|  |  |  |
| --- | --- | --- |
| **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 |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| 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

# 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 updated. 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

# Aanbevelingen

# Bronnenlijst

# Bijlage