

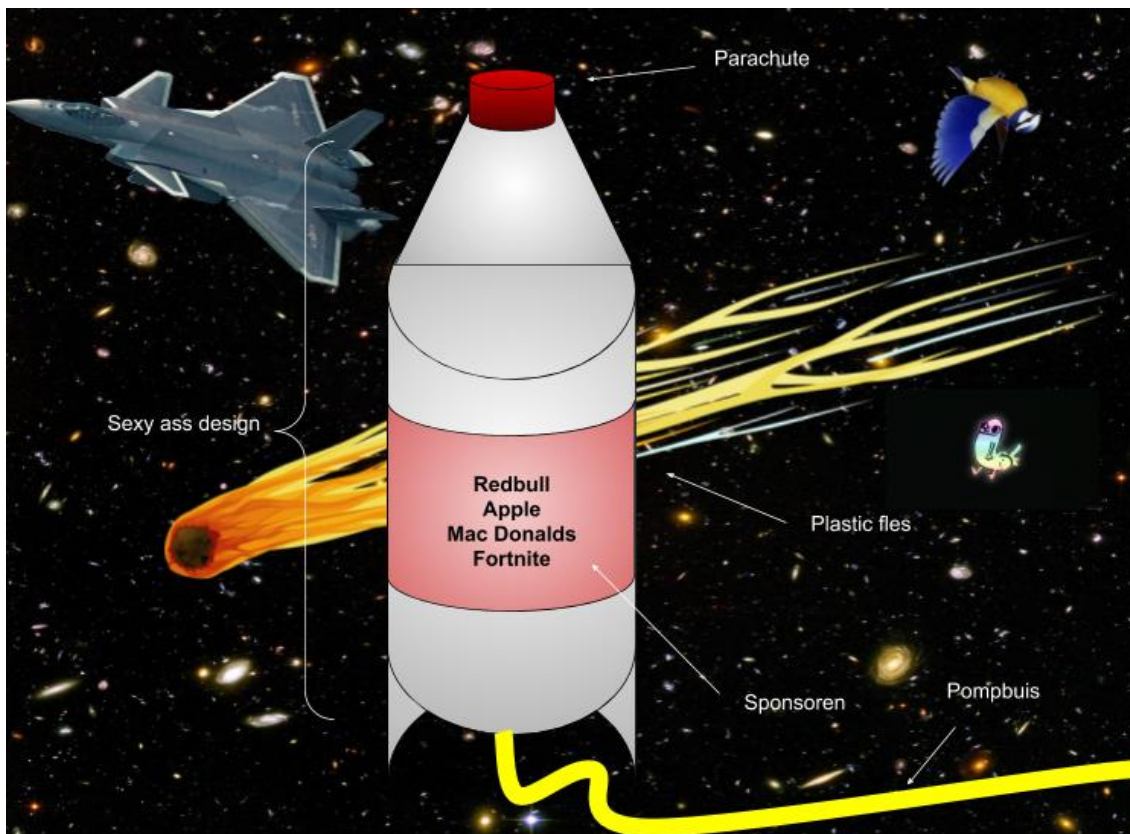
Ontwerp en realisatie van een waterraket

Profielwerkstuk van

Paul Groen, 6C

Timo Hoogenbosch, 6E

08-01-2021



Inhoudsopgave

Inleiding	4
Theorie	5
De raket	5
Gewichtsverdeling van de raket	5
Krachten tijdens een raketvlucht	6
Het meten van de vlucht	8
Het bepalen van de snelheid	9
Het bepalen van de hoogte	9
Ontwerp en realisatie	11
De raket	11
Het ontwerp	11
De realisatie	14
De boordcomputer	17
De hardware	17
De software	21
Het lanceerplatform	33
Het ontwerp	33
De realisatie	34
Lancering	41
Vorbereiding	41
Lancering 1	42
Analyse	42
Herstel van de raket	44
Lancering 2	45
Conclusies	46
Evaluatie	47
Bronvermelding	48
Appendix met bijlagen	48
Broncode van de boordcomputer	50
Link naar video lancering	50
Pakket van Eisen	50
Raket	50
Boordcomputer	50
Lanceerplatform	50
Model	50
Plan van Aanpak	51
Mogelijke problemen	51
Logboek	51

Inleiding

Sinds juni 2020 zijn we bezig geweest met de vraag: wat komt er kijken bij het ontwerpen en realiseren van een herbruikbare waterraket voorzien van elektronische meet- en regelapparatuur? In dit verslag geven we antwoord op deze vraag, van de natuurkunde waar we rekening mee moeten houden tot aan het ontwerpen, bouwen, lanceren en evalueren van de raket.

We willen een waterraket gaan bouwen, een raket die voortgestuwd wordt door lucht- en waterdruk. Ons doel wordt om de raket zo hoog mogelijk te laten vliegen. Deze zal dus aan een aantal eisen¹ moeten voldoen, zo moet de raket aerodynamisch en stabiel zijn, zodat hij zo recht mogelijk omhoog gaat. De raket zal een parachute bevatten zodat hij ook weer kan landen zonder te beschadigen. In de raket zal zich een Arduino® als boordcomputer bevinden, die tijdens de vlucht de gegevens van een versnellingsmeter en een barometer zal opslaan op een SD-kaart, en bovendien de besturing van de raket zal regelen (zoals het ontplooiën van de parachute tijdens de afdaling) door een servomotor aan te sturen. Ook zullen we een geschikt lanceerplatform moeten bouwen zodat de raket veilig gelanceerd kan worden. Voordat we de raket lanceren zullen we eerst een voorspelling maken van hoe hoog de raket zal komen, door de vlucht te modelleren in Coach 7. Na de vlucht zullen we de gegevens uit de versnellingsmeter gebruiken om de vlucht te reconstrueren, en te vergelijken met het model.

Om dit allemaal te realiseren zullen we een aantal obstakels moeten overkomen, zoals: hoe zorgen we voor een zo hoog mogelijke stuwkracht, maar een zo laag mogelijk gewicht? Hoe maken we de raket aerodynamisch? Hoe bouwen we een parachute-ontplooingsmechanisme? Hoe zorgen we ervoor dat de boordcomputer alles correct timet?

¹ Het pakket van eisen staat in de bijlage op pagina 45.

Theorie

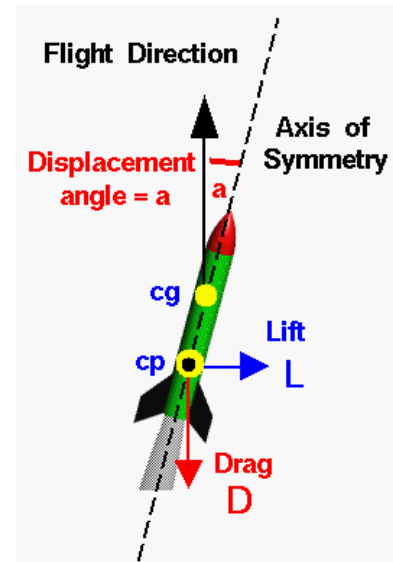
In dit deel van het verslag wordt op basis van natuurwetten uitgelegd hoe de raket (zo goed mogelijk) vliegen kan. Alles waarmee rekening mee moet worden gehouden bij het ontwerpen van de raket wordt hier beschreven.

De raket

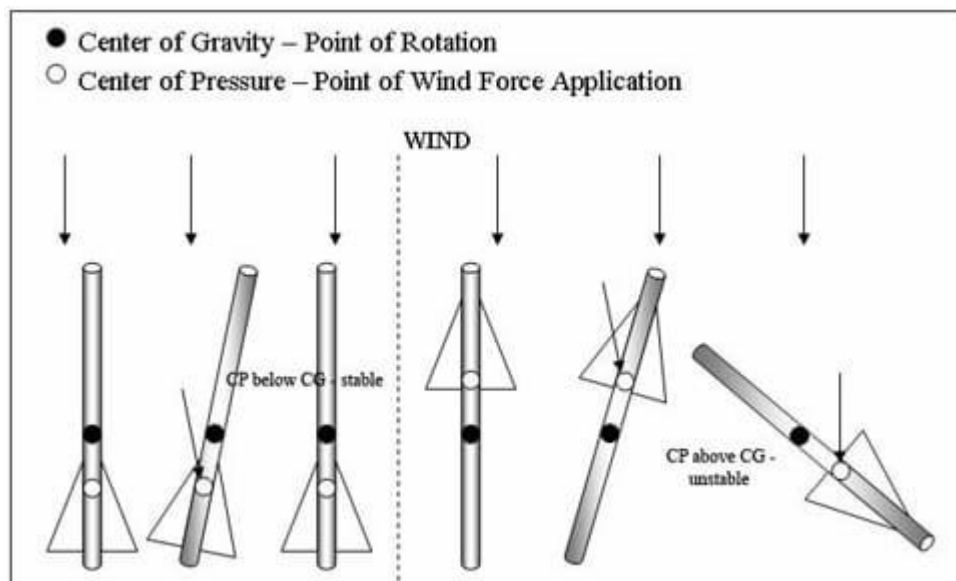
Gewichtsverdeling van de raket

Om de raket stabiel en recht te laten vliegen moet het drukmiddelpunt onder het zwaartepunt liggen². Het zwaartepunt is het punt waarop de zwaartekracht werkt, en dit is het punt waar een object omheen zal draaien als er aan één kant van het zwaartepunt een kracht wordt uitgeoefend. Het drukmiddelpunt is het punt waarop alle aerodynamische krachten werken. De aerodynamische krachten die een rol spelen zijn de luchtweerstandskracht en de liftkracht.

Als een raket tijdens de vlucht schuin gaat vliegen ten opzichte van de bewegingsrichting, zal er een liftkracht uitgeoefend worden op het drukmiddelpunt van de raket. Als het drukmiddelpunt boven het zwaartepunt ligt zal de raket hierdoor omklappen. Als het drukmiddelpunt boven het zwaartepunt ligt zal de raket zichzelf stabiliseren.



Figuur 1 Krachten die tijdens de vlucht op de raket werken. Bron: NASA



Figuur 2 Effect positie drukmiddelpunt en zwaartepunt op stabiliteit raket. Bron: teachengineering.org

Bepaling van het zwaartepunt

Het zwaartepunt van de raket kan bepaald worden door het punt te vinden waar de raket op gebalanceerd kan worden.

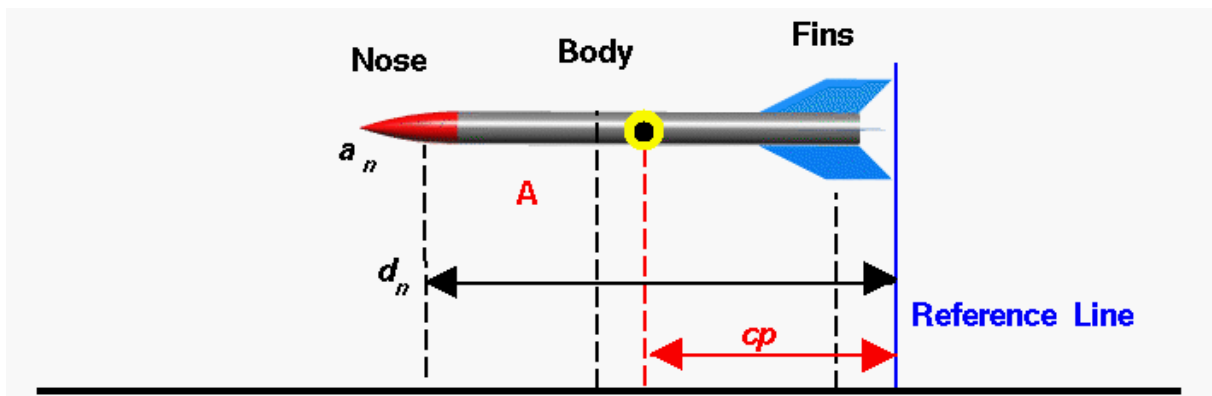
² White, J. (2006). Using Thrust, Weight & Control: Rocket Me into Space. TeachEngineering.org. Geraadpleegd op 27 december 2020, van https://www.teachengineering.org/lessons/view/cub_rockets_lesson03.

Bepaling van het drukmiddelpunt

Het bepalen van het drukmiddelpunt gaat met behulp van de volgende formule³:

$$c_p * A = d_n * a_n + d_b * a_b + d_f * a_f + \dots \quad (1.1)$$

Hier is c_p de afstand van de referentielijn tot het drukmiddelpunt van de hele raket en A de oppervlakte van een 2D-projectie van de zijkant van de raket, zoals in Figuur 3. Dit wordt gelijkgesteld aan de som van de afstanden d van de referentielijn tot het drukmiddelpunt van elk afzonderlijk onderdeel maal de oppervlakte a van elk afzonderlijk onderdeel.



Figuur 3 Bepaling van het drukmiddelpunt. Bron: NASA.

Het drukmiddelpunt van symmetrische vormen ligt in het midden. Van de overige onderdelen wordt het drukmiddelpunt anders bepaald. Aangezien dit punt hetzelfde is als het zwaartepunt van een 2D-projectie van een object, kan deze bepaald worden door het zwaartepunt te bepalen van zo'n projectie in het computerprogramma Fusion 360. Door een foto van de afzonderlijke onderdelen over te trekken in dit programma, hier een object van te maken, en vervolgens het zwaartepunt te bepalen kan het drukmiddelpunt bepaald worden. Eveneens kan met Fusion 360 de oppervlakte van complexe figuren bepaald worden.



Figuur 4 Het drukmiddelpunt van de top van de raket, bepaald met behulp van Fusion 360.

Krachten tijdens een raketvlucht

De krachten die van toepassing zijn op de beweging van de raket zijn de stuwkracht, de zwaartekracht en de luchtweerstandskracht.

Stuwkracht⁴

De stuwkracht wordt geleverd door een stroom water die de straalbuis verlaat. Een formule voor de stuwkracht kan worden afgeleid uit impulsbehoud.

$$p_{\text{water dat de raket verlaten heeft}} + p_{\text{raket}} = 0 \quad (p = mv)$$

³ National Aeronautics and Space Administration (NASA). (z.d.). Rocket Center of Pressure. NASA Glenn Research Center. Geraadpleegd op 10 december 2020, van <https://www.grc.nasa.gov/www/k-12/rocket/rktcp.html>

⁴ Finney, G. A. (2000). Analysis of a water-propelled rocket: A problem in honors physics. American Journal of Physics, 68(3), 223–227. <https://doi.org/10.1119/1.19415>

Kracht is een verandering van impuls over tijd.

$$F = \frac{dp}{dt} = \frac{d(mv)}{dt}$$

De impuls van de totale hoeveelheid water die de raket heeft verlaten verandert constant, omdat de hoeveelheid water die de raket heeft verlaten constant toeneemt. De massastroom van het water dat de straalbuis verlaat is $\frac{dm}{dt}$. De snelheid waarmee het water de raket verlaat is v .

$$\begin{aligned} F &= \frac{d(mv)}{dt} = \frac{dm}{dt} v \\ F_{stuw} &= \frac{dm}{dt} v \end{aligned} \quad (2.1)$$

De massastroom van een vloeistof wordt gegeven door:

$$\frac{dm}{dt} = \rho \frac{dV}{dt} = \rho A v \quad (2.2)$$

Hierin is ρ de dichtheid van het water, A de doorsnede van de straalbuis en v de snelheid waarmee het water stroomt. De energie die nodig is om het water te versnellen is afkomstig uit de opgebouwde druk in de fles. Dit kunnen we afleiden met de vergelijking van Bernoulli.

$$P_1 + \frac{1}{2} \rho v_1^2 + \rho g h_1 = P_2 + \frac{1}{2} \rho v_2^2 + \rho g h_2$$

Hier stellen we punt 1 als het oppervlak van het water in de raket en punt 2 als de uitgang van de straalbuis. Het energieverschil dat ontstaat door het verschil in waterhoogte (term $\rho g h$) verwaarlozen we. Het energieverschil dat ontstaat doordat het oppervlak van het water een kleine snelheid heeft ten opzichte van de raket (term $\frac{1}{2} \rho v_1^2$) verwaarlozen we ook. De statische druk aan de uitgang van de straalbuis is gelijk aan de atmosferische druk, omdat het water nu in open lucht is. Dit stellen we P_{atm} . De snelheid die het water heeft aan het uiteinde van de straalbuis stellen we $v_{straalbuis}$. Hieruit volgt dus:

$$\begin{aligned} P &= P_{atm} + \frac{1}{2} \rho_{water} v_{straalbuis}^2 \\ v_{straalbuis} &= \sqrt{\frac{2(P - P_{atm})}{\rho_{water}}} \end{aligned} \quad (2.3)$$

Als we nu vergelijkingen (2.2) en (2.3) invullen in (2.1) krijgen we voor de stuwkracht:

$$F_{stuw} = 2(P - P_{atm})A \quad (2.4)$$

De druk verandert naarmate de vlucht vordert. De druk kunnen we bepalen afhankelijk van de hoeveelheid water die is uitgestoten. We gebruiken de adiabatische expansie van lucht⁵.

$$P V^\gamma = P_0 V_0^\gamma \quad (2.5)$$

⁵ Wikipedia contributors. (2020a, augustus 31). Adiabatic process. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/Adiabatic_process

$$P = P_0 \left(\frac{V_0}{V} \right)^\gamma$$

Hierin zijn P_0 en V_0 de druk in de fles en het volume van de lucht vlak voor de lancering. P en V zijn de druk in de fles en het volume van de lucht op een willekeurig tijdstip na de lancering. γ is de adiabatistische index en die nemen we op $\frac{7}{5}$. Dit is de adiabatistische index voor droge lucht bij 20°C.

Luchtweerstandskracht

De luchtweerstandskracht wordt gegeven door de formule:

$$F_{lw} = \frac{1}{2} C_w \rho A v |v| \quad (2.6)$$

We nemen $v|v|$ i.p.v. v^2 zodat het teken voor de richting behouden blijft. De C_w voor de raket is niet bekend. Deze moet namelijk experimenteel bepaald worden in een windtunnel. Voor het model is een C_w van 0.5 gekozen, omdat we denken dat dit een redelijke waarde is. Zodra er daadwerkelijk gevlogen is zullen we proberen een betere C_w te bepalen.

Zwaartekracht

Voor de zwaartekracht is gegeven:

$$F_z = mg \quad (2.7)$$

Als zwaartekrachtsversnelling nemen we 9.80665 m/s² volgens de *Standaard Zwaartekracht*⁶. De massa van de raket hebben we gemeten op 600g.

