|  |  |  |
| --- | --- | --- |
| **Verslag** | | Versie: 0.2  Filenaam:  Datum opgesteld: 17-02-2017  Datum laatste aanpassing: 22-05-2017 |
| Naam coördinator(s) | Martin van Malten  Wouter Middel  Rob van Steene  Joan Schrasser  Arthur Kluitmans | Voor goedkeuring:  Datum: |
| Naam student  Contact | Teun Broeren  [tjbroere@avans.nl](mailto:tjbroere@avans.nl) | |
| Naam student  Contact | Sander Heijmans  [sajp.heijmans@student.avans.nl](mailto:sajp.heijmans@student.avans.nl) | |
| Naam student  Contact | Jos de Koning  [jakoning1@avans.nl](mailto:jakoning1@avans.nl) | |
| Naam student  Contact | Jeroen Reeskamp  [jreeskam@student.avans.nl](mailto:jreeskam@student.avans.nl) | |
| Naam student  Contact | Damy van Valenberg  [dfb.vanvalenberg@student.avans.nl](mailto:dfb.vanvalenberg@student.avans.nl) | |

|  |  |  |
| --- | --- | --- |
| **Versie** | **Wijzigingen** | **Datum** |
| 0.1 | Eerste opzet Verslag | 17-02-2017 |
| 0.2 | Wijzigingen lay-out | 20-02-2017 |
| 1.0 | LIDAR vooronderzoek toegevoegd | 19-04-2017 |
| 1.3 | LIDAR vergelijkingen toegevoegd | 21-04-2017 |
| 1.1 | Manifold info toegevoegd | 15-05-2017 |
| 1.2 | Sweep SDK info toegevoegd | 20-05-2017 |
| 1.3 | Aanbeveling & conclusie Manifold toegevoegd | 22-05-2017 |
| 1.4 |  |  |

|  |
| --- |
| Handtekening voor goedkeuring: |

# Voorwoord

# Samenvatting

Inhoud

[Voorwoord 3](#_Toc483216663)

[Samenvatting 4](#_Toc483216664)

[2 Inleiding 6](#_Toc483216665)

[2.1 Over Project Inspectie Drone 6](#_Toc483216666)

[2.2 Opdracht 6](#_Toc483216667)

[3 Vooronderzoek 7](#_Toc483216668)

[3.1 Websecurity 7](#_Toc483216669)

[3.2 DJI QT SDK 7](#_Toc483216670)

[3.3 LIDAR 8](#_Toc483216671)

[4 Eisen 11](#_Toc483216672)

[5 LIDAR & Vision 12](#_Toc483216673)

[5.1 LIDAR 12](#_Toc483216674)

[5.2 3D-vision 22](#_Toc483216675)

[6 Websecurity ontwerp 23](#_Toc483216676)

[7 Manifold & Sweep SDK 23](#_Toc483216677)

[7.1 Corrupt 23](#_Toc483216678)

[7.2 Onboard-SDK 23](#_Toc483216679)

[7.3 Sweep 24](#_Toc483216680)

[8 Sweep-SDK 24](#_Toc483216681)

[9 Testcases 24](#_Toc483216682)

[10 Conclusie 24](#_Toc483216683)

[11 Aanbevelingen 24](#_Toc483216684)

[11.1 Manifold 24](#_Toc483216685)

[12 Bronnenlijst 25](#_Toc483216686)

[13 Bijlage 25](#_Toc483216687)

# Inleiding

## Over Project Inspectie Drone

Drones kunnen op veel verschillende gebieden worden ingezet. Een van die gebieden is de industriële automatisering. Actemium houdt zich bezig met industriële automatisering en onder andere project inspectie drone.

Drones worden al ingezet om metingen uit te voeren op verschillende locaties. Dit gebeurt echter meestal buiten (*outdoor*) omdat de systemen die drones gebruiken (gps, magnetometer) binnen niet of in mindere mate functioneel zijn. Toch zijn er *indoor* een hoop toepassingen te bedenken waar een drone erg waardevol voor zou kunnen zijn.

## Opdracht

Bedrijven maken gebruik van grote opslagruimten met veel stellingen, om bijvoorbeeld kaas, wijn of andere producten op te slaan. Bij het opslaan van voedsel is het van belang dat het milieu (temperatuur, luchtvochtigheid, lichtintensiteit etc.) in de gaten kan worden gehouden om de kwaliteit van het product te garanderen.

Een drone kan hier worden ingezet om metingen te doen op gewenste punten en gewenste tijd. Hiervoor moet de drone echter wel binnen zijn weg kunnen vinden.

Daarvoor zijn indoor positioning systemen nodig, zodat de drone zijn weg kan vinden in de ruimte. Hiervoor wordt een camerasysteem en een aantal andere sensoren gebruikt die gemonteerd zijn aan de drone. Het voordeel van dit systeem is dat de ruimte in principe niet hoeft te worden voorzien van een *indoor position* systeem. Dit maakt het systeem toegankelijker voor bedrijven en minimaliseert de installatie tijd.

# Vooronderzoek

## Websecurity

## DJI QT SDK

## LIDAR

#### LIDAR

In dit onderzoek wordt gekeken gedaan naar hoe de indoor locatie-bepaling het beste gedaan kan worden. Om de drone autonoom te laten vliegen, is het erg belangrijk dat de drone zijn eigen locatie weet, en deze real-time wordt geüpdatet.

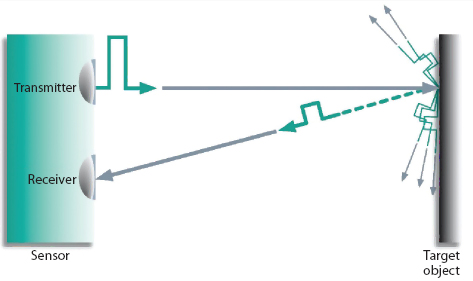
In dit onderzoek wordt er gekeken naar LIDAR zelf, hoe te realiseren, wat de mogelijkheden zijn op gebied van vision en de opties die al aanwezig zijn op de drone (ultrasoon & 3D-vision) maar er wordt ook gekeken naar andere mogelijkheden (LIDAR) die misschien wel meer toepasbaar, nauwkeurig, goedkoper etc. zijn.

#### LIDAR zelf realiseren

Er kan gekozen worden om een complete LIDAR-module te kopen en deze dan onder de drone te monteren, maar er kan ook gekozen worden om deze module zelf te maken om zo de kosten te drukken. Ook kan er veel van geleerd worden wanneer deze module zelf gemaakt wordt.

Een afstandsmeting met behulp van LIDAR kan op verschillende manieren gedaan worden, hieronder staan een aantal opties beschreven.

* Tijdmeting (ToF): hierbij wordt de tijd tussen het vertrekken van een laser puls/patroon en het terugkomen van deze puls/patroon gemeten. Aan de hand van deze tijd kan de afstand berekend worden (tijd \* lichtsnelheid/ 2). Deze methode wordt “Time of Flight” (ToF) genoemd. Deze methode wordt door verschillende bestaande LIDAR-modules gebruikt. Maar ook voor alleen afstandsmetingen, dus zonder de mapping/beeldverwerking van LIDAR.

