

# **PVA Automated Systems, Project codename Beezkneez**

**Mark Steijger**  
0938713

22 juni 2021



# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>2</b>
<b>2</b>	<b>System/Subsystem Specification</b>	<b>3</b>
2.1	Identification . . . . .	3
2.2	System overview . . . . .	3
2.3	Document overview . . . . .	3
2.4	Referenced documents . . . . .	3
<b>3</b>	<b>System/Subsystem Design Description</b>	<b>4</b>
3.1	System wide design decisions . . . . .	4
3.1.1	Camera . . . . .	4
3.1.2	Server . . . . .	5
3.1.3	Simulatie . . . . .	6
3.2	System architectural design . . . . .	6
3.2.1	System components . . . . .	6
3.2.2	System architecture . . . . .	6
3.3	System architectural design . . . . .	9
<b>4</b>	<b>Software Design Description</b>	<b>10</b>
<b>5</b>	<b>Software Requirement Specification</b>	<b>11</b>
5.1	requirements . . . . .	11
<b>6</b>	<b>Interface Design Description</b>	<b>13</b>
6.1	Interface design . . . . .	13
6.2	Interface identification and diagrams . . . . .	13
<b>7</b>	<b>Software Test Plan</b>	<b>14</b>
7.1	Software . . . . .	14
7.2	Hardware . . . . .	14
<b>8</b>	<b>System Test Report</b>	<b>15</b>
<b>9</b>	<b>Risico Analyse</b>	<b>16</b>
<b>10</b>	<b>Conclusie</b>	<b>17</b>

# 1 Inleiding

Dit document dient als een techniesch naslagwerk voor AutomatedBeezzzz project, hierin worden de verschillende aspecten die nodig zijn geweest voor het uiteindelijk complete product besproken. [?]

## **2 System/Subsystem Specification**

### **2.1 Identification**

Het volledige systeem bestaat uit de volgende subsystemen.

1. Camera
2. Server
3. Simulatie

### **2.2 System overview**

Het doel van het systeem is het nabootsen van het gedrag dat bijen uitvoeren bij het verzamelen van eten. Hierbij wordt voor de bijen gebruik gemaakt van zowel hardware als gesimuleerde bijen. Deze bijen zullen vanuit hun start locatie, de korf, eten zoeken en hierna vervolgens andere bijen op de hoogte brengen van de locatie van deze voedselbron.

Het project zal uitgevoerd worden door de AutomatedBeezzzz projectgroep

### **2.3 Document overview**

Dit document is geschreven om het gehele systeem van de drone simulatie die de bijen nabootst te omschrijven. Hierbij komt kijken de architectuur, eisen, designs en een testplan waar het systeem aan moet voldoen.

### **2.4 Referenced documents**

Versies van dit document:

- Versie 1: 6-19-21

## 3 System/Subsystem Design Description

### 3.1 System wide design decisions

In dit hoofdstuk worden, zoals de titel al verraad, de design decisions behandeld. Hieronder vallen bijvoorbeeld input/outputs van het systeem, gedrag keuzes en andere onderdelen die voor het gehele systeem gelden

#### 3.1.1 Camera

Aan de camera zit een aantal eisen. Het meest voor de hand liggende eis is dat de resolutie hoog genoeg moet zijn om bepaalde objecten te kunnen herkennen. Dit is niet de enige eis, zo moet de camera ook kleur kunnen herkennen en modulair zijn. Deze camera zal aangesloten moeten worden aan een microcontroller zodat de fotos die de camera maakt bewerkt kan worden. Dit betekent dat de microcontroller krachtig genoeg moet zijn om fotos op te vragen, verwerken en berekeningen erop uitvoeren. Ook moet hij vervolgens de verwerkte data doorsturen naar een centrale server. Dit betekent dat er een manier moet zijn op de microcontroller om deze data te verzenden. Dit zijn de hardware eisen aan de camera kant, als dit allemaal voldaan wordt is het mogelijk om de camera te gebruiken voor de simulatie van bijen.