Totaalvergelijkingen

Op elk moment van de vlucht worden de krachten op de raket beschreven door de volgende vergelijkingen (van de grond af wordt als positieve richting gedefinieerd).

$$F_{som} = \begin{cases} F_{stuw} - F_z - F_{lw} & : \text{als er zich nog water in de raket bevindt} \\ -F_z - F_{lw} & : \text{als al het water opgebruikt is} \end{cases} \quad (3.1)$$

$$F_{stuw} = 2(P - P_{atm})A_{straalbuis} \quad (3.2)$$

$$F_z = mg \quad (3.3)$$

$$F_{lw} = \frac{1}{2} C_w \rho_{lucht} A_{frontaal} v |v| \quad (3.4)$$

$$P = P_0 \left(\frac{V_0}{V} \right)^\gamma \quad (3.5)$$

$$m = m_{droog} + m_{water} \quad (3.6)$$

$$m_{water} = V_{water} \rho_{water} \quad (3.7)$$

$$\frac{dV_{water}}{dt} = A_{straalbuis} v_{water \text{ t.o.v. straalbuis}} \quad (3.8)$$

Het meten van de vlucht

Het verrichten van metingen aan de vlucht van de raket is nuttig, omdat daarmee bepaald kan worden op welk moment de parachute open moet. Sommige grootheden, zoals snelheid, kunnen niet direct worden gemeten maar moeten worden afgeleid uit andere grootheden.

⁶ Wikipedia contributors. (2020d, november 23). *Standard gravity*. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/Standard_gravity.

De grootheden die we uit de vlucht van de raket willen halen zijn:

- Versnelling
- Snelheid
- Hoogte
- Oriëntatie

Sommige grootheden kunnen op meerdere manieren bepaald worden.

Het bepalen van de snelheid

Uit de versnelling is het mogelijk de snelheid te bepalen met behulp van integraalrekening.

$$\Delta v = \int a(t) dt$$

Hieruit volgt dat de formule voor de snelheid afgeleid kan worden door een continue formule voor de versnelling te integreren. Tijdens het meten van de vlucht hebben we echter niet te maken met een continue formule voor de versnelling.

Tijdens de vlucht wordt de versnelling gemeten, en met die meting wordt bijgehouden wanneer die meting is gedaan. Op die manier kan uit het verschil in tijd tussen twee metingen toch de snelheid worden afgeleid.

Tabel 1 Voorbeeld van de versnelling a op tijdstip t .

$t \text{ (ms)}$	$a \text{ (ms}^{-2}\text{)}$
0	0
100	2
200	4

Nu kunnen we numeriek de snelheid bepalen.

$$v = v + a * \Delta t \quad (4.1)$$

Het bepalen van de hoogte

De hoogte is een cruciale grootheid om rekening mee te houden tijdens de vlucht van de raket. Er zijn twee manieren waarop de hoogte bepaald kan worden.

Integreren van de snelheid

De hoogte kan bepaald worden door het integreren van de snelheid. Deze techniek is de dubbele integratie van de versnelling (zie *Het bepalen van de snelheid* op pagina 9).

$$\Delta x = \int v(t) dt = \iint a(t) dt$$

Net als bij de bepaling van de snelheid is het niet mogelijk de formule voor de snelheid te integreren, omdat de metingen tijdens de vlucht niet continu zijn. We gebruiken dus weer de numerieke rekenmethode.

$$x = x + v * \Delta t \quad (4.2)$$

Zo kan uit alleen de meting van de versnelling zowel de snelheid als de hoogte bepaald worden.

Berekening met de luchtdruk

De hoogte kan ook berekend worden met behulp van de gemeten luchtdruk en de barometrische formule⁷.

$$P = P_b * \left(\frac{T_b}{T_b + L_b * (h - h_b)} \right)^{\frac{g * M}{R * L_b}}$$

In deze formules zijn de symbolen gedefinieerd als:

P_b : Referentiedruk in Pa

T_b : Referentietemperatuur in K

L_b : Temperatuurafname in K/m

h : Hoogte waarop de druk berekend wordt in m

h_b : Referentiehoogte in m

R : Gasconstante: $8,3144598 \text{ Jmol}^{-1} \text{ K}^{-1}$

g : Standaard zwaartekrachtversnelling op aarde: $9,80665 \text{ ms}^{-2}$

M : Molaire massa van de lucht van de aarde: $0,0289644 \text{ kg} * \text{mol}^{-1}$

De referentiedruk wordt voor de vlucht ingesteld op de druk die gemeten wordt als de raket aan de grond is. Als referentietemperatuur wordt 15°C genomen. Als temperatuurafname wordt -0.0065 K/m genomen. Beide waarden zijn volgens de Standaardatmosfeer (ISA - 1976)⁸ voor een hoogte van 0 m. We nemen aan dat de temperatuurafname niet zal veranderen tijdens de vlucht van de raket.

Deze formule kunnen we herschrijven om het hoogteverschil te krijgen:

$$\Delta h = \frac{T_b \left(\left(\frac{P}{P_b} \right)^{\frac{R * L_b}{g * M}} - 1 \right)}{L_b} \quad (4.3)$$

We kunnen nu de hoogte van de raket berekenen aan de hand van een vastgestelde referentiedruk P_b een gemeten druk P .

Als we alle waarden (m.u.v. de druk en de referentiedruk) invullen en afronden op significantie 4 (de barometer die wij gebruiken meet in hPa nauwkeurig, dus als de druk in de orde grootte 10^5 is, dan is de significantie 4 (100 000)) dan krijgen we de volgende formule:

$$\Delta h = 44330 * \left(1 - \left(\frac{P}{P_b} \right)^{0.1903} \right) \quad (4.4)$$

Dit is ook de formule die geïmplementeerd is in de Adafruit BMP280 softwarelibrary⁹ die gebruikt wordt om de barometer aan te sturen.

⁷ Wikipedia contributors. (2020b, augustus 31). Barometric formula. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/Barometric_formula

⁸ Wikipedia contributors. (2020c, november 23). International Standard Atmosphere. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/International_Standard_Atmosphere

⁹ Adafruit_BMP280_Library – https://github.com/adafruit/Adafruit_BMP280_Library/blob/master/Adafruit_BMP280.cpp: float readAltitude() (line 373).

Ontwerp en realisatie

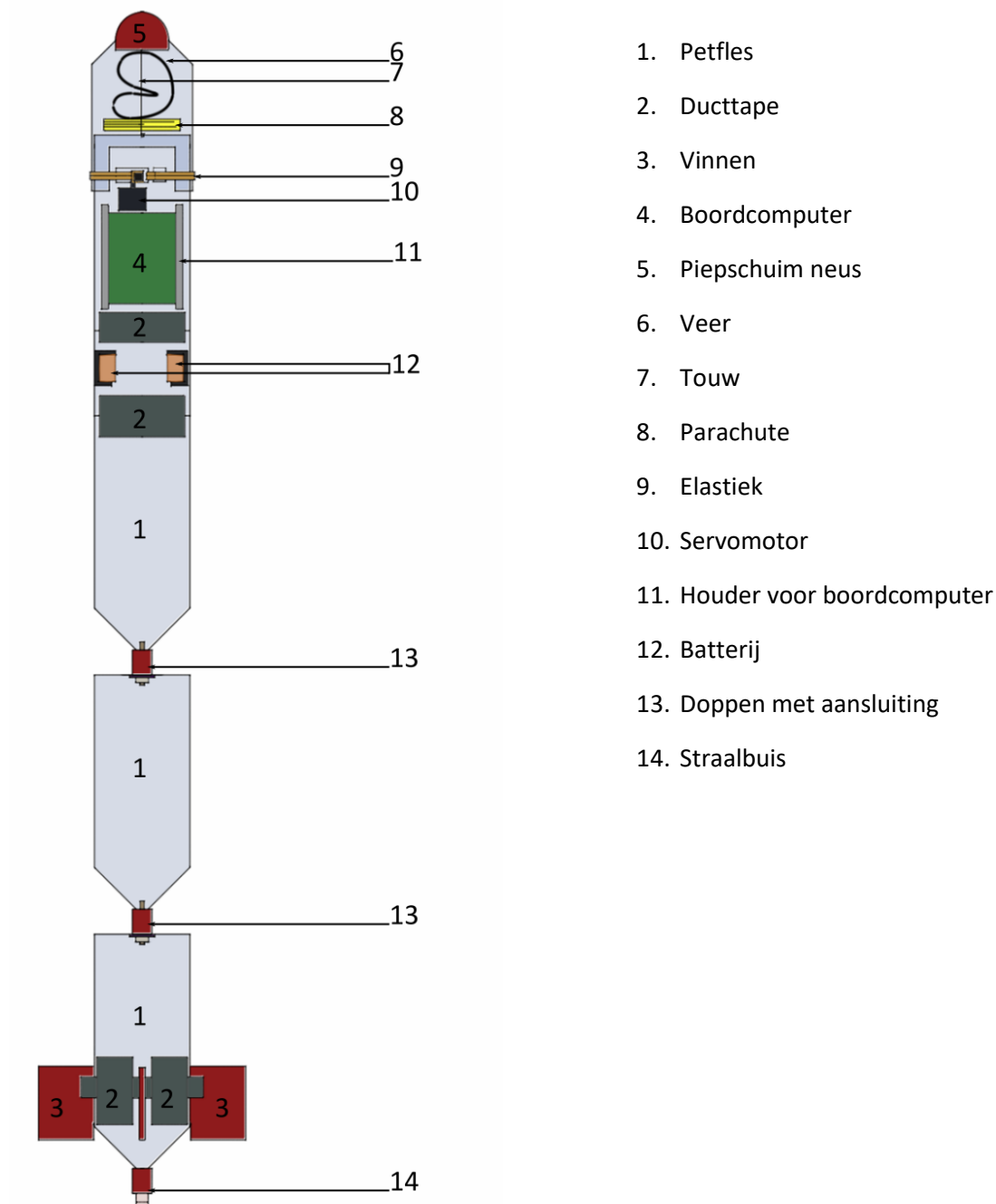
In dit deel van het verslag wordt per onderdeel (Raket, boordcomputer, lanceerplatform en het model) beschreven wat er moet gebeuren om de waterraket te bouwen.

De raket

De raket moet voldoen aan meerdere eisen: de raket moet herbruikbaar zijn en zowel makkelijk als goedkoop te bouwen zijn, en vervolgens nog van de grond kunnen komen natuurlijk. De waterraket bestaat uit drie aaneengesloten flessen, met daarop een boordcomputer met parachute.

Het ontwerp

Figuur 5 is de ontwerptekening van de raket.



Figuur 5 Ontwerptekening van de raket.

Als basis zijn drie 1.5 liter flessen op elkaar gemonteerd (Figuur 6), hierin zal het water zitten en zal de lucht worden gepompt om de raket voort te stuw. De flessen moeten stevig, luchtdicht en drukbestendig aan elkaar worden bevestigd. Door de doppen en onderkant van de flessen is een gat geboord en een draadstang gehaald. Hier kan de lucht en het water doorheen. Aan elk uiteinde houden een moeren, een sluitring en een rubber ring, die om de draadstang zitten, de flessen stevig en luchtdicht aan elkaar, afgebeeld in Figuur 7 en Figuur 8.



Figuur 6 Drie verbonden flessen.



Figuur 7 Flesverbinding met draadstang en moeren, tussen de middelste ringen komt rubber en de dop van de ene fles komt tegen de onderkant van de andere fles, zoals is te zien in Figuur 8

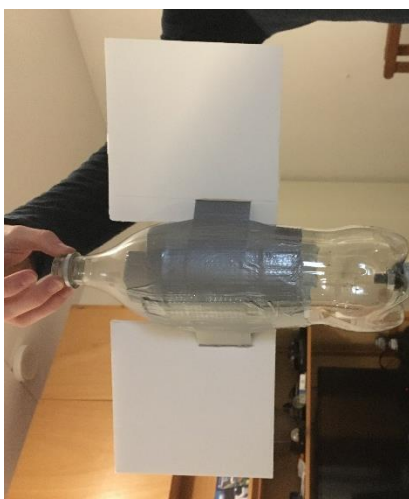


Figuur 8



Figuur 9 Straalbuis

Aan het uiteinde van de raket komt de straalbuis, hieruit komt het water dat de raket voortstuwt. Ook deze moet luchtdicht en drukbestendig worden gemonteerd en aansluiten op het lanceerplatform. Een kraanstuk is verwerkt in de dop van de laatste fles, maar een normale petfles dop bleek te weinig ruimte te hebben hiervoor; de dop kon niet meer helemaal op de fles worden gedraaid. Daarom is uiteindelijk de dop van een terpentinefles gebruikt; deze dop is een stuk hoger en zo is er dus ruimte voor het kraanstuk. Met rubber en lijm is de verbinding luchtdicht gemaakt.



Figuur 10 Vinnen, 2 van de 4 bevestigd.



Figuur 11 Top van de raket. Van links naar rechts: Boordcomputercompartiment, batterijcompartiment, fles 3.

Op de onderste fles zitten vier vinnen (Figuur 10), deze stabiliseren de raket tijdens de vlucht. De vinnen moeten zo licht mogelijk zijn. Ze zijn vierkant met een afmeting van 15 bij 15 centimeter.

Boven op de raket komen de flessen met boordcomputer en parachutesysteem. De omhulling is als volgt opgebouwd:

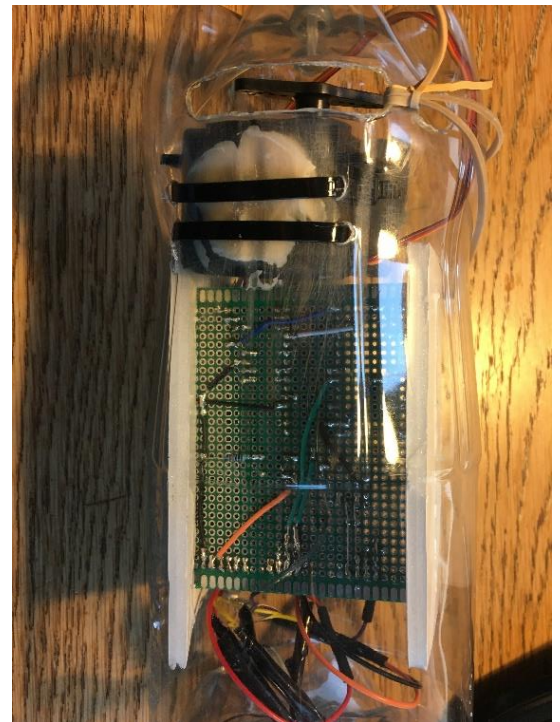
Boven op de derde fles is een halve onderkant van een fles ondersteboven met ducttape gemonteerd. Hierin komen de batterijen. Deze zitten met ducttape vast aan de binnenwand van de fles. Door een gat in de top van de fles gaat de bedrading van batterij naar boordcomputer, zie Figuur 11.

Hier bovenop wordt een fles met de top eraf ondersteboven bevestigd. Hierin komt de boordcomputer met servomotor (Figuur 12). De boordcomputer is met stukken foamboard in de fles geklemd. Daarboven is de servomotor met montagekit en tie-wraps vastgebonden. Door een spleet in de fles kan de servomotor uitsteken. Daar kan een elastiek aan vast worden gehaakt, die op gelijke hoogte is vastgebonden aan de fles.

Als laatste komt daar bovenop de neus van de raket, bestaande uit de bovenste helft van een fles met de punt afgesneden. Op de plek van de punt is als punt een half piepschuim ei geklemd. In de neus van de raket zitten de parachute en de veer. Beide zijn aan een lijn vastgeknoopt waarvan de uiteinden aan het neusstuk en de rest van de raket zijn bevestigd. Het neusstuk past over het boordcomputercompartiment, en heeft een gat groot genoeg om de spleet waar de servomotor uitsteekt vrij te houden. Het neusstuk wordt vervolgens, met de gespannen veer erin, op het boordcomputercompartiment geduwd en vastgehouden door de elastieken die om de fles heen aan de servomotor worden gehaakt. Zodra de servomotor wordt geactiveerd schieten de elastieken los, en daarbij duwt de veer het neusstuk met de parachute los. Zo ontplooit de parachute en landt de raket zacht.¹⁰



Figuur 13 Neusstuk



Figuur 12 Boordcomputercompartiment, met boordcomputer en daarboven servomotor.



Figuur 14 Top van de raket in zijn geheel. Neusstuk zit vast door de elastieken, met erin de parachute opgevouwen en daarboven de veer.

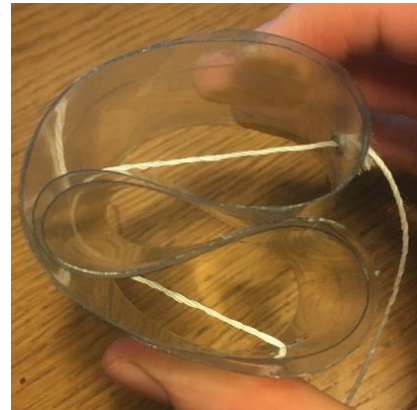
¹⁰ USWaterRockets. (2013, 12 februari). *Build a Parachute Deploy System for Water Rockets Tutorial* [Video]. YouTube. <https://www.youtube.com/watch?v=vqTKXXpD9IU>

De parachute is gemaakt van een vuilniszak¹¹, gevouwen en geknipt in de juiste vorm (Figuur 15). Op elke hoek is ducttape geplakt, daardoor een gat geknipt of geprikt en vliegertouw gehaald en vastgeknoopt. Deze acht touwtjes komen bij elkaar en worden tot een lus geknoopt. Hieraan komt het touw dat aan de veer en het neusstuk vastzit en het touw dat aan de rest van de raket is gebonden.

De veer (Figuur 16) bestaat uit het uitgeknipte middenstuk van twee flessen. Door deze in een C-vorm te vouwen fungeert hij als veer. Door de veer is een touw heen gehaald dat aan de parachute en het neusstuk zit.



Figuur 15 Parachute prototype. Rechts uitgeknipt en links met touw en lus.