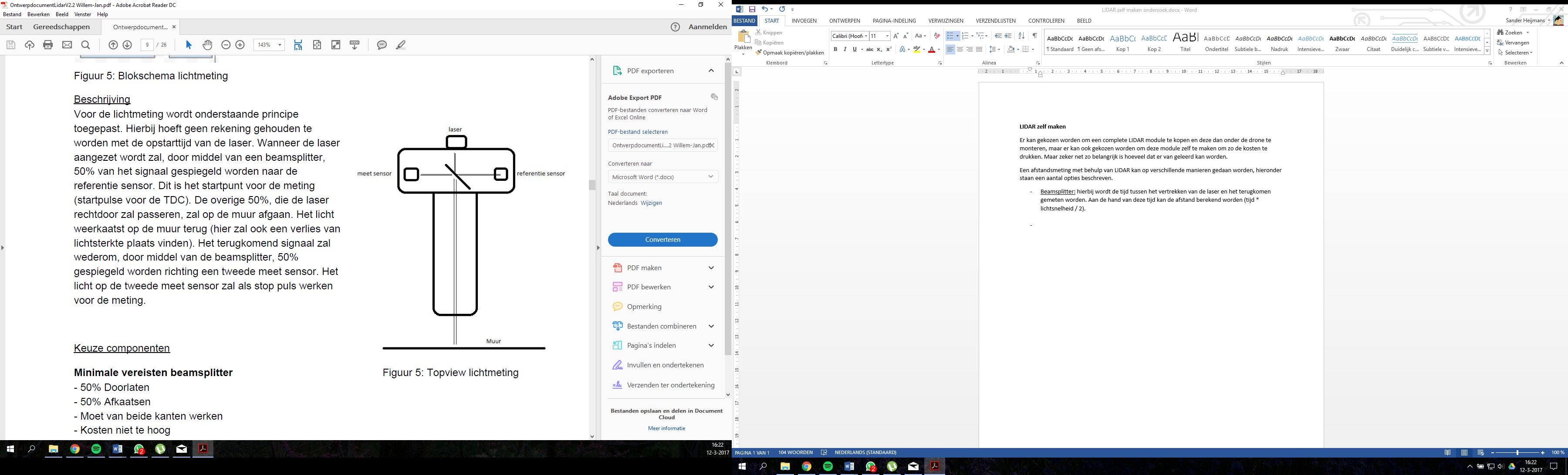


Figuur 1 bron:http://machinedesign.com/sensors/sensor-sense-pulse-ranging-technology

De laser puls die verstuurd wordt kan bestaan uit een enkele puls maar ook uit een bepaald uniek patroon dat herkend wordt door de ontvanger. Door een uniek patroon aan het signaal mee te geven kan er onderscheid gemaakt worden tussen pulsen en dit geeft de gebruiker veel meer informatie. Het kan zijn dat de laser deels gereflecteerd wordt door een object en deels doorgelaten wordt en daarna pas gereflecteerd wordt. Dit zorgt ervoor dat er meerdere afstandsmetingen gedaan kunnen worden met een enkele laser-puls/patroon. Een voorbeeld van een object dat een deel van de laser kan doorlaten is een blad van een boom. Het versturen van een patroon in plaats van een puls maakt de zender/ontvanger wel complexer.

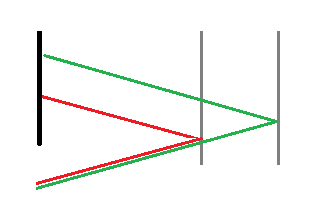
Het verzenden van een patroon zorgt er ook voor dat de ontvanger minder gevoelig wordt voor storingsbronnen omdat er een bepaald patroon moet zijn voordat de ontvanger erop reageert.

Deze methode kan met een Beam splitter gerealiseerd worden. Deze methode is hieronder schematisch beschreven. Hierbij wordt 50% van de laser naar de referentie sensor gespiegeld en de andere 50% gaat naar het object toe. Van het lichtsignaal dat terugkaatst wordt ook weer 50% naar de meetsensor gespiegeld en 50% terug naar de laser. De tijd wanneer de referentie sensor een signaal krijgt tot wanneer de meetsensor een signaal krijgt kan gemeten worden. Nu is de tijd bekend en kan de afstand berekend worden met de lichtsnelheid.



* Hoekmeting: Het principe is hieronder weergeven. De hoek waarin de laser verstuurd wordt is bekend, aan de hand van waar de laser op de camera valt kan dan de afstand berekend worden.

De rode en groene lijn stellen de lasers voor, object 1 en 2 zijn objecten waar de laser op reflecteert. Object 3 is de camera van de LIDAR-module. Hierbij is te zien dat de laser hoger op de camera valt wanneer een object zich verder van de LIDAR-module bevindt.



1

2

3

Bron: User manual Sweep V1.0

***Distance measurement***

*Sweep employs a time of flight ranging method. This technique involves transmitting a packet of micro pulses of light in a unique pattern. When this light bounces off an object and returns to the receiving detector, a correlation algorithm is used to identify the unique light pattern from ambient noise. Each light packet is different from the last, which allows multiple Sweep sensors to operate adjacent to each other without interference. The light packets that Sweep uses can vary in length, which can affect accuracy of range measurements, as well as the maximum range and update rate. Under normal operation, Sweep limits the maximum time per measurement to a value determined by the sample rate set using the LR command (see LR packet structure description). If not enough light is returned from the environment, the measurement fails, and a 1 is returned as the range value. On the other hand, if a lot of light is returned from the environment, the correlation algorithm can reach its maximum accuracy early, and can return a range value more quickly. This is what makes the update rate of Sweep variable. The value of setting a slower sample rate using the LR command is that more light will be gathered from a target, and the range measurements will be more accurate. The exact accuracy is determined by many factors, including the target surface characteristics and ambient noise, so we cannot give an exact number for relative accuracy between the different LR settings.*

***Angle measurement***

*Sweep uses an optical encoder to measure the angle of the rotating sensor head. The angle that is recorded for a range data point is the angle the sensor is at when the measurement is completed. The beginning of the scan and zero degrees is located where the status LED project out of the base of the sensor, as indicated in Figure 7.*

Voorbeelden van LIDAR producten:

* Adafruit VL53L0X Time of Flight Distance Sensor (1D sensor)
* GEOslam (indoor mapping van gebouwen)
* TIMMS indoor mapping (ook indoor mapping van gebouwen)

#### Conclusie

Vorig jaar is door een groep derdejaars ES-studenten een project uitgevoerd, waarin zij zelf een LIDAR-module wilde maken. Van dit project is het ontwerpdocument ontvangen en bekeken. Uit dit ontwerpdocument bleek dat het de groep niet is gelukt om zelf een LIDAR-module te realiseren. Aangezien dit het hele project van die groep was en het bij ons maar een deel van het project is, achten wij de kans van slagen klein.

Na overleg met de opdrachtgevers zijn de eisen waar de LIDAR-module aan moet voldoen duidelijker geworden, en is gebleken dat het niet haalbaar is om zelf een LIDAR-module te realiseren.

In plaats van zelf een LIDAR-module te realiseren, is er besloten om te gaan kijken wat er allemaal beschikbaar is op de markt, wat er toepasbaar is bij dit project, wat er voldoet aan de criteria etc.

In hoofdstuk *3 Vergelijkingen LIDAR-units* valt hier meer over te lezen.

