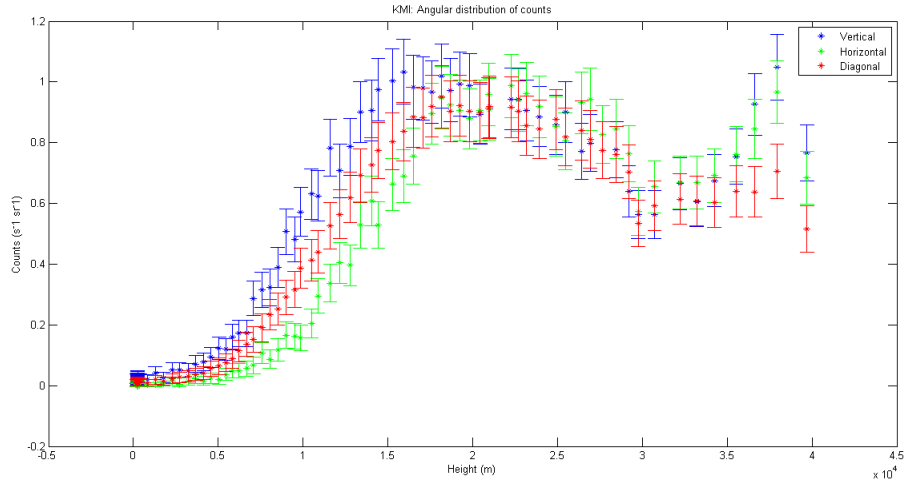


Figure 17: Hoekafhankelijkheid van de invallende deeltjes tijdens de KMI vlucht

Dit is het resultaat dat we verwachten. De verticale counts gebeuren initieel meer dan de schuine en deze gebeuren dan weer meer dan de horizontale. Vanaf ongeveer 20 km komen het aantal counts langs elke richting bij elkaar en worden ze nagenoeg gelijk. Dit is wat we verwachten als we kijken naar het dichtheidsverloop van de atmosfeer:

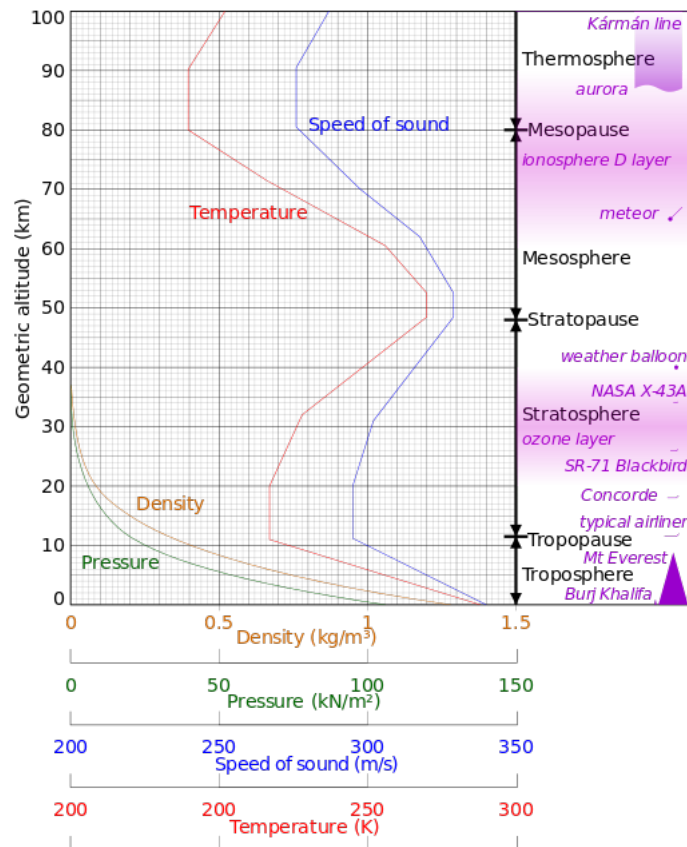


Figure 18: U.S. Standard Atmosphere model

We zien dat de dichtheid van de atmosfeer boven de 20 km praktisch gezien 0 is. Deeltjes die horizontaal invallen op de detector moeten een langere baan afleggen door de atmosfeer. Op hoogtes onder de 20

km zal dit ervoor zorgen dat ze meer kans hebben om tegen moleculen in de atmosfeer te botsen dan deeltjes die verticaal door de atmosfeer komen. Hetzelfde geldt voor deeltjes die schuin door de atmosfeer bewegen. Deze moeten een langere weg afleggen dan de verticale deeltjes maar een minder lange weg dan de horizontale.

Hoger dan 20 km zal de dichtheid van de atmosfeer geen rol meer spelen en zullen alle mogelijkheden ongeveer evenveel kans hebben om ergens anders tegen te botsen dan tegen onze Geiger Muller tubes. Wat wel moet worden opgemerkt is dat naar het maximum toe de schuine counts zich wat meer afscheiden van de horizontale en verticale counts. Door de foute vlaggen blijven de resultaten steeds zeer dicht bij elkaar.

Wat wel belangrijk is is dat op grote hoogtes de primaire deeltjes worden gedetecteerd in plaats van deeltjes afkomstig van een shower. Of dit de reden is waarom er blijkbaar iets minder deeltjes schuin invallen is niet geweten.

## 3.2 KNMI vlucht

Voor de KNMI vlucht moet er vermeld worden dat er een Geiger Muller tube defect was. Dit heeft als gevolg dat de hoekafhankelijkheids metingen niet goed konden worden uitgevoerd.

Ook moet er worden vermeld dat tijdens de vlucht de detector verschillende keren is uitgevallen wat ook te zien is in de data. Dit falen van de detector wordt in wat meer detail uitgelegd in een komend hoofdstuk.

Dit heeft als gevolg dat er gebieden zijn waar er spaties in de data zijn en waardoor sommige resultaten van deze vlucht niet goed kunnen worden vergeleken met andere vluchten.

### 3.2.1 Hoogte en Temperatuur profielen

Zoals bij de vorige vlucht werd het hoogte profiel berekend met de barometrische formule van het U.S. Standard Atmosphere Model.

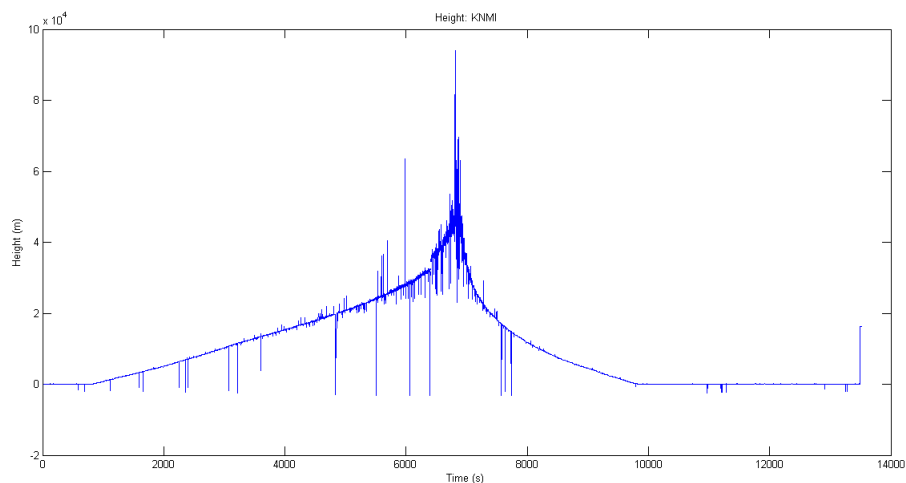


Figure 19: Hoogte profiel van KNMI vlucht

Als dit hoogte profiel vergeleken wordt met die van de KMI vlucht dan kan besloten worden dat de detector iets hoger ging bij de vlucht van het KNMI. Het maximum ligt rond de 50 km.

Het temperatuursprofiel zag er als volgt uit:

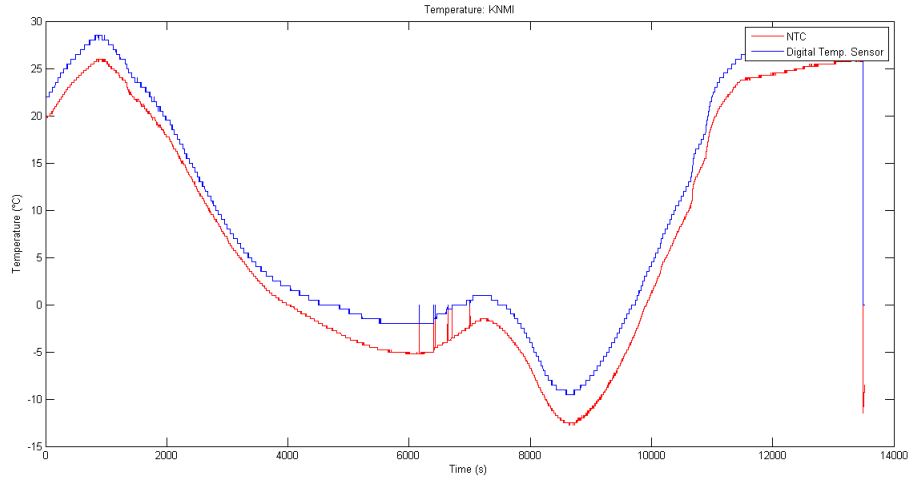


Figure 20: Temperatuursprofiel van KNMI vlucht

Hierop is te zien dat het absolute minimum wat lager ligt dan bij de KMI vlucht. Dit omdat de omhulzing van de detector anders was waardoor de isolatie van de temperaturen buitenaf anders was.

Bij de KMI vlucht zat de detector in een grote piepschuimen doos met dikke wanden terwijl bij de KNMI vlucht de detector in een kleine kartonnen doos zat met dunne wanden waarbinnen een aantal dunne piepschuimen plaatjes waren tegen geplakt.

Het absolute minimum lag rond de  $-13^{\circ}\text{C}$  waardoor we in de gevarenzone van een van onze componenten kwamen. De DC/DC converter die verantwoordelijk is om de 5V spanning om te zetten naar de 500V die onze Geiger Muller tubes nodig hebben heeft een minimum temperatuur van  $-10^{\circ}\text{C}$ . Doordat het absolute minimum niet zoveel lager lag dan het minimum van deze component bleef het werken zoals verwacht. Door een lage temperatuurstest die in een volgend hoofdstuk wordt besproken kon geconcludeerd worden dat deze DC/DC converter blijft werken tot zeker  $-60^{\circ}\text{C}$ .

### 3.2.2 Hoogte afhankelijkheid van de flux

De totale counts in functie van de hoogte voor de vlucht van het KNMI zien er als volgt uit:

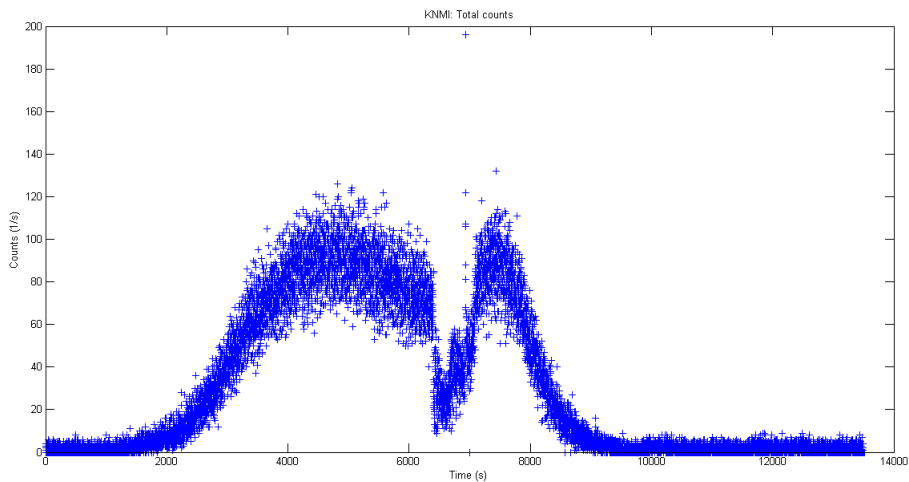


Figure 21: Totale flux in functie van de tijd voor de KNMI vlucht

Dit wordt weergegeven om twee zaken te visualiseren die in het begin van dit hoofdstuk werden aangehaald. Doordat er een Geiger Muller tube defect was zien we bij het maximum ongeveer 30-40 minder counts dan bij de vlucht van het KMI. Dit hadden we verwachten omdat we weten dat elke Geiger Muller tube dit bedrag van counts bijdraagt aan het totaal.

Een ander probleem was het falen van de detector gedurende een deel van de vlucht. Dit is duidelijk zichtbaar rond 6000s. Er is een zeer zichtbare spleet waar er geen of weinig data werd verzameld. Dit falen gebeurde nog voor de detector op zijn maximum hoogte was. Dit wordt in meer detail uitgelegd in een volgend hoofdstuk.

Ook hier wordt enkel de data van de opstijging gebruikt zoals bij al de andere vluchten.