Figuur 16 Gespannen veer in C-vorm

De realisatie

Lijst met benodigde spullen:

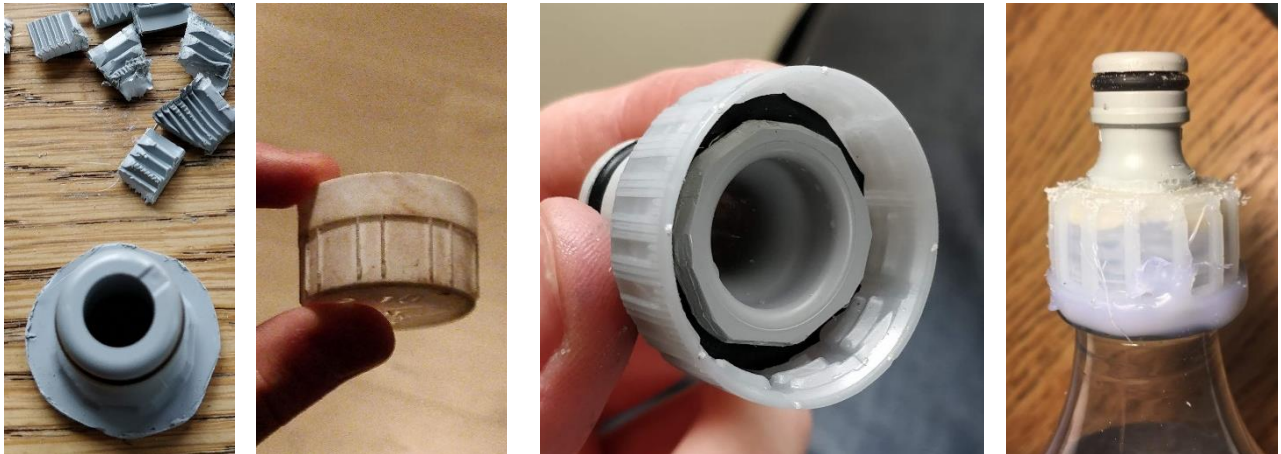
- 5+ petflessen 1.5 liter
- 2x dop petfles
- 1x dop terpentinefles
- Rubber van een fietsbinnenband
- 4x moer
- 2x holle draadstang
- 4x sluitring
- Plastic vuilniszak
- Touw
- Montagekit
- 2x tie-wrap
- 2x elastiek
- Piepschuim ei
- Foamboard
- Ducttape
- 1x Stanley mes
- 1x Kniptang
- 1x Schaar
- Lijmpistool
- Boor met passende boorstukken, eventueel soldeerbout

Alle onderdelen bij elkaar exclusief gereedschap kosten ongeveer € 20,-.

¹¹ SG WaterRockets. (2017, 30 juli). *Large DIY water rocket parachute* [Video]. YouTube. <https://www.youtube.com/watch?v=H-SNqZcMnX8>

In het midden van de onderkant van twee flessen en in het midden van twee doppen is een gat geboord. Dit gat is bewerkt met een mes totdat de draadstang er exact doorheen paste. Door de hals van de fles gaat de draadstang naar binnen en door het gat gestoken. Dan gaat aan de andere kant van het gat de dop over de draadstang en wordt op dezelfde wijze met sluitring, rubber en moer gemonteerd. Op de dop wordt een andere fles vastgedraaid. Dit proces wordt herhaald totdat drie flessen aan elkaar zitten.

Dan de straalbuis. De terpentinedop had een kinderslot, die is eerst verwijderd. Dan is met de kniptang de bovenkant van het kraanstuk bewerkt tot zijn diameter klein genoeg was om in de dop te passen. Vervolgens is met boren en knippen een gat gemaakt in de terpentinedop tot het kraanstuk erin paste. Met een schaar zijn uit een binnenband van een fiets rubber ringen geknipt en om zowel het kraanstuk als tussen kraanstuk en dop geplaatst, deze vulden gaten tussen de verbinding van dop en kraanstuk. Toen bij testen dit niet luchtdicht bleek, is tot slot het kraanstuk met een lijmpistool in de dop gelijmd en de dop luchtdicht gelijmd op de fles. Na testen bleek dit met voldoende lijm wel luchtdicht te zijn. Overblijvend lijm boven op het kraanstuk is weggehaald met een mes en schuurpapier.



Figuur 17 De realisatie van het kraanstuk. Van links naar rechts:

Bewerkte kraanstuk, terpentinedop, kraanstuk prototype in dop met rubber, dop met kraanstuk luchtdicht gemaakt met lijmpistool

De vinnen zijn vier 15x15 centimeter vierkante stukken foamboard uitgesneden met een stanleymes. Vervolgens zijn met het mes gleuven gesneden in de vinnen waardoor ducttape past. Door de gleuven is dubbelgevouwen ducttape gehaald, en dit is strakgetrokken en vast gemaakt aan de raket met nog meer ducttape.

De flessen voor de top van de raket waren makkelijk in vorm te snijden met het Stanley mes. Een enkele laag ducttape was genoeg om de flessen aan elkaar te binden. De batterijclips zijn aan de binnenwand geplakt met naar buiten dubbelgevouwen ducttape. De foamhouders voor de boordcomputer zijn met het mes uitgesneden, stuk voor stuk tot ze net pasten met de boordcomputer vast, die door gesneden gleuven in het foam paste.

Een gat is geboord door de top van het batterijcompartiment waardoor bedrading kan. De soldeerbout bleek ook uitstekend te werken voor het maken van gaten in plastic, maar het inademen van de gassen die vrijkomen is ongezond. Boven de boordcomputer is de servomotor geplaatst.

De gaten voor de tiwrap zijn met de soldeerbout gemaakt, net als de spleet voor de servo en het gat voor de elastieken. De gaten door de veer zijn met de boor gemaakt en er is vliegertouw doorheen gehaald. Dit touw, gebonden aan de lus van de parachute, zit vast aan de raket en aan het neusstuk door hem door twee met de soldeerbout gemaakte gaten te knopen (Figuur 19). Bij het neusstuk is dit gat van buiten afgedicht met tape.



Figuur 18 Prototype van een vin.



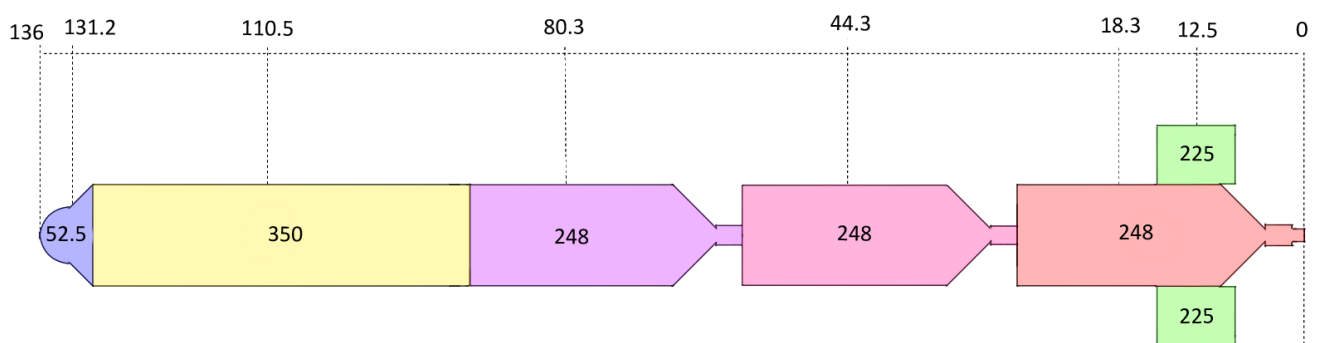
Het neusstuk is volledig in vorm gesneden met het mes, en heeft aan de onderkant nog een aantal sneetjes (te zien in Figuur 13) gekregen zodat het makkelijker over het boordcomputercompartiment past. Het piepschuim ei is met een mes doormidden gesneden, en is gebruikt als neus door hem te forceren in het gat in het neusstuk.

De parachute is uitgeknipt met een schaar, en de gaten door de hoeken zijn geboord. Het touw is met een simpele dubbele knoop door de gaten gehaald, en over de knopen is weer eens tape geplakt.

Om te zorgen dat de raket stabiel vliegt moet het drukmiddelpunt onder het zwaartepunt zitten. In Figuur 20 staan de oppervlakten per onderdeel van de raket en de afstand van de drukmiddelpunten van de afzonderlijke onderdelen tot de referentielijn. Hoe we deze bepaald hebben staat bij *Bepaling van het drukmiddelpunt* op pagina 6. Uiteindelijk hebben we het drukmiddelpunt bepaald op 55 cm vanaf de referentielijn. Het zwaartepunt hebben we bepaald op 82 cm van de referentielijn. De gewichtsverdeling van de raket is dus goed.



Figuur 19 Verbinding touw aan raket.



Figuur 20 Oppervlakteverdeling voor bepaling drukmiddelpunt. De waarden in het figuur zijn oppervlakten in cm² en de waarden aangegeven met de stippellijn is de afstand in cm vanaf de referentielijn tot aan het drukmiddelpunt van de raketonderdelen.

De boordcomputer

De boordcomputer is een cruciaal onderdeel van de raket. Deze doet metingen aan de vlucht en moet op tijd de parachute openen. De boordcomputer moet in de raket en dus in een 1.5L fles passen. De software die op de boordcomputer draait moet de metingen verrichten en moet die metingen ook verwerken.

De hardware

Het ontwerp

De boordcomputer moet een aantal taken vervullen, namelijk:

- Het bepalen wanneer de parachute open moet,
- Het openen van de parachute,
- Metingen verrichten aan de vlucht,
- Het opslaan van deze metingen zodat ze later ingezien kunnen worden.

Als hart van de boordcomputer wordt een Arduino® Nano gebruikt. Arduino is een open-source elektronicaplatform gebaseerd op makkelijk te gebruiken hardware en software.¹² Deze kan geprogrammeerd worden om de taken van de boordcomputer uit te voeren.

Er worden twee sensoren gebruikt om grootheden uit de omgeving te meten, namelijk:

- Een versnellingsmeter/gyroscop, om de versnelling en hoeksnelheid te meten. De MPU6050 van InvenSense wordt gebruikt.
- Een barometer, om de hoogte te meten. De BMP280 van Bosch Sensortec wordt gebruikt.

Deze sensoren zijn gekozen omdat ze goedkoop zijn en omdat er veel open-source softwarelibraries voor zijn.

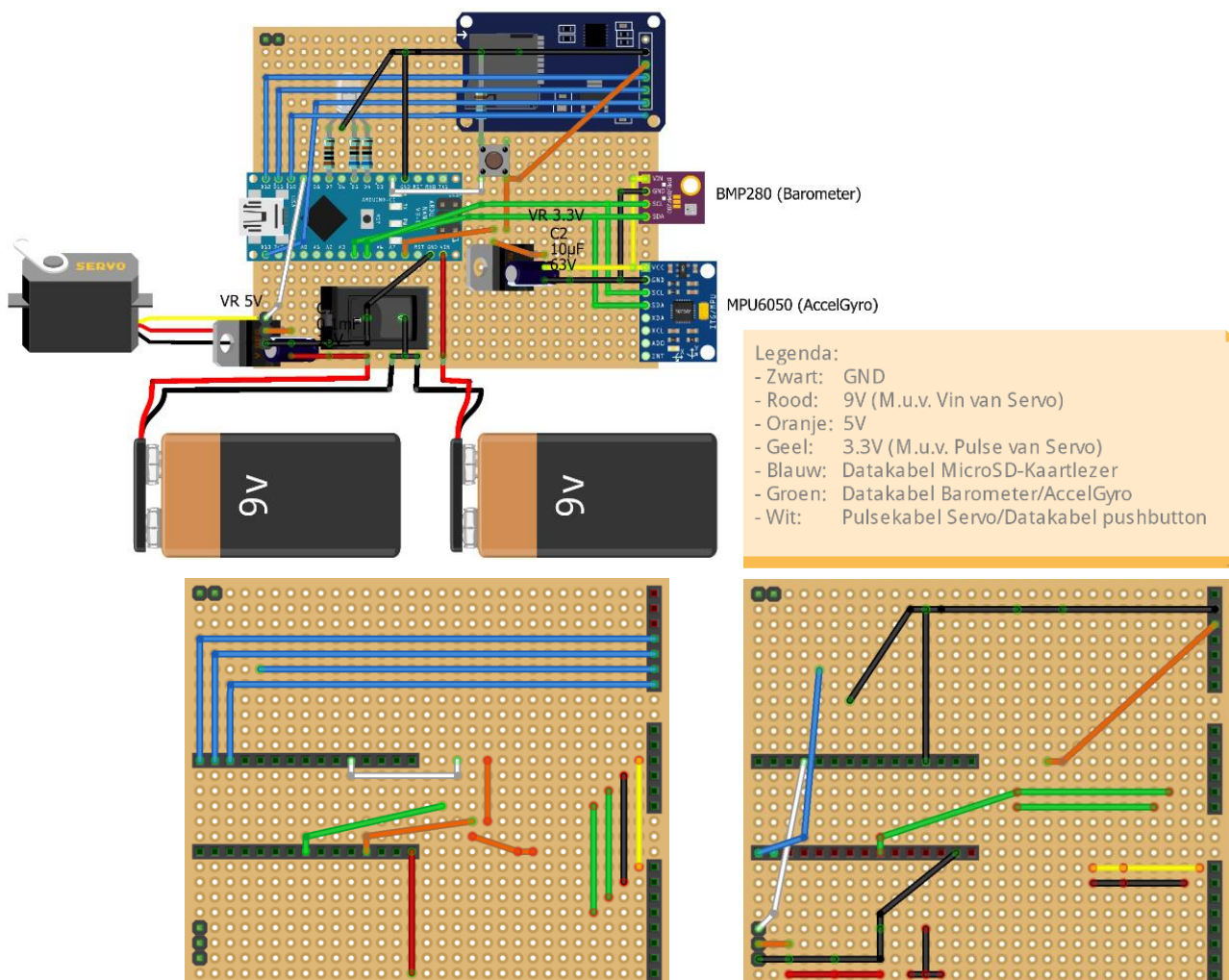
Daarnaast zijn er nog een aantal andere onderdelen nodig, namelijk:

- Een Micro SD-kaartlezer, zodat de meetgegevens opgeslagen kunnen worden.
- Een servomotor om de parachute te ontplooien.

De batterijen moeten genoeg stroom kunnen leveren en licht zijn.

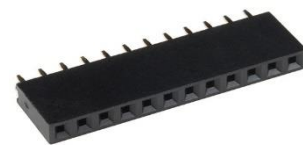
¹² Arduino. (z.d.). What is Arduino? Geraadpleegd op 25 november 2020, van <https://www.arduino.cc/en/Guide/Introduction/>

In Figuur 21 volgt de ontwerptekening van de boordcomputer op een prototyping board.



Figuur 21 Ontwerp van de hardware van de boordcomputer. Van linksboven naar rechtsonder: volledig ontwerp, legenda, bovenkant alleen bedrading, onderkant alleen bedrading. Afbeelding gegenereerd met Fritzing.


Alle onderdelen worden op een prototyping board gesoldeerd die in de raket past, zo passen alle onderdelen in een compacte ruimte. Sensoren en andere grote onderdelen worden niet direct op het bord gesoldeerd. Er worden headerpins (zie Figuur 22 als voorbeeld) op het bord gesoldeerd zodat deze onderdelen later gemakkelijk ook weer verwijderd kunnen worden zonder ze te desolderen.





Figuur 22 Headerpins (female).
Bron: SparkFun Electronics.

Alle onderdelen worden onderling verbonden met draadjes. De uiteinden hiervan kunnen door de gaten in het bord worden gestoken en aan de onderkant vast worden gesoldeerd. Hier kunnen ze ook aan de headerpins vast worden gesoldeerd. Om te voorkomen dat draadjes elkaar overlappen worden sommige bovenlangs en sommige onderlangs getrokken (zie Figuur 21 ter verduidelijking).

De stroom wordt geleverd door twee 9V batterijen die allebei een ander deel van het circuit voeden. Het circuit is in tweeën geknipt omdat één batterij niet genoeg stroom kan leveren om het hele circuit te voeden. De minpool van beide batterijen gaat door een aan/uitknop zodat het circuit aan- en uitgezet kan worden. De stroom gaat door verschillende spanningsregelaars (in Figuur 21

aangegeven met ) om te zorgen dat alle componenten de juiste hoeveelheid spanning krijgen. Zo heeft de Arduino 9V nodig, hebben de servo en de Micro SD-kaartlezer 5V nodig en hebben de sensoren 3.3V nodig. De Arduino bevat een 5V spanningsregelaar, dus om de spanning voor de sensoren omlaag te brengen is enkel een 3.3V spanningsregelaar nodig.

De knop (in Figuur 21 aangegeven met ) en de RGB LED (in Figuur 21 aangegeven met ) zijn aangebracht zodat de boordcomputer tussen verschillende modi kan wisselen. De kleur van de LED geeft aan in welke modus de boordcomputer zich bevindt. Welke modi er zijn en wat ze inhouden wordt verder uitgewerkt onder het kopje *De software* op pagina 21.

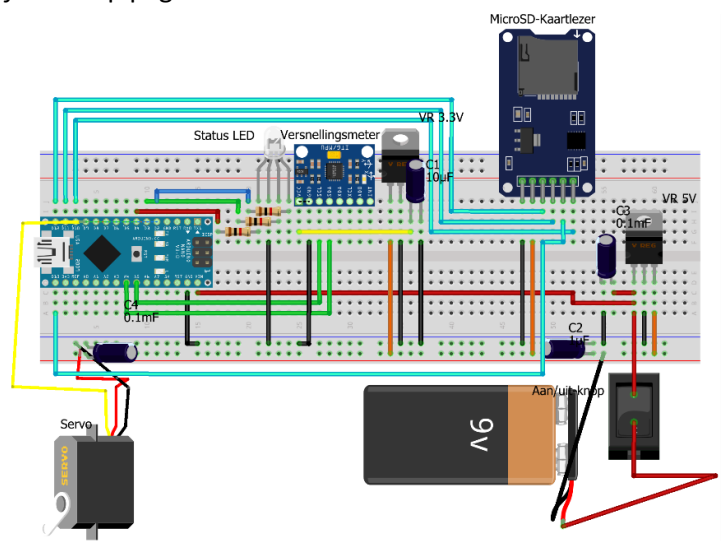
De realisatie

Er is begonnen met het bouwen van het circuit op een breadboard. Dit is gedaan om het circuit te kunnen testen zonder dat solderen nodig was. Hiervoor is er eerst een ontwerp gemaakt (Figuur 23).

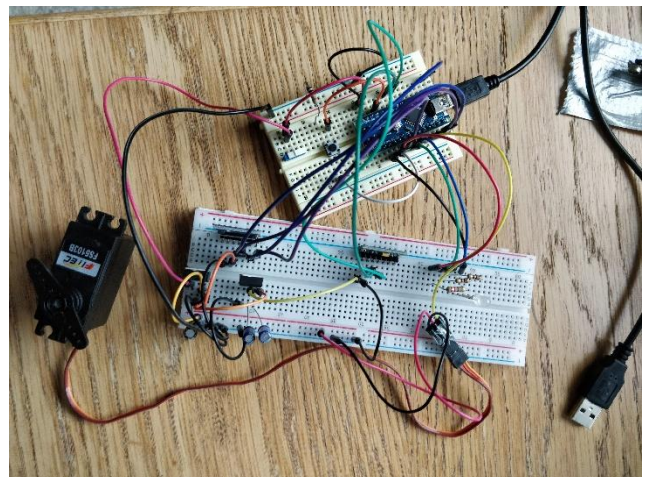
In deze vorm past de boordcomputer niet in de raket. Hij moet dus op een kleinere en permanentere manier geassembleerd worden op een prototyping board. Het uiteindelijke ontwerp hiervan staat in Figuur 21.

