Gamotica

Communicatie Java applicatie

Auteurs:

Cordell Stirling(S1154605)

Jelle Smelt(S1159072)

Jeremy de Groot(S1114990)

Rick Stoffer(S1162883)

Jacco Kalter(S1158783)

Jesper Kiers(S1156710)

**Versiebeheer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Omschrijving** | **Opmerkingen** |
| **0.1** | 18-02-2021 | Start onderzoek, invullen gegevens, opmaak, introductie |  |
| **0.2** | 19-02-2021 | Uitwerken hoofdstuk 2 Introductie en hoofdstuk 3 Methoden |  |
| **0.3** | 24-02-2021 | Uitwerken hoofdstuk 4.2 |  |
| **0.4** | 02-03-2021 | Uitwerking 4.3 |  |
| **0.5** | 19-03-2021 | Uitwerking 4.1 |  |
| **0.6** | 22-03-2021 | Uitwerking 4.4 |  |
| **0.7** | 23-03-2021 | Uitwerking 4.5 |  |
| **0.8** | 25-03-2021 | Uitwerking 5.1, uitbreiden van literatuurlijst |  |
| **0.9** | 26-03-2021 | Uitwerking 5 |  |
| **1.0** | 26-03-2021 | Kwaliteit controle |  |

**Distributie**

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Rol | Datum | Versie |
| Cordell Stirling | Teammanager | 26/03/2021 | 1.0 |
| Jacco Kalter | Teamlid | 26/03/2021 | 1.0 |
| Jelle Smelt | Teamlid | 26/03/2021 | 1.0 |
| Jesper Kiers | Teamlid | 26/03/2021 | 1.0 |
| Rick Stoffer | Teamlid | 26/03/2021 | 1.0 |
| Jeremy de Groot | Teamlid | 26/03/2021 | 1.0 |

**Goedkeuring**

|  |  |  |  |
| --- | --- | --- | --- |
| Naam | Rol | Datum | Paraaf |
| Wille Conen | Opdrachtgever | 26/03/2021 |  |
|  |  |  |  |

Inhoud

[**1 Abstract** 1](#_Toc67666269)

[**2 Introductie** 2](#_Toc67666270)

[2.1 Onderwerp 2](#_Toc67666271)

[2.2 Context 2](#_Toc67666272)

[2.3 Doelstelling 2](#_Toc67666273)

[2.4 Vraagstelling 2](#_Toc67666274)

[2.4.1 Hoofdvraag 2](#_Toc67666275)

[2.4.2 Deelvragen 2](#_Toc67666276)

[**3 Methoden** 4](#_Toc67666277)

[3.1 Literatuurstudie 4](#_Toc67666278)

[3.1.1 zoekstrategie 4](#_Toc67666279)

[3.1.2 Betrouwbaarheid 4](#_Toc67666280)

[3.1.3 Validiteit en betrouwbaarheid 5](#_Toc67666281)

[3.2 Prototyping 5](#_Toc67666282)

[**4** **Resultaten** 6](#_Toc67666283)

[4.1 Welke communicatie mogelijkheden zijn er? 6](#_Toc67666284)

[4.1.1 Communicatie door WiFi 6](#_Toc67666285)

[4.1.2 Communicatie door USB 6](#_Toc67666286)

[4.1.3 Communicatie door Bluetooth 6](#_Toc67666287)

[4.1.4 Communicatie door Serial-Ports 6](#_Toc67666288)

[4.2 Wat is een Java-library 7](#_Toc67666289)

[4.3 Welke libraries maken communicatie tussen systemen mogelijk? 8](#_Toc67666290)

[4.3.1 Java Communications API 8](#_Toc67666291)

[4.3.2 jSerialcomm 9](#_Toc67666292)

[4.3.3 jRxTx 9](#_Toc67666293)

[4.3.4 PureJavaComm 9](#_Toc67666294)

[4.4 Welke libraries komen het meest overeen met de eisen van het project? 9](#_Toc67666295)

[4.4.1 Java Comm API 9](#_Toc67666296)

[4.4.2 jSerialcomm 10](#_Toc67666297)

[4.4.3 jRxTx 10](#_Toc67666298)

[4.4.4 PureJavaComm 10](#_Toc67666299)

[4.5 Welke library gaan we in het project gebruiken? 10](#_Toc67666300)