Hetzelfde uitmiddellings proces wordt gedaan voor de vlucht van het KNMI als bij de vlucht van het KMI en vorige vluchten. Dit toegepast op de totale hoeveelheid counts van de enkele Geiger Muller tubes geeft ons volgend resultaat:

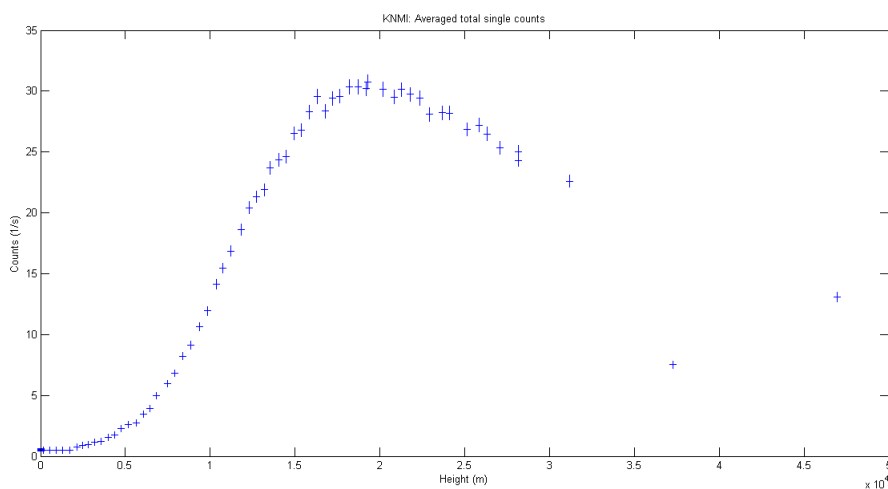


Figure 22: Uitgemiddelde counts van de enkele Geiger Muller tubes

Hierop is duidelijk de spleet in de data te zien nog voor de maximum hoogte werd bereikt. De hoeveelheid counts is gelijkend aan de vlucht van het KMI. Ook hier wordt het Pfozzer maximum uit berekend.

Zoals bij de vlucht van het KMI wordt de zwarte straler functie gefit door het maximum van het verloop. Dit heeft het volgende resultaat:

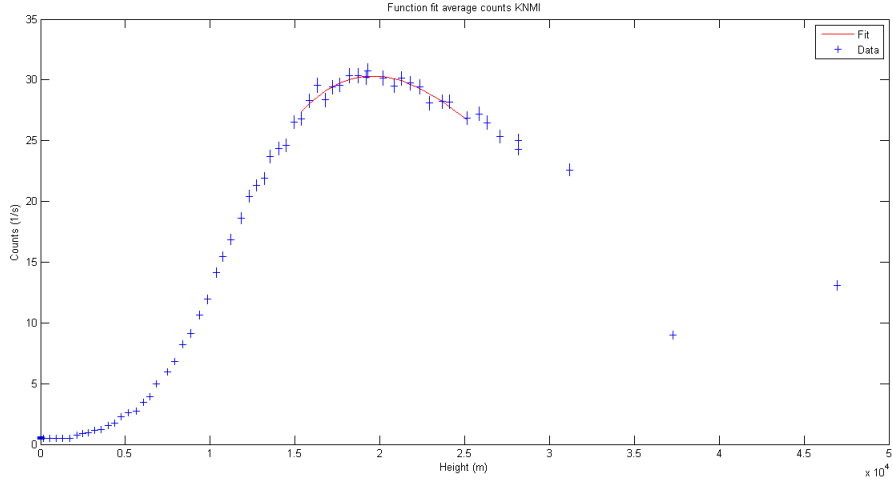


Figure 23: Functie gefit door het maximum van de gemiddelde counts van de KNMI vlucht

Bij deze vlucht lag het Pfozter maximum op  $(19.560 \pm 0.015)$  km hoogte en bedroeg de flux  $(30.33 \pm 0.31) \frac{1}{s}$ .

### 3.2.3 Hoekafhankelijkheid van de flux

Zoals eerder vermeld was er een Geiger Muller tube defect. Dit verkleint de combinaties voor schuine coincidenties van 4 naar 2 en de combinaties voor verticale coincidenties van 1 naar 0. Dit heeft als gevolg dat de hoekafhankelijkheid van de flux er als volgt uit ziet:

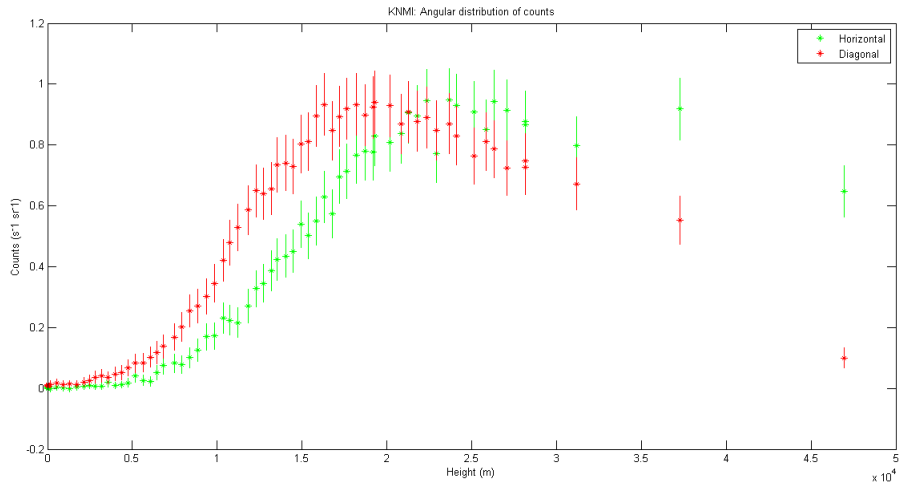


Figure 24: Hoekafhankelijkheid van de invallende deeltjes tijdens de KNMI vlucht

Wat ook hier op te merken valt is dat het aantal counts onder de 20 km hetzelfde patroon volgt als deze bij de vlucht van het KMI, wat eerder werd verklaard.

Over de counts tussen 30 km en de maximum hoogte zou ik liever geen uitspraak willen doen omdat dit het gebied was waar de detector faalde en waar er in het volgende hoofdstuk dieper op wordt ingegaan.

### 3.2.4 Detector falen tijdens vlucht

Na het uitlezen van de detector na de recuperatie ervan werden er meer dan 50 lege bestanden aangetroffen en nog extra bestanden waar er maar een aantal datapunten werden in opgeslagen. Sommige datapunten waren zelfs niet volledig wat betekent dat er iets is gebeurd tijdens het wegschrijven van de data.

Dit gedrag werd niet geobserveerd tijdens de vlucht van het KMI. Het grootste verschil tussen de vlucht van het KMI en het KNMI was dat de detector in een andere behuizing zat. Tijdens de vlucht van het KMI zat de detector in een grote dikke piepschuimen doos wat voor goede isolatie zorgde. Tijdens de vlucht van het KNMI zat de detector in een kleine dunne kartonnen doos waar aan de zijden dunne piepschuimen wanden aan werden gekleefd.

Als we de temperatuursprofielen van beide vluchten vergelijken krijgen we het volgende te zien:

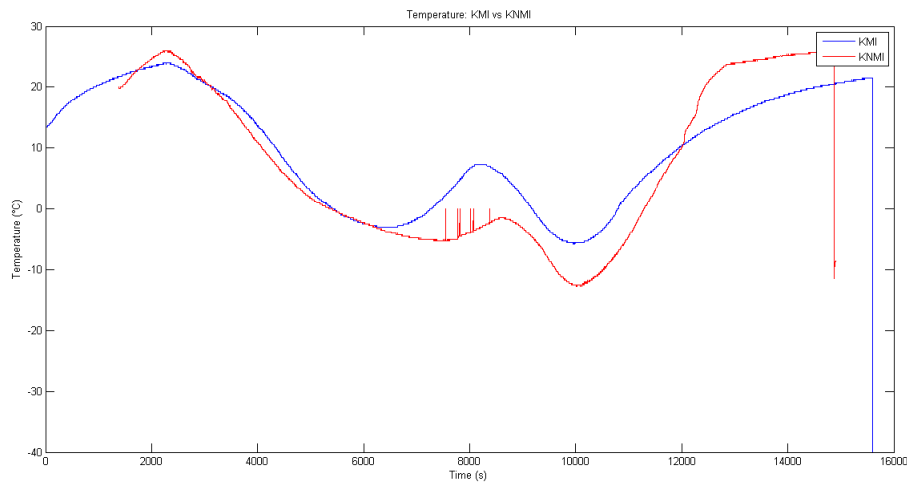


Figure 25: Temperatuur van de NTC tijdens KNMI en KMI vlucht

Hierbij werd de temperatuur van het KNMI verschoven zodat beide curven konden worden vergeleken.

De temperatuur in de doos tijdens de vlucht van het KNMI was iets lager dan bij de vlucht van het KMI.

Met deze informatie in het achterhoofd kunnen we een beter beeld krijgen over wat er zou zijn kunnen gebeurd.

Er zijn 2 momenten in de software wanneer de detector een nieuw databestand aanmaakt. Dit is bij het opstarten en wanneer het 500 datapunten heeft opgeslagen in een bestaand bestand. Aangezien er zeer veel lege bestanden waren aangemaakt kunnen we besluiten dat deze niet zijn aangemaakt doordat er in de voorgaande bestanden 500 datapunten werden opgeslagen. Hierdoor kunnen we besluiten dat de bestanden zijn aangemaakt doordat de detector zeer veel keer in de opstart sectie van de software kwam.

Dit gedrag wordt ook geobserveerd wanneer de detector voor een langdurige tijd metingen doet op de grond. Specifieker nog, het gebeurt tegen het einde van de levensduur van de batterijen. Wanneer de batterijen bijna op zijn dan kunnen ze soms nog net genoeg energie leveren om de microcontroller op te starten, maar kunnen deze niet aanhouden. Hierdoor zal de microcontroller voortdurend aan- en afschakelen totdat de batterijen zelfs niet meer genoeg energie kunnen leveren om de microcontroller op te starten.

Na het voortdurend aan- en afschakelen van de microcontroller tijdens de vlucht van het KNMI gingen de metingen gewoon door, waardoor we kunnen besluiten dat de batterijen niet volledig op zijn geraakt in het midden van de vlucht.

Een belangrijke eigenschap van batterijen is dat batterijen zeer temperatuursafhankelijk zijn [8] [9] [10]. De batterijen die dagdagelijks worden gebruikt zijn meestal getest op kamertemperatuur en zullen wel degelijk de spanning en energie kunnen leveren die op de verpakking staat, maar dit wordt een heel ander feit wanneer deze batterijen worden gebruikt in extremere situaties zoals de zeer koude temperaturen die tijdens de vlucht worden gemeten.

Een algemeen temperatuurs afhankelijk verband voor batterijen ziet er als volgt uit:

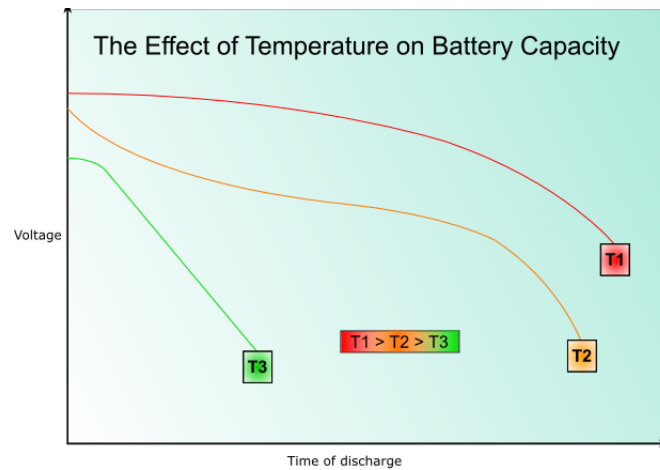


Figure 26: Algemeen temperatuurs afhankelijkheid van batterij capaciteit

Hierop is te zien dat de spanning en de tijd tot volledige ontlading zeer sterk veranderd met temperatuur. Deze foto is om een algemeen idee te geven wat voor de meeste batterijen geldt.

De batterijen dat we gebruikten tijdens de vlucht waren Ultra245 Lithium batterijen van Duracell. Dit zijn batterijen die grote stromen in zeer korte pulsen kunnen geven wat zeer handig is voor bijvoorbeeld camera flash.