# Eisen

De eisen zijn beschreven in onderstaande tabel. In de Prioriteit kolom, een M, S, C of W is te zien. Dit geeft de prioriteit van een eis aan. Een M (*Must have*) geeft aan dat aan deze eis voldaan moet worden. Een S (*Should have*) geeft aan dat deze eisen zeer gewenst zijn maar zonder deze eisen is het product wel bruikbaar. Een C (*Could have*) staat voor eisen die aan bod komen als er genoeg tijd is. Een W (*Won’t have*) geeft aan dat deze eisen niet aan bod komen in het project, maar kunnen in de toekomst bij een vervolgproject interessant zijn.

|  |  |  |
| --- | --- | --- |
| **#** | **Eis** | **Prioriteit** |
| 1 | De drone moet autonoom kunnen vliegen. | M |
| 2 | De drone moet een noodstop bevatten. D.m.v. deze noodstop krijgt de piloot de volledige controle. | M |
| 3 | De drone moet vaste obstakels kunnen ontwijken. | M |
| 4 | De drone moet zijn locatie bijhouden. | M |
| 5 | De drone moet zijn locatie versturen naar een User Interface. | M |
| 6 | In de User Interface is te zien waar de drone zich bevindt. | M |
| 7 | De verbinding tussen de drone en User Interface moet beveiligd zijn. | M |
| 8 | De drone moet indoor kunnen navigeren naar vooraf bepaalde punten | S |
| 9 | De drone moet bewegende obstakels kunnen ontwijken | C |
| 10 | De drone kan metingen uitvoeren van de omgeving op vooraf bepaalde punten | C |
| 11 | De drone moet 3D ruimte genereren uit de data | W |
| 12 | De drone moet meetpunten verzamelen van de ruimte d.m.v. LIDAR (laser) | W |

# LIDAR & Vision

## LIDAR

In dit onderzoek zullen verschillende LIDAR-units met elkaar vergelijken worden.

Er zullen verschillende units vergelijken worden, van goedkoop tot extreem duur.

Onderstaand is te zien waaraan een unit zal moeten voldoen.

#### LIDAR

* Nauwkeurigheid: In overleg met de opdrachtgevers is bepaald dat de nauwkeurigheid minimaal 1m moet zijn. Minder is beter.
* Prijs: Er is geen maximale prijs afgesproken. Er wordt naar de prijs gekeken in combinatie met de andere criteria.
* Gewicht: Omdat de LIDAR-unit op de drone geplaatst moet worden, mag het gewicht niet te zwaar zijn. De DJI Matrice 100 kan maximaal 600gr gewicht meedragen.
* Benodigde stroom: Omdat de LIDAR-unit op de drone geplaatst moet worden, mag het stroom verbruik niet te hoog zijn. Een hoog stroom verbruik zal betekenen dat de accu’s sneller leeg zijn waardoor er minder vliegtijd is, of dat er extra accu’s geplaatst moeten worden. Hoe lager het verbruik, des te beter.
* 2D/3D: 3D heeft de voorkeur. Er moet een mogelijkheid zijn om een ruimte 3D in kaart te brengen, dit mag ook bestaan uit meerdere 2D kaarten die op verschillende hoogtes (plane levels) gemaakt zijn.

#### LIDAR-units

In onderstaande lijst zijn alle LIDAR-units te zien die beschikbaar zijn op de markt voor commercieel gebruik en die toepasbaar zijn op ons project. Er zijn bijvoorbeeld ook units beschikbaar die meer bedoeld zijn voor autonome voertuigen, vliegtuigen of UAV-drones, maar deze wegen >3kg en vallen daardoor direct af. Dit wil zeggen dat er tijdens het kiezen van welke units er vergeleken zouden worden, er al een voorselectie is gemaakt. Dit is gedaan zodat het overzichtelijk blijft en de vergelijking duidelijk is.

* RPLIDAR A1 - 360 degree **2D** Laser Scanner Development Kit <https://www.seeedstudio.com/RPLIDAR-360-degree-Laser-Scanner-Development-Kit-p-1823.html>
* $399
* Omnidirectional laser scan
* User configurable scan rate
* Plug & Play
* Motor system power supply: 5-10V

Digital system power supply: 5V

* Detection range: 0.2-6m
* Sample rate: 2000Hz
* Resolution: 0.2% (<1%) of current distance
* Angular resolution: 1°
* 3.3V-TTL UART for communication
* Debug GUI tool & SDK
* Weight: 170gr (core)
* RPLIDAR A2 - The Thinnest LIDAR (**2D**)

<https://www.seeedstudio.com/RPLIDAR-A2---The-Thinest-LIDAR-p-2687.html>

* $449
* Omnidirectional laser scan
* User configurable scan rate
* Detection range: 0.15-6m
* <0.5% for <1.5m

<1% of current distance (>1.5m)

* Angular resolution: 0.9°
* Sample rate: 4000Hz
* Scan rate: 5-15Hz (typ.: 10Hz)
* Power supply: 5V
* Power current: off: 200mA, on: 400mA
* DE-LIDAR TF01

<https://www.seeedstudio.com/DE-LIDAR-TF01-p-2789.html>

* $139.99
* Detection range: 0.3-10m @90% Reflection）
* Detection frequency: 500Hz
* Weight: <50gr
* Accuracy: centimeter-level
* Power supply: 7.4-12V
* Power consumption: 1W
* Minimum detectable object size @5m: 4-6cm (1%)
* Resolution: 0.5cm
* Sweep V1 360° Laser Scanner

<http://www.robotshop.com/en/sweep-v1-360-laser-scanner.html>

* $349
* Range: 40m
* Sample rate: 1000Hz
* Rotation frequency: 2-10Hz
* Power supply: 5V
* Power consumption: 450mA
* Single plane scanner (2D)
* Weight: 120gr
* Resolution: 1%
* 2 serial ports for low level microcontrollers
* Velodyne LiDAR Puck LITE (VLD-16)

<http://velodynelidar.com/vlp-16-lite.html>

* + $7999
  + Weight: 590gr
  + Range: 100m
  + 300.000 points per second
  + Power consumption: 8W
  + 3D
  + Accuracy: 3cm (typ.)
  + Rotation rate: 5 – 20Hz
  + Power supply: 9 – 18V
* LIDAR-Lite 3 Laser Rangefinder

<https://www.sparkfun.com/products/14032>

* + $150
  + Range:0-40m
  + Accuracy: <5m: ±2.5cm, >5m: ±10cm
  + Power: 4.75-5 VDC; 6 V Max
  + Current consumption: 105ma, idle; 130ma, continuous
  + Sample rate: 1-500Hz
  + Weight: 22gr
  + Update rate: 270Hz typ.
  + Repetition rate: ~50Hz typ., 500Hz max
  + Power supply: 5V
* Adafruit VL53L0X Time of Flight Distance Sensor

<https://www.adafruit.com/product/3317>

* $14.95
* ~30 to 1000mm
* Power supply:3-5V
* Weight: 1.3gr
* Only sensor
* Need to build a rotating platform for full LIDAR use
* Riegl VUX-1UAV

<http://products.rieglusa.com/product/all-categories-unmanned-scanners/vux-1uav-survey-grade-unmanned-laser-scanners>

* + $??? (expensive)
  + 3-920m
  + Accuracy: 10mm
  + Sample rate: 50-550kHz
  + Weight: 3.5kg
  + Used in UAV
  + 240GB SSD to store data
  + 200 scans/second
* YellowScan Mapper

<http://www.yellowscan.fr/products/yellowscan-mapper>

* + Price: N/A
  + Accuracy: 10cm
  + Scan frequency: 40kHz
  + Weight: 2.1kg
  + Power consumption: 10W
* GEO-MMS SAASM LIDAR SENSOR

<http://geodetics.com/product/geo-mms-saasm/>

* Prijs op aanvraag
* Sample rate: 125Hz
* Power supply: 10-30V
* Power consumption: >2A
* Single plane scanner (2D)
* Weight: <2.26kg
* Uses VLP-16 PUCK
* LeddarVu

<http://leddartech.com/modules/leddarvu/>

* €520
* Sample rate: 100Hz
* Range: 22m
* Power supply: 12V
* Power consumption: 2W
* Fixed beam, no rotating parts
* Accuracy: 5cm
* Distance precision: 6mm
* Weight: 107g

#### Vergelijkingstabel

Op onderstaande pagina is het overzicht van de verschillende LIDAR-units te vinden. Deze tabel is gemaakt zodat er snel en duidelijk gezien kan worden wat de specificaties, voordelen en nadelen van elke LIDAR-unit zijn.