Uit onderzoek is gebleken dat de Picam V2 voldoet aan de eisen die is gesteld bij de SSS. Om deze reden hebben wij voor deze camera gekozen. Elke moderne webcam zou werken maar communicatie met een microcontroller wordt onnodig ingewikkeld. Ook zijn deze webcams van een hogere prijsklasse. Als er gekozen wordt voor een Picam voor de camera, is het uiteraard gebruikelijk om dan een raspberry pi te gebruiken als microcontroller. Dit is de enige microcontroller die plug and play werkt met een Picam. Vervolgens wordt er gekeken naar de eisen van de Pi, en er is gebleken dat deze eraan voldoet. De Pi 4 is krachtig genoeg om afbeeldingen te verwerken en heeft een ingebouwde bluetooth module om deze data weer op te sturen naar de server. Dit is dus waarom er gekozen is voor een Raspberry Pi 4.

### **3.1.2 Server**

Omdat het een gecentraliseerd systeem is worden de meeste acties bepaald door een centraal subsysteem, in dit geval is dat de server. Vanuit de server worden de drones aangestuurd en wordt informatie van en naar de simulatie en camera verstuurd. Voor alle communicatie wordt een MQTT broker gebruikt.

#### **Gedrag**

Met gedrag worden de keuzes bedoeld die het systeem maakt, hiermee worden de basis van het systeem bepaald.

1. Bij opstart word een drone naar de voedsel bron gestuurd.
2. Wanneer de drone deze locatie heeft bereikt zal hij terugkeren naar het startpunt
3. De Drone danst om de locatie te delen met de andere drones
4. Meerdere drones worden gestuurd om het voedsel op te halen
5. Wanneer het laatste voedsel uit de bron is gehaald en alle drones binnen zijn worden een nieuwe scout gestuurd

#### **Inputs/outputs**

De server krijgt vanuit meerdere punten informatie binnen en verstuurd het ook naar meerdere subsystemen. Deze inputs en outputs worden uitgebreid in het hoofdstuk Interface Design Decisions.

**Input** Camera informatie

**Input** Simulatie informatie

**Output** Drone gedrag

**Output** Simulatie informatie

### 3.1.3 Simulatie

De simulatie is een eenvoudig manier om een overzicht te krijgen over het gehele project. Dit komt omdat het net als met de echte server en drone, samenhangt met de rest van het geheel. Zo neemt de simulatie als input de coördinaten van de drones. Net als de echte server verwerkt de simulatie de coördinaten om uiteindelijk als output signalen door te geven aan de gesimuleerde drones. Deze signalen zijn de commandos die de drones uit moeten voeren. Dit gebeurt allemaal op een hoge snelheid gezien de simulatie op een sterke Windows laptop afgebeeld wordt. De limiet hieraan is dus de snelheid van de ingekomen locatie data. Om deze data veilig te ontvangen wordt er een MQTT server gebruikt die vergrendeld is met inloggegevens. Gezien de simulatie zelf lokaal op een PC is, zijn hier verder geen beveiligingsrisico's aan gekoppeld. Omdat het een simulatie is, wordt er een groot belang gehecht aan toegankelijkheid van data. De beste, en soms wel enige, manier om een simulatie te testen is door gebruik te maken van verschillende variabelen en states waar een simulatie zich in kan bevinden. Zo kan een simulatie gebruikt worden om een zo realistisch mogelijk gesimuleerde omgeving weer te geven.