In de datasheet [7] hiervan vinden we volgende temperatuurs plot:

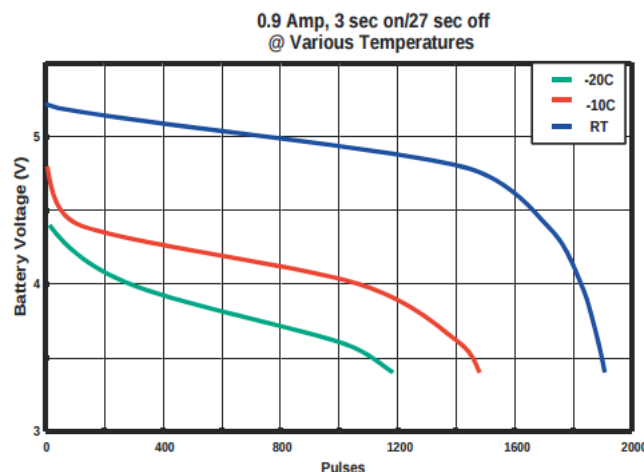


Figure 27: Temperatuurs afhankelijkheid van de gebruikte batterijen

Hierop is te zien dat ook het gedrag van deze batterijen zeer drastisch veranderd bij lage temperaturen. Wat moet worden opgemerkt is dat deze plot op de x-as weergeeft hoeveel pulsen er kunnen gegeven



worden van 0.9A in 3 seconden tijd waarbij de batterij daarna 27 seconden af staat. Dit is niet wat er in onze detector gebeurt, maar het verloop van de spanning van de batterij is wel representatief voor ons geval.

Deze drastische verandering in capaciteit van de batterijen bij lagere temperaturen komt doordat de interne weerstand stijgt. Het aantal reacties per seconde in batterijen is uiteraard temperatuurs afhankelijk doordat de moleculen minder energie hebben en hierdoor minder reacties aangaan.

Uit deze informatie heb ik een sterk vermoeden dat de batterij tijdens de KNMI vlucht een periode had dat het kouder werd dan tijdens de KMI vlucht. Wat als gevolg had dat de batterij niet genoeg energie kon leveren in die periode.

In de weggeschreven datapunten staat er een tijd. Deze tijd reset elke keer als de microcontroller heropstart. Hierdoor kunnen we zien bij welke temperaturen de microcontroller zich telkens heropstarte.

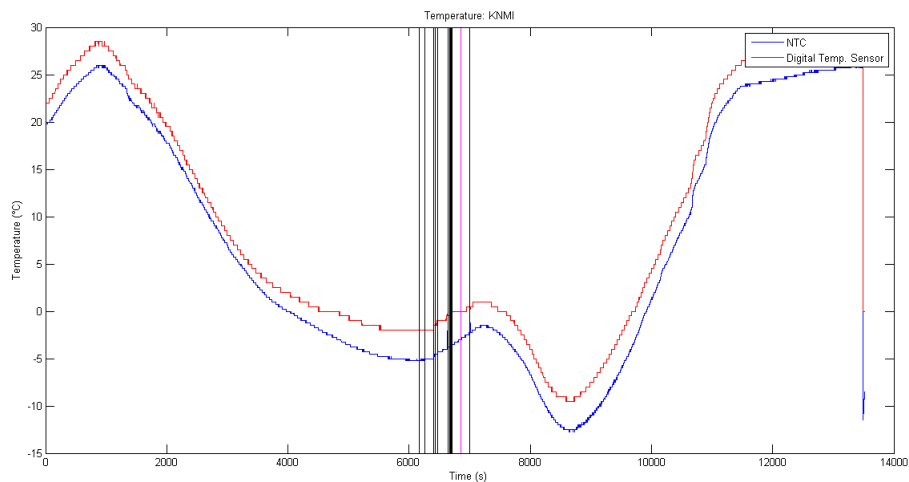


Figure 28: Temperatuur van NTC en digitale sensor tijdens KNMI vlucht met resets aangeduid (zwarte lijnen)

Op deze figuur duiden de zwarte lijnen de punten aan waar er een reset plaatsvond. De magenta lijn is de plaats waar de detector op zijn maximum hoogte was. Let op dat op deze figuur de 2 verschillende temperatuur sensors werden geplot. De NTC en de digitale temperatuursensor.

Deze plot is zeer eigenaardig en lijkt het eerder vermoeden teniet te doen. Zoals te zien gebeuren de resets inderdaad in een minimum van het temperatuursprofiel, maar het gebeurt in een lokaal minimum en niet in het absoluut minimum. Uit vorige bedenkingen zouden we de resets ten minste verwachten in het absoluut minimum.

De temperaturen waarbij de resets zich voordeden waren vrijwel allemaal lagere temperaturen dan bij de KMI vlucht, maar dit geeft geen verklaring waarom het niet gebeurde tijdens het absoluut minimum.

Op 29 Juli 2015 werd er een afkoelingstest uitgevoerd met de nieuwe PCB die gemaakt werd voor het REXUS/BEXUS programma. Op deze PCB staan voornamelijk dezelfde componenten als deze op de HACORD-III detector die faalde tijdens de KNMI vlucht.

De afkoelingstest gebeurde aan de hand van een vriezer van het biodepartement. Deze vriezer gaat tot -80 °C. Hiermee wilden we hoofdzakelijk testen tot welke temperatuur de hoge spannings DC/DC converter kan gaan omdat deze maar gerate is tot -10 °C terwijl de andere componenten tot -40 °C of -55 °C gerate zijn. Deze test werd uitgevoerd voor het REXUS/BEXUS programma, maar kan ons meer

info geven over de mogelijke oorzaken van falen tijdens de KNMI vlucht.

Op de geteste PCB staan al dezelfde componenten als op de HACORD-III dector. Als het temperatuursprofiel van de test wordt geplot wordt volgende grafiek bekomen:

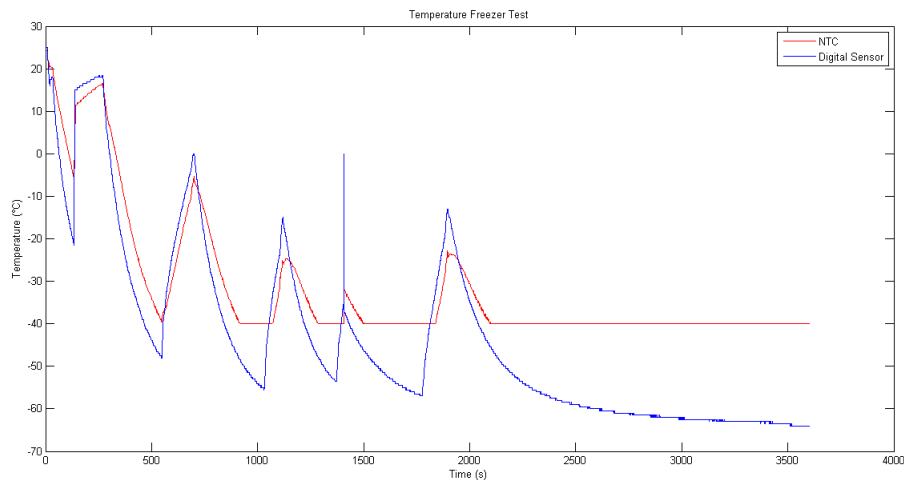


Figure 29: Temperatuurs profiel van de biovriezer test

Allereerst wordt er opgemerkt dat de NTC maar tot  $-40^{\circ}\text{C}$  gaat, waardoor deze op dit punt een constante temperatuur weergeeft. De digitale temperatuur sensor kan volgens de datasheet tot  $-55^{\circ}\text{C}$  gaan, maar kon tijdens de test tot lagere temperaturen.

Wat ook moet worden vermeld zijn de plotse pieken op de temperatuursplot. Dit zijn de momenten waarop de vriezer open werd gedaan om de detector te inspecteren op falen. Soms werd de detector geheel uit de vriezer gehaald waarna deze met een thermografische camera werd geïnspecteerd. Hierdoor stijgt de temperatuur snel. Andere keren werd de vriezer enkel open gedaan waarbij de detector enkel visueel werd geïnspecteerd, dit zijn de kleinere pieken op de temperatuursplot. De thermografische camera kon niet lager dan  $-25^{\circ}\text{C}$  waardoor de detector niet meer uit de vriezer gehaald moest worden.

De minimale temperatuur was  $-64^{\circ}\text{C}$  waarbij de gehele detector stopte met werken. Deze temperatuur is vele malen kleiner als de temperaturen van zowel de KMI als de KNMI vlucht. Tijdens deze test werd er gebruik gemaakt van een andere batterij dan tijdens de KMI en KNMI vluchten. Deze batterij werd ook nog eens apart geïsoleerd zodat de batterij wat warmer bleef. Ook had de batterij een hogere beginspanning waardoor verlaging van de spanning door afkoeling minder problemen vormt dan bij de batterijen die een begin spanning hebben die al dicht bij de minimale operationele spanning zitten.

Doordat de geteste PCB al de componenten bevat als de HACORD-III detector en deze nog steeds werkt op deze lage temperaturen kan er vrijwel met zekerheid worden gezegd dat de HACORD-III niet faalde door componenten die de temperaturen niet aankonden. Omdat het enige verschil tussen beide detectoren de gebruikte batterij was biedt deze test een vrij goede indicatie dat het probleem hierbij lag tijdens de vlucht.

Een volgende test zou dezelfde afkoelingstest zijn met dezelfde batterijen als tijdens de KNMI vlucht. Door een gebrek aan tijd werd deze test niet meer uitgevoerd. Dit zal ook geen probleem zijn tijdens de vlucht voor het REXUS/BEXUS project van de ESA omdat we daar niet onze batterijen zullen gebruiken. De batterijen die worden gebruikt tijdens die vlucht zullen zeer goed geteste batterijen zijn die altijd al werden gebruikt voor zulke lage temperatuur vluchten.

### 3.3 Vergelijking resultaten met andere vluchten

In volgende tabel worden de berekende Pfozter maxima van de voorbije vluchten naast elkaar gezet om ze te kunnen vergelijken.

Jaar	h (km)	$s_h$ (km)	F ( $\frac{1}{s}$ )	$s_F$ ( $\frac{1}{s}$ )
2012	18.920	0.011	29.42	0.36
2013	19.111	0.010	29.81	0.29
2014 (ASGARD)	18.697	0.010	29.29	0.34
2014 (KMI)	19.759	0.010	30.48	0.31
2015 (KNMI)	19.560	0.015	30.33	0.31

### 3.4 Besluit

Wat we verwachten is dat het Pfozter maximum zal stijgen aangezien we naar een minimum van de zonnecyclus gaan. Dit betekent minder zonnevlekken waardoor er dus minder magnetische velden zullen zijn die de kosmische stralen zullen wegbuigen van de aarde. De zonnecyclus voorspelling is te zien in volgende figuur:

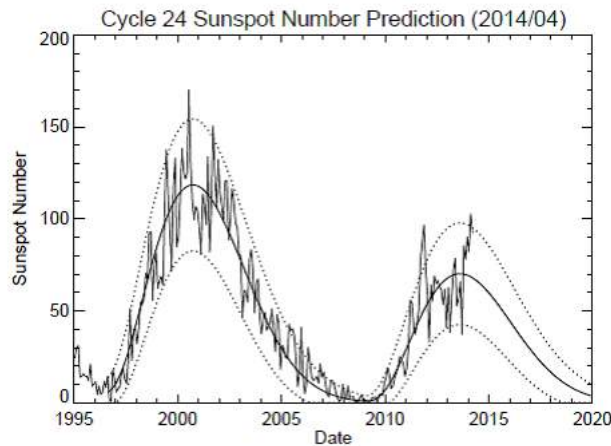


Figure 30: Voorspelling van aantal zonnevlekken

Doordat de stijging van de flux zo klein is, kunnen we niet echt vaststellen of er daadwerkelijk een verandering van de flux intensiteit is geweest over de jaren. Hierdoor kunnen we ook niet zeggen of we de verandering van de zonneactiviteit zien in onze resultaten.

## 4 ESA REXUS/BEXUS Programma

### 4.1 Inleiding

#### 4.1.1 Organisatie

Het REXUS/BEXUS (Rocket EXperiments for University Students/ Balloon EXperiments for University Students) programma geeft de mogelijkheid aan studenten van universiteiten in heel Europa om een wetenschappelijk of technologisch experiment uit te voeren op een onderzoeksraket of ballon. Elk jaar worden er 2 raketten en 2 ballonnen gelanceerd die een maximum van 20 experimenten kunnen dragen.

Het REXUS/BEXUS programma werd gerealiseerd onder een bilateraal akkoord tussen de Duitse Luchtvaart Centrum (DLR) en de Zweedse Nationale Ruimte Organisatie (SNSB) [11]. Het zweedse deel van de payload wordt ter beschikking gesteld aan studenten van andere Europese landen door een samenwerking

met de Europese Ruimtevaart Organisatie (ESA).

EuroLaunch, een samenwerking tussen het Esrange Ruimte Centrum van SSC en de Mobiele Raketten Basis (MORABA) van DLR, is verantwoordelijk voor de organisatie en operatie van de lanceer voertuigen tijdens de campagne. Experts van DLR, SSC, ZARM en ESA bieden technische ondersteuning aan de studenten teams doorheen het project.

REXUS en BEXUS worden gelanceerd van SSC, Esrange Ruimte Centrum in het noorden van Zweden (Kiruna).

#### **4.1.2 Waarom interessant voor ons ?**

Doordat de vluchten georganiseerd door het KMI enkel voor studenten zijn van de middelbare school en omdat we geen school vonden in België die dit project mee wilden doen hadden we geen mogelijkheid meer voor een ballonvlucht.

