|  |  |  |
| --- | --- | --- |
| **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 onderzoek toegevoegd | 19-04-2017 |
| 1.1 | Manifold info toegevoegd | 15-05-2017 |
| 1.2 | Sweep SDK info toegevoegd | 20-05-2017 |
|  |  |  |

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

# Voorwoord

# Samenvatting

Inhoud

[Voorwoord 3](#_Toc475107899)

[Samenvatting 4](#_Toc475107900)

[1 Inleiding 6](#_Toc475107901)

[1.1 Over Project Inspectie Drone 6](#_Toc475107902)

[1.2 Opdracht 6](#_Toc475107903)

[2 Vooronderzoek 7](#_Toc475107904)

[2.1 Websecurity 7](#_Toc475107905)

[2.2 DJI QT SDK 7](#_Toc475107906)

[2.3 LIDAR 7](#_Toc475107907)

[3 Eisen 8](#_Toc475107908)

[4 Websecurity ontwerp 9](#_Toc475107909)

[5 Software besturing ontwerp 9](#_Toc475107910)

[6 Testcases 9](#_Toc475107911)

[7 Conclusie 9](#_Toc475107912)

[8 Aanbevelingen 9](#_Toc475107913)

[9 Bronnenlijst 9](#_Toc475107914)

[10 Bijlagen 9](#_Toc475107915)

# 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

### Inleiding

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.

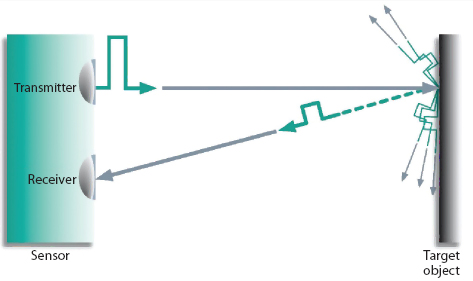
In dit onderzoek wordt er gekeken naar LIDAR zelf, hoe te realiseren, wat de mogelijkheden zijn op gebied van vision en de opties die al aanwezig zijn op de drone (ultrasoon & 3D-vision) maar er wordt ook gekeken naar andere mogelijkheden (LIDAR) die misschien wel meer toepasbaar, nauwkeurig, goedkoper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

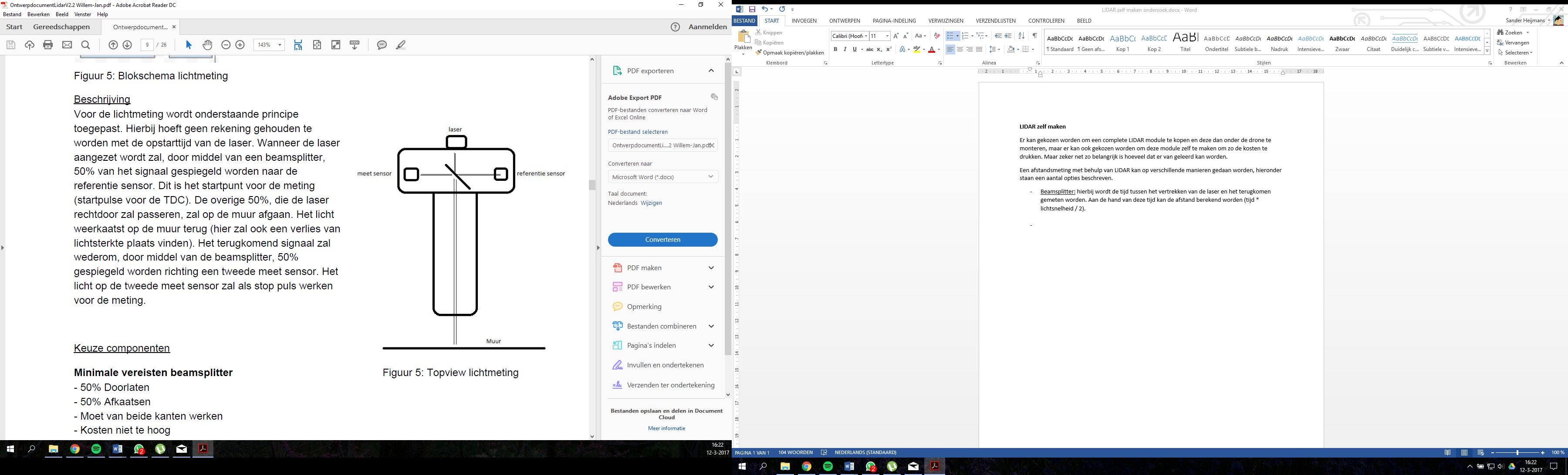


Figuur 1 bron:http://machinedesign.com/sensors/sensor-sense-pulse-ranging-technology

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.

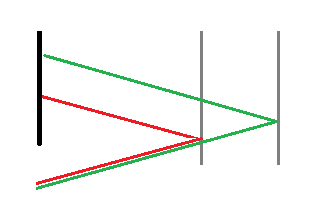
Het verzenden van een patroon zorgt er ook voor dat de ontvanger minder gevoelig wordt voor storingsbronnen omdat er een bepaald patroon moet zijn voordat de ontvanger erop reageert.

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 ze 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 |

# Websecurity ontwerp

# 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>

# Testcases

# Conclusie

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