## 3.2 System architectural design

### 3.2.1 System components

Hier wordt een opsomming gegeven van alle hardware en software componenten gegeven

SW Camera

SW MQTT Broker

SW simulatie

SW Server

HW Picam

HW Crazyflie

### 3.2.2 System architecture

Diagram met hardware en software componenten van het gehele systeem en de relatie tussen deze onderdelen

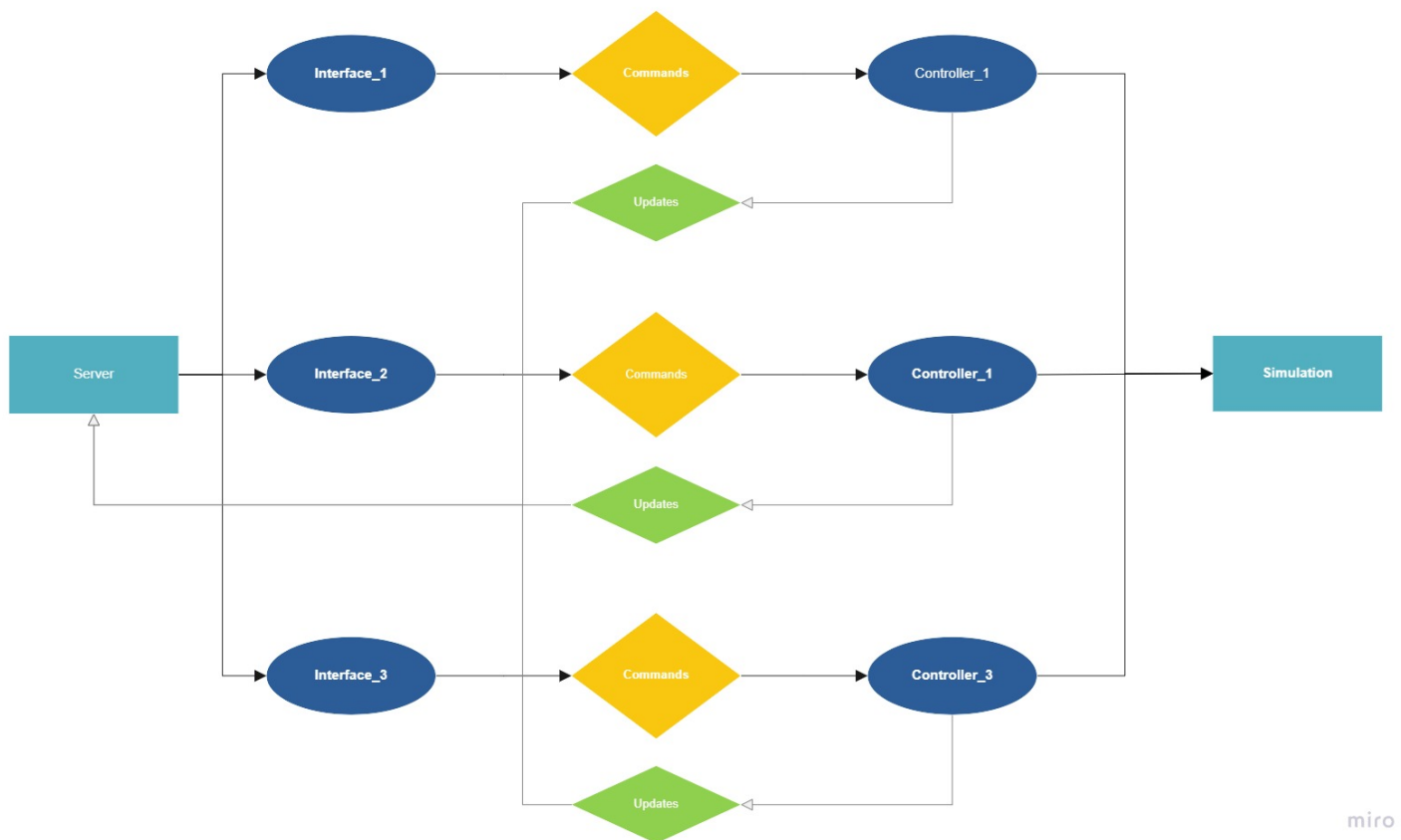
#### Camera

Aangezien het project inhoudt dat een aantal bijen gesimuleerd wordt, is het nodig om hun zintuigen ook te simuleren. Dit gebeurt door middel van een camera aan een raspberry pi. Hier zullen uiteraard eisen aan zitten, maar zullen in de SRS vermeldt worden. Om deze camera draadloos te laten werken is er ook een batterij pack nodig.