Dit programma gaf ons de mogelijkheid om toch nog een ballonvlucht te kunnen hebben met onze detector. Deze vlucht zal langer duren dan vluchten met een weerballon waardoor er meer data zal zijn en dus de statistiek hierop verbeterd zal worden.

Wat ook interessant is is dat de vlucht in het noorden van Zweden plaatsvindt, boven de poolcirkel. Dit zorgt ervoor dat we op een heel andere breedtegraad gaan meten wat een effect gaat hebben op de flux. We verwachten hier een hogere flux dan op de breedtegraad van België doordat de magnetische veld dichtheid hoger is en deze de kosmische stralen harder zal afbuigen.

#### **4.1.3 Selectie proces en verder verloop van het programma**

Het experiment werd ingestuurd en uit alle inzendingen werden er 8 experimenten van Europa gekozen, waaronder dit experiment. Hierna moesten we naar ESTEC in Nederland wat de grootste basis is van de ESA in Europa. Op ESTEC moesten we een presentatie geven over het experiment voor de andere studenten en een panel van 15 experts verspreid over de organisaties die vooraf werden genoemd.

Na de presentatie kwamen er zeer kritische vragen van het panel omtrent de haalbaarheid, duur, doel en veiligheid van ons experiment. Uit de 8 teams die werden gekozen mochten er uiteindelijk 5 meedoen aan het programma.

Sinds December werd er non stop gewerkt aan dit project omdat er om de 2 maanden een document moest worden ingediend waarin elk detail van het experiment werd besproken net zoals het in de ruimtevaart aan toe gaat. Er waren geregelde samenkomsten waarbij we weer presentaties moesten geven om de vooruitgang van het experiment te tonen en voor eventueel hulp te krijgen bij problemen.

Omdat ik verantwoordelijk was voor de elektronica en software moest ik een cursus solderen volgens de regels van de ruimtevaart volgen van de ESA waarvoor ik ook een certificaat krijg op het einde van het programma.

De vlucht staat gepland voor de eerste 10 dagen van Oktober 2015 waarbij het hele team (Ik, Jesse Van Muiden, Jeroen Van Houtte en Jeroen Peeters) naar Zweden moeten om het experiment uit te voeren. Hierna zal er nog een bijeenkomst zijn in Januari op ESTEC waarbij we onze resultaten moeten presenteren en eventueel een fouten analyse moest er iets zijn fout gegaan.

Aangezien dit een vrij groot project was voor een student die nog steeds lessen moest volgen en examens moest meedoen heb ik hier veel van mijn tijd ingestoken in plaats van de data analyse wat oorspronkelijk het idee was van deze bachelor thesis.

Wat volgt zijn enkel de zaken die ik zelf heb gedaan voor dit programma, de contributies van mijn teamgenoten worden hier niet besproken.

## 4.2 Software

### 4.2.1 E-Link: Draadloze verbinding met experiment via ethernet

**E-Link systeem van SSC** Op de gondel zal er een speciaal ontwikkeld systeem staan ontwikkeld door SSC genaamd de E-Link. Dit zorgt ervoor dat al de experimenten verbonden kunnen worden met een lokaal netwerk zodat een grondstation door ethernet protocollen te gebruiken contact kan maken met het experiment.

Elk experiment dat deze verbinding wilt benutten zal een RJ45 kabel moeten verbinden tussen zichzelf en de E-Link. De E-Link staat in contact met een station op de grond doormiddel van een radio-ontvanger en -zender. Dit station stuurt al de ontvangen data door op het lokale netwerk naar de experimentators zodat zij dit met hun software kunnen uitlezen. Ook kunnen de experimentators commandos versturen langs hetzelfde netwerk. Dit is dus een bidirectionele verbinding. De grondstation software dat voor dit experiment werd ontwikkeld zal in een later hoofdstuk worden uitgelegd.

**UDP of TCP?** Wanneer er data wordt verstuurd langs het internet (of in dit geval ethernet) dan wordt er voornamelijk gebruik gemaakt van 1 van de volgende 2 protocollen: User Datagram Protocol (UDP) en Transmission Control Protocol (TCP). Deze protocollen werken bovenop de Internetprotocol (IP) laag. Om deze thesis niet te technisch te maken zullen deze protocollen vrij kort en oppervlakkig worden uitgelegd omdat dit voldoende is om de keuze van protocol te rechtvaardigen.

TCP wordt meestal tesamen met IP genoemd in de naam TCP/IP. Dit duid al aan dat TCP de meest gebruikte onder de protocollen is en zo goed als onafscheidelijk is van het IP. Voor een pakketje data te versturen gebruikmakende van TCP zullen volgende stappen worden genomen:

1. De data verstuurer stuurt een pakketje naar de eindbestemming waarin je verzoekt om een verbinding te openen (SYN packet)
2. De eindbestemming stuurt op ontvangen twee pakketjes terug: Een erkenning van ontvangen (ACK packet) en ook een pakket waarin het verzoekt om een verbinding te openen met de data verstuurer (SYN packet)
3. De data verstuurer stuurt op ontvangen van deze pakketjes een erkenning van ontvangen naar de eindbestemming (ACK packet)
4. De data verstuurer stuurt nu de echte data pakketjes naar de eindbestemming waarbij telkens de checksum wordt nagekeken om ervoor te zorgen dat de eindbestemming elk pakketje heeft aangekregen, als er een pakketje ontbreekt zal de eindbestemming dit pakketje opvragen van de data verstuurer
5. Als alle pakketjes zijn verstuurd en ontvangen zal de data verstuurer een einde connectie pakket sturen (FIN packet)
6. De eindbestemming zal op ontvangen van het einde connectie pakketje weer een erkenning van ontvangen sturen (ACK packet)
7. De eindbestemming stuurt nu een einde connectie pakket naar de data verstuurer (FIN packet)
8. De data verstuurer stuurt op ontvangen van dit einde connectie pakket een erkenning van ontvangen terug (ACK packet)

Zoals te zien worden er vrij veel stappen doorlopen vooraleer de data kan worden verstuurd en tijdens het versturen van pakketten. Dit heeft een aantal voordelen waardoor duidelijk is waarom dit een zeer veel gebruikt protocol is wanneer de connectie correct geopend kan worden:

1. **elk** pakketje dat de data verstuurer stuurt zal aankomen bij de eindbestemming
2. **elk** pakketje zal in de juiste volgorde aankomen bij de eindbestemming

3. **elk** pakketje zal correct aankomen bij de eindbestemming aangezien er integriteits checks worden gedaan bij het ontvangen van pakketjes

Deze voordelen zijn essentieel voor het correct werken van het internet. Al deze stappen kunnen worden uitgevoerd doordat elk pakketje dat verzonden wordt gebruikmakende van TCP bevestigd wordt met een TCP header. Ik zal deze header niet in detail bespreken maar in essentie is dit een extra 160 bits die alle relevante informatie bevat om het pakketje op correcte wijze en volgorde op de eindbestemming te krijgen. Dit gecompliceerde proces komt uiteraard met wat nadelen (meestal zijn deze nadelen niet nadelig genoeg om een ander protocol te gaan gebruiken):

1. De header op elk pakketje is minstens 160 bits lang en kan groter worden wanneer er extra informatie nodig is
2. Het aantal stappen om pakketjes te kunnen versturen
3. Doordat TCP garandeert dat **elk** pakketje volledig en in de juiste volgorde aankomt kan er een wachtrij gevormd worden in de software met pakketjes die nog moeten worden verstuurd wanneer er een probleem is met een van de verzonden pakketjes en deze opnieuw moet worden gestuurd

TCP wordt onder andere gebruikt voor email, het world wide web en om bestanden te versturen.

Het andere protocol dat we kort bekijken is UDP. De stappen die moeten worden gevolgd voor een pakketje te versturen gebruikmakende van UDP zijn:

1. De data verstuurder stuurt de data pakketten met het juiste adres (IP en poort) op het netwerk en hoopt dat het aankomt

Zoals te zien is UDP een zeer minimalistisch systeem vergeleken met TCP. Waar het pakketje naartoe moet zit in de header dat bij elk pakketje wordt bijgestoken zoals bij TCP. Deze header is tussen de 32 en 64 bits lang. UDP heeft de volgende voordelen:

1. Data pakketten kunnen zeer snel verzonden worden
2. De header is klein

De lijst met nadelen is iets langer:

1. Er is geen enkele garantie over het aankomen van de pakketjes
2. Er is geen enkele garantie over het volledig aankomen van de pakketjes
3. Er is geen enkele garantie over het aankomen in juiste volgorde van de pakketjes

Deze nadelen lijken op het eerste zicht genoeg om UDP te schrappen en altijd TCP te gebruiken, maar het eerste voordeel is wel een zeer grote. Doordat er geen erkenning van ontvangst pakketjes worden rondgestuurd of integriteits checks van de pakketten worden gedaan (dit is wel mogelijk om te implementeren maar dat moet de gebruiker zelf doen, UDP biedt wel een mogelijkheid om checksums mee in de header te sturen, maar de verwerking hiervan moet door de gebruiker gedaan worden) zal er zich nooit een wachtrij vormen in de software zoals bij TCP en zal dit als gevolg hebben dat pakketjes ontzettend snel verstuurd kunnen worden en de software niet lang op dit deel van het programma zal zitten.

UDP wordt onder andere gebruikt voor doeleinden die zeer snel moeten werken bijvoorbeeld telefonie (Skype) of videostreaming. Als er bij deze zaken eens een pakketje niet aankomt is dit niet zo erg, maar als er grote onderbrekingen zijn doordat je protocol **elk** pakketje in de juiste volgorde wilt versturen dan zal dit zeer irritant worden voor de gebruiker.

**UDP als protocol voor het experiment.** Wat er moet worden opgemerkt is dat het versturen van de data langs het E-Link systeem een backup is op de data die lokaal zal worden weggeschreven naar een SD kaart. In het geval dat het experiment niet te recupereren valt of dat de SD kaart kapot is gegaan tijdens de vlucht zal de data worden gebruikt die langs het E-Link systeem werd verstuurd. Het is dus cruciaal dat het gebruikte protocol zo weinig mogelijk de detector verstoort en dat het dus zo snel mogelijk werkt. Het maakt op zich niet uit dat er eens een datapakketje niet aankomt omdat er 1 pakketje per seconde wordt verstuurd door de detector.

Met deze zaken in gedachte werd er gekozen om gebruik te maken van UDP. Er zullen zowieso connectie problemen ontstaan omdat dit een draadloos systeem is. Hierdoor is het veel te gevaarlijk om het risico te lopen dat de detector vast blijft zitten op de TCP socket wanneer het data wilt versturen terwijl de verbinding is weggefallen (wat het niet oneindig lang zal proberen te doen, maar dit is nog steeds een te groot risico).

Bij een stabiele niet-draadloze verbinding (RJ45 kabel) werd UDP getest met pakketjes van 100, 200 en 300 bytes. Er werd een pakketje verstuurd om de seconde en er werd een totaal aantal van 15621 pakketjes verstuurd, dit is gelijkend aan de KMI vlucht van de detector. Er werd geen enkel verlies van pakketjes geobserveerd. Dit is doordat de grootte van de pakketjes en de verstuursfrequentie relatief klein zijn in vergelijking met ander genoemde toepassingen.

Hierdoor kunnen we vaststellen dat bij een stabiele connectie er geen verlies zou moeten zijn aan data, maar doordat ESA een systeem gebruikt dat ze zelf hebben ontwikkeld kan dit niet op voorhand worden getest.

#### 4.2.2 Druksensor backup systeem

**Initieel doel als start systeem.** Tijdens onze eerste presentatie voor het expertenpanel van o.a. ESA, DLR en SSC werd er ons gevraagd hoe het experiment zou starten met meten. Op dat moment startte het experiment door gebruik te maken van een schakelaar die moest worden omgeschakeld. Dit was niet goed genoeg voor het panel en er werd gevraagd dat het systeem moet starten op basis van een commando gestuurd langs het E-Link systeem en dat er hier ook een backup systeem voor moest worden verzonden voor het geval de E-Link connectie problemen heeft.

Het backup systeem dat we bedachten werkt op basis van de gemeten druk in drie druksensoren die een besluit maken op basis van een majority vote systeem.

Het concept werkt als volgt: Elke sensor meet voor een periode van 10s de druk en middelt hier na 10s over uit (dit leek een zeer grote fout te zijn wat te zien zal zijn wanneer de testvlucht wordt besproken). Wanneer het systeem geen startsignaal heeft ontvangen langs het E-Link systeem zal een overkoepelend systeem elke druk sensor apart uitlezen en kijken of de druk lager is dan de uitgemiddelde basismeting van die sensor. Als de druk lager is in minstens 2 van de 3 sensoren (majority vote) dan zal dit overkoepelende systeem een start signaal sturen naar het experiment en zal het beginnen meten.