Lijst met benodigde spullen¹³:

- 1x Arduino® Nano
- 1x MPU6050
- 1x BMP280
- 1x Micro SD-kaartlezer
- 1x servomotor
- 2x batterijclip
- 1x prototypingboard
- 51x headerpin (female)
- 5x headerpin (male)
- 1x weerstand 1K Ω
- 2x weerstand 6.8K Ω
- 1x weerstand 10K Ω
- 1x condensator 0.1mF
- 1x condensator 10 μ F
- 1x spanningsregulator 5V
- 1x spanningsregulator 3.3V
- 1x Aan/uitknop



Figuur 23 Ontwerp van de boordcomputer op een breadboard. Afbeelding gegenereerd met Fritzing.



Figuur 24 De boordcomputer geassembleerd op een breadboard.

¹³ Alle onderdelen zijn besteld bij bitsandparts.nl en hobbyelektronica.nl. Hartelijk dank aan Etienne Meulen voor het leveren van de RGB LED en de weerstanden.

- 1x pushbutton
- 1x RGB LED
- Genoeg draad
- Soldeerbout en soldeer
- Kabelstriptang en kniptang
- Multimeter

Alle onderdelen bij elkaar (exclusief gereedschap) kosten ongeveer € 30,-.

Bij het assembleren is als volgt te werk gegaan.

1. Als eerste zijn de headerpins op het bord gesoldeerd.
2. Vervolgens is het circuit voor de stroomlevering – bestaande uit de batterijdraden, de aan/uitknop en de draden naar de Arduino en de servo – aangelegd.
3. Dan zijn alle kleine onderdelen, zoals de LED, condensatoren, de knop en weerstanden aangebracht.
4. Daarna zijn de grotere onderdelen aangelegd, zoals de spanningsregelaars.
5. Vervolgens is alle bedrading aangelegd.

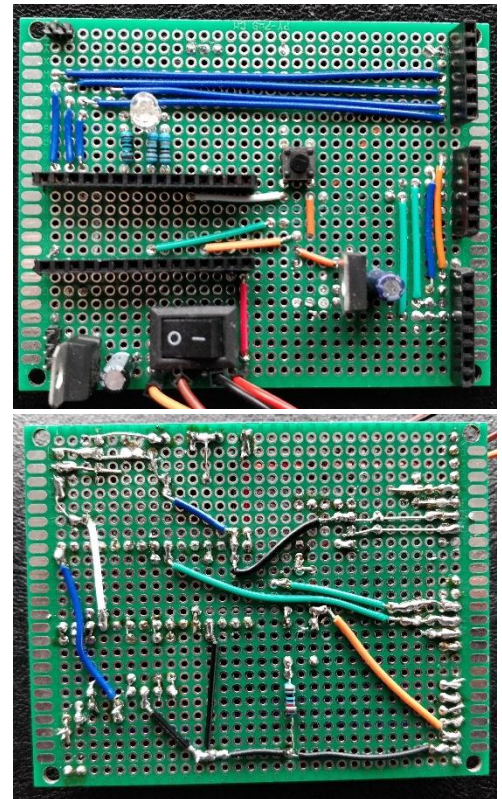
Tijdens het assembleren is het belangrijk om alle verbindingen te testen met een multimeter, om er zeker van te zijn dat de gesoldeerde verbindingen ook functioneel zijn.

Uiteindelijk wordt de boordcomputer in de raket geklemd. Meer informatie hierover en over het parachute-ontplooiingssysteem staat bij *De raket* op pagina 11.

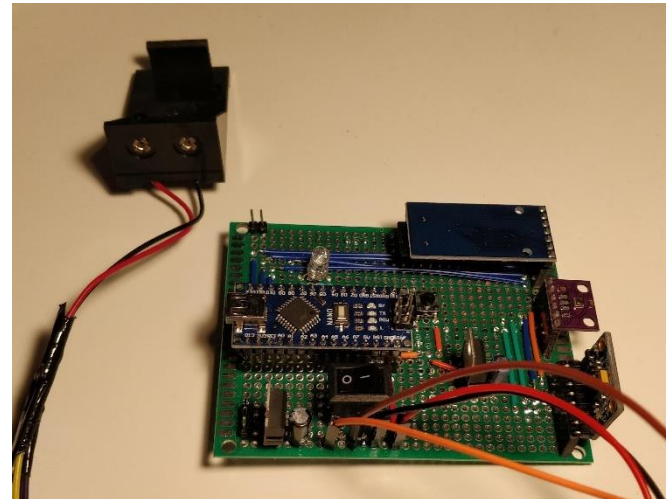
Moeilijkheden en problemen

Tijdens het testen van de boordcomputer in breadboard-vorm werd het duidelijk dat één 9V batterij niet genoeg stroom kon leveren voor het hele circuit. Het eerste plan was om twee 9V batterijen parallel te schakelen zodat de hoeveelheid stroom die geleverd kon worden verdubbeld werd, maar dit werd afgeraden door een geconsulteerde expert¹⁴ omdat twee ongelijke batterijen elkaar kunnen hinderen. Er werd aangeraden het circuit in tweeën te knippen, zodat één batterij de servo voedt, en een andere batterij de rest van het circuit.

Er zijn twee versies gemaakt van de boordcomputer. Waarom er is gekozen om een tweede boordcomputer te maken zal nu worden besproken.



Figuur 25 Boven- en onderkant van de uiteindelijke versie van de boordcomputer.



Figuur 26 De uiteindelijke versie van de boordcomputer, met Arduino en sensoren in de headerpins, en 1 batterijhouder.

¹⁴ Bijzondere dank aan Remy Hanswijk.

De eerste versie is gebouwd op een goedkopere printplaat van lagere kwaliteit. Dit zorgde ervoor dat de verbindingpunten soms loslieten. De uiteindelijke versie is gebouwd op een printplaat van hogere kwaliteit met verbindingpunten die niet op de printplaat liggen maar er doorheen gaan (zie Figuur 27).

Bij de eerste versie zijn er draden die elkaar overlappen. Door de hitte van de soldeerbout kwam het soms voor dat de plastic isolatie van de draden smolt, en als de draden dan overlaptten zorgde dat voor kortsluiting. Bij de uiteindelijke versie zijn er geen overlappende draden.

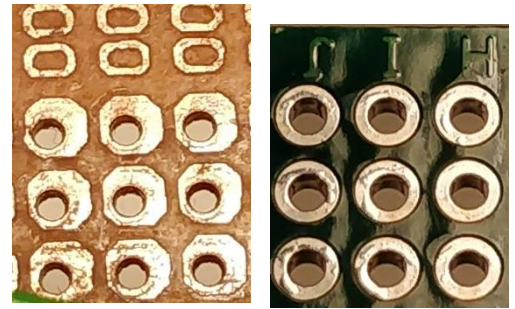
Bij de eerste versie werd er een soldeerbout gebruikt met een te brede punt waardoor het onmogelijk was nette precieze soldeerverbindingen te vormen. Daarnaast werd er vloeibaar soldeerflux gebruikt. Dit zorgde ervoor dat de printplaat erg vuil en vet bleef, ook na grondig schoonmaken. Bij de nieuwe versie is soldeer met een kern van soldeerflux gebruikt¹⁵, zodat de soldeerverbindingen veel schoner zijn.

Verder werkte de eerste versie niet omdat er overal in het circuit punten waren waar kortsluiting plaatsvond. Hierdoor was het niet mogelijk de Arduino verbinding te laten maken met de sensoren. Dit probleem is geprobeerd te verhelpen door de verbindingen in kwestie te desolderen en opnieuw netter te solderen. Tijdens het hersolderen zijn alle verbindingen grondig getest met een multimeter¹⁶ maar de problemen konden niet verholpen worden. Bij dit proces is een expert geconsulteerd¹⁷.

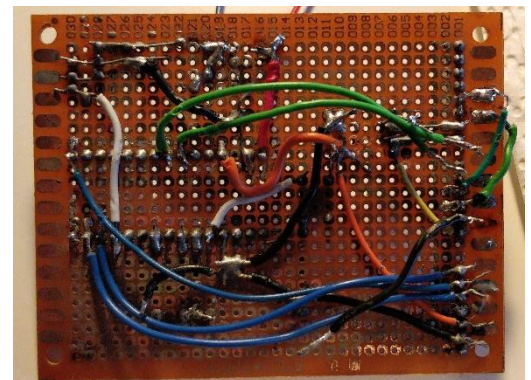
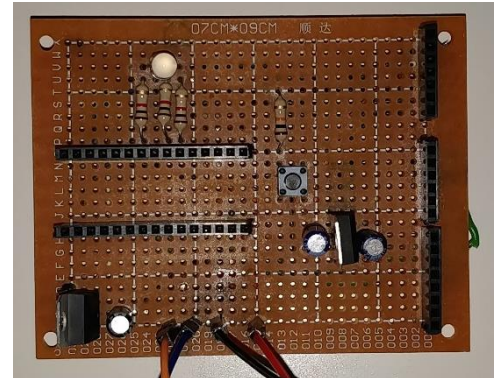
De software

Op de boordcomputer moet software draaien om alles aan te sturen. De software meet gedurende de vlucht continu de versnelling op drie assen, de hoeksnelheid om drie assen en de hoogte die bepaald is met een barometer.

De volledige broncode is te vinden in de appendix (zie *Broncode van de boordcomputer* op pagina 50).



Figuur 27 Vergelijking tussen printplaat eerste versie (links) en uiteindelijke versie (rechts).



Figuur 28 Boven- en onderkant van de eerste versie van de boordcomputer.

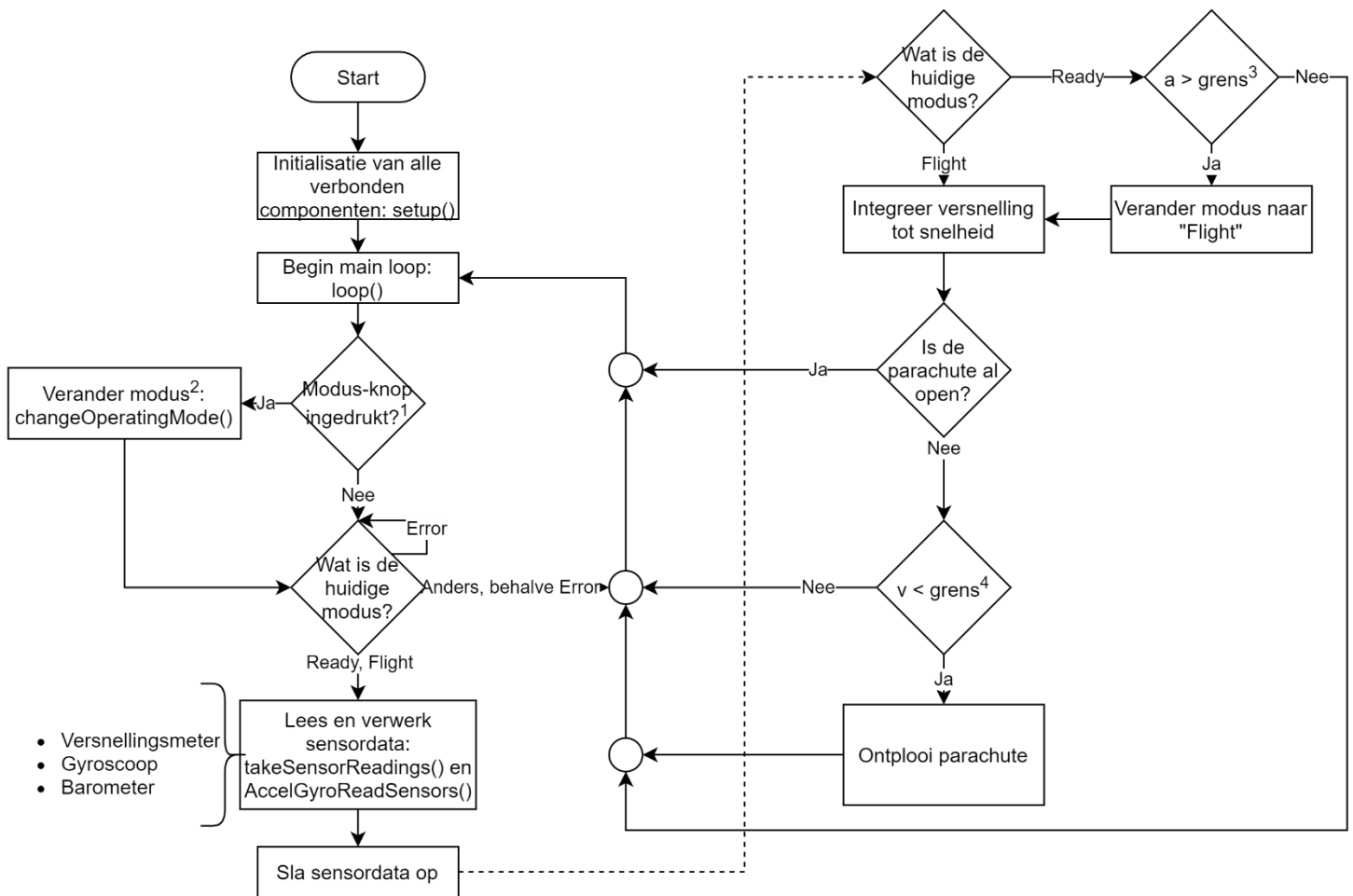
¹⁵ Bijzondere dank aan Remy Hanswijk voor het uitlenen van zijn soldeer.

¹⁶ Bijzondere dank aan Remy Hanswijk voor het uitlenen van zijn multimeter.

¹⁷ Bijzondere dank aan Remy Hanswijk.


Het ontwerp

De structuur van de software is weergegeven in Figuur 29.



Figuur 29 Flowchart van de software die draait op de boordcomputer. Enkele punten aangegeven met superscript zijn hieronder verder toegelicht. Afbeelding gegenereerd met draw.io.

Toelichting bij enkele punten aangegeven in het diagram:

- Met de modus-knop wordt de pushbutton () bedoeld in Figuur 21. Door deze knop voor een seconde ingedrukt te houden kan tussen modi gewisseld worden (zie voor meer informatie over modi ook *Verschillende modi* op pagina 23). De kleur van de LED op de boordcomputer correspondeert met de huidige modus. Deze modi zijn:
 - Not Ready (paars): de boordcomputer is aan het initialiseren.
 - Standby (blauw): de boordcomputer wacht totdat er op de modus-knop wordt gedrukt.
 - Ready (groen): de boordcomputer slaat meetgegevens op en wacht op de lancering.
 - Flight (uit): de boordcomputer slaat meetgegevens op en bepaalt of de parachute open moet.
 - Error (rood): er heeft een fout plaatsgevonden. De boordcomputer moet gereset worden.
- De modi veranderen op de volgende manier bij het indrukken van de pushbutton:
 - Not ready of Error → verandert modus niet
 - Standby → Ready
 - Ready of Flight → Standby

3. Als de verticale versnelling boven een bepaalde drempelwaarde komt wordt de modus overgeschakeld naar Flight. Van dan af aan zal voor de verticale snelheid niet meer een snelheid van 0 m/s worden aangenomen. De snelheid zal worden bepaald door de verticale versnelling te integreren. Ook zal van dan af aan worden gecheckt of de raket zijn hoogtepunt heeft bereikt.
4. Als de bepaalde verticale snelheid (een snelheid omhoog wordt als positief beschouwd) onder een bepaalde drempelwaarde komt wordt aangenomen dat de raket zijn hoogtepunt heeft bereikt, en zal de parachute ontplooid worden. In het hoogtepunt heeft de raket namelijk een verticale snelheid van 0 m/s. Er wordt gebruik gemaakt van een drempelwaarde om rekening te houden met een mogelijke meetfout van de versnelling.

Resetprocedure

Om de boordcomputer te resetten en de data op de SD-kaart te verwijderen moet als volgt te werk worden gegaan. Zet eerst de boordcomputer uit. Houd de pushbutton ingedrukt. Zet de boordcomputer aan, de LED zal nu geel worden. Blijf de pushbutton ingedrukt houden totdat de LED paars wordt. Laat vervolgens de pushbutton los. Als de LED blauw wordt is de resetprocedure correct uitgevoerd. Waarom deze resetprocedure noodzakelijk is wordt uitgelegd onder het kopje *Preventie van gegevensverlies* op pagina 30.

De realisatie

Er zullen hier steeds codesnippets worden weergegeven om te illustreren hoe de code werkt. De volledige broncode is te vinden in de appendix (*Broncode van de boordcomputer* op pagina 50).

Alle broncode is geschreven in C++ omdat dit de taal is die gebruikt wordt voor het programmeren van zogenaamde *embedded systems*. De broncode is op te delen in drie delen: definitie van variabelen, *setup* (waarin alle componenten worden geïnitieerd) en *loop* (de code die constant herhaald wordt; hier vindt het meten plaats). De code zal niet van boven naar beneden besproken worden. De verschillende onderdelen zullen apart besproken worden, zoals het uitlezen en rekenen aan de versnellingsmeter, het uitlezen van de barometer en het bepalen van het hoogtepunt. Bovendien zal niet alle code besproken worden, omdat de functie van grote delen van de code voor de hand ligt.