Server

asd

Simulatie



Het diagram hierboven weergeeft de communicatie tussen de simulatie en de server. In de simulatie wordt het gedrag van bijen met behulp van 3 ontworpen drones gesimuleerd. Om dat mogelijk te maken zijn voor de input en de output de volgende ontwerp keuzes gemaakt:

#### INPUT:

1. De input voor de simulatie is altijd een opdracht naar een controller. Deze opdracht is een actie die een drone moet uitvoeren waarbij een positie meegegeven wordt naarr de volgende verplaatsingsstap. Dit wordt



mogelijk gemaakt door te communiceren tegen een opgebouwde interface. Deze interface communiceert dan vervolgens tegen een controller die een gesimuleerde drone bestuurd.

**OUTPUT:**

1. De output bij deze simulatie is verplaatsing van de gesimuleerde drone naar een geplande positie.
2. Daarnaast is er ook een andere continue output van de simulatie. Dat is een bericht die afkomstig is van een controller. Een controller stuurt om de bepaalde tijd een bericht naar de server met de informatie over de positie van de bestuurd drone.

### 3.3 System architectural design

Uiteraard is het project een samenhangend geheel. Dit betekent dat er onderdelen zijn die afhankelijk zijn van andere onderdelen. Deze onderdelen zullen vermeld worden.

#### System components

Diagram met hardware en software componenten van het gehele systeem en de relatie tussen deze onderdelen. J. Goeie, moeten we doen

Zoals op de afbeelding te zien is, is de camera afhankelijk van een Raspberry Pi. Dit komt omdat de beeldverwerking moet worden gedaan door een microprocessor. Zo is er ook een samenhang tussen ...

## 4 Software Design Description

In dit hoofdstuk worden de diagrammen behandeld die over het systeem of zijn subsystemen gaan

## **5 Software Requirement Specification**

### **5.1 requirements**

Aan de camera zit een aantal eisen. Het meest voor de hand liggende eis is dat de resolutie hoog genoeg moet zijn om bepaalde objecten te kunnen herkennen. Dit is niet de enige eis, zo moet de camera ook kleur kunnen herkennen om objecten van de achtergrond te onderscheiden en modulair zijn. Deze camera zal aangesloten moeten worden aan een microcontroller zodat de fotos die de camera maakt bewerkt kan worden. Dit betekent dat de microcontroller krachtig genoeg moet zijn om fotos op te vragen, verwerken en berekeningen erop uitvoeren. Ook moet hij vervolgens de verwerkte data doorsturen naar een centrale server. Dit betekent dat er een manier moet zijn op de microcontroller om deze data te verzenden. Dit zijn de hardware eisen aan de camera kant, als dit allemaal voldaan wordt is het mogelijk om de camera te gebruiken voor de simulatie van bijen.

#### **Required states and modes**

In het geval dat het systeem actief is, is ook de camera en de server actief. Aangezien de drone constant zijn locatie moet weten binnen het systeem, is het niet mogelijk om het gehele systeem idle te maken. Voor de specifieke drone zelf kan het wel een aantal states hebben. Zo kan de drone inactief zijn. Dit wordt shutdown genoemd, en de drone ligt dan op de grond. Het kan actief worden gemaakt, wat start genoemd wordt, en de drone staat dus aan, maar beweegt niet per se. Er is een take off stand, waarbij de drone opstijgt. De drone kan dan idle gemaakt worden, en in de lucht blijven zweven. De drone kan op move worden gezet, en vooruit vliegen. De drone heeft een turn mogelijkheid, waarbij het draait. Voor dit specifiek project is er ook een dance functie nodig, waarbij de drone een beweging maakt dat op een dans lijkt om het gedrag van bijen na te bootsen. En ten slotte kan de drone ook weer dalen, oftewel landing.