Het volgende UML diagram laat zien hoe het geïmplementeerd was op de microcontroller:

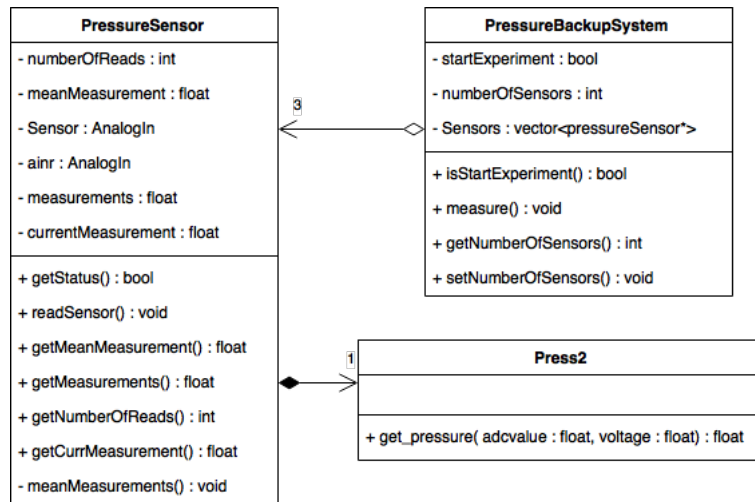


Figure 31: UML diagram van het oude backup start systeem

**Testvlucht.** In de paasvakantie mocht de detector mee met een weerballon van het KNMI om extra data te verzamelen. Van deze vlucht werd gebruik gemaakt om het backup start systeem te testen. Het experiment gebruikte het start systeem uiteraard nog niet om daadwerkelijk te starten, maar de resultaten van het systeem werden telkens weggeschreven zodat hier achteraf naar gekeken kon worden. Op deze iteratie van de detector stond nog maar 1 druksensor dus het effect van het majority vote systeem kon nog niet geanalyseerd worden.

Volgende plot werd gemaakt aan de hand van de data van deze vlucht. De hoogte werd uitgezet in functie van de tijd en de datapunten werden rood gekleurd als het backup systeem het experiment niet zou starten en ze werden groen als het systeem het wel zou starten.

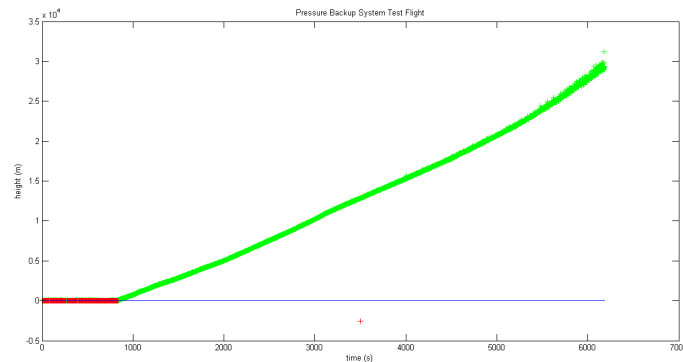


Figure 32: Resultaten van het oude backup start systeem

De horizontale lijn is de referentie waarde die werd berekend na de eerste 10s van drukmetingen.

Hierop is mooi te zien hoe het systeem het experiment start wanneer het in de lucht zit. Maar eerder werd al aangehaald dat er een grote fout zat in onze redenering en dit is duidelijk te zien wanneer we inzoomen op het gedeelte waar de detector nog op de grond staat:



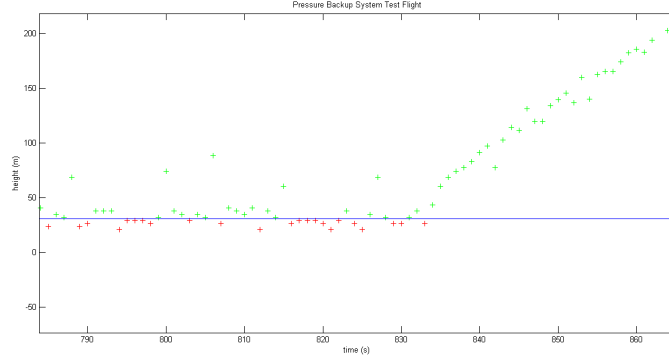


Figure 33: Ingezoomd op het grondgedeelte in de resultaten van het oude backup start systeem

Hierop is te zien dat het systeem het experiment zal starten zelfs wanneer het nog op de grond staat. Dit resultaat is uiteraard logisch want we gebruiken als referentie waarde het gemiddelde van metingen over 10s wat uiteraard zal resulteren in drukwaarden die lager dan dit gemiddelde liggen wanneer de druk wordt gemeten in 50% van de metingen. Dit maar om aan te tonen dat zelfs wanneer een handjevol studenten over hetzelfde probleem nadenken er nog steeds zeer domme fouten kunnen worden gemaakt en dat testen van cruciaal belang is.

**Nieuw doel als stop systeem.** Als oplossing voor de voormalig genoemde fout werd er besloten om de maximale waarde van de druk over de beginmeting van 10s te gebruiken als referentie.

Dit werd niet meer geïmplementeerd of getest aangezien ESA ons op het critical design review (CDR) heeft verteld dat ze geen backup start systeem meer willen op onze detector, maar dat de detector zou moeten starten vanaf dat de spanning aan staat.

Als aanpassing op het huidige systeem wilden ze dat het een backup stop systeem werd. De hoge spanning in het experiment zou afgezet moeten worden vanaf dat de detector een bepaalde hoogte heeft bereikt bij de daling van de ballon.

Deze aanpassing werd gevraagd aangezien de mensen die ons experiment gaan recupereren geen gevaar mogen lopen door ons experiment. De hoge spanning zal hierdoor dus worden afgezet door dit systeem in het geval dat de E-Link connectie niet zou werken tijdens de vlucht.

Het systeem zal nog steeds gebruik maken van een majority vote systeem. Het doel van dit systeem is dat het de hoge spanning af zet na dat de detector op een bepaalde hoogte is geweest en tijdens de daling van de detector. Dit zijn twee zaken die gedetecteerd moeten worden.

Elke keer als de detector een meting doet zal het de drukwaarde opslaan in een buffer. Als het 10 waarden heeft opgeslagen zal het deze waarden uitmiddelen en een standaarddeviatie berekenen:

$$\bar{p} = \frac{\sum_{i=1}^{10} p_i}{10} \quad (4.2.1)$$

$$s_p = \sqrt{\frac{\sum_{i=1}^{10} (\bar{p} - p_i)^2}{10 \cdot (10 - 1)}} \quad (4.2.2)$$

Hierna zal het vorige gemiddelde worden vergeleken met de net berekende. Als de detector aan het dalen is dan zal de druk stijgen. Als aan volgende conditie wordt voldaan dan zal de microcontroller detecteren dat we aan het dalen zijn:

$$\bar{p}_2 - s_{p_2} > \bar{p}_1 + s_{p_1} \quad (4.2.3)$$

met  $\bar{p}_2$  de net berekende gemiddelde waarde en  $\bar{p}_1$  de vorig berekende waarde. Deze conditie moet 5 keer achter elkaar voldaan worden opdat de software zeker weet dat de detector aan het dalen is.

Detecteren of de detector onder een bepaalde hoogte zit (of boven een bepaalde druk) wordt met volgende conditie gecontroleerd:

$$\bar{p}_2 > p_t \quad (4.2.4)$$

met  $p_t$  de threshold druk waarboven de detector moet zijn opdat het druksysteem de hoge spanning mag afzetten. Ook deze conditie moet 5 keer achter elkaar worden voldaan voor een definitief antwoord te krijgen.

Als de detector zeker is dat het zowel aan het dalen is als dat het onder de vooraf bepaalde hoogte zit, dan zal het in geval van verbroken draadloze communicatie de hoge spanning afzetten.

Deze logica werd allereerst getest op de data van de KMI vlucht. Dit gaf volgende resultaten:

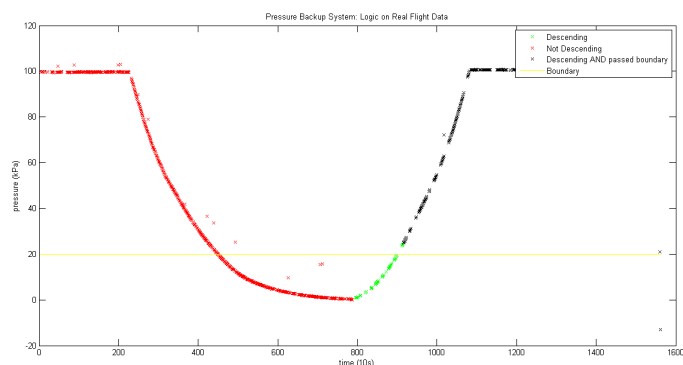


Figure 34: Logica van het backup stop systeem getest op de data van de KMI vlucht

Zoals te zien zal de daling alsook de threshold conditie correct worden gedetecteerd. Als deze twee gevallen tesamen gebeuren dan zal het systeem een signaal geven aan de hoge spanning waarbij deze wordt afgezet. Dit is te zien met de zwarte kruisjes op de plot.

Deze logica werd geschreven voor de microcontroller en werd getest met een vluchtsimulatie programma. Deze vluchtsimulatie wordt in een volgend hoofdstuk in detail besproken. De volgende figuur wordt verkregen als we een simulatie van de KMI vlucht gebruiken:

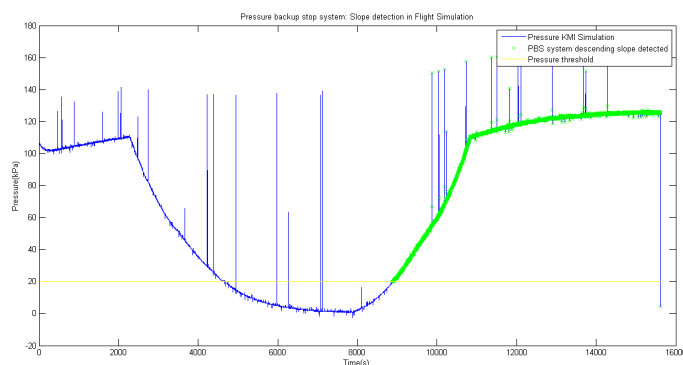


Figure 35: Backup stop systeem daling detectie op de microcontroller getest met een 1 Hz simulatie van de KMI vlucht

Zoals te zien op de figuur wordt de daling correct gedetecteerd door de software. Hetzelfde kunnen we nu testen met de threshold druk en hierbij wordt volgende figuur verkregen:

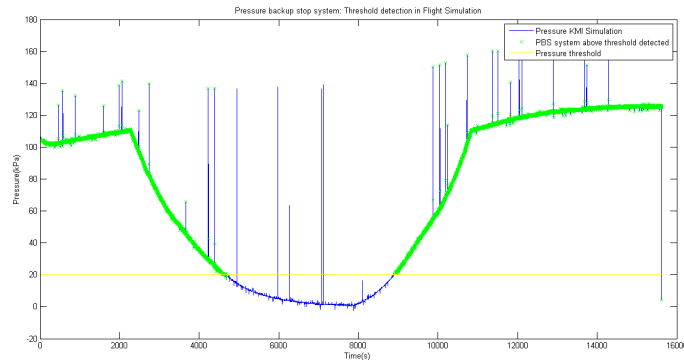


Figure 36: Backup stop system threshold detectie op de microcontroller getest met een 1 Hz simulatie van de KMI vlucht

Ook hier wordt er correct gedetecteerd wanneer de detector boven de vooraf bepaalde druk zit (in dit geval 20 kPa).