Definitie pins

Componenten worden aangestuurd via pins van de Arduino. Voor deze componenten moet gedefinieerd worden met welke pin ze werken. Voorbeeld:

```
1. #define PUSHBUTTON_PIN 2 // Pushbutton operates at digital pin D2.
```

Verschillende modi

De boordcomputer heeft verschillende modi waartussen gewisseld kan worden.

```
1. /** Defines different operating modes for the flightcomputer.
2.  * - NotReady      : The flightcomputer is initializing and is not ready to
   record flight.
3.  * - Standby       : The flightcomputer is prepared to start recording.
4.  * - Ready         : The flightcomputer is recording and waiting for launch.
5.  * - Flight        : The flightcomputer has recorded launch and is now
   logging and waiting for apogee.
6.  * - Error         : An error has occurred and the flightcomputer has
   stopped recording.
7. */
```



```

8. enum OperatingMode
9. {
10.     NotReady,
11.     Standby,
12.     Ready,
13.     Flight,
14.     Error
15. };
16.
17. OperatingMode currentMode = OperatingMode::NotReady;

```

Wisselen tussen modi gaat door de pushbutton voor een seconde ingedrukt te houden.

```

1. int counter = 0;
2. // If button is held for 1000ms, change mode.
3. while (digitalRead(PUSHBUTTON_PIN))
4. {
5.     counter += 1;
6.     delay(1);
7.
8.     if (counter > 1000)
9.     {
10.        changeStatusLed(HIGH, LOW, HIGH); // Purple
11.
12.        delay(1000);
13.
14.        switch (currentMode)
15.        {
16.            case Standby:
17.                changeOperatingMode(OperatingMode::Ready);
18.                break;
19.            case Ready:
20.                changeOperatingMode(OperatingMode::Standby);
21.                break;
22.            case Flight:
23.                changeOperatingMode(OperatingMode::Standby);
24.                break;
25.        }
26.        break;
27.    }
28. }

```

De verschillende routines om van modus te wisselen zijn gedefinieerd in void changeOperatingMode(OperatingMode mode). Welke acties plaats moeten vinden in de functie void loop() wordt bepaald aan de hand van de huidige modus.

```

1. void loop()
2. {
3.     // ...
4.     // Routines for different operating modes.
5.     switch (currentMode)
6.     {
7.         case Standby:
8.             // Do nothing
9.             break;
10.        case Ready:
11.        case Flight:

```



```

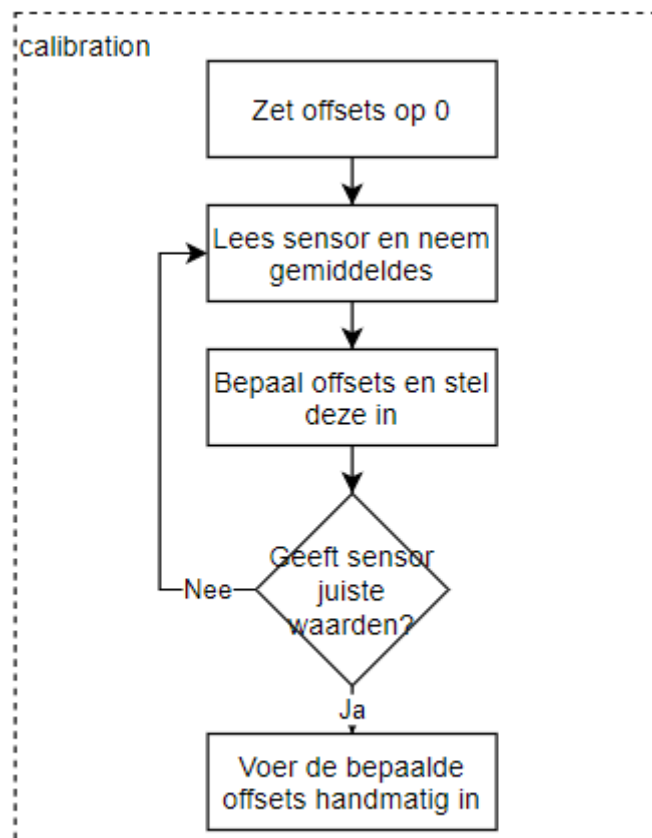
12.      // ...
13.      break;
14.      case Error:
15.          // ...
16.          break;
17.      default:
18.          // ...
19.          break;
20.  }
21. }

```

Versnellingsmeter/gyroscoop

De versnellingsmeter moet eerst gekalibreerd worden. Voor de kalibratie is de kalibratiecode van Luis Ródenas met wijzigingen van Andr s Kun¹⁸ gebruikt. De code is voor deze boordcomputer ook nog lichtelijk aangepast (de volledige broncode is te vinden in de appendix *Broncode van de boordcomputer* op pagina 50). De wijzigingen die zijn gedaan staan bovenaan in de broncode.

De werking¹⁹ van de kalibratie staat beschreven in het volgende schema.



Figuur 30 Werking van de kalibratie van de versnellingsmeter/gyroscoop. Afbeelding gegenereerd met draw.io. Origineel van mjwhite8119.github.io.

¹⁸- Gebruikte code: https://github.com/Protonerd/DIYino/blob/master/MPU6050_calibration.ino

- Originele blogpost: *Arduino Sketch to automatically calculate MPU6050 offsets* -

<https://www.i2cdevlib.com/forums/topic/96-arduino-sketch-to-automatically-calculate-mpu6050-offsets/>

- Informatie over het gebruik van *Calibrating & Optimising the MPU6060* -

<https://wired.chillibasket.com/2015/01/calibrating-mpu6050/>

¹⁹ White, M. (2019, 26 juli). *The MPU6050 Explained*. Programming Robots. Geraadpleegd op 24 oktober 2020, van <https://mjwhite8119.github.io/Robots/mpu6050>

De sensor moet stilliggen. Als eerste worden de zogenaamde *offsets* ingesteld op 0. Deze offsets geven aan hoe veel de gemeten waarden afwijken van de waarde die ze zouden moeten hebben. Ze moeten namelijk 0 m/s^2 zijn in alle richtingen behalve recht naar beneden (recht naar beneden moet het namelijk gelijk zijn aan de zwaartekrachtsversnelling (9.81 m/s^2)) en een hoeksnelheid van $0^\circ/\text{s}$ rond alle assen. Vervolgens wordt de sensor meerdere keren uitgelezen, en van deze waarden wordt het gemiddelde genomen. Er wordt gekeken naar de gemiddelde waarde om meeton nauwkeurigheden van de versnellingsmeter te vermijden. Van hoe veel metingen het gemiddelde genomen wordt, wordt als volgt gedefinieerd in de code:

```
1. int buffersize = 1000;
```

De eerste honderd metingen worden niet gebruikt om meeton nauwkeurigheden veroorzaakt door het net opstarten van de sensor tegen te gaan.

Als de gelezen waarden afwijken van de verwachte waarden met een afwijking (int *acel_deadzone*) groter dan 10, dan worden de *offsets* aangepast. (De functie van *ACCEL_RANGE* in de code hieronder wordt iets verderop uitgelegd).

```
1. // Error allowed.
2. int acel_deadzone = 10;
3.
4. // For vertical acceleration.
5. if (abs(int(16384 / pow(2, ACCEL_RANGE)) - mean_az) > acel_deadzone)
6. // Offset = offset + (gravity - sensor_measured_reading) / 8
7.
   az_offset = az_offset + (int(16384 / pow(2, ACCEL_RANGE)) - mean_az) / acel_
   _deadzone;
8.
9. // For non-vertical acceleration
10. if (abs(mean_ax) > acel_deadzone)
11.   ax_offset = ax_offset - mean_ax / acel_deadzone;
```

De berekende *offset* nadert de gewenste *offset*. Het proces wordt herhaald totdat de gemeten waarden voldoen aan de ingestelde toelaatbare afwijking.

Vervolgens worden de berekende *offsets* ingesteld voor gebruik tijdens de vlucht.

```
1. // AccelGyro calibration
2. accelgyro.setXAccelOffset(-2810);
3. accelgyro.setYAccelOffset(1547);
4. accelgyro.setZAccelOffset(1507);
5. accelgyro.setXGyroOffset(51);
6. accelgyro.setYGyroOffset(27);
7. accelgyro.setZGyroOffset(-82);
```

Bij de kalibratie moet één van de assen de zwaartekrachtsversnelling meten. Het is dus wel noodzakelijk om te specificeren welke van de assen deze versnelling moet meten. Welke as dit moet zijn is als volgt gedefinieerd in de code:

```
1. /** Defines which axis forms a right angle with the ground */
2. // #define ACCEL_X_AXIS_DOWN
3. // #define ACCEL_Y_AXIS_DOWN
```

```
4. #define ACCEL_Z_AXIS_DOWN
```

Op basis hiervan wordt de verticale versnelling als pointer gedefinieerd.

```
1. // Variables to store accelgyro data in.
2. int16_t ax, ay, az;
3. // Vertical acceleration (pointer).
4. int16_t *av;
5.
6. #ifndef ACCEL_X_AXIS_DOWN
7. av = &ax;
8. #elif defined(ACCEL_Y_AXIS_DOWN)
9. av = &ay;
10. #elif defined(ACCEL_Z_AXIS_DOWN)
11. av = &az;
12. #endif
```

Ook tijdens de vlucht is het noodzakelijk om te weten welke as de zwaartekrachtsversnelling meet. Hier moet namelijk rekening mee gehouden worden om de versnelling ten opzichte van de grond te berekenen.

De uitgelezen waarden van de versnellingsmeter/gyroscop moeten eerst verwerkt worden voordat ze bruikbaar zijn. De versnellingsmeter/gyroscop geeft altijd een waarde tussen -32 750 en +32 750. Welke versnelling/hoeksnelheid hiermee correspondeert hangt af van de ingestelde *range*²⁰.

Tabel 2 Range en sensitivity van de MPU6050. LSB = Least Significant Bit; g = zwaartekrachtversnelling; bron: InvenSense.

Gyro Full Scale Range	Gyro Sensitivity	Accel Full Scale Range	Accel Sensitivity
(°/sec)	(LSB/°/sec)	(g)	LSB/g
±250	131	±2	16384
±500	65.5	±4	8192
±1000	32.8	±8	4096
±2000	16.4	±16	2048

De gebruikte range wordt in de code gedefinieerd met:

```
1. /* Accel range
2. * 0 = +/- 2g
3. * 1 = +/- 4g
4. * 2 = +/- 8g
5. * 3 = +/- 16g
6. */
7. #define ACCEL_RANGE 3
8. /* Gyro range
9. * 0 = +/- 250 °/sec
10. * 1 = +/- 500 °/sec
11. * 2 = +/- 1000 °/sec
```

²⁰ TDK InvenSense. (z.d.). MPU-6050 | TDK. Geraadpleegd op 24 oktober 2020, van <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>

```

12. * 3 = +/- 2000 °/sec
13. */
14. #define GYRO_RANGE 0
15.
16. accelgyro.setFullScaleAccelRange(ACCEL_RANGE);
17. accelgyro.setFullScaleGyroRange(GYRO_RANGE);

```

Er is voor de range $\pm 16g$ gekozen omdat de versnelling van een raket tijdens de lancering erg groot is. Er is voor de range $\pm 250^\circ/s$ gekozen omdat de raket tijdens de vlucht niet met een hoge snelheid zal roteren.

De versnelling in de richting van een as kan nu berekend worden met de volgende formule:

$$a_{as} = \frac{a_{gemeten} * g}{16384 * (\frac{1}{2})^{n_{range}}} \quad (5.1)$$

En voor de hoeksnelheid om een as:

$$\omega_{as} = \frac{\omega_{gemeten}}{131 * (\frac{1}{2})^{n_{range}}} \quad (5.2)$$

De zwaartekrachtsversnelling is gedefinieerd in m/s^2 . De versnelling wordt berekend met een precisie van cm/s^2 . In de berekening van de versnelling moet de zwaartekrachtsversnelling dus eerst omgezet worden in cm/s^2 door te vermenigvuldigen met 100.

```

1. ax = int16_t(ax / (16384.0 / pow(2, ACCEL_RANGE)) * GRAVITY * 100);

```

```

1. gx = int16_t(gx / (131.0 / pow(2, GYRO_RANGE)));

```

Dit wordt voor de 3 assen x, y en z gedaan.

Omdat de versnellingsmeter in rust de zwaartekrachtsversnelling meet, moet de zwaartekrachtsversnelling van de verticale versnelling worden afgetrokken om de versnelling ten opzichte van de grond te krijgen.

```

1. (*av) = int16_t((*av) - (GRAVITY * 100));

```

Er is gekozen om geen rekening te houden met de oriëntatie van de raket, omdat ervan uit wordt gegaan dat de raket tijdens de stijgende vlucht recht omhoog zal gaan. Deze benadering bespaart erg veel tijd en opslagruimte die nodig zou zijn om een rotatiematrix te implementeren, voor een systeem dat bij benadering recht omhoog zal gaan.

Barometer

Om uit de luchtdruk de hoogte te bepalen is de luchtdruk bij een referentiehoogte nodig (zie *Berekening met de luchtdruk* op pagina 10). Deze referentiedruk moet dus ingesteld worden.

```

1. float bmpSeaLevel; // in hPa.
2. bmpSeaLevel = bmp.readPressure() / 100; // bmp.readPressure() gives a value in Pa.

```

Om nu de hoogte te bepalen gebruiken we de `bmp.readAltitude(float seaLevel)` methode. Deze methode implementeert formule (4.4) uit *Berekening met de luchtdruk* op pagina 10.

```
1. altitude = bmp.readAltitude(bmpSeaLevel);
```

Integratie van de versnelling

Voor de integratie van de versnelling gebruiken we de numerieke rekenmethode zoals beschreven in *Het bepalen van de snelheid* op pagina 9. De volgende code implementeert formule (4.1).

```
1. time = millis();
2. velocity = velocity + (*av) * double(time - previousTime) / 1000.0;
3. previousTime = time;
```

Bepalen wanneer de raket gelanceerd is

Voor het bepalen wanneer de raket gelanceerd is, wordt een drempelwaarde voor de verticale versnelling gedefinieerd.

```
1. // In cm/s^2
2. #define LAUNCH_ACCELERATION_TRIGGER_THRESHOLD 500
```

Als de versnelling gelijk is aan de drempelwaarde of deze overschrijdt, dan zal de boordcomputer dat detecteren als een lancering.

```
1. if ((*av) >= LAUNCH_ACCELERATION_TRIGGER_THRESHOLD)
2. {
3.     changeOperatingMode(OperatingMode::Flight);
4.     // ...
5. }
```

Het gebruiken van een drempelwaarde is noodzakelijk, omdat de versnellingsmeter in rust een meetonnauwkeurigheid heeft van $\pm 5 \text{ cm/s}^2$. Bovendien is het handig om een drempelwaarde in te stellen zodat de raket niet een lancering waarneemt als de raket slecht een klein beetje heeft bewogen. De drempelwaarde van 500 cm/s^2 is groot genoeg dat de raket nog verschoven kan worden zonder dat er een lancering wordt waargenomen, en klein genoeg dat er toch correct een lancering wordt waargenomen.

Pas als een lancering gedetecteerd is zal worden begonnen met het integreren van de snelheid. Daarvoor werd de snelheid namelijk als 0 m/s beschouwd, onafhankelijk van wat de versnellingsmeter als versnelling aangaf, om te voorkomen dat er ongewild een snelheid zou worden waargenomen.

Bepaling van het hoogtepunt

Het hoogtepunt wordt bepaald aan de hand van de verticale snelheid. Zodra deze lager is dan de maximale gemeten snelheid en onder een drempelwaarde van 100 cm/s ligt, dan zal de parachute geopend worden.

```
1. /** Velocity in cm/s */
2. double velocity = 0;
3. double maxVelocity = 0;
4. #define PARACHUTE_VELOCITY_TRIGGER_THRESHOLD 100
5.
6. if (velocity < maxVelocity && velocity < PARACHUTE_VELOCITY_TRIGGER_THRESHOLD)
7. {
```

```
8. // Open parachute
9. }
```

Het is noodzakelijk om een drempelwaarde te gebruiken omdat de versnellingsmeter waarschijnlijk niet perfect de versnelling zal meten tijdens de vlucht. Er ontstaat dus enige mate van afwijking van de daadwerkelijke snelheid. Als de raket al op zijn hoogtepunt is geweest, zonder dat de boordcomputer een hoogtepunt heeft waargenomen en dus de parachute nog niet ontplooid heeft, zal deze versnellen met de punt omlaag. De versnellingsmeter zal dit waarnemen als een positieve versnelling en de verticale snelheid zal dus positief toenemen. Om te zorgen dat de boordcomputer ook bij een afwijking toch in de buurt van het hoogtepunt de parachute ontvouwt, wordt er dus gewerkt met een drempelwaarde die bepaalt wanneer de parachute open moet.

Opslaan van de gegevens

Alle gegevens opgeslagen op een Micro SD-kaart.

```
1. logFile = SD.open("ROCKET_1.txt", FILE_WRITE);
2. logFile.println(altitude);
```

Preventie van gegevensverlies

Tijdens de vlucht is het mogelijk dat door trillingen de voedingskabels naar de boordcomputer los raken. Hierdoor valt de stroom tijdelijk uit. Het is niet de bedoeling dat als de stroom weer terug is de boordcomputer de al opgeslagen data verwijderd. De boordcomputer moet in dat geval direct weer verder gaan met het opslaan van sensordata. Daarom is er een speciale procedure geïmplementeerd om gegevensverlies te voorkomen. Deze procedure is te vinden in de void setup() functie.

De hieronder gedeclareerde variabele stelt of de boordcomputer normaal moet starten of moet starten in *recovery mode*. Deze dingen houden het volgende in:

- Normaal starten: creëer het bestand waarin de sensordata opgeslagen zal worden en ga in standby modus.
- Recovery mode: open het al bestaande bestand, en begin direct met het opslaan van sensordata.

```
1. /** Defines whether flight computer should assume an accidental reset has occurred.
2.  * If true, recovery sequence shall be initiated on boot. */
3. bool recoverAfterReset = true;
```

Hoe er wordt bepaald of de boordcomputer in recovery mode moet starten of niet gaat als volgt. Als het bestand al bestaat wordt er gekeken of de pushbutton wordt vastgehouden voor 2000ms. Is dit het geval, dan wordt het al bestaande bestand verwijderd en zal er normaal verder gegaan worden. Als de pushbutton niet wordt ingedrukt zal de boordcomputer ervan uitgaan dat er per ongeluk een reset heeft plaatsgevonden en zal vervolgens meteen verder gaan met het opslaan van sensordata in het bestand. Als het bestand in de eerste plaats niet bestond wordt er altijd normaal verder gegaan.

```
1. // Check if file exists.
2. // If yes, check to see if recovery should be initiated.
3. // If no, boot normally.
4. if (SD.exists("ROCKET_1.txt"))
5. {
6.     changeStatusLed(HIGH, HIGH, LOW); // Yellow
```

```

7.
8.     int counter = 0;
9.
10.    // Check if pushbutton is being held for 2000ms.
11.    // If yes, delete file and boot normally.
12.    // If no, initiate recovery sequence.
13.    while (digitalRead(PUSHBUTTON_PIN))
14.    {
15.        counter += 1;
16.        delay(10);
17.
18.        // If button is held for long enough, delete file.
19.        if (counter >= 200)
20.        {
21.            changeStatusLed(HIGH, LOW, HIGH); // Purple
22.            delay(2000);
23.            recoverAfterReset = false;
24.            SD.remove("ROCKET_1.txt");
25.            break;
26.        }
27.    }
28.}
29.else
30.{
31.    recoverAfterReset = false;
32.}
33.logFile = SD.open("ROCKET_1.txt", FILE_WRITE);

```

De normale sequentie:

```

1. // Normal boot
2.
3. // Data headings
4. logFile.println("t;ax;ay;az;gx;gy;gz;v;h;accelStatus;baroStatus");
5. logFile.println("ms;cm/s^2;cm/s^2;cm/s^2;deg/s;deg/s;deg/s;cm/s;m;");
6.
7. // After succesful setup, go into standby.
8. changeOperatingMode(OperatingMode::Standby);

```

De recovery sequentie:

```

1. // Recovery sequence
2. logFile.println("##### RECOVER AFTER RESET #####");
3. logFile.close();
4.
5. changeOperatingMode(OperatingMode::Ready);
6. changeOperatingMode(OperatingMode::Flight);

```

Moeilijkheden en problemen

Hier zullen nog enkele moeilijkheden en problemen besproken worden waar we tegenaan gelopen zijn bij het programmeren van de boordcomputer.