[**5** **Discussie** 12](#_Toc67666301)

[5.1 Resultaten prototyping 12](#_Toc67666302)

[5.2 Conclusie 13](#_Toc67666303)

[5.3 Discussie 13](#_Toc67666304)

[**6** **Literatuurlijst** 15](#_Toc67666305)

# **1 Abstract**

In dit document wordt het onderzoek met betrekking tot de communicatie tussen de verschillende apparaten vastgelegd. Als eerst wordt er een introductie opgesteld waar de hoofdvraag wordt gesteld samen met de deelvragen. Onze hoofdvraag luid: Welke library of libraries gaan wij gebruiken om communicatie tussen verschillende systemen te realiseren?   
Als eerst hebben wij onze deelvragen uitgewerkt om onze selectie van de library en de connectie te verduidelijken. Er word uitgelegd wat een Java library is en hoe het functioneert. Wij hebben naar de mogelijkheden van connectie gekeken, de voor- en nadelen uitgewerkt en daarna een keuze gemaakt. Na het zoeken naar relevante libraries hebben we een selectie van vier libraries gemaakt waaruit wij er één moesten uitkiezen. Wij hebben de voor- en nadelen van de vier libraries genoteerd en vanuit daar een keuze gemaakt. De gekozen library is JSerialcomm.

# **2 Introductie**

## 2.1 Onderwerp

In dit document beschrijven wij het onderzoek naar de verschillende Java-libraries die communicatie tussen verschillende apparaten mogelijk maakt. Er wordt gekeken naar de verschillende opties voor libraries die gebruikt kunnen worden, deze libraries worden vergeleken met elkaar en waaruit komt een conclusie welke Java-library er gebruikt wordt om ons project te realiseren. Er wordt dan gekeken naar de verschillende opties voor libraries die gebruikt kunnen worden, deze libraries worden vergeleken met elkaar waaruit een conclusie komt welke Java-library er gebruikt wordt om ons project te realiseren.

## 2.2 Context

Voor dit project is er een Java-library vereist. Om een goed geïnformeerde keuze te maken moet er onderzoek gedaan worden naar de beschikbare libraries die wij kunnen gebruiken voor ons project om communicatie met domotica systemen mogelijk te maken.

In het belang van de projectgroep wordt dit onderzoek opgesteld. Met het onderzoek kan de groep de keuze onderbouwen en het project beter in kaart brengen. Door het onderzoek vast te leggen wordt er een extra laag aan documentatie toegevoegd voor toekomstige onderhoud.

## 2.3 Doelstelling

Met dit onderzoek willen we in kaart brengen wat de best beschikbare libraries zijn en welke het beste aansluiten op het project. Binnen het onderzoek wordt er vergeleken tussen de verschillende libraries en wordt er duidelijk gemaakt welke library er uiteindelijk gekozen is om te gebruiken in het project.

## 2.4 Vraagstelling

### 2.4.1 Hoofdvraag

Voor het domotica systeem is een Java-library vereist om de communicatie tussen verschillende systemen af te handelen:

Welke library of libraries gaan wij gebruiken om communicatie tussen verschillende systemen te realiseren?

### 2.4.2 Deelvragen

Om te besluiten welke library we gaan gebruiken moet er eerst onderzocht worden welke soort communicatiemogelijkheden er zijn met java-libraries:

Op welke manier willen we de aansluiting regelen tussen de libraries en het domotica systeem?

Voordat we kijken naar verschillende Java-libraries is het goed om te weten wat een Java-library precies is:

Wat houdt een Java-library in en wat kan je met een library?

Nadat er duidelijk gemaakt is wat een Java-library is kunnen we verschillende libraries uitkiezen die communicatie beschikbaar maken tussen verschillende apparaten:

Welke libraries komen in aanmerking voor ons project die communicatie tussen systemen mogelijk maakt?

Nadat er een selectie van libraries is kan er vergeleken worden tussen de verschillende libraries:

Welke libraries komen het meest overeen met de eisen van het project om communicatie beschikbaar te maken tussen systemen?

Na deze stappen kunnen we de volgende vraag stellen:

Welke library gaan we gebruiken in het project om de functies van het domotica systeem te realiseren?