Als beide resultaten gecombineerd worden zien we wanneer het backup stop systeem een stop signaal zou sturen naar de hoge spannings converter. Deze figuur ziet er als volgt uit:

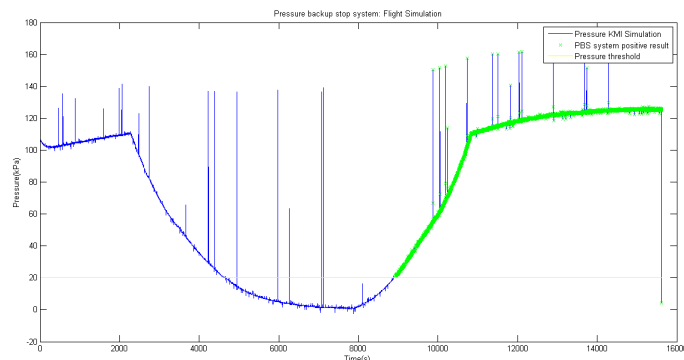
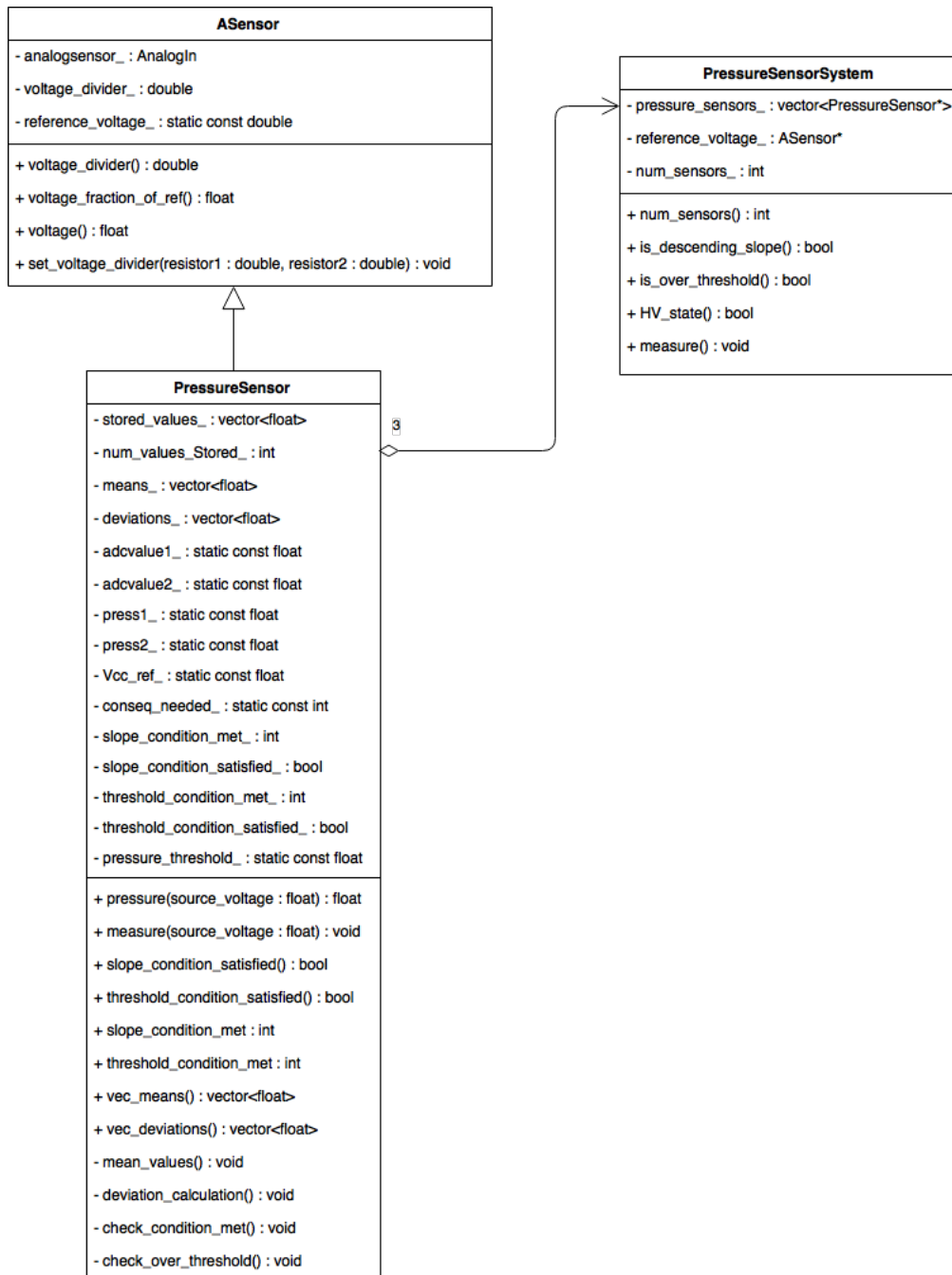


Figure 37: Backup stop systeem op de microcontroller getest met een 1 Hz simulatie van de KMI vlucht

Dit is het gewenste resultaat. De hoge spanning wordt enkel afgezet als het aan het dalen is en boven een vooraf bepaalde druk zit en als er geen netwerk verbinding meer is met het experiment. Al deze tests werden oorspronkelijk gedaan met een 10 Hz simulatie zodat het sneller was om verbeteringen te maken op de software, maar hierdoor werd de ruis meer onderdrukt. Zoals te zien werkt de software ook bij een 1 Hz simulatie wat de abnormale ruispieken van de echte vlucht data zeer goed simuleert. Dit wordt beter uitgelegd in een volgend hoofdstuk.

Het nieuwe UML diagram ziet er als volgt uit:



### 4.2.3 Grondstation

**Doel van het grondstation.** Om gebruik te kunnen maken van het E-Link systeem dat op de gondel staat is er een grondstation nodig. Deze software werd geschreven met als doel om de interactie met het experiment betrouwbaar en intuïtief te maken zodat ook teamleden die de software niet hebben ontwikkeld hiermee kunnen werken.

Het grondstation heeft 3 doelen: Commandos versturen naar het experiment, ontvangen data lokaal opslaan voor het geval dat het experiment niet te recupereren is en ten slotte het visualiseren van de inkomende data.

Zoals eerder al vermeld gebeurt de communicatie volgens het UDP. Een belangrijk aspect van het sturen van een commando is het weten dat het commando is aangekomen. Omdat er wordt gebruik gemaakt

van UDP doordat dit het snelste is en het minste overhead heeft is er inherent geen acknowledgment aanwezig. Dit werd zelf geïmplementeerd bovenop de standaard UDP methoden. Telkens er een pakketje wordt verstuurd vanuit het grondstation zal het experiment het pakketje terugsturen wanneer het is aangekomen. Elke commando zal minder dan 10 bytes zijn waardoor de kans op het verlies van een pakketje op de heen of terug weg relatief klein is.

De data wordt elke seconde verstuurd vanuit het experiment naar het grondstation. Deze data zal lokaal weggeschreven worden zodat mocht de data van het experiment niet recupereerbaar zijn we nog steeds genoeg data hebben voor een goede analyse te doen.

Het laatste aspect van het grondstation is het realtime visualiseren van de inkomende data. Dit laat ons zien hoe de detector zich gedraagt en geeft ons de mogelijkheid om in te grijpen met een commando moest er iets fout zijn (bijvoorbeeld de hoge spanning dat om een of andere reden uitgeschakeld werd). Dit geeft ons ook de mogelijkheid om te zien hoe goed onze data is en deze te vergelijken met andere vluchten.

Doordat het grondstation gebruiksvriendelijk moet zijn voor de andere teamleden werd er een GUI (Graphical User Interface) gemaakt. Dit laat de andere teamleden toe om commandos te sturen zonder dat ze iets moeten typen in een terminal of dergelijke.

**Keuze van programmeertaal** Het grondstation werd eerst ontwikkeld in Python. Deze keuze werd gemaakt omdat Python een taal is waarin zeer snel een werkend prototype kan ontwikkeld worden in vergelijking met C++. Zeker als het op GUI's aankomt. Voor het realtime plotten werd er gebruik gemaakt van de library matplotlib. Dit maakt grafieken van publiceerbare kwaliteit zoals die van Matlab. Dit werd een probleem wanneer het werd toegepast op realtime inkomende data. Elke seconde komt er een nieuw datapakketje binnen waarna de plot hertekend moest worden. Doordat de library mooie plots maakt gebruikt het veel resources en als dit elke seconde moest worden gedaan dan liep de GUI vast en kon er geen commandos meer verstuurd worden.

Hierdoor werd er gekozen om gebruik te maken van C++ omdat dit iets lager level is en er een minder zwaar plotting library ter beschikking was. Voor de GUI werd er gebruik gemaakt van het Qt library/framework. Dit is een zeer gebruiksvriendelijke library dat het maken van een GUI simpel maakt. De plotting library die uiteindelijk werd gebruikt was Qwt. Dit is een licht plotting library dat makkelijk te implementeren is in een GUI gemaakt met Qt.

De verdere details over de implementatie van het grondstation zal achterwege gelaten worden. Dit was enkel maar om aan te tonen welke programmeer talen er werden gebruikt en waarom.

De huidige versie ziet er als volgt uit:

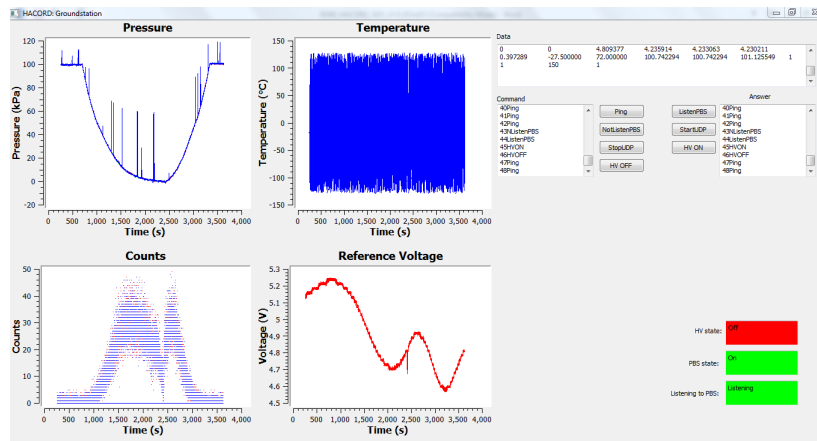


Figure 38: De GUI van de grondstation software getest met de simulatie

Hierop is te zien hoe aan de linkerhelft van het scherm al de data kan worden gevisualiseerd en aan de rechterkant staan alle communicatie aspecten. Er zijn een aantal knoppen ingesteld die een specifiek commando sturen, dit maakt het makkelijk om mee te werken voor andere teamleden. De tekstkader boven de commandoknoppen laat de inkomende ruwe datapakketten zien en de twee tekstkaders naast de commandoknoppen zijn uitgaande en inkomende commandos respectievelijk.

Op de plots is de inkomende data van de KMI simulatie te zien, de temperatuur plot is wartaal omdat de temperatuurs metingen (nog) niet in de simulatie zijn geïmplementeerd.

Er zijn zoals te zien 7 commandos geïmplementeerd:

Commando	Betekenis
Ping	Stuurt een Ping naar het experiment waarop het experiment simpelweg een Ping terugstuurt. Dit is om de connectie te testen.
ListenPBS en NotListenPBS	Dit geeft de mogelijkheid om het backup systeem aan/af te zetten. Als we zien dat het backup systeem voor het gewenste moment de hoge spanning al wilt afzetten dan kunnen we ervoor zorgen dat het experiment hier niet naar luistert.
HVON en HVOFF	Deze commandos zorgen ervoor dat we de hoge spanning kunnen afzetten of aanzetten.
StartUDP en StopUDP	Hiermee kan het versturen van datapakketten gestopt of gestart worden. Dit is belangrijk omdat de ESA vroeg dat we ervoor kunnen zorgen dat er radiostilte kan worden gevraagd.

Wat er ook opvalt aan de verzonden en aangekregen commandos is dat er telkens een nummer mee wordt gestuurd. Dit is een identifier dat elk commando meekrijgt wanneer het wordt verstuurd. Omdat het experiment het exacte commando dat het ontvangt terugstuurt kunnen we hierdoor zien welk commando werd ontvangen en welke niet.

#### 4.2.4 Vlucht Simulatie

Voor het REXUS/BEXUS programma werd de software volledig herschreven waarbij er nieuwe zaken werden toegevoegd zoals UDP en het backupsysteem om de hoge spanning automatisch uit te zetten in het geval dat er geen contact kan worden gemaakt met de detector vanaf het grondstation.

Al deze nieuwe software moet getest worden. De eerste iteratie van het backupsysteem werd getest op een echte vlucht, de vlucht met het KNMI. Dit is uiteraard zeer lastig en tijdsrovend om telkens een echte vlucht te hebben waarbij de detector tot soms een week verdwenen is en er dus ontzettend veel tijd zit tussen het analyseren van de data en het vernieuwen van de software.

Om onder andere deze last uit het ontwikkelings proces te halen werd er een vlucht simulatie programma geschreven gebaseerd op data van echte vluchten. De hoofdzak van deze vlucht simulatie is dat de signalen die in onze microcontroller binnen komen bijna exact dezelfde signalen zijn als tijdens een echte vlucht. Er moeten analoge en digitale signalen gecreerd worden die gelijken op echte sensor spanningen of tellingen van de geiger muller tubes.

Deze signalen zullen worden gegenereerd door een raspberry PI. Dit is een volwaardige computer ter grootte van een broekzak, maar heeft een set general purpose IO pins (gpio) waarmee digitale signalen gecreerd of gelezen kunnen worden [15]. De raspberry PI heeft geen Analog Out pins waardoor er een ander truukje moet gebruikt worden om echte analoge signalen te genereren.

Er zijn twee obstakels met het simuleren van de echte signalen. Dit zijn: het vooraf vermelde probleem dat er geen Analog Out pins staan op de Raspberry PI en dat simultane digitale pulsen niet mogelijk zijn met high level software commandos. Allereerst zal dit laatste probleem worden besproken.

**Simultaneiteit van digitale pulsen.** Als we echte kosmische stralen meten met Geiger Muller tubes dan detecteren we onder andere coincidenties tussen verschillende tubes. Dit zijn signalen die instantaan worden gegenereerd in minstens 2 Geiger Muller tubes. Hierdoor zullen er minstens 2 pins van de microcontroller HIGH staan waarbij dit wordt gemeten tijdens de uitlezing van de pins en dit als een coincidentie in de software wordt opgevat.