Elke kleur staat voor een aantal punten. Op deze manier is het makkelijk om een vergelijking te maken.

* + De groene kleur geeft aan dat dit positief is 🡪 5 Pt.
  + De lichtgroene kleur geeft aan dat dit positief is, maar nog niet optimaal 🡪 4 Pt.
  + De gele kleur geeft aan dat dit neutraal is 🡪 3 Pt.
  + De oranje kleur geeft aan dat dit negatief is, maar nog niet heel beroerd 🡪 2 Pt.
  + De rode kleur geeft aan dat dit negatief is 🡪 1 Pt.
  + De witte kleur geeft aan dat dit criterium niet van toepassing is 🡪 0 Pt.

#### Criteria

In het begin van dit onderzoek is al genoemd welke criteria belangrijk is bij het uitzoeken van een LIDAR-unit. Tijdens het onderzoeken en selecteren van verschillende LIDAR-units bleek dat er nog meer criteria belangrijk was. Dit is hieronder te vinden.

Elke criterium heeft een weging, van 1x tot 3x, te zien achter de betreffende criteria. Dit komt doordat bepaalde criteria belangrijker is dan andere. Hoe zwaarder de weging, hoe belangrijker het criterium.

* ***Price (3x)***: De prijs is een belangrijke factor. Hoe lager de prijs, des te beter. Echter wordt hier wel met een schuin oog gekeken naar de andere criteria
* ***2D/3D (1x)***: Een aantal LIDAR-units zijn in staat meteen een 3D-map te maken. Dit scheelt tijd en dus geld wanneer het gebied in kaart gebracht moet worden.
* ***Detection range (3x)***: De range geeft aan toe welke maximale afstand de unit nauwkeurig kan meten. Wordt deze maximale afstand overschreden, want kan het zijn dat de metingen niet meer de opgegeven nauwkeurigheid halen. Een hogere range is dus beter.
* ***Weight (3x)***: Het gewicht speelt bij dit project een belangrijke rol. Omdat de unit op een drone geplaatst zal worden, mag het gewicht niet te hoog zijn.
* ***Resolution/Accuracy (3x)***: In overleg met de opdrachtgevers is bepaald dat de nauwkeurigheid minimaal 1m moet zijn. Minder is beter.
* ***Angular Resolution (1x):*** Dit geldt alleen bij de units die 360° meten. Het geeft aan in hoeveel “stappen” er wordt gemeten. Bij een angular resolution van 1° wordt er dus 360 x 1° gemeten.
* ***Sample rate (2x)***: De sample rate geeft aan met welke frequentie de data wordt doorgestuurd. Een hogere sample rate geeft dus betere data, dus hoger is beter.
* ***Scan rate (2x)***: De scan rate geeft aan met welke frequentie er gescand wordt. Bij een unit die 360° meet geeft het dus ook aan hoe snel deze ronddraait. Een hogere frequentie is nauwkeuriger en dus beter.
* ***Supply voltage (2x)***: Het is van belang dat we weten wat de voedingsspanning is. De Matrice 100 heeft standaard 5V outputs aan boord, die je dus kan gebruiken. Bij een afwijkende spanning moet er zelf een step-up/-down circuit gemaakt worden.
* **Power consumption (3x)**: De accu’s van de drone hebben maar een beperkte capaciteit, en al het extra vermogen wat nodig is moet hieruit gehaald worden. Dus betekent dat bij een hoger verbruik, de vliegtijd korter wordt. Lager is dus beter.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Product / Requirements  weging | Price  *3x* | 2D/3D  *1x* | Detection range  *3x* | Weight  *3x* | Resolution / Accuracy  *3x* | Angular resolution *1x* | Sample rate  *2x* | Scan rate  *2x* | Supply voltage  *2x* | Power consumption  *3x* | Total score |
| RPLIDAR A1 | $399 | 2D | 0.2 – 6m | 170gr | <1% of current distance | 1° | 2000Hz | 5.5Hz  Max.: 10Hz | Motor: 5-10V (according speed)  Digital: 5V | Motor: 100mA: 500mW  Scanner: 200mA: 1W | **85** |
| RPLIDAR A2 | $449 | 2D | 0.15 – 6m | 190gr | <1% of current distance | 0.9° | 4000Hz | 5-15Hz (typ.: 10Hz | 5V | On: 400mA: 2W  Off: 200mA: 1W | **84** |
| DE-LIDAR TF01 | $140 | 2D | 0.13 – 10m | 50gr | 4-6cm @5m (1%) | 1° | 500Hz | N/A | 7.4-12V | 1W | **83** |
| Sweep V1 | $349 | 2D | Max. 40m | 120gr | 1% | 0,5° | 1000Hz | 2-10Hz | 5V | 400mA: 2W | **88** |
| Velodyne VLD-16 | $7999 | 3D | Max. 100m | 830gr  590gr (LITE) | Typ.: 3cm | 0,1 – 0,4° | 300kHz | 5-20Hz | 9-18V | 8W | **71** |
| LIDAR-Lite v3 | $150 | 1D | Max. 40m | 22gr | <5m: ±2.5cm ≥5m: ±10cm | - | Typ.: 270Hz  Fast mode: 650Hz  >1000Hz for short range | 50Hz default  500Hz max. | 5V | Idle: 105mA, 525mW Continuous: 130ma, 650mW | **88** |
| Adafruit VL53L0X Time of Flight Distance Sensor | $14.95 | 1D | 30 – 1000mm | 1.3gr | 3-12% | - | - | - | 3-5V | N/A | **50** |
| Riegl VUX-1UAV | N/A | 2D | 3 – 920m | 3.5kg | 10mm | - | 50 – 550kHz | 200Hz | N/A | N/A | **51** |
| YellowScan Mapper | N/A | 2D | N/A | 2.1kg | 10cm | - | 40kHz | N/A | N/A | 10W | **43** |
| LeddarVu8 | €520 | Fixed 3D beam | Max 215m | 128.5g | 5cm | N/A | N/A | 100Hz | 12V | 2W | **73** |