# **3 Methoden**

## 3.1 Literatuurstudie

### Zoekstrategie

Ons zoekstrategie bestaat uit het onderzoeken wat een Java-library inhoud. Om hierachter te komen worden verschillende relevante zoektermen gebruikt bestaande uit onder andere:

* Java
* Library
* Definition
* What is a Java library?

Uit de zoekresultaten komen verschillende resultaten die relevant zijn voor ons om te onderzoeken wat een Java-library is en hoe we een Java-library kunnen gebruiken.

Om verschillende Java-libraries te vinden die in aanmerking komen voor ons project moeten we zoeken op libraries die communicatie beschikbaar maakt, dit zullen we onder andere doen met de volgende zoektermen:

* Java
* Library
* Serial
* Communication

De zoekresultaten zullen verschillende Java-libraries bevatten die communicatie tussen apparaten mogelijk maakt, elke met een eigen manier van werken. Uit deze resultaten kunnen we een geïnformeerde selectie maken van libraries die voldoen aan de eisen voor het project.

### Betrouwbaarheid

Om de betrouwbaarheid van de verschillende bronnen te checken worden er een aantal vragen gesteld waar de bron aan moet voldoen om als een betrouwbare bron te worden beschouwd.

Deze vragen bestaan uit:

* Wat is het doel van de informatiebron?
* Bevat de bron feiten of meningen?
* Is de bron toepasbaar?
* Is de bron herzien?
* Voldoet de informatie aan de informatiebehoefte?
* Is de auteur een hedendaagse deskundige op het vakgebied?
* Heeft de auteur meer geschreven over het onderwerp?
* Is er een overzicht van gebruikte bronnen?
* Bevat de bron de full-tekst of is het een samenvatting of verwijzing?
* Hoe recent is de informatie?
* Is de informatie nog geldig of is deze verouderd?
* Zijn er inmiddels nieuwe inzichten over het onderwerp?

Wanneer er positief antwoord uit de vragen komen over een bron, word een bron als betrouwbaar beschouwen en word de informatie van de bron gebruikt in het onderzoek.

### Validiteit en betrouwbaarheid

Om de validiteit en betrouwbaarheid van de informatie van een bron te garanderen, wordt de te gebruiken informatie opnieuw onderzocht. Wanneer informatie van de desbetreffende bron bij meerdere bronnen te vinden is en deze bronnen voldoen ook aan de betrouwbaarheidscheck wordt de informatie als betrouwbaar en feitelijk beschouwen.

## 3.2 Prototyping

### 3.2.1

Binnen het project is het van belang dat het systeem met verschillende apparaten kan communiceren, binnen deze communicatie zijn een aantal handelingen die verricht moeten kunnen worden:

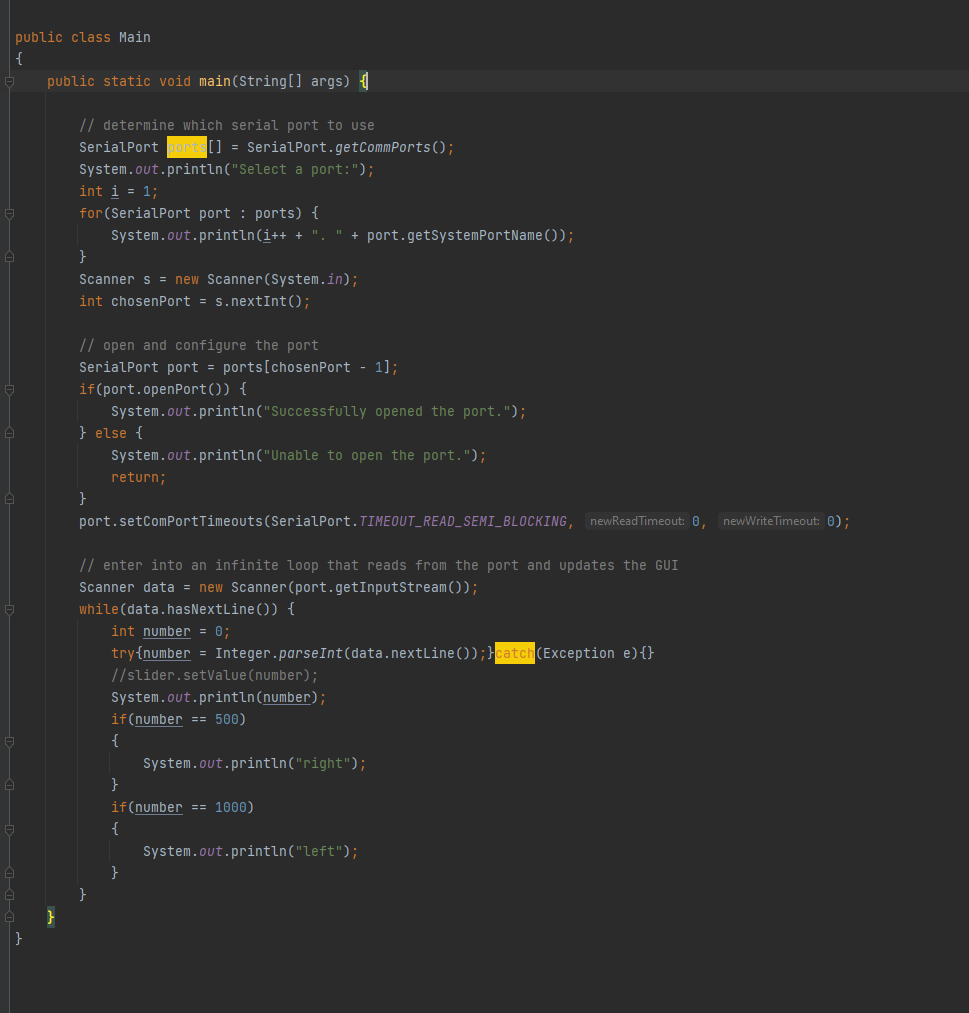
* Metingen(sensoren) verrichten en doorsturen naar de controller van de applicatie.
* Actuatoren besturen en uitlezen vanuit het centrale systeem.
* Verschillende invoer en selecties kunnen doorvoeren.

De verschillende libraries worden geselecteerd op functionaliteiten, onder deze functionaliteiten moeten de te verrichte handelingen staan om in aanmerking te komen voor selectie voor het project. Wanneer de handelingen verricht kunnen worden door de library, word deze in de voorlopige selectie gezet.

De voorlopige selectie van libraries wordt vervolgens geselecteerd op communicatiemogelijkheden. Wanneer een library in aanmerking komt voor het project worden naast de vereiste eigenschappen de extra eigenschappen van die library genoteerd om te gebruiken in de vergelijking met de andere libraries.

### 3.2.2 Resultaten prototyping

Om te testen of jSerialcomm een goede verbinding met de arduino kan maken is er een test gemaakt met 2 knoppen om uit te lezen welke van de knoppen word ingedrukt. Aan het begin worden de beschikbare ports opgevraagd, waaruit een keuze moet worden gemaakt. De tester voert dan de port in waar de arduino op is aangesloten. Wanneer de tester op één van de knoppen op de arduino drukt, doet het programma een check op wat de waarde is van de opgestuurde data, deze data is door de tester gedefinieerd. Wanneer de data gelijk is aan één van de twee waardes voor links of rechts word dit geprint in de console.



Met als resultaat:



Door op deze manier te bevestigen dat de port te bereiken is met jSerialcomm is bewezen dat het een toegankelijke library is om te gebruiken binnen ons project.

# **Resultaten**

## Welke communicatiemogelijkheden zijn er?

Om te bepalen welke Java-library we gaan gebruiken moet er eerst onderzocht worden welke mogelijkheden er hierbinnen zijn. De verschillende mogelijkheden worden naast elkaar gezet en er wordt bekeken welke mogelijkheid het beste past bij ons systeem.

### Communicatie door WiFi

De Arduino heeft de mogelijkheid om een verbinding via WiFi te maken.

Dit wordt mogelijk gemaakt door bijvoorbeeld de WiFi.h library, maar hiervoor is een Arduino shield nodig om een verbinding te kunnen ontvangen op de Arduino. Deze library ondersteunt WEP en WPA2 persoonlijke encryptie maar geen WPA2 enterprise encryptie. Een nadeel van verbinden met WiFi is dat het SSID ten alle tijden beschikbaar moet zijn anders is de verbinding niet mogelijk. De WiFi library is grotendeels vergelijkbaar met de Ethernet library waarbij veel van de function calls hetzelfde zijn.