Allereerst werd de vluchtsimulatie met een Arduino UNO gemaakt (dit is een zeer gebruiksvriendelijke adaptatie van een ATMEGA 328PU microcontroler). Er werd uiteindelijk voor een Raspberry PI gekozen omdat de Arduino UNO niet genoeg geheugen had voor de vlucht simulatie, zelfs met veel optimalizaties. Bij deze computers/microcontrollers zijn er een basisset commandos die op een niet abstracte manier de pins van de microcontroller HIGH kunnen zetten. Om 2 pins "simultaan" HIGH te zetten zou de naieve implementatie zijn om achter elkaar het commando hiervan op te roepen op verschillende pins. Dit ziet er dan als volgt uit:

---

```
DigitalWrite(1, HIGH);  
DigitalWrite(2, HIGH);
```

---

Waarbij 1 en 2 de namen van de pins zijn. Voor bijna alle praktische toepassingen is deze implementatie genoeg voor "simultane" pulsen te genereren. Maar voor ons is dit niet snel genoeg. Elk commando zal tussen de 7 en 14 microseconden nodig hebben om een pin HIGH te zetten. Wanneer de eerste pin HIGH wordt gezet zal dit een uitlees functie oproepen op de detector microcontroller. Deze functie zal kijken welke pin er nog op HIGH staat zodat de coincidenties kunnen geregistreerd worden. Het "probleem" in de detector software is dat de uitlees functie ongeveer 5 microseconden nodig heeft om uitgevoerd te worden. Dit is niet genoeg tijd om de tweede pin op HIGH te zetten. Hierdoor wordt er door de naieve implementatie geen coincidentie geregistreerd waardoor dit niet geschikt is voor een realistische vlucht simulatie.

Op de Raspberry PI hebben we hetzelfde probleem. Dit wordt opgelost door gebruik te maken van Port Manipulation [14]. Dit is een lagere level methode om de pins van de microcontroller te controleren. Dit proces zal worden uitgelegd voor de Arduino, maar is volledig analoog voor de Raspberry PI.

De Arduino heeft 3 ports genaamd: B (digitale pins 8 tot 13), C (analoge input pins) en D (digitale pins 0 tot 7). Elke port wordt gemanipuleerd door 3 registers genaamd: DDR (bepaald of de pin een input of een output is), PORT (bepaald of de pin HIGH of LOW staat) en PIN (leest pins uit dat input pins zijn).

Stel dat we pin 7, 5 en 3 willen gebruiken als output en om deze pins simultaan HIGH te zetten, dan worden volgende stappen gezet:

Allereerst weten we dat pin 7, 5 en 3 in de D port zitten. Hierna moeten deze pins als outputs worden

ingesteld door gebruik te maken van de Data Direction Register (DDR). Dit ziet er dan als volgt uit:

---

```
DDRD = B10101000;
```

---

Zoals duidelijk te zien is, is dit een zeer lage level manier van interageren met de pins van de microcontroller.

Hierna kunnen deze pins simultaan HIGH worden gezet door gebruik te maken van de PORT register:

---

```
PORTD = B10101000;
```

---

Dit laatste commando is het equivalent van de naieve implementatie:

---

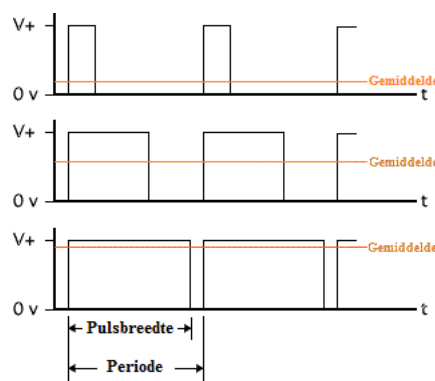
```
DigitalWrite(7, HIGH);  
DigitalWrite(5, HIGH);  
DigitalWrite(3, HIGH);
```

---

maar is ook voor onze doeleinden simultaan doordat er veel minder klokcycles nodig zijn om de pin HIGH te zetten. Dit is omdat er veel meer bagage zit in de hoge level functies.

Er zijn veel details weggelaten over registers en port manipulation omdat anders de thesis technisch te ingewikkeld wordt en dit uiteindelijk geen meerwaarde biedt bij het uitleggen en het oplossen van het simultaneiteits probleem van digitale pulsen.

**Analoge signalen met digitale pulsen.** Het volgende probleem is het genereren van een analoog signaal door enkel gebruik te maken van digitale signalen. Dit wordt gedaan met een concept genaamd Pulse Width Modulation (PWM) of Pulsebreedtemodulatie [12]. Dit is een modulatietechniek waarbij dat de duty cycle van een blokgolf wordt aangepast zodat de gemiddelde spanning de gewenste spanning is. Dit kan onder andere worden gebruikt om de intensiteit van een LED aan te passen simpelweg door gebruik te maken van enkel HIGH of LOW digitale signalen. De PWM ziet er als volgt uit:



Om deze modulatie te gebruiken in de software moet er een duty cycle worden meegegeven. De Raspberry Pi splitst de 3.3V op in 256 stukken. In de software ziet dit er als volgt uit:

---

```
gpioPWM(24, 255); // Duty Cycle 100%  
gpioPWM(24, 128); // Duty Cycle 50%  
gpioPWM(24, 0); // Duty Cycle 0%
```

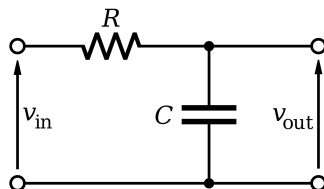
---

Deze commandos manipuleren de duty cycle van pin 24 waardoor de gemiddelde spanning respectievelijk 3.3V, 1.65V en 0V is. Hoe de PWM waarden worden berekend zal later concreet worden uitgelegd.

Het probleem is dat nu de gemiddelde spanning wel kan worden gemanipuleerd, maar dat het signaal nog steeds een digitaal signaal is dat ofterwel HIGH staat of LOW. Als dit signaal wordt aangeboden in



een Analoge Input pin van een microcontroller dan zal de microcontroller nog steeds geen echte analoge spanning lezen buiten LOW of HIGH (in ons geval is dit 0V en 3.3V respectievelijk). Om een echt analoge signaal te kunnen aanbieden aan de microcontroller zal dit PWM signaal gefilterd moeten worden op de hoge frequenties waarmee de pin van HIGH naar LOW wordt geschakeld waardoor er effectief wordt uigemiddeld over het signaal en het gemiddelde analoge signaal het echte signaal wordt. Dit wordt gedaan met een RC filter, of specifieker een Low Band Pass filter [13]. Voor onze doeleinden is een condensator van 100  $\mu\text{F}$  en een weerstand van 4.7 k $\Omega$  genoeg. Verdere uitleg over filter design wordt achterwege gelaten aangezien dit maar een eerste orde filter is en hier niets speciaal bij gebeurt. De schakeling ziet er als volgt uit:



Deze twee concepten worden gebruikt om een realistische vlucht simulatie te creëren waarbij alle toekomstige software mee kan worden getest. Ook kan er met een vlucht simulatie voorbereid worden op verschillende scenarios die zich kunnen voordoen, van stroomuitval tot netwerk storingen.

De vlucht simulatie wordt gemaakt door een Python script dat al de data files van een vlucht uitleest en hiermee een C++ programma schrijft dat kan worden gecompileerd en gerund op de Raspberry PI. Voor de PWM waarden wordt volgende formule gebruikt:

$$PWM = \text{round}\left(\frac{V_{data} \cdot N_{stappen} \cdot \text{Spanningsdeler}}{V_{max}}\right) \quad (4.2.5)$$

Waarbij round een functie is die afrond naar boven als het getal na de komma groter of gelijk is aan 5 en anders naar beneden afrond.

De verschillende factoren van deze formule zullen worden uitgelegd met een voorbeeld:

Stel de druksensoren hebben een output van 4.5V, dit is onze  $V_{data}$ . De maximale spanning  $V_{max}$  is 3.3V voor onze microcontroller. Alle spanningen die gelijk zijn of groter dan 3.3V worden als hetzelfde beschouwd. Hierdoor kan de 4.5V niet rechtstreeks in de microcontroller worden gevoerd. Met een spanningsdeler kan deze spanning tot onder de maximale spanning van 3.3V geschaald worden zodat de input nooit groter is dan 3.3V. In de software van de microcontroller zal de spanning dan uiteindelijk worden herberekend naar de echte waarde. Aangezien we een geheel getal nodig hebben moeten we weten hoeveel volt elke stap teweegbrengt. De maximale spanning is 3.3V en deze is verdeeld in 255 stappen (waarbij de 256e stap 0V is). Hierdoor kunnen we het aantal volt per stap berekenen. De spanings gedeelde waarde van de dataset wordt dan gedeeld door deze factor om zo op het gehele getal te komen dat de duty cycle voorstelt. Voor onze 4.5V wordt dit:

$$PWM = \text{round}\left(\frac{4.5V \cdot 255 \cdot 0.5}{3.3V}\right) = 174 \quad (4.2.6)$$

De PWM waarde van 174 zal een gemiddelde spanning van 2.25V genereren die door de filter als input kan gebruikt worden voor de microcontroller. In de microcontroller software zal door gebruik te maken van de spannings deler de echte spanning van 4.5V berekend worden.

Het Python script zal voor elk datapunt (ongeveer 15000 voor de KMI vlucht en 14000 voor de KNMI vlucht) andere argumenten invullen in een functie. Al deze functie calls zullen in een switch constructie staan doordat dit veel beter wordt geoptimaliseerd dan if-else if-else constructies voor een groot aantal verschillende statements. Om de microcontroller en de Raspberry PI in sync te houden met elkaar zal de microcontroller een digitaal signaal sturen telkens wanneer het klaar is om het volgende datapunt te ontvangen. De Raspberry PI zal dit signaal gebruiken als trigger om de teller van de 15000 lijnen switch

constructie te incrementeren om zo het volgende datapunt te simuleren.

**Verificatie van simulatie.** Uiteraard moet er nu geverifieerd worden dat de simulatie nagenoeg hetzelfde is als de echte vlucht data. Dit wordt gedaan door de druk van de simulatie te vergelijken met de druk van de dataset waaruit de simulatie is opgebouwd. Voor de KMI data ziet dit er als volgt uit:

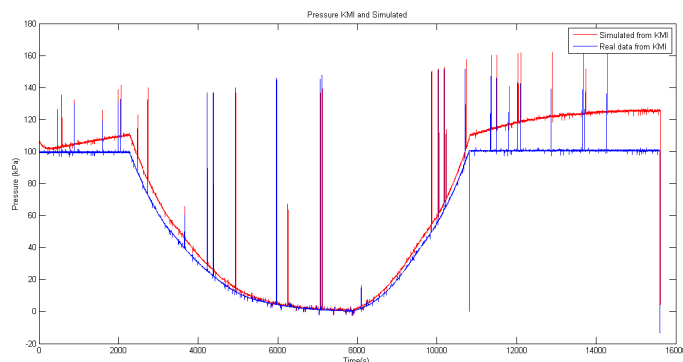


Figure 39: Druk over tijd van de echte vlucht vergeleken met een 1 Hz vlucht simulatie

In dit geval werden de datapunten van de simulatie op dezelfde snelheid (1Hz) doorgestuurd naar de microcontroller als tijdens de echte vlucht (tijdens de echte vlucht werd er elke seconde een meting gedaan). Wat duidelijk is is dat de simulatie niet exact gelijk is aan de echte data. Dit komt doordat er uiteraard zelf nog ruis zit op de gegenereerde spanningen. Wat belangrijk is is dat de gesimuleerde data door dezelfde fases gaat als de echte data. Het heeft ook nagenoeg constante druk bij landing en voor het opstijgen. Het daalt en stijgt ook op dezelfde wijze als bij de echte data. Hierdoor kan worden geconcludeerd dat de simulatie wel degelijk nuttig is om nieuwe software mee te testen. Het feit dat de simulatie telkens wat zal verschillen door ruis kan een voordeel zijn zodat de data waarop de software wordt getest niet telkens identiek dezelfde is.