#### Vergelijkingstabel

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Product / Requirements | Price  *3x* | 2D/3D  *1x* | Detection range  *3x* | Weight  *3x* | Resolution/ Accuracy  *3x* | Angular resolution *1x* | Sample rate  *2x* | Scan rate  *2x* | Supply voltage  *2x* | Power consumption  *3x* | Total score |
| RPLIDAR A1 | 12 | 2 | 9 | 12 | 12 | 3 | 6 | 6 | 8 | 15 | **85** |
| RPLIDAR A2 | 9 | 2 | 9 | 12 | 12 | 4 | 8 | 6 | 10 | 12 | **84** |
| DE-LIDAR TF01 | 15 | 2 | 9 | 15 | 12 | 3 | 4 | 2 | 6 | 15 | **83** |
| Sweep V1 | 12 | 2 | 12 | 12 | 12 | 4 | 6 | 6 | 10 | 12 | **88** |
| Velodyne VLD-16 | 3 | 5 | 15 | 3 | 15 | 5 | 10 | 8 | 4 | 3 | **71** |
| LIDAR-Lite v3 | 15 | 1 | 12 | 15 | 6 | 0 | 4 | 8 | 10 | 15 | **86** |
| Adafruit VL53L0X Time of Flight Distance Sensor | 15 | 1 | 3 | 15 | 3 | 0 | 0 | 0 | 10 | 3 | **50** |
| Riegl VUX-1UAV | 3 | 2 | 3 | 3 | 15 | 0 | 10 | 10 | 2 | 3 | **51** |
| YellowScan Mapper | 3 | 2 | 3 | 6 | 12 | 0 | 10 | 2 | 2 | 3 | **43** |
| LeddarVu8 | 6 | 2 | 12 | 12 | 12 | 1 | 2 | 8 | 6 | 12 | **73** |

#### Vergelijkingstabel

#### Keuze LIDAR-unit: **Sweep V1**

Zoals te zien in bovenstaande tabellen haalt de Sweep V1 de meeste punten. Dit betekent dat deze LIDAR-unit het beste voldoet aan onze criteria.

Hieronder meer informatie over deze unit.

#### Bedrijf achter Sweep V1

Het bedrijf achter de Sweep V1 is Scanse.io (<http://scanse.io/>).

Dit is een nieuw bedrijf, opgericht puur voor de Sweep V1. Dit is het enige product wat ze verkopen.

Het is begonnen met een Kickstarter-project (<https://www.kickstarter.com/projects/scanse/sweep-scanning-lidar>). Hier is te zien dat er in totaal $272,990 opgehaald is, terwijl $230,000 het doel was: dit is dus ruim overschreden. Op de Kickstarter pagina zijn ook een heleboel filmpjes te zien met informatie over de verschillende toepassingen. Op een van deze filmpjes is te zien dat de Sweep V1 ook op een drone wordt gebruikt, net als bij dit project.

Tijdlijn Sweep V1:

* 2 maart 2016 : Project gestart
* 8 april 2016 : Kickstarter passeert 100% van doel
* 19 januari 2017 : Begonnen met shipping
* 25 januari 2017 : Shipping is gepauzeerd (onbekende reden)
* 13 februari 2017 : Fout in design verbeterd
* 6 april 2017 : Minor tweaks, shipping begint volgende week
* 20 april 2017 : Shipping has begun (eindelijk)
* 25 april 2017 : Sweep V1 ontvangen

Ondanks de pauze en verbeteringen die doorgevoerd zijn, zijn er al een heel aantal Sweep V1-units verstuurd. Op de site is te zien dat er een Community is opgericht, waarin gebruikers hun ervaringen/problemen delen en elkaar helpen.

#### Bestelling

Op het moment dat wij de Sweep V1 wilde bestellen, was deze niet leverbaar. Na contact te hebben gehad met Scanse.io, bleek dat dit kwam doordat er een probleem was gevonden in het design, en deze verbeterd moest worden. Helaas hadden ze geen units meer liggen die eerder verzonden konden worden, dus moest er gewacht worden op de volgende levering.

Scanse.io verkoopt de Sweep V1 in Europa via Robotshop.com (<http://www.robotshop.com/en/sweep-v1-360-laser-scanner.html>). Via de opdrachtgevers van dit project is er contact geweest met Robotshop.com. Robotshop.com werkte uitstekend mee, en raadde aan om de unit te bestellen via het CA-warehouse, i.p.v. het EU-warehouse. De verzendkosten zijn dan wel hoger, maar de levertijd is korter (14 dagen korter) en zekerder. Er is dus gekozen om de Sweep V1 te laten bestellen bij het CA-warehouse, waardoor deze eerder aankomt in NL.

#### Uitlezen Sweep

Om een map te maken van het gebied/ruimte/magazijn waar de drone zich bevindt, is het (natuurlijk) nodig om de data uit te lezen.

Op de volgende link (<https://s3.amazonaws.com/scanse/Sweep_user_manual.pdf>) is heel duidelijk beschreven hoe de data wordt verstuurd (encoding & decoding), hoe de packages eruit zien, structuur, en het format.

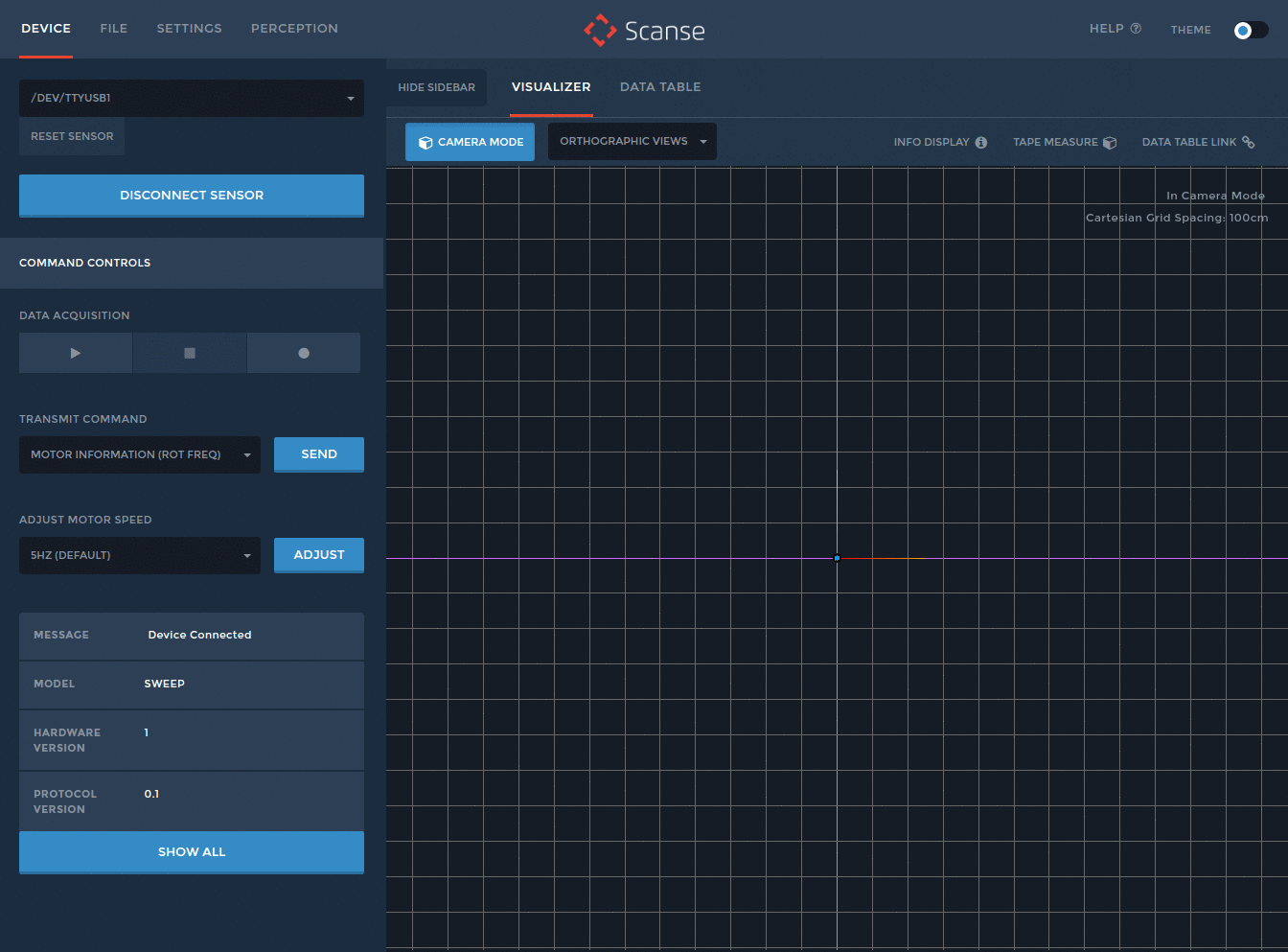
#### Sweep Visualizer

Op de site van Scanse.io is er een downloadlink voor de Sweep Visualizer. Dit is een programma ontwikkeld door Scanse.io, speciaal voor de Sweep V1. Op de volgende link (<https://vimeo.com/157948695>) is kort te zien hoe dit werkt.