Rekenkundige typeconversie

Een probleem waar tegenaan gelopen werd, was dat de berekende versnelling onrealistische waarden gaf. Dit probleem was te herleiden tot een vergissing in type conversie. In C++ moeten

variabelen van verschillende typen eerst geconverteerd worden als ermee wordt gerekend. Dit heet *standard conversion*²¹ en gebeurt automatisch, maar als men het zich niet realiseert kan het voor onverwacht gedrag zorgen. Om te zorgen dat al het gedrag ook is zoals verwacht, moeten variabelen eerste gecast worden naar het juiste datatype.

```
1. velocity = velocity + (*av) * double(time - previousTime) / 1000.0;  
2. [double] = [double] + [int16] * double([uint32] - [uint32]) / [double] ;
```

In plaats van:

```
1. velocity = velocity + (*av) * (time - previousTime) / 1000.0;
```

Gegevensverlies

Tijdens lancering 1 (zie *Lancering 1* op pagina 42) zijn de gegevens van de vlucht verloren gegaan. Om dit probleem in de toekomst te voorkomen zijn er extra maatregelen genomen om gegevensverlies te voorkomen. De veranderingen van de code zijn beschreven onder *Preventie van gegevensverlies* op pagina 30. Bij lancering 1 bestond ook al een dergelijke check, maar deze was minder grondig. Het was enkel nodig de knop 1 keer voor onbepaalde tijd in te drukken. Daarnaast was het voor de boordcomputer niet mogelijk om uit zichzelf terug te krabbelen en meteen verder te gaan met het verwerken van de data.

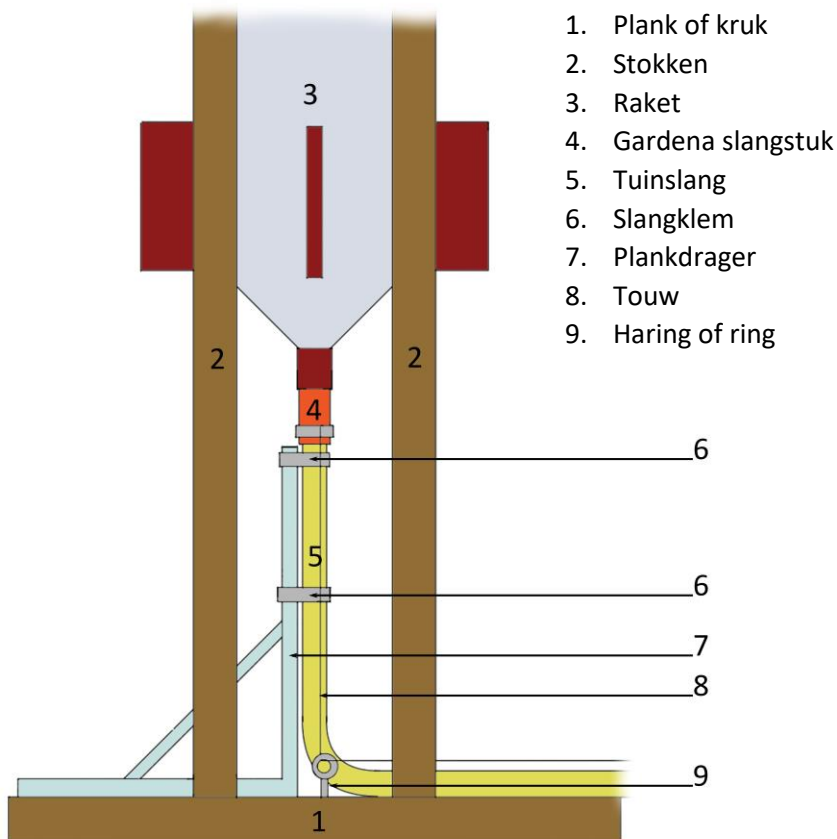
²¹ Microsoft. (2019, 2 oktober). *Standard conversions*. Microsoft Docs. Geraadpleegd op 24 november <https://docs.microsoft.com/en-us/cpp/cpp/standard-conversions?view=msvc-160>

Het lanceerplatform

De raket kan niet functioneren zonder lanceerplatform. Deze houdt de raket recht bij zowel stilstand als lancering. Het lanceerplatform moet ruimte voor aanvoer van gepompt lucht bieden, en veilig op afstand de raket kunnen lanceren. Bij de lancering moet het platform de raket zo recht mogelijk omhoog leiden maar tegelijkertijd de lancering zo min mogelijk belemmeren. Het platform moet tot slot wel goedkoop en makkelijk te bouwen zijn, vervoerbaar zijn en snel op te zetten zijn.

Het ontwerp

In Figuur 31 volgt de ontwerptekening van het lanceerplatform.



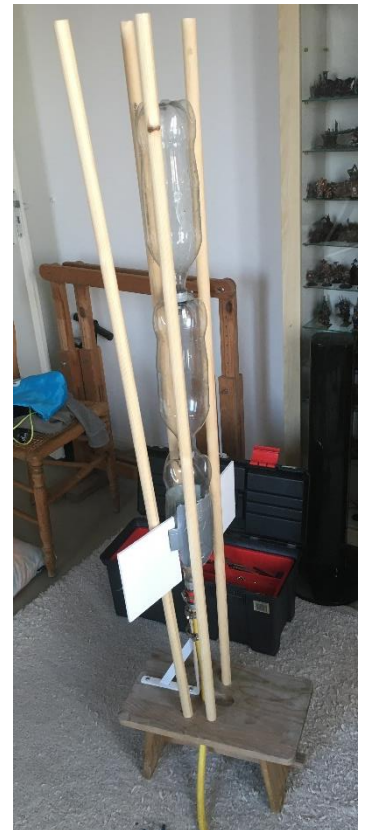
Figuur 31 Ontwerp van het lanceerplatform.

De kern van het lanceerplatform bestaat uit een simpele plank of kruk met daarop 3 of 4 houten stokken die de raket rechthouden (Figuur 32).

Verder is er een slang nodig die fungeert als luchttoevoer. Hieraan moet een pomp en de raket passen.

Aan het ene uiteinde van de slang zit een pomp met drukmeter die minimaal 7 bar moet kunnen aangeven. De pomp wordt bevestigd op een ventiel dat moet worden vastgemaakt op de slang, luchtdicht en drukbestendig (Figuur 33).

Aan het andere uiteinde van de slang moet deze luchtdicht en drukbestendig worden bevestigd aan de raket, met de mogelijkheid deze connectie op afstand te verbreken bij de lancering. Door een plankendrager (Figuur 34) vast



Figuur 32 Het lanceerplatform.



Figuur 33 Ventiel aan slang.

te schroeven aan het platform en daaraan met slangklemmen de slang te binden wordt de raket stevig en recht op de slang vastgehouden.

Slangstukken en kraanstukken zijn ontworpen om luchtdicht en drukbestendig op elkaar aan te sluiten, het kraanstuk als straalbuis van de raket past dan ook goed op het slangstuk van het lanceerplatform. Om deze los te maken bij lancering is een touw bevestigd aan het slangstuk. Door het touw naar beneden te trekken wordt het slangstuk naar beneden en dus losgetrokken van het kraanstuk van de raket en kan de raket losschieten. We bevestigden het touw stevig aan het slangstuk met behulp van bijvoorbeeld een slangklem. Dit touw is naar beneden gericht en gaat door een gat in het lanceerplatform naar de grond, waar het met behulp van een haring een bocht van 90° maakt. Zo kan er veilig op afstand aan getrokken worden. Om een haring stevig in de grond te zetten is het handig als de lancering plaatsvindt op bijvoorbeeld een grasland.



Figuur 34 Plankendrager

Als voorzorgsmaatregel moet er de mogelijkheid zijn de druk van de slang en raket te halen zonder de raket te lanceren met behulp van een abort-valve, een punt halverwege de slang waar deze als nodig kan worden opengebroken. Dit punt bestaat uit twee slanguiteinden met slangstukken bevestigd met een slangkoppeling. Deze kan eveneens als het lanceermechanisme op afstand losgemaakt worden met een touw bevestigd aan een van de slangstukken. Het is noodzakelijk dat de slang op de plek van de abort-valve stevig aan de grond bevestigd is zodat er aan het touw getrokken kan worden zonder dat de slang mee getrokken wordt. Weer kunnen haringen hierbij van pas komen, direct om de slang of met touw de slang klemmend.



Figuur 35 Kraanstuk met slangklemmen bevestigd aan plankendrager.

De realisatie

Lijst met benodigde spullen:

- 1x Oude houten kruk 40x20x20 cm
- 2x Houten stok \varnothing 28mm, 270 cm
- 1x plankdrager 15x25 cm
- 1x Gardena tuinslang ~15 m
- 4x Gardena slangklem
- 3x Gardena slangstuk
- 1x Gardena kraanstuk
- 1x ventiel van een fietsbinnenband
- 1x Handpomp met drukmeter
- Touw van een vlieger
- Boor met passende boorstukken
- 4x Lange moer \varnothing 6.0 mm, 140 mm
- 2x Korte moer \varnothing 3.0 mm, 20 mm
- Lijmpistool
- Schroevendraaier
- 1x kurk
- Schuurpapier
- 3x haring



Figuur 36 Haring waardoor een touw een 90° hoek maakt.



Figuur 37 Abort-valve, onder gesloten en boven open.

Alle onderdelen bij elkaar exclusief gereedschap kosten ongeveer € 60,-.

Als basis van het platform is een oude houten kruk gebruikt, makkelijk te bewerken en voldoende oppervlakte om de palen, plankdrager en raket te houden. Deze kruk had toevallig een gleuf in het midden, ideaal om de slang doorheen te laten lopen (Figuur 38).

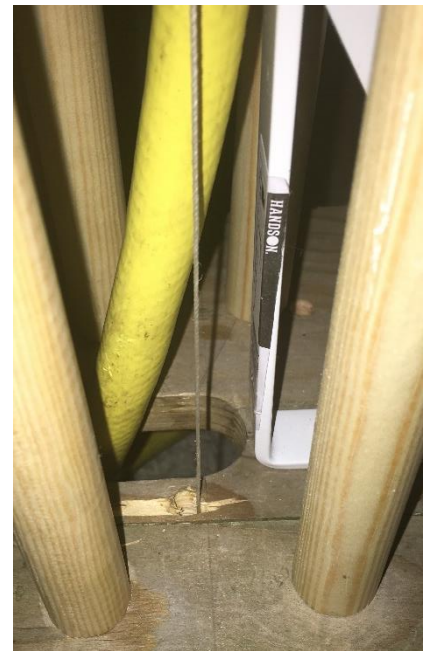
De gebruikte stokken zijn twee 270 cm lange stokken doormidden gezaagd tot vier 135 cm lange stokken. De stokken kunnen rechtop boven op de kruk worden bevestigd door een gat met gelijke diameter te boren in de kruk en ze daarin vast te lijmen. De stokken hadden echter een diameter van 28 mm en vanwege gebrek van een passend boorstuk werden de stokken met behulp van lange moeren bevestigd. Als ze er zijn dan zijn passende paalhouders een steviger maar wel duurder alternatief. Let bij het boren op dat de stokken zo recht mogelijk bevestigd moeten worden, dit was bij ons op eentje na gelukt en deze corrigeerden we door onderaan elastieken te plaatsen (te zien in Figuur 35).

Na de stokken te hebben bevestigd is de plankdrager (Figuur 34) bevestigd. Plaats hem zo, dat zodra de slang eraan vast wordt geklemd deze exact in het midden tussen de palen zit. De plankdrager is bevestigd met een schroevendraaier en twee korte schroefjes in de van tevoren geboorde gaten.

Een Gardena tuinslang van 15 meter wordt gebruikt om druk op te voeren in de raket (Figuur 39).

Aan het ene uiteinde komt de pomp. Hiervoor moet een ventiel luchtdicht en drukbestendig in de tuinslang worden gemonteerd.

Bij de eerste poging was een stuk van een fietsband met ventiel met behulp van slangklemmen, tape en een overvloed aan lijm uit een lijmmachine aan een uiteinde van de tuinslang met slangstuk gemonteerd (Figuur 40). Genoeg lijm bleek te voldoen om de verbinding luchtdicht te maken, maar bij het uittesten bleek de binnenband niet drukbestendig genoeg te zijn om de 7 bar te halen; al bij een schamele 2.5 bar begaf de band het met een luide knal (Figuur 41). Een alternatieve verbinding was nodig.



Figuur 38 Stokken, slang door gleuf en touw door gat.



Figuur 39 Gebruikte tuinslang



Figuur 40 Oud model ventiel aansluiting, plek van het kraanstuk getekend



Figuur 41 Geklapte plek oud ventiel aansluiting

Bij het uiteindelijke product (Figuur 33) is het ventiel losgeknipt van de binnenband. Er is een gat geboord in een kurk groot genoeg voor het ventiel om erin te passen. Deze is hierin met een lijmpistool vastgelijmd, en de kurk met ventiel is vervolgens in een van de slanguiteinden geduwd en vastgelijmd met het lijmpistool. Deze veel simpelere verbinding voldoet wel aan de drukbestendigheid-normen. Op het ventiel kan bij lancering de handpomp worden bevestigd aan de slang en lucht de raket in worden gepompt.



Figuur 42 Realisatie ventiel aansluiting

Aan het andere slanguiteinde is een slangstuk bevestigd. Deze sloot in eerste instantie echter niet goed aan op het kraanstuk van de straalbuis, waarschijnlijk door verdikkingen en uitsteeksels ontstaan bij het monteren van het kraanstuk op de raket. Als oplossing hiervoor is zorgvuldig wat van de aansluiting van het slangstuk geschuurd met schuurpapier, totdat deze wel paste. Let hierbij op dat te ver schuren kan leiden tot het niet langer functioneren van de aansluiting.

Om de tuinslang af te maken is deze in tweeën gezaagd, aan beide nieuwe uiteindes zijn slangstukken bevestigd, en vervolgens zijn deze bevestigd met een slangkoppeling. Wat vliegertouw is gewikkeld rond een slangklem en deze is bevestigd aan een van de slangstukken met een schroevendraaier. Op deze wijze is de abort-valve gemaakt, door aan het touw te trekken zullen de slanguiteinden loskomen en zo de druk van de raket afhalen. We klemmen de abort-valve met haringen aan de grond vast.



Figuur 43 Abort-valve

Vervolgens is de tuinslang geklemd aan de plankdrager. De slang moet zo recht mogelijk omhoog wijzen, laat hiervoor het slangstuk net boven het uiteinde van de plankdrager steken (te zien in Figuur 35). Klem vervolgens de slang vast met twee slangklemmen op de uiteindes en een schroevendraaier om ze te sluiten. De slang hoort nu netjes rechtop tussen de palen te staan en de raket past erop.

Dan is het lanceringsmechanisme gemaakt. Wat vliegertouw is gewikkeld om een slangklem en deze is vervolgens onderaan het slangstuk bevestigd en met een schroevendraaier gesloten. Dit touw naar beneden trekken hoort te resulteren in het openen van het slangstuk en dus het loskomen van de raket.

Vervolgens hoort het vliegertouw een 90° hoek te maken om op afstand aan te kunnen trekken. Om het touw te leiden naar de haring is er een gat geboord recht onder het touw om doorheen te gaan (Figuur 38). Het is aan te raden het touw eerst door het gat te halen voor het vast te klemmen aan het slangstuk. Onder de kruk loodrecht onder het geboorde gat wordt voor lancering de haring (Figuur 36) geplaatst in de grond.

Vluchtmodel

Het vluchtmodel implementeert de vergelijkingen beschreven in *Krachten tijdens een raketvlucht* op pagina 6. De parachute wordt niet meegenomen in het model. Met deze vergelijkingen wordt de somkracht bepaald. Hieruit kan met de derde wet van Newton $F_{som} = ma$ de versnelling op de raket worden bepaald. Het model bepaalt op numerieke wijze de snelheid en hoogte uit de versnelling. Uitleg over de numerieke bepalingen is te vinden in *Het bepalen van de snelheid* en *Het bepalen van de hoogte* (specifiek *Integreren van de snelheid*) op pagina 9.

Naast de snelheid en de hoogte wordt ook de afname van het watervolume numeriek bepaald. De vloeistofstroom wordt gegeven door vergelijking (3.8). Voor het numeriek bepalen van het watervolume kunnen we de vergelijking als volgt opschrijven:

$$\Delta V_{water} = A_{straalbuis} v_{water} * dt \quad (6.1)$$

Als we nu vergelijking (2.3) invullen en de formule omschrijven voor het totale watervolume krijgen we:

$$V_{water} = V_{water} - A_{straalbuis} \sqrt{\frac{2(P - P_{atm})}{\rho_{water}}} * dt \quad (6.2)$$

Nu kunnen we alle vergelijkingen implementeren in het model. Het model wordt geschreven in Coach 7. Het model heeft de stopconditie $h < 0$ m.