Wat een van de voordelen is van een simulatie opgebouwd uit een echte dataset is dat ook ruis en zeer eigenaardig gedrag van analoge sensors wordt gesimuleerd. Zoals te zien op volgende figuur worden de abnormale pieken zeer goed gesimuleerd. Deze ruis simulatie is handig om ervoor te zorgen dat de software (zoals het backup stop systeem) niet enkel werkt bij ideale sensor responsen, maar ook bij ruisige signalen:

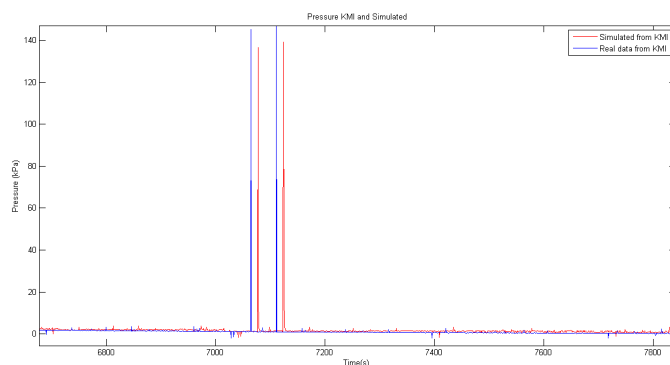


Figure 40: Zeer scherpe pieken in de echte data worden zeer goed gesimuleerd bij 1 Hz simulatie

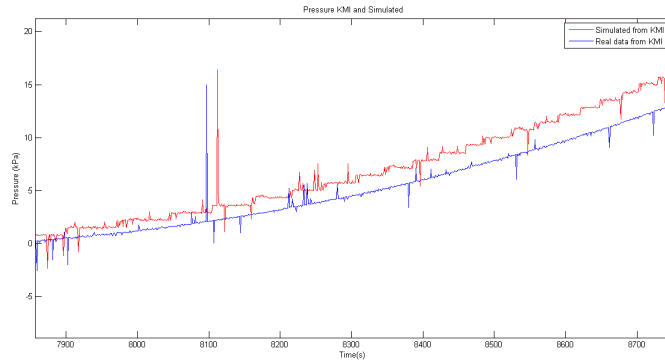


Figure 41: Ook kleinere ruis pieken worden zeer goed gesimuleerd bij 1 Hz simulatie

Een ander voordeel aan een vluchtsimulatie gebruiken is het feit dat we een vlucht van iets meer dan 4 uur (KMI vlucht) betrouwbaar kunnen simuleren in een tijdsspanne van 30 minuten. Dit maakt het zeer snel om nieuwe iteraties van de software te testen in elk gebied van de vlucht (ascent, float en descent).

Wat wel moet worden opgemerkt is dat hoe sneller de datapunten worden gegenereerd hoe minder goed de ruis wordt gesimuleerd. Dit is te zien op volgende figuur waarbij de datapunten met een frequentie van 10 Hz werd gesimuleerd:

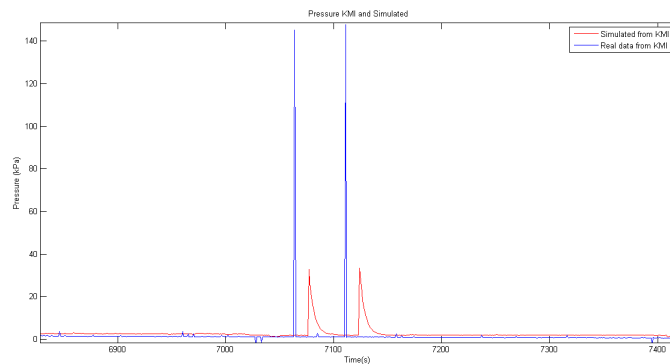


Figure 42: Zeer scherpe pieken in de echte data worden minder goed gesimuleerd bij 10 Hz simulatie

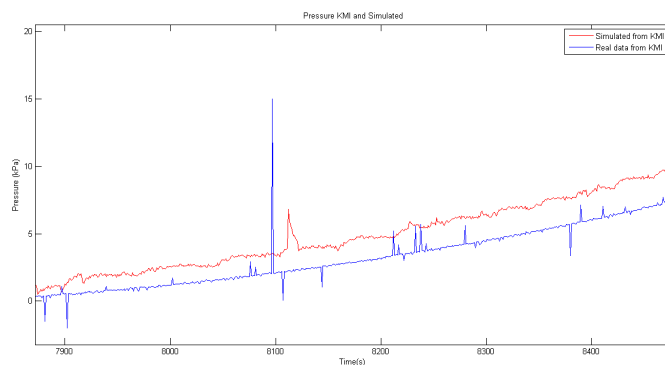


Figure 43: Ook kleinere ruis pieken worden minder goed gesimuleerd bij 10 Hz simulatie

Dit gebeurt doordat er niet genoeg tijd is voor de condensator om volledig opgeladen te worden wanneer

er een scherpe piek ontstaat in de data. Hierdoor zal deze 10 Hz simulatie de ruis van de data effectief uitdoven.

Deze zeer snelle simulatie kan worden gebruikt als eerste test voor de nieuwe software, maar uiteindelijk moet elke software iteratie getest worden met de 1 Hz simulatie zodat de software ook getest wordt met de juist gesimuleerde ruis.

Zoals eerder vermeld kan de simulatie gebruikt worden om verschillende error scenarios te testen. De voornaamste scenarios die moeten worden getest is het gedrag van het backup stop systeem op stroom uitval tijdens verschillende fasen van de vlucht. Ook zal er worden getest hoe goed de software reageert wanneer de ethernet verbinding onderbroken wordt of wanneer er een file error ontstaat. Door al deze scenarios te kunnen uittesten en ons hierop te kunnen voorbereiden zal ESA genoeg vertrouwen krijgen in ons experiment. De simulatie kan ook worden gebruikt om de werking van het grond station te testen.

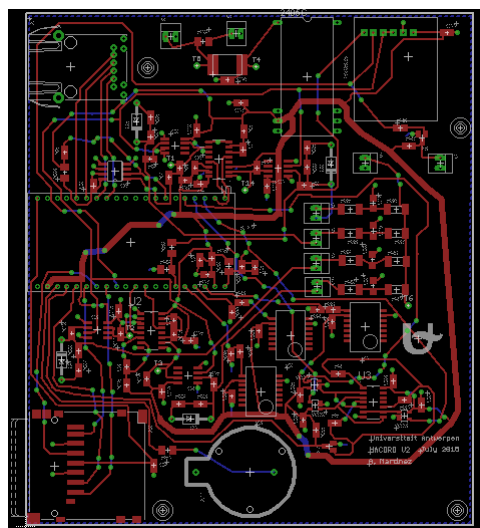
## 4.3 Elektronica

### 4.3.1 Nieuwe PCB

De ESA wou dat we een nieuwe PCB ontworpen en niet met de bestaande vlogen. Hierdoor konden we ook de benodigde aanpassingen doen aan de detector zonder al te veel problemen.

De nieuwe PCB werd ontworpen door mijzelf in nauwe samenwerking met Ir. Wim Beaumont die me genoeg aanleerde over PCB design om dit experiment succesvol te kunnen uitvoeren.

Er werden verschillende iteraties ontworpen door mijzelf waar na elke iteratie werd besproken met Wim welke aanpassingen ik nog zou moeten doorvoeren. De uiteindelijke PCB ziet er als volgt uit:



Dit design werd naar een fabrikant gestuurd en werd gedurende de eerste maand van Juli 2015 gesoldeerd in het labo van de onderzoeksgroep EDF door mijzelf en Wim.

### 4.3.2 Aanpassingen t.o.v. HACORD-III

Op de nieuwe PCB staan 2 extra druksensoren. Deze werden gevraagd door ESA zodat we 3 druksensoren hebben en we een majority vote systeem konden gebruiken voor het backup stop systeem.

Er moesten DC/DC converters voorzien worden om gebruik te kunnen maken van de inkomende 28V van de batterijen die op de gondel staan. Deze werden op de nieuwe PCB gezet, alsook de mogelijkheid om nog steeds onze eigen batterijen te gebruiken in een toekomstige vlucht waarbij er geen batterijen voorzien zijn voor de experimenten.

Omdat de vlucht aanzienlijk langer duurt dan de vluchten met een weerballon was er meer opslagruimte nodig. Hiervoor werd er een SD connector op de PCB gezet. Waar wel aan gedacht moest worden is dat SD kaarten op 3.3V werken en niet op de 5V die de rest van de componenten gebruiken. Aangezien de SD kaart maar ongeveer 100 mA nodig heeft kan dit worden aangestuurd door de microcontroller. Deze heeft een onboard regulator die de inkomende spanning omzet naar een uitgaande 3.3V en die tot 950 mA kan leveren. Als we rekenen op een 150 bytes/s en het is een 5 uur durende vlucht dan zouden we 2.7 MB opslagruimte nodig hebben. De microcontroller heeft ongeveer 1.5 MB opslagruimte. De grootte van de SD kaart zal dus uiteindelijk niets uitmaken aangezien zelfs de kleinste SD kaart genoeg opslagruimte zal zijn.

Omdat het experiment verbinding moet maken met een lokaal netwerk moet hiervoor ook een RJ45 connector worden voorzien.

De laatste belangrijke aanpassing is het toevoegen van 3 watchdogs. Een watchdog is een IC dat binnen een bepaald tijdsinterval (in ons geval 2s) een verandering van state aan de ingang verwacht. Als deze verandering niet gebeurt zal de uitgang van state veranderen en dit kan worden gebruikt voor onder andere de microcontroller te resetten. Dit betekent dat elke 2s in de software van de microcontroller de watchdogs moeten getriggert worden.

Er worden gebruik gemaakt van 3 watchdogs die in een majority vote systeem zullen werken. Deze zijn in elk mogelijke combinatie verbonden aan NOR gates waarvan de resultaten zijn verbonden aan een transistor die verbonden is aan de reset pin van de microcontroller. Als de transistor aan zijn gate een spanning krijgt zal deze de reset pin verbinden met ground waardoor de microcontroller reset.

Dit mechanisme is om ervoor te zorgen dat in het geval van een grote software error, waardoor de microcontroller vast zit, de microcontroller hestart wordt.

## 5 Algemeen Besluit

In deze thesis werd elk aspect van een fysica experiment aangeraakt. Data analyse, fouten analyse (detector fouten), detector design op fysica, elektronica en software niveau en samenwerking in een team om de doelen van het experiment te bereiken.

Uit de data analyse werd er besloten dat het Pfozter Maximum hoger ligt dan vorige jaren, maar doordat het over 3 jaar met 1 count per seconde gestegen is is dit resultaat niet noodzakelijk te wijten aan het observeren van de komende zonne minimum.

Uit de fouten analyse van een falende detector werd er besloten dat het probleem hoogst waarschijnlijk bij de gebruikte batterij lag. Doordat de batterij te koud werd kon het niet genoeg energie leveren om het experiment aan te houden. De batterijen kwamen in een toestand waarbij ze net genoeg energie konden leveren om de microcontroller op te starten, maar niet om het gehele experiment werkend te houden.

Door het REXUS/BEXUS programma kwamen elektronica en software design aan bod. Alle keuzes die werden gemaakt tijdens het ontwerp van de elektronica en software moesten gedetailleerd verantwoord worden in documenten. Deze documenten werden zeer kritisch gelezen door de mensen die verantwoordelijk zijn voor het REXUS/BEXUS programma.

## References

- [1] CERN, *Cosmic rays: particles from outer space*, <http://home.web.cern.ch/about/physics/cosmic-rays-particles-outer-space> geraadpleegd op 01/08/15.
- [2] NASA, *Cosmic Rays*, <http://helios.gsfc.nasa.gov/cosmic.html> geraadpleegd op 12/08/15.
- [3] Hyper Physics, *Cosmic Rays*, <http://hyperphysics.phy-astr.gsu.edu/hbase/astro/cosmic.html> geraadpleegd op 01/08/15.
- [4] P. K. F. Grieder, *Cosmic Rays at Earth: Researcher's Reference Manual and DataBook*, p20, 2001, Elsevier.
- [5] Jeroen Van Houtte, *Kosmische straling in de atmosfeer*.
- [6] ARM mbed, [mbed.org](http://mbed.org) geraadpleegd op 01/08/15.
- [7] Duracell, Ultra 245 Datasheet, [http://professional.duracell.com/downloads/datasheets/product/Specialty/Specialty\\_Photo\\_Ultra\\_245.pdf](http://professional.duracell.com/downloads/datasheets/product/Specialty/Specialty_Photo_Ultra_245.pdf) geraadpleegd op 01/08/15.
- [8] Dr Kevin F Scott, *How well do dry batteries perform in the cold?*, <http://drkfs.net/coldbattery.htm> geraadpleegd op 01/08/15.
- [9] Battery University, *Charging at High and Low Temperatures*, [http://batteryuniversity.com/learn/article/charging\\_at\\_high\\_and\\_low\\_temperatures](http://batteryuniversity.com/learn/article/charging_at_high_and_low_temperatures) geraadpleegd op 01/08/15.
- [10] Dr. Anne Marie Helmenstine, *Why do batteries discharge more quickly in cold weather ?*, <http://chemistry.about.com/od/howthingsworkfaqs/f/coldbattery.htm> geraadpleegd op 01/08/15.
- [11] REXUS/BEXUS programme, <http://rexusbexus.net/> geraadpleegd op 01/08/15.
- [12] Arduino, PWM method, <https://www.arduino.cc/en/Tutorial/PWM> geraadpleegd op 01/08/15;
- [13] Texas Instruments, *Using PWM output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Controller*, <http://www.ti.com/lit/an/spra88a/spra88a.pdf> geraadpleegd op 01/08/15.
- [14] Arduino, *Port Registers*, <https://www.arduino.cc/en/Reference/PortManipulation> geraadpleegd op 01/08/15.
- [15] pigpio, GPIO library for Raspberry PI, <http://abyz.co.uk/rpi/pigpio/cif.html> geraadpleegd op 01/08/15.
- [16] IBM, IBM experiments in soft fails in computer electronics (1978-1994), from Terrestrial cosmic rays and soft errors, IBM Journal of Research and Development, Vol. 40, No. 1, 1996. Retrieved 16 April 2008.
- [17] NASA, *U.S. Standard Atmosphere 1976*, 1976