Van deze Sweep Visualizer is een Windows, Mac en Ubuntu 14.04 (64-bit) versie gemaakt. Dit komt goed uit voor dit betreffende project, want de Manifold op de drone maakt ook gebruik van Ubuntu (Linux).

De Sweep V1 unit kan aangesloten worden op een computer en daarna verbinding maken met dit programma.

Met de Visualizer is het mogelijk om data te “recorden/opnemen.” Dit wil zeggen dat de data die wordt ontvangen voor een bepaalde tijd, opgeslagen kan worden. Uit onderzoek is gebleken dat deze data op wordt geslagen als .JSON-file. Aangezien er in de software al gebruik wordt gemaakt van JSON, is dit een groot en nuttig voordeel.



In de volgende link (<https://support.scanse.io/hc/en-us/sections/207173067-Sweep-Visualizer>) is te zien wat alle mogelijkheden zijn met dit programma.

#### Sweep SDK

Ook is er vanuit Scanse.io een SDK (Software Development Kit) beschikbaar gesteld d.m.v. GitHub.

Deze SDK bevat o.a.:

* Libsweep: low-level ABI/API-stable C library

Dit is de library die alle methodes etc. die de Sweep nodig heeft om te communiceren met een pc bevat.

* SweepPy: Python bindings

Bevat methodes voor communicatie met Python. Werkt met Python2 en Python3.

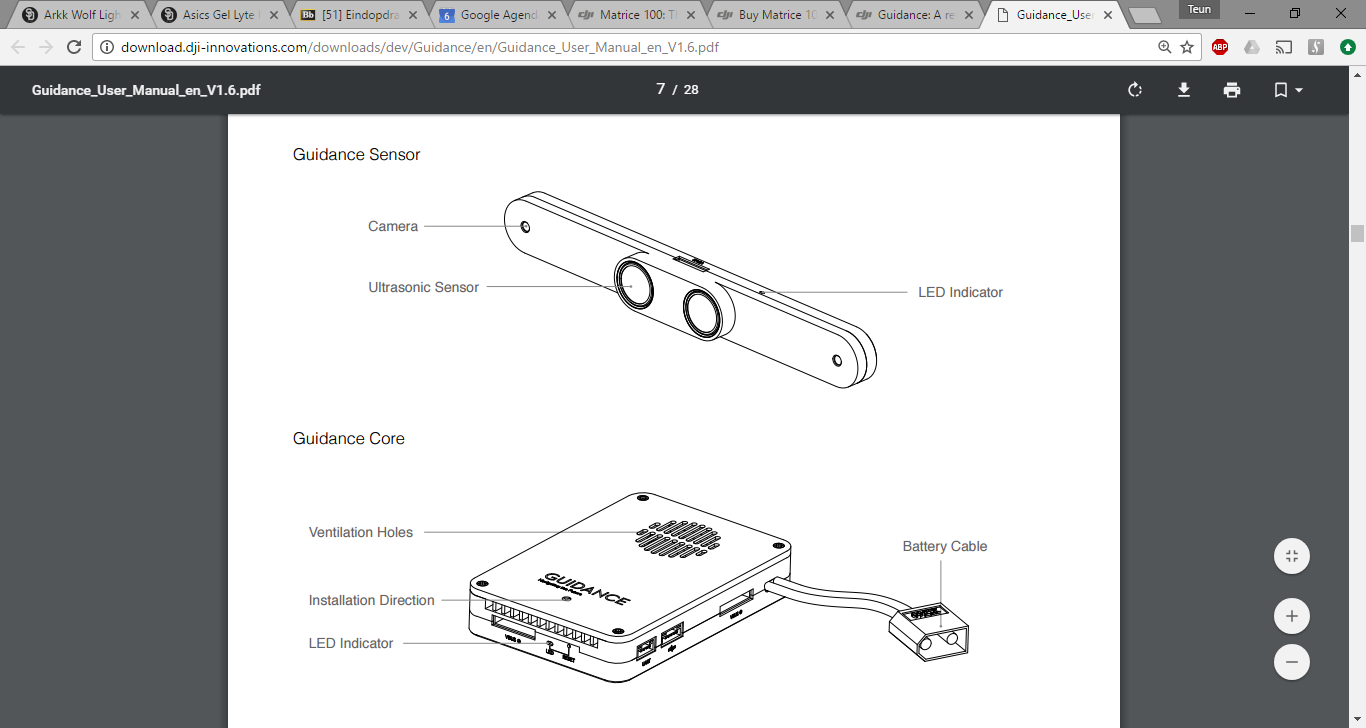
* SweepJs: NodeJS bindings

Dit bevat de methodes etc. voor communicatie met Linux. Zorgt ervoor dat de Sweep V1 kan worden aangestuurd via Linux. Bevat methodes zoals *startScanning(), stopScanning(), sweep.getMotorSpeed(), sweep.setMotorSpeed(number), sweep.getSampleRate(), sweep.setSampleRate(number)* etc.

## 3D-vision

Naast LIDAR is er ook de mogelijkheid om gebruik te maken van guidance/vision systeem van DJI zelf (<https://www.dji.com/guidance>).

Guidance is een visueel detectiesysteem van DJI. Dit systeem bestaat uit:

* Guidance Core x1
* Verzamelt de data via VBUS-poorten
* Stuurt verwerkte data door naar de DIJ Matrice flight control system
* Guidance Sensor x5
  + Verzamelt beelden en ultrasone data
  + Heeft 2 camera’s
  + Heeft 2 ultrasone sensoren
  + Stuurt data door naar Score via VBUS-kabel
* VBUS Cable x5
  + Verbindt Sensor met Core
* VBUS Cable (long) x1
  + Reserve kabel
* CAN-bus Cable x1
  + Verbindt Core met DJI Matrice flight controller

#### Conclusie 3D/Guidance

Met de Guidance set van DJI is het mogelijk om 3D beelden te maken.

Hier heeft DJI zelf geen applicatie voor, maar op hun site vermelden ze het volgende:

“People can program their own application to get information from the Guidance, which can sense the 3D world by calculating the dense depth images in real-time. The depth of most pixels in the field of view are depicted to an accuracy of within a few centimeters. This gives Guidance the capability to interact with the real world, and enables developers to build all kinds of customized 3D applications.”

Hieruit blijkt dus dat het mogelijk is om 3D beelden te creëren. Helaas is het echter niet mogelijk, zoals wel beweerd wordt in bovenstaande tekst, om dit real-time te doen.