Initialisatie en beginvoorwaarden van het model:

```

1. 'Alle gegevens staan in SI-units, behalve P_drukmeter, nl. in bar.
2. 'Natuurconstanten
3. g := 9.80665
4. P_atm := 101325
5. rho_lucht := 1.225
6. rho_water := 1000
7.
8.
9. 'Gegevens van de raket
10. m_droog := 0.6
11. V_water := 1.5 / 1000
12. V_totaal := 4.5 / 1000
13.
14. A_frontaal := 0.3^2 / (4 * pi)
15. r_straalbuis := 0.0045
16. C_w := 0.5
17. P_drukmeter := 7
18.
19. A_straalbuis := pi * r_straalbuis^2
20. P_0 := P_drukmeter * 100000 + P_atm
21. V_lucht_0 := V_totaal - V_water
22.
23. P := P_0
24. m_totaal := 0
25.
26.
27. 'Initialisatie overige variabelen
28. F_som := 0
29. F_lw := 0
30. F_s := 0
31. F_g := 0
32.
33. a := 0
34. v := 0
35. h := 0
36.
37.
38. 'Modelgegevens
39. t := 0
40. dt := 0.01

```

Programmacode van het model:

```

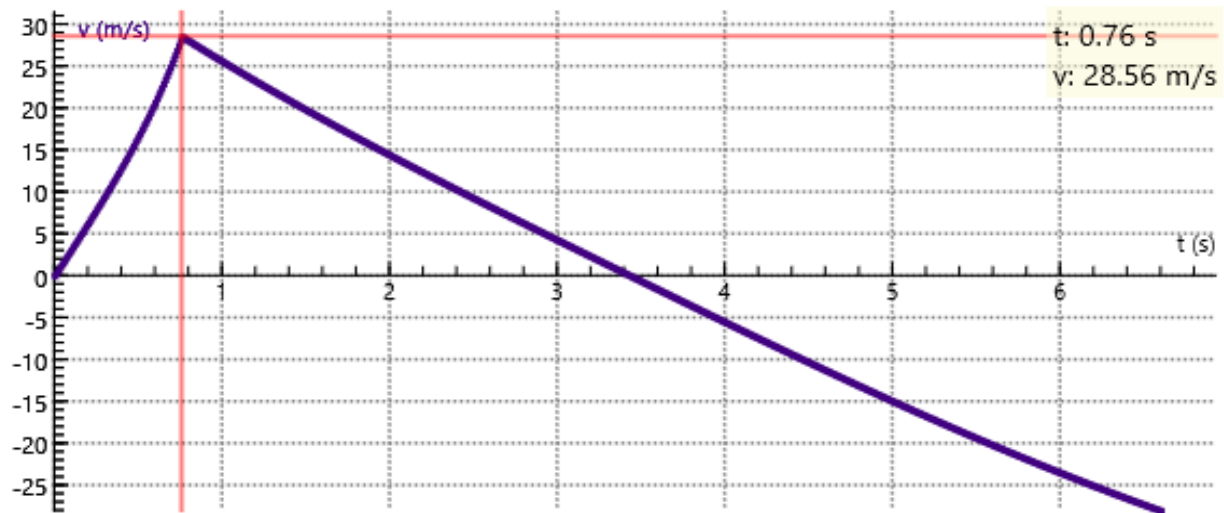
1. 'Stopconditie : h < 0
2. m_totaal := m_droog + V_water * rho_water
3.
4. F_s := 2 * (P - P_atm) * A_straalbuis
5. F_lw := 0.5 * C_w * A_frontaal * rho_lucht * v^2 * sign(v)
6. F_g := m_totaal * g
7.
8. F_som := F_s - F_lw - F_g
9.
10. a := F_som / m_totaal
11. v := v + a * dt
12. h := h + v * dt
13.
14. V_water := V_water - A_straalbuis * sqrt(2 * (P - P_atm) / rho_water) * dt
15. V_lucht := V_totaal - V_water
16. P := P_0 * (V_lucht_0 / V_lucht)^(7/5)
17.
18. If V_water <= 0 Then
19.     V_water := 0
20.     V_lucht := V_totaal
21.     P := P_atm
22. EndIf
23.
24. t := t + dt

```

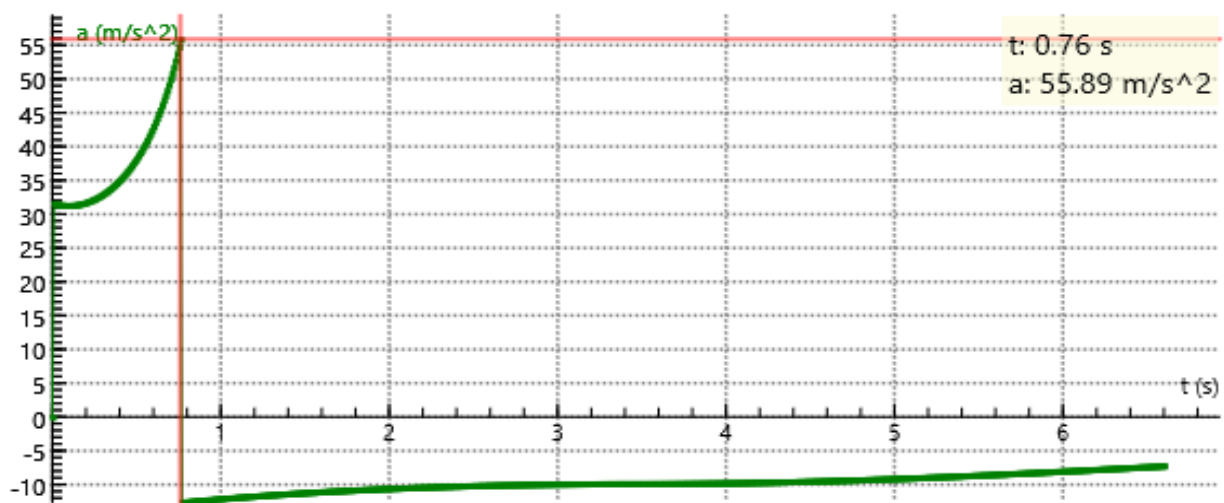
Hier volgen de grafieken die gegenereerd zijn door het model.



Figuur 44 Hoogte over tijd.



Figuur 45 Snelheid over tijd.



Figuur 46 Versnelling over tijd.

Uit het model blijken de volgende gegevens.

- Hoogst ervaren versnelling: 56 m/s² of ongeveer 5.7 g.
- Hoogst bereikte snelheid: 29 m/s.
- Duur van de eerste versnelling: 0.76 s.
- Hoogtepunt: 46 m na 3.5 s.

Het model zal met de daadwerkelijke vlucht worden vergeleken.

Lancering

Op 2 januari 2021 was het zover: de raket werd voor het eerst gelanceerd. Wat kwam erbij kijken en waar liepen we tegen op?



Figuur 47 Pomp rechtsonder



Figuur 48 De lancering

Voorbereiding

Alles dat we nodig hadden voor de lancering:

- Vervoer
- 6x 1.5 liter flessen met water
- 1x houten schraag
- Lanceerlocatie
- Laptop met USB-kabel voor boordcomputer
- Gehoorbescherming
- Veiligheidsbrillen
- Trechter
- Ducttape

De locatie waar we lanceerden was een groot vlak grasland, om enig risico op een landing op een verkeerde plek te voorkomen en gras is een goede plek om haringen stevig in de grond te zetten. We kozen een dag met zo min mogelijk wind en regen. We maakten gebruik van een houten schraag om de tuinslang omhoog te houden (zoals te zien in Figuur 48) om te verzekeren dat water van de raket niet onnodig wegstroomt. Hiervoor moest de slang net hoger dan het waterniveau van de raket worden gehouden, deze schraag werkte toevallig goed hiervoor maar bijvoorbeeld een stevig staande stoel zou hetzelfde kunnen.

Als voorbereiding voor de lancering plaatsten we het lanceerplatform en de schraag, met de slang over de schraag en volledig uitgerold met pomp aan het ventiel (Figuur 47). Vervolgens zijn de haringen geplaatst: een onder het lanceerplatform waardoor het touw om te lanceren een 90° hoek maakte, en twee die de abort-valve klemde zodat hieraan getrokken kon worden terwijl deze op zijn plek bleef. De touwen van het lanceerplatform en de abort-valve werden parallel aan de tuinslang naar de pomp toe uitgerold, klaar om te gebruiken. Toen vulden we de raket met water. We draaiden de onderste fles van de raket los en vulde de raket met een trechter met iets meer dan 1.5 liter water om vervolgens de fles weer terug te draaien op de raket. Met een duim op de straalbuis vermeden we dat water wegliep en we zetten de raket op het lanceerplatform aan het slangstuk vast. Voor de lancering moet de boordcomputer gekalibreerd worden. Via het klepje in het boordcomputercompartiment sloten we de boordcomputer aan met een usb-kabel op een laptop.

Vanaf de laptop werden de sensoren gekalibreerd door de kalibratiecode te uploaden. Nadat de kalibratie klaar was werd de boordcomputer in de “Ready” modus gebracht. Zodra de boordcomputer gereed was zetten we de top van de raket vast op de onderkant met tape. Gehoor- en oogbescherming op en de raket was klaar om te lanceren.

Lancering 1

Op een veilige afstand begonnen we met pompen, de drukmeter in de gaten houdend. Toen we een druk van 6 bar bereikten zagen we dat de boordcomputer ongewild al naar “Flight” modus was overgeschakeld, dus moest de lancering worden gestopt, zodat de boordcomputer goed ingesteld kon worden. We trokken aan het touw van de abort-valve en alle druk, en hiermee ook water, ontsnap zonder de raket te lanceren. Met de boordcomputer nu wel juist ingesteld, dichtten we de abort-valve weer en vulden we opnieuw de raket met water. We gingen weer pompen, maar toen we ditmaal de 5 bar voorbijgingen steeg de druk niet langer door wat we hoorden als een klein lek bij het ventiel. Omdat telkens toen we pompte de druk afnam tot het weer 5 bar was zat er niets anders op dan de raket te lanceren. We trokken aan het lanceringstouw en de raket spoot de lucht in tot een hoogte die we op 20 meter schatten, om vervolgens rechtsomkeert te maken en met zijn neus hard op de grond te landen.

Analyse

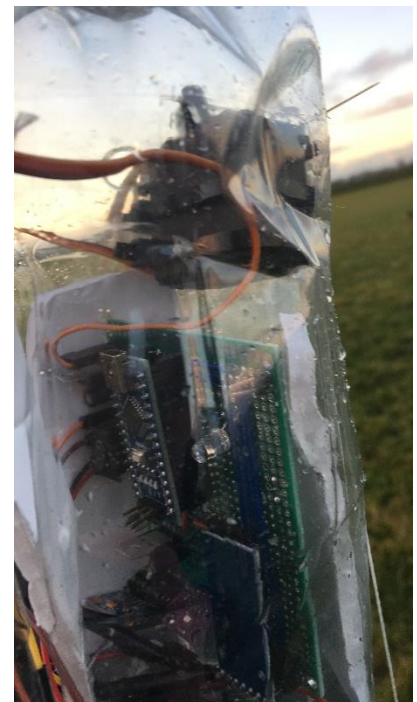
Tijdens de vlucht was er een probleem met de boordcomputer waardoor de parachute niet was uitgeklapt. We vermoeden dat tijdens de vlucht per ongeluk de stroom met de boordcomputer is verbroken. De harde val resulteerde in een scheur van de onderste fles (Figuur 49). De foamhouders van de boordcomputer waren gescheurd maar hadden zijn val voldoende opgevangen; op een paar licht gebogen onderdelen na was de boordcomputer schadeloos. Na de boordcomputer te hebben aangesloten op de computer bleek geen data over te zijn van de lancering en vlucht. Enkel de definitie van de groot- en eenheden stond in het bestand. De reden waarom er geen data is opgeslagen is onbekend. Mogelijk is de boordcomputer per ongeluk opnieuw opgestart, en heeft hij, ondanks een check om dit probleem te voorkomen, het bestand met de data verwijderd. Het zou ook mogelijk kunnen zijn dat door een probleem met de stroom de sensoren vanaf het begin al geen contact maakten met de Arduino.

Nu volgt een natuurkundige analyse van de vlucht.

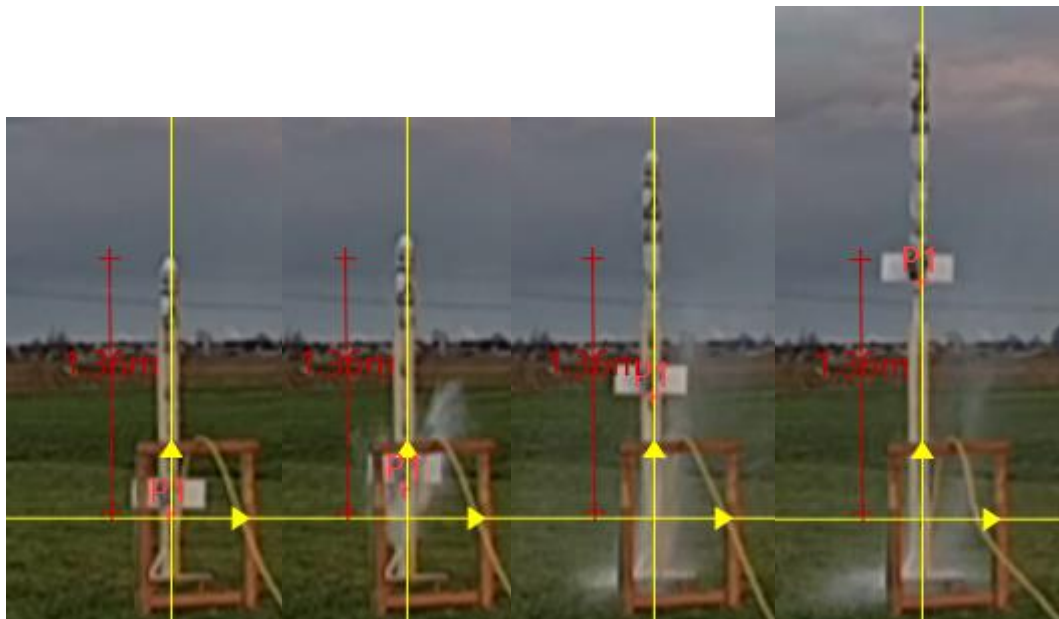
Een video van de lancering van de raket is geanalyseerd. Een link naar deze video is te vinden in de bijlage.



Figuur 49 Scheur in de onderste fles

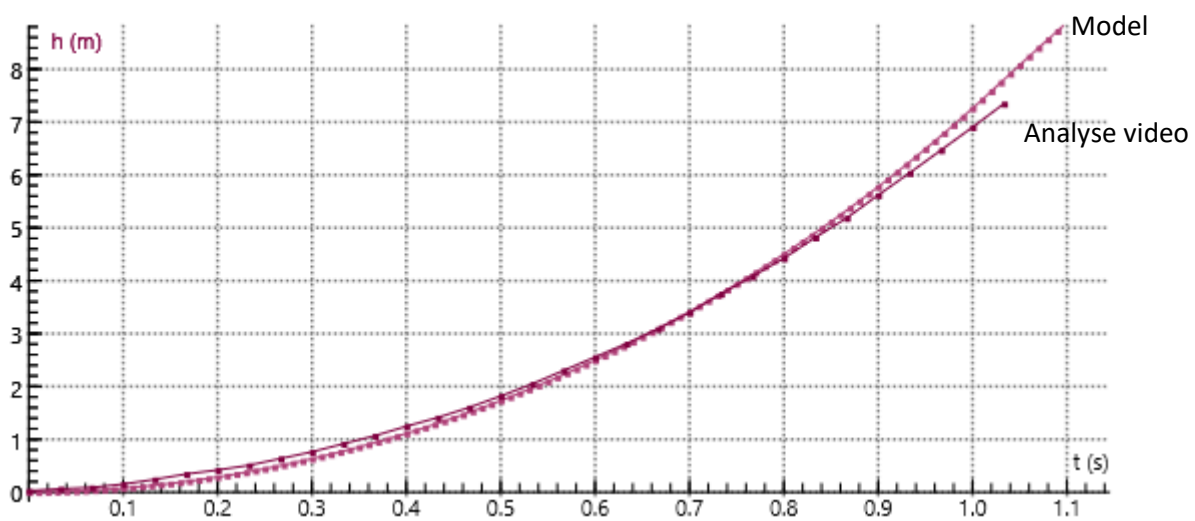


Figuur 50 Schade boordcomputercompartiment



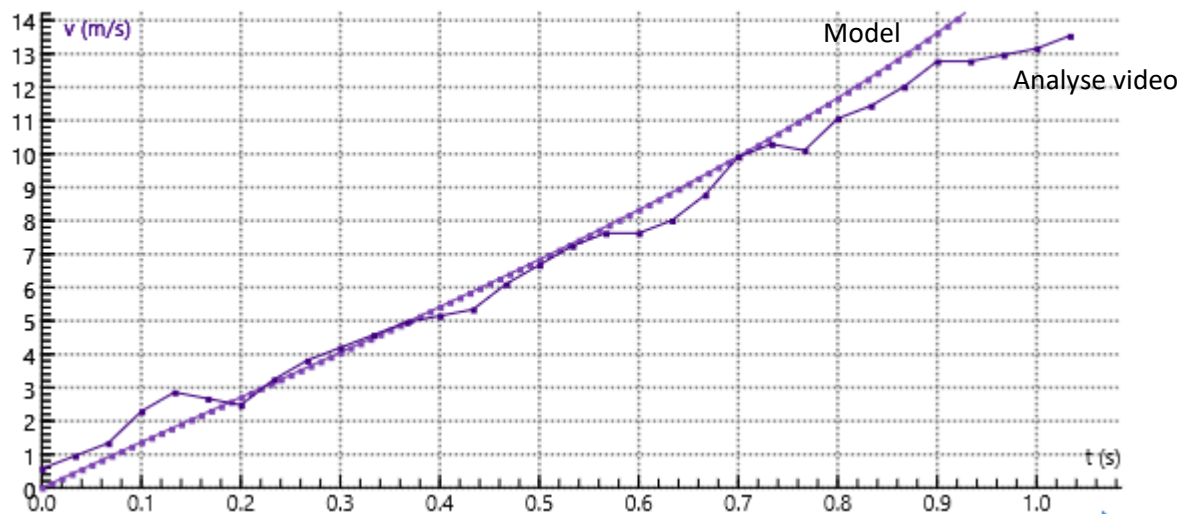
Figuur 51 Video van de lancering van de raket. Let op: slechts enkele frames zijn weergegeven. De vier frames die zijn weergegeven spannen een tijd van 0.4 seconden.

In Figuur 51 is te zien dat de raket mooi recht omhoog gaat. Als we in het model de waarden voor deze vlucht invullen, namelijk $P_{\text{drukmeter}} = 4$ bar en $V_{\text{water}} = 1.5$ L dan zien we dat het model voor de hoogte goed overeenkomt met de lancering van de raket.



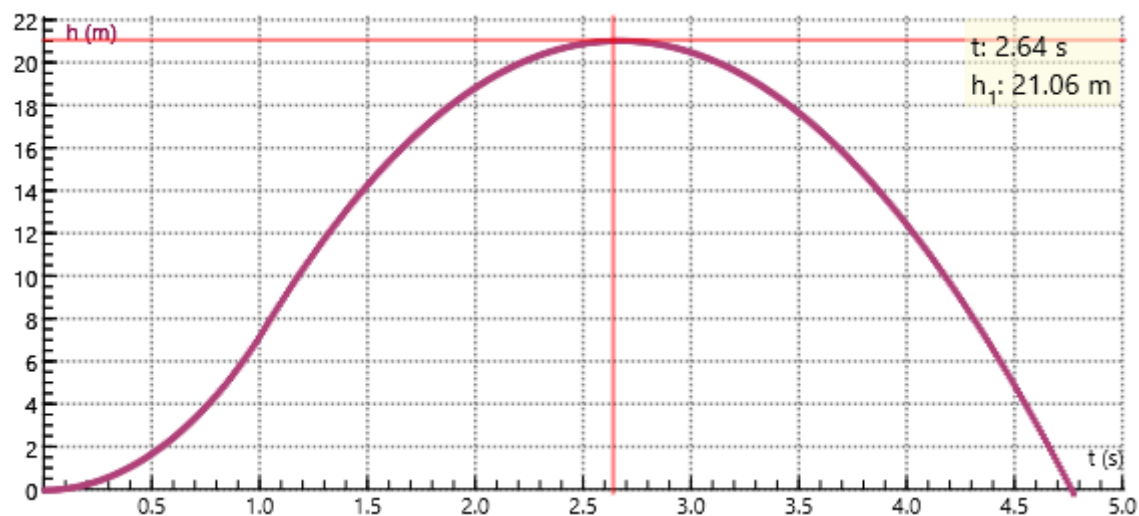
Figuur 52 Hoogte over tijd. Vergelijking model en video.