#### **System internal interface requirements**

##### **System internal data requirements**

Er zijn een aantal variabelen die duidelijk gemaakt moeten worden over het systeem. Beginnend bij de camera zal deze objecten herkennen, zoals de drones, etensbronnen en bijenkorf. De locatie van deze objecten zal opgestuurd worden naar de centrale server. De centrale server verwerkt deze lokaal en verstuurt vervolgens data om de drones aan te sturen. Deze data is de states waarin de drone zich in kan bevinden, en zijn dus al benoemd onder "Required states and modes".

#### **Design and construction constraints**

De grootste beperkingen binnen het project is het budget. Ondanks dit is het mogelijk geweest eromheen te werken en een kostenanalyse op te zetten die positief uitkomt. Verdere beperkingen zijn aantal drones die geleverd zijn, oftewel het

beschikbare hardware. Hierdoor zou er een mogelijke compromis gesloten moeten worden om een aantal drones te simuleren. Ook is er niet volledige kennis over een systeem als deze voor elke groepslid, waardoor er kennis opgedaan moest worden en uiteindelijk niet genoeg tijd overbleef. De kennis met tijd balans is dus een grote beperking binnen het project.

## **6 Interface Design Description**

Dit hoofdstuk behandelt in detail alle interfaces en communicatie de het systeem en daarmee subsystemen met elkaar hebben.

### **6.1 Interface design**

### **6.2 Interface identification and diagrams**

## 7 Software Test Plan

De test plan zal worden opgesplitst in een software-matige kant en een hardware-matige kant. Zo kan voor ieder testomgeving omschreven worden wat er gebeurt binnen de omgeving, en de resultaat van de testen.

### 7.1 Software

Om de testen op te zetten is het natuurlijk nodig om voor de drones bijbehorende software te maken. Dit betekent dat er een vorm van aansturing moet zijn, oftewel een interface om de drones mee aan te sturen. Deze interface bestuurt de benodigde commandos naar de individuele drones. De server draait een python server om de interface te runnen. Het is mogelijk om dit te doen op een laptop die Windows draait. Deze code wordt via python ook verbonden met een MQTT server, die dan verbonden is met een raspberry pi of Windows laptop waar een camera aan hangt. Hierop wordt code uitgevoerd om beeld te verwerken. Dit gebeurt ook in python. Ten slotte is er nog een simulatie aanwezig van twee of drie drones die op een scherm afgebeeld worden in WeBots. Dit gebeurt op een Windows laptop die in staat is deze software uit te voeren. Alle code is geschreven door de gehele project team.

### 7.2 Hardware

Voor de hardware wordt er gebruik gemaakt van twee drones van de merk CrazyFlie 2.0. De architectuur van deze drones is te vinden in de documentatie van CrazyFlie. De gehele simulatie zal deze drones aan moeten staan, aangezien dit de visuele indicatie is van de gedrag van bijen. De centrale server is een python script die op een Windows pc draait. Deze laptop hoeft niet krachtig te zijn, gezien zijn voornaamste taken locaties ontvangen, omrekenen en commandos versturen naar de drones zijn. Ook deze heeft geen down-time tijdens de simulatie, want anders weet een bij/drone niet wat het moet doen. Ten slotte is er een apparaat met een camera nodig. Er zou gekozen kunnen worden voor een raspberry pi met een PiCam v2, maar het werkt net zo goed met een laptop met Windows waar een goedkope webcam aan is gesloten. Daarom is er uiteindelijk gekozen voor een laptop met een webcam. Deze webcam kan fotos en films in full HD maken. Beide de laptop en de webcam zijn constant aan, gezien de drones ten alle tijden zijn locatie moet weten.



## **8 System Test Report**

In dit hoofstuk worden de testresultaten behandeld

## 9 Risico Analyse

## 10 Conclusie