De afbeeldingen die uit de Guidance-set komen, die zijn real-time. Om van deze afbeeldingen 3D beeld te maken, moeten deze verwerkt worden in OpenCV. Hier wordt de diepte uit de afbeeldingen gehaald, zodat er een 3D beeld gemaakt kan worden. De diepte uit deze beelden, 3D beeld creëren en dit terug sturen naar de Matrice 100 duurt dermate lang dat het niet meer real-time is.

Dit betekent de totale beeldverwerking te lang duurt om de drone om deze manier van een ‘map’ van de ruimte te voorzien.

Hierdoor valt deze optie helaas af. Het was ideaal geweest als de beeldverwerking wel real-time plaats kon vinden. Dan was er een duidelijk en stabiel beeld van de omgeving geweest. Daardoor vervalt deze optie.

# Webservice & GUI

## Webservice

Uit zichzelf verzameld de drone alleen data van zijn huidige positie en eventuele meting data van de aangesloten apparatuur. Om de drone acties uit te laten voeren is het van belang dat de gebruiker een verbinding kan maken met de drone. Via deze verbinding moet de gebruiker onder andere de data van de drone kunnen opvragen en nieuwe commando’s naar de drone kunnen sturen.

Er is gekozen om de webservice met een REST-API te bouwen omdat dit tegenwoordig de standaard is voor het bouwen van webapplicaties en het zorgt voor schone en overzichtelijke code.

|  |
| --- |
| If(path==/api/position"){ |
|  | if (request.getMethod() == "GET") { |
|  | getPosition(response); |
|  | } else if (request.getMethod() == "PUT") { |
|  | // extract data |
|  | setPosition(request, response); |
|  | } |

Afhankelijk van de methode waarmee de methode de URL wordt aangeroepen in de code wordt er een functie op de webserver aangeroepen die het verzoek verder afhandelt.

Voor het versturen van data is gekozen voor een JSON opmaak omdat dit makkelijker in gebruik is vergeleken met XML en ook compacter waardoor berichten sneller verzonden kunnen worden.

**Voorbeeld van positie data van de drone:**

{"position": [

     {"Xaxis": "45"},

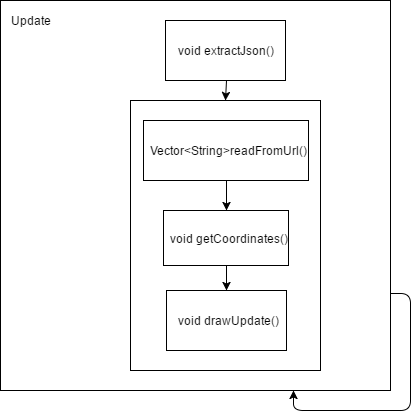
     {"Yaxis": "123"},

     {"Zaxis": "2"}

   ]

 }

}}

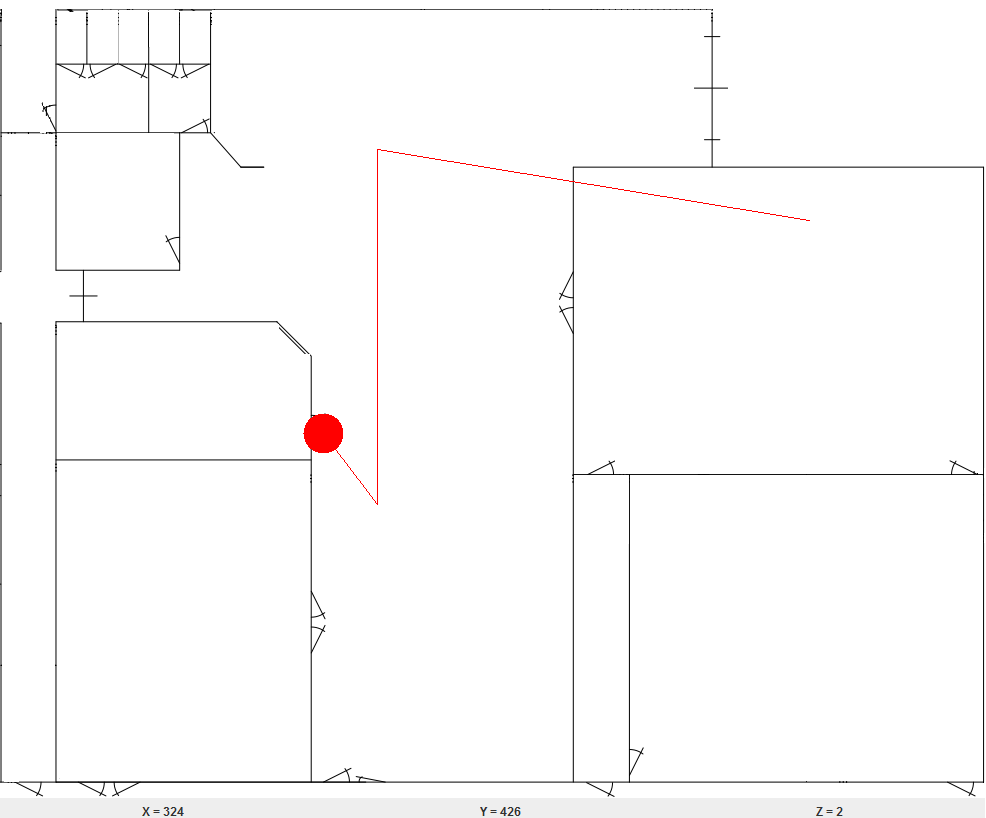


## Gui

Om gebruik te kunnen maken van de webservice is er een applicatie gemaakt die verbinding maakt met de webservice en data kan opsturen en opvragen. De applicatie heeft verschillende taken:

* Weergeven van het pad dat de drone heeft afgelegd
* Huidige positie van de drone weergeven
* Meetwaarde weergeven die gemaakt zijn op de vooraf ingestelde positie
* Nieuw pad instellen voor de drone
* Drone commando’s geven (stop, start, pause, resume)

Van deze taken zijn tot nu toe alleen de eerste twee geïmplementeerd en is er een begin gemaakt aan het instellen van een pad voor de drone om te volgen.

De GUI-applicatie probeert een SSL-verbinding te maken met de server die draait op de drone. Als dit gelukt is wordt met een vaste interval de huidige positie van de drone opgevraagd van de webserver. Deze data komt terug in een JSON-format en wordt omgezet naar een instantie van de coordinate-klasse. De klasse wordt toegevoegd aan een lijst met eerder opgehaald coördinaten en doorgestuurd naar de draw functie waar alle coördinaten op een plattegrond getekend worden. De drone zelf wordt op de meest recente plek getekend zodat je in een oogopslag kan zien waar de drone zich bevindt.

# Websecurity

## Probleem:

De drone moet vanuit een GUI kunnen worden gevolgd en commando’s verwerken. Deze gui draait op een van de computers aanwezig bij het bedrijf. Momenteel kan iedereen de data van de drone uitlezen en commando’s uitvoeren, hiervoor is alleen een verbinding nodig met het draadloze netwerk. Dit is niet wenselijk. Daarom is het de bedoeling dat de verbinding tussen de GUI en de drone beveiligd wordt. Hierdoor is het niet meer mogelijk voor 3e partijen om de drone commando’s te sturen of om mee te luisteren met de data stream.

## Oplossing:

De standaard voor het versleutelen van een verbinding is Secure Sockets Layer (SSL). SSL werkt door middel van public en private key encryption. Dit houdt in dat er een public key is waarmee de data versleuteld wordt, vervolgens kan deze alleen ontsleuteld worden met de private key, die alleen in het bezit is van de ontvanger.