Ook het model voor de snelheid komt in het begin van de vlucht goed overeen met de daadwerkelijke vlucht.



Figuur 53 Snelheid over tijd. Vergelijking model en video.

Als we nu het model volgen zien we dat die een hoogtepunt aangeeft van ongeveer 21 m, en dit komt overeen met waarop we de hoogte geschat hebben, namelijk ongeveer 20 m.



Figuur 54 Hoogte over tijd voorspeld door het model.

Het hoogtepunt dat het model voorspelt verandert slechts met enkele meters als we de C_w variëren. Bij een C_w van 0.5 is het hoogtepunt volgens het model op 21m en bij een C_w van 1.5 is het hoogtepunt volgens het model op 19m. Omdat we het echte hoogtepunt op het oog geschat hebben, kunnen we dus niet de conclusie trekken of de gekozen C_w van 0.5 een accurate waarde is.

Herstel van de raket

Om de raket na de crash te herstellen moesten een aantal onderdelen vervangen en een aantal gerepareerd worden. Alle flessen die het reservoir vormen zijn vervangen. Op de nieuwe onderste fles moest weer een straalbuis gemaakt worden. We hebben de straalbuis van de kapotte fles gerecycled, op de dop van de terpentinefles na. De verbindingen tussen de andere flessen zijn ook gerecycled, op de rubberen ringen na: die zijn opnieuw gemaakt. Ook alle vinnen zijn gerecycled. Er zijn nieuwe foamhouders voor de boordcomputer gemaakt. Ter reparatie zijn op verschillende

plekken ducttape aangebracht. Ook de elastiekjes van het parachutesysteem zijn vervangen. Voor de rest was er geen schade.

Lancering 2

Een tweede lancering is gepland. Zodra deze voltooid is zullen de resultaten worden gepubliceerd.

Conclusies

Tot en met januari 2021 zijn we bezig geweest met het ontwerpen en realiseren van een waterraket met meet- en regelapparatuur. Voordat we zijn begonnen hebben we aan alle onderdelen een aantal eisen gesteld.

Ten eerste de raket. Net zoals we gesteld hadden is deze herbruikbaar, mits hij niet te hard op de grond landt. Bij lancering 1 kwam de raket hard neer omdat de parachute niet open ging, maar was daarna na enkele goedkope reparaties opnieuw te gebruiken. De raket heeft een drooggewicht van 600g, en dit is lager dan de eis van 800g. Bij lancering 1 heeft de raket niet de geëiste hoogte van 50m bereikt. Dit kwam doordat de druk niet hoger opgebouwd kon worden door een lek. Uit lancering 1 blijkt dat de raket stabiel is, en dit komt doordat hij aan de eis voldoet dat zijn zwaartepunt boven zijn drukmiddelpunt moet liggen. De uiteindelijke kosten voor het bouwen van de raket (exclusief benodigd gereedschap, maar inclusief de kosten van de boordcomputer, namelijk ongeveer € 30,-) bedraagt ongeveer € 50,- en dit is lager dan de eis van € 75,-. Ook heeft de raket een parachute. Al klapte deze bij de lancering niet uit, uit testen werd duidelijk dat hij wel werkt. De raket heeft een boordcomputer die kan meten hoe hoog hij komt, al is bij lancering 1 de data verloren gegaan.

Ten tweede de boordcomputer. Door zijn kleine afmetingen past hij perfect in een 1.5L colafles. De boordcomputer is ook in staat om te bepalen wanneer de raket zijn hoogtepunt heeft bereikt. Ondanks dat dit bij lancering 1 niet gelukt is, is uit testen gebleken dat het wel werkt. Voor het ontplooiën van de parachute geldt hetzelfde. Het kalibreren van de boordcomputer is binnen 15 minuten geklaard, en op dit aspect voldoet de boordcomputer dus ook aan de eisen.

Ten derde het lanceerplatform. We hebben het lanceerplatform meerdere keren kunnen gebruiken, en hij voldoet dus aan die eis. Bij de lancering hebben we alles met de auto vervoerd, het lanceerplatform voldoet dus aan de eis dat hij vervoerbaar moet zijn met de auto. De lengte van de tuinslang is lang genoeg om te voldoen aan de eis dat de raket op minimaal 15m afstand gelanceerd kan worden. Bij lancering 1 bleek dat het mechanisme om de lancering op afstand af te breken perfect werkt. De kosten voor het lanceerplatform (exclusief gereedschap) komen uit op ongeveer € 60,- en dit voldoet aan de eis van maximaal € 75,-. Ook bleek uit lancering 1 dat het lanceerplatform ruim binnen een half uur op te zetten is.

Ten vierde het model. Uit lancering 1 bleek dat het model de uiteindelijke hoogte van de raket redelijk voorspelde. De hoogte uit het model was namelijk 21m en we schatten de hoogte die de raket heeft bereikt bij lancering 1 op 20m.

In zijn geheel voldoet de waterraket dus ruim aan haast alle eisen; wat nog niet voldoet doet dit als het goed is wel bij lancering 2. We zijn erin geslaagd uit te vinden en uiteen te zetten wat er komt kijken bij het ontwerpen en realiseren van een herbruikbare waterraket voorzien van elektronische meet- en regelapparatuur.

Evaluatie

In zijn geheel zijn wij zeer tevreden met hoe het is gegaan; veel ging zoals gepland en eventuele tegenslagen waren allemaal tot op een zeker punt opgelost.

Hoe kunnen we de raket nog verbeteren en wat zouden we anders doen als we de raket opnieuw bouwen mochten? Onze grootste uitdaging en nu toch nog het grootste zwakste punt van de raket is de drukbestendigheid, zoals bij het ventiel. Al hebben we alles gedaan wat we konden om dit optimaal te maken, als we het opnieuw moeten doen zullen we flink de tijd nemen om dit nog beter aan te pakken. Onze raket kan ook nog herbouwd worden met meer oog op aerodynamica, met een groter parachute en de stokken van het lanceerplatform hadden beter kunnen worden gemonteerd met paalhouders. Een goed alternatief voor kraanstuk als straalbuis had er een kunnen zijn die was geprint met een 3d-printer. Daarnaast zou het goed zijn geweest om de verbindingen tussen de batterij en de boordcomputer beter te maken met bijvoorbeeld krimpkousen, dit was namelijk het zwakste punt van de boordcomputer.

Op deze verbeterpunten na zijn wij ervan overtuigd dat onze manier van een waterraket bouwen de beste manier was voor het bouwen met grotendeels huis-, tuin- en keukenspullen. Sommige keuzes die we gemaakt hebben vonden we bovenal goed werkend. Om te beginnen het gebruik van ducttape en een lijmpistool voor veel van de verbindingen. Ducttape bleek uitermate geschikt als goedkope, goed vervangbare en zeer sterk materiaal om verbindingen mee te maken. De lijm van het lijmpistool heeft ons geholpen om zowel stevige maar bovenal luchtdichte verbindingen te maken. Onze vinnen enkel met strakgespannen ducttape bevestigen was fijn door de makkelijke reparaties en minimale mogelijkheid voor beschadiging. Ook het gebruik van Gardena producten was ideaal omdat ze al luchtdicht en drukbestendig ontworpen waren. Onze flesverbindingen maakte het heel makkelijk voor ons om de flessen te vervangen en te vullen, doordat deze door de verbindingen los te draaien waren. Ons heel scherpe Stanley mes was ook ideaal voor het snijden van flessen, en het kopen van een binnenband van een fiets was slim omdat we zowel het ventiel als de rubber goed gebruiken konden voor het maken van sluitringen. Onze abort-valve bleek uitermate nuttig tijdens onze eerste lancering, en haringen deden ideaal hun taak als houders van de abort-valve en het touw van het lanceermechanisme. Het idee om de boordcomputer met verschillende modi te laten werken was ook een goed idee. Daardoor was het mogelijk de boordcomputer makkelijk te testen en in te stellen. Daarnaast was het een goed idee om opnieuw te beginnen met de boordcomputer, omdat hij er nu veel netter uit ziet en veel beter werkt.

Zodra we onze waterraket herbouwen met alle kennis die we nu hebben is het gegarandeerd dat we een uitstekend werkende waterraket ervan overhouden, gebouwd op budget met grotendeels huis-, tuin- en keukenspullen.

Bronvermelding

Accelerometer Basics. (z.d.). SparkFun. Geraadpleegd op 17 oktober 2020, van <https://learn.sparkfun.com/tutorials/accelerometer-basics/>

Adafruit Industries. (2015, 31 juli). *Adafruit BMP280 Barometric Pressure + Temperature Sensor Breakout*. Geraadpleegd op 17 oktober 2019, van <https://learn.adafruit.com/adafruit-bmp280-barometric-pressure-plus-temperature-sensor-breakout>

Air Command Water Rockets. (2019a, november 26). *Nozzles*. Geraadpleegd op 27 september 2020, van <http://www.aircommandrockets.com/nozzles.htm>

Air Command Water Rockets. (2019b, november 26). *Water rocket construction - Advanced tutorials*. Geraadpleegd op 27 september 2020, van http://www.aircommandrockets.com/construction_2.htm

Air Command Water Rockets. (2019c, november 26). *Water rocket construction - Basics*. Geraadpleegd op 27 september 2020, van <http://www.aircommandrockets.com/construction.htm>

Air Command Water Rockets. (2019d, november 26). *Water rocket recovery guide*. Geraadpleegd op 30 september 2020, van http://www.aircommandrockets.com/recovery_guide.htm

Apogee Components. (z.d.). *Technical Publication 16 - What Type of Fin Shape is Best : Apogee Rockets, Model Rocketry Excitement Starts Here*. Geraadpleegd op 20 september 2020, van https://www.apogeerockets.com/technical_publication_16

Arduino. (z.d.). *What is Arduino?* Geraadpleegd op 25 november 2020, van <https://www.arduino.cc/en/Guide/Introduction/>

Bluett, S. (2015, 21 januari). *Calibrating & Optimising the MPU6050*. chillibasket. Geraadpleegd op 24 oktober 2020, van <https://wired.chillibasket.com/2015/01/calibrating-mpu6050/>

Circuito.io. (z.d.). *How to wire Servo - Generic Metal Gear (Micro Size), RGB LED Clear Common Anode, SparkFun MPU-6050 - Accelerometer and Gyro, Adafruit BMP280 - Barometric Pressure & Altitude Sensor, Micro SD module to Arduino Nano*. Geraadpleegd op 25 augustus 2020, van <https://www.circuito.io/app?components=10218,10333,10820,11022,11028,999979,1671987>

Finney, G. A. (2000). Analysis of a water-propelled rocket: A problem in honors physics. *American Journal of Physics*, 68(3), 223–227. <https://doi.org/10.1119/1.19415>

Microsoft. (2019, 2 oktober). *Standard conversions*. Microsoft Docs. Geraadpleegd op 24 november <https://docs.microsoft.com/en-us/cpp/cpp/standard-conversions?view=msvc-160>

NASA. (z.d.). *Rocket Stability*. Geraadpleegd op 27 december 2020, van <https://www.grc.nasa.gov/www/k-12/rocket/rktstab.html>

National Aeronautics and Space Administration (NASA). (z.d.). *Rocket Center of Pressure*. NASA Glenn Research Center. Geraadpleegd op 10 december 2020, van <https://www.grc.nasa.gov/www/k-12/rocket/rktcp.html>

National Physics Laboratory, & de Podesta, M. (2007, juni). *A guide to building and understanding the physics of Water Rockets*. National Physics Laboratory. Geraadpleegd op 14 augustus 2020, van https://www.npl.co.uk/skills-learning/outreach/water-rockets/wr_booklet_print.pdf

- Ponder, M. (2013). *Designing Your Own Model Rocket*. Ohio State University 4-H Youth Development. Geraadpleegd op 14 augustus 2020, van <https://ohio4h.org/sites/ohio4h/files/imce/Designing%20Your%20Own%20Model%20Rocket.pdf>
- SG WaterRockets. (2017, 30 juli). *Large DIY water rocket parachute* [Video]. YouTube. <https://www.youtube.com/watch?v=H-SNgZcMnX8>
- TDK InvenSense. (z.d.). *MPU-6050 | TDK*. Geraadpleegd op 24 oktober 2020, van <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>
- Todd Hampson. (2010, 29 mei). *How to make a Robinson Coupling* [Video]. YouTube. <https://www.youtube.com/watch?v=AedBMPJnQFQ>
- USWaterRockets. (2013, 12 februari). *Build a Parachute Deploy System for Water Rockets Tutorial* [Video]. YouTube. <https://www.youtube.com/watch?v=vqTKXXpD9IU>
- White, J. (2006). *Using Thrust, Weight & Control: Rocket Me into Space*. TeachEngineering.org. Geraadpleegd op 27 december 2020, van https://www.teachengineering.org/lessons/view/cub_rockets_lesson03
- White, M. (2019, 26 juli). *The MPU6050 Explained*. Programming Robots. Geraadpleegd op 24 oktober 2020, van <https://mjwhite8119.github.io/Robots/mpu6050>
- Wikipedia contributors. (2020a, augustus 31). *Adiabatic process*. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/Adiabatic_process
- Wikipedia contributors. (2020b, augustus 31). *Barometric formula*. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/Barometric_formula
- Wikipedia contributors. (2020c, november 23). *International Standard Atmosphere*. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/International_Standard_Atmosphere
- Wikipedia contributors. (2020d, november 23). *Standard gravity*. Wikipedia. Geraadpleegd op 11 december 2020, van https://en.wikipedia.org/wiki/Standard_gravity

Appendix met bijlagen

Broncode van de boordcomputer

De volledige broncode is te vinden op

https://github.com/Hoog3059/PWS_Boordcomputer/blob/master/src/final.h.

De volledige kalibratiecode is te vinden op

https://github.com/Hoog3059/PWS_Boordcomputer/blob/master/src/accelgyro_calibrate.h.

Link naar video lancering

De video van de eerste lancering is te vinden op <http://tiny.cc/5tr7tz>.

Pakket van Eisen

Raket

De raket...

- Moet herbruikbaar zijn.
- Moet een drooggewicht van minder dan 800g hebben.
- Moet een hoogte van minimaal 50m kunnen bereiken.
- Moet zijn zwaartepunt boven zijn drukmiddenpunt hebben.
- Moet niet al te duur na te bouwen zijn (niet meer dan €75,-).
- Moet een parachute hebben om weer zacht naar beneden te komen.
- Moet een boordcomputer hebben om te meten hoe hoog hij komt.
- Moet binnen een half uur kunnen worden opgezet voor gebruik.

Boordcomputer

De boordcomputer...

- Moet in een 1.5L fles passen.
- Moet kunnen bepalen wanneer de raket zijn hoogtepunt is gepasseerd.
- Moet na het bereiken van het hoogtepunt de parachute kunnen ontplooiën.
- Moet binnen 15 minuten kunnen worden klaargemaakt voor gebruik.

Lanceerplatform

Het lanceerplatform...

- Moet herbruikbaar zijn.
- Moet de raket recht houden.
- Moet vervoerbaar zijn met de auto.
- Moet ervoor zorgen dat de raket op een afstand van minimaal 15m gelanceerd kan worden.
- Moet een mechanisme hebben waardoor de lancering op afstand afgebroken kan worden.
- Mag niet meer dan €75,- kosten.
- Moet binnen een half uur kunnen worden opgezet voor gebruik.

Model

Het model...

- Moet ongeveer aan kunnen geven hoe hoog de raket zal komen bij bepaalde parameters.

Plan van Aanpak

Ons plan is om een raket te bouwen uit flessen en deze met luchtdruk en water voort te stuw, die vervolgens uit zichzelf een parachute ontplooit en zo zacht landt.

De hele raket is opgebouwd uit verschillende delen: de romp van de raket, de boordcomputer en alles eromheen en het lanceerplatform. Aan sommige onderdelen zullen we samen werken, voor andere onderdelen zullen we de taken verdelen. Samen zullen we aan de romp van de raket werken. De boordcomputer wordt aan Timo toegewezen en het lanceerplatform wordt aan Paul toegewezen.

We houden ongeveer de volgende planning aan:

- Voor het einde van september moeten de grote lijnen van het ontwerp uitgewerkt zijn. Dit betekent onderzoek doen naar wat er gebouwd moet worden en hoe het gebouwd gaat worden.
- Voor 14 oktober moeten alle onderdelen besteld zijn, zodat er daarna begonnen kan worden met bouwen.
- Voor het einde van oktober moet de boordcomputer af zijn en werken, en er moet op zijn minst begonnen zijn met het verslag. Bovendien moet de romp van de raket al zo goed als af zijn.
- Voor het einde van november moet het lanceerplatform af zijn. Ook moet het parachutesysteem gemaakt zijn en werken.
- Voor 12 december moet het bouwen en assembleren van de raket volledig af zijn. Ook moeten alle onderdelen getest zijn.

Planning voor de uitstelling door corona:

- Voor 14 december moet de raket gelanceerd zijn, en moeten de onderdelen theorie en realisatie zo goed als af zijn.
- Voor 17 december moet het verslag volledig af zijn, dus ook conclusies en evaluatie.

Planning na de uitstelling door corona:

- Op 2 januari 2021 wordt de raket gelanceerd, en voor deze datum moeten de onderdelen theorie en realisatie zo goed als af zijn.
- Voor 8 januari 2021 moet het verslag volledig af zijn, dus ook conclusies en evaluatie.

Mogelijke problemen

Van veel onderdelen zijn we nog onzeker of het ontwerp dat we bedacht hebben zal werken.

Van het aansluitpunt van het ventiel aan de tuinslang zijn we niet zeker of het de druk zal kunnen houden. Als blijkt dat het de druk niet kan houden zullen we een heel ander ontwerp moeten kiezen.

We zijn niet zeker of de versnellingsmeter en barometer precies genoeg zullen zijn om te bepalen wanneer de parachute open moet. Als blijkt dat ze dat niet zijn dan kunnen we niet verder.

Logboek

Zie volgende bladzijdes.