## Certificaten

SSL werk doormiddel van certificaten. Een SSL-certificaat bevat wat informatie over het ‘subject’. Dit gaat over de eigenaar van het certificaat, zoals de website van het certificaat.

Certificaten kunnen vertrouwd worden door een cliënt, dat betekend dat ze er vanuit gaan dat de source van het certificaat betrouwbaar is.

Een certificaat kan vertrouwd worden door dit op de cliënt zelf aan te geven dat het certificaat vertrouwd is.

Het is ook mogelijk om je als cliënt aan te melden bij bijvoorbeeld de certificaten ‘service’ van google. Google zal dan bepalen welke certificaten vertrouwd worden. Dit word in praktijk veel gebruikt.

Voor de drone is alleen een self-signed certificaat nodig. Dit is dus niet vertrouwd door bekende services zoals google. Dit is niet nodig omdat het certificaat alleen lokaal gebruikt wordt. En dus niet in het world wide web.

Bij het communiceren middels dit certificaat kan de cliënt een waarschuwing krijgen dat het certificaat niet vertrouwd is. Deze waarschuwing kan genegeerd worden.

Voor de veiligheid van de verbinding is het wel belangrijk dat gecontroleerd word of het certificaat echt hoort bij de drone, en niet bij een ‘fake server’.

Bij de drone wordt er gebruik gemaakt van OpenSSL. Dit kan als volgt worden geïnstalleerd op Linux:

* *sudo apt-get install openssl*
* *Het genereren van een self signed certificaat op linux:*

*openssl req -new -newkey rsa:2048 -nodes -keyout yourdomain.key -out yourdomain.csr*

* *Het genereren van een certificaat op MAC.*

*open Keychain Access*

*navigeer Keychain Acces > Certificate Assistant > Create A Certificate…*

*Kies dan de optie Self Signed Root.*

*Vul bij de andere velden de informatie naar wens in.*

Nu is er een beveiligde verbinding mogelijk. Bij het leggen van de verbinding tussen de client en de drone is het wel belangrijk dat er gecontroleerd wordt of de client wel gemachtigd is om verbinding te maken met de drone. Hiervoor zal het wachtwoord gebruikt worden waarmee ingelogd wordt via in de GUI

# Manifold & Sweep SDK

In dit project is er gebruik gemaakt van een Manifold computer. Deze is ontwikkeld door DJI. De Manifold was al aanwezig voordat de huidige projectgroep aan dit project ging werken.

## Corrupt

Toen de Manifold overhandigd werd was deze corrupt. Er was geen back-up van het systeem of de data die erop stond. De al ontwikkelde software voor de Manifold was dus weg. Er zijn een aantal stappen uitgevoerd om dit probleem op te lossen.

De Manifold leek operationeel te zijn, maar dit bleek niet zo te zijn. De computer startte op, maar er bleek een probleem met de rechten te zijn over het filesystem. Het sudo commando werkte bijvoorbeeld niet. Er verscheen dan een foutmelding dat de file sudo niet van de juiste user group was. In de handleiding van DJI staat dat het wachtwoord van root random gegenereerd wordt. Het was dus ook niet mogelijk om in te loggen als root en de rechten goed te zetten.

Als eerste is er geprobeerd de originele image van DJI terug te flashen. Dit had geen effect. De Manifold gedroeg zich nog steeds hetzelfde. Om te controleren of het flashen van de originele image effect had is er in het RFS van de image een nieuw leeg bestand aangemaakt. Na het flashen van deze aangepaste image verscheen dit bestand ook op de Manifold. Dit was een bevestiging dat het flashen wel werkt.

Hierna is er geprobeerd om met chmod de rechten van alle bestanden in de /bin folder aan te passen. De rechten van de bestanden zijn gezet op user 0 en group 0. Daarna is de image nog een keer geflasht. Dit keer werkte sudo weer normaal. Als laatste is er een nog een update en upgrade uitgevoerd, waarna het hele systeem weer goed werkte.

## Onboard-SDK

Om te communiceren met de drone via de Manifold levert DJI een SDK. Via deze SDK kan je verschillende sensoren uitlezen van de drone, maar je kan hiermee ook de drone aansturen.

De SDK is beschikbaar als source via de website van DJI. Deze moet vervolgens zelf gecompileerd worden. DJI heeft een guide om dit te compileren, maar deze werkt niet volledig. Dit kwam doordat ze in de guide gebruik maken van een verouderde versie van QT, terwijl er in dit project gebruikt wordt gemaakt van de laatste versie van QT.

Na het updaten naar versie 5 van QT moest het pad van de *qmake* nog worden aangepast. Dit gebeurde niet automatisch. Na het maken van een nieuwe symbolic link ging dit wel goed en konden we de SDK ook succesvol compileren.

Met *QT Creator* is er een project aangemaakt en de SDK geïmporteerd. Met een example van de SDK konden de verschillende sensoren van de drone worden uitgelezen.

## Sweep

Scanse levert bij de Sweep ook een Visualizer programma. Dit is te downloaden van hun website. Hiermee kan je eenvoudig en snel de Sweep uitlezen op je computer. Daarnaast kan je via de Visualizer ook de Sweep updaten. Om te ontwikkelen voor de Sweep heb je de SDK nodig.

#### Sweep-SDK

De SDK is te downloaden van GitHub. Deze moet vervolgens gecompileerd worden. Dit was gelukt op een virtual machine (VM). De SDK werkte naar behoren met de Sweep LIDAR. We konden verschillende waardes uitlezen. Ditzelfde is ook geprobeerd op de Manifold, maar daar miste een driver voor de FTDI die wordt gebruikt met de Sweep. Er is geprobeerd om de drivers van FTDI te installeren, maar dit was niet gelukt. Daarom was er besloten om gebruik te maken van een andere FTDI, de CP210. Helaas waren er ook problemen met deze FTDI. De CP210 werkt prima met de Visualizer van Scanse. Je kan de Sweep gewoon uitlezen, maar als deze wordt gebruikt met de SDK kan er geen verbinding worden gemaakt. De SDK wacht op een commando, maar dit wordt nooit gestuurd. Er kan geen verschil worden gevonden tussen de FTDI’s, behalve dat er een andere IC op zit. Er was ook contact opgenomen met de developers van de SDK: <https://github.com/scanse/sweep-sdk/issues/72>

# Conclusie

## Manifold

Er hebben veel problemen plaats gevonden met de Manifold of de software die op de Manifold moest draaien. Veel software is eerst succesvol gecompileerd in een VM, maar compileren op de Manifold was vaak niet succesvol. De Manifold is intussen verouderd en werd ook een tijd niet meer geleverd. Er mist vaak te veel software, waardoor er veel niet meer lukt om te compileren of drivers zijn niet beschikbaar.

Desondanks het aantal tegenslagen die er hebben plaatsgevonden tijdens dit project, is er een duidelijk beeld gevormd hoe dit project verder gerealiseerd moet worden. Het doel is zeker haalbaar en dit project moet worden voortgezet.

# Aanbevelingen

## Manifold

Wij adviseren om te onderzoeken of het mogelijk is een volledige Ubuntu distributie op de Manifold te installeren.

De Manifold mist momenteel te veel drivers, waardoor het bijvoorbeeld niet mogelijk is de Sweep erop aan te sluiten. Als dit niet lukt, moet er misschien gekozen worden voor een andere embedded computer die beter geschikt is voor deze toepassing. Dit zou een van de taken van de volgende projectgroep zijn.

# Bronnenlijst

# Bijlage