Gezien bij een wifi-oplossing een constante verbinding niet te garanderen is in gebieden met slecht bereik, kan dit invloed hebben op de gebruiksvriendelijkheid en de interactie met de microcontrollers. Om deze reden zullen wij geen wifi-oplossing gebruiken om de communicatie mogelijk te maken.

### Communicatie door USB

De Arduino heeft de mogelijkheid om een verbinding te maken via USB. De Arduino waarmee we werken, de Arduino Uno Rev3, heeft een USB B port, hiervoor hebben we dus een USB type A/B kabel nodig om een verbinding te maken met een USB A port.

Het gebruik van communicatie door USB is gemakkelijk, omdat het een kwestie van inpluggen is. Het nadeel hiervan is dat een kabel benodigd is. De lengte van deze kabel draagt bij aan het gebruiksgemak en defecten aan de kabel zal de verbinding dwarsbomen.

### 4.1.3 Communicatie door Bluetooth

De mogelijk is er om een bluetooth verbinding te hebben met de Arduino, maar daar is een extra component voor nodig. Bijvoorbeeld een HC-05 of 06 bluetooth Module.   
Bluetooth zou een goede mogelijkheid zijn omdat het redelijk gemakkelijk is om te gebruiken en goed werkt met draadloze signalen op korte afstanden, omdat wij een controller maken hebben wij ook geen langeafstand signalen nodig.   
Het nadeel van het gebruiken van bluetooth is de prijs. Een HC – 05 kost rond de 6 euro plus het extra codeer en aansluiting werk wat we dan moeten doen is dit geen efficiënte keuze om te gebruiken.

### Communicatie door Serial-Ports

Serial communicatie is standaard geïntegreerd in de Arduino en word gebruikt om feedback van de Arduino naar het aangesloten apparaat te sturen. Alle Arduino bordjes hebben minimaal één serial port om op aan te sluiten. Bij het gebruik van serial communicatie worden poorten 0(RX) en 1(TX) gebruikt en kunnen deze dus ook niet meer gebruikt worden voor digital input of output. De Arduino omgeving heeft een ingebouwde serial monitor om te output van de Arduino uit te lezen.

Het is mogelijk om zowel bedraad als draadloos een serial verbinding te gebruiken, al is een draadloze verbinding niet beveiligd en word dit dus ook niet aangeraden om in de praktijk toe te passen.

De Serial communicatie is voor het project het meest voor de hand liggend, de verbinding is ingebouwd en meegeleverd in het pakket wat ontvangen wordt en is dus makkelijk te implementeren. Draadloze serial communicatie is (voor nu) niet een optie, gezien de toepassing van de microcontroller binnen ons systeem is het geen goed idee om een onbeveiligde draadloze verbinding hiervoor te gebruiken. Dus daarom gaan wij de controller verbinden met de computer via USB en leest de gegevens met JSerialcomm uit.

Om deze reden gebruiken wij een usb voor de fysieke verbinden, terwijl we hiermee direct de connectie voor jserialcomm ook mogelijk maken om de data uit te lezen.

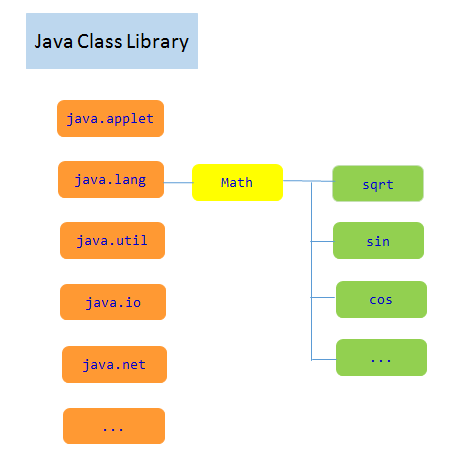
## 4.2 Wat is een Java-library

Een Java-library is een collectie van classes die geschreven zijn door iemand anders, deze classes worden gedownload wanneer de library geïmporteerd wordt en zijn hierna bruikbaar in een project. Door gebruik te maken van Java-libraries maak je het jezelf mogelijk om gebruik te maken van code die geschreven is door andere mensen, dit heeft een aantal voordelen:

* + De code is getest, dit betekent dat de code dat de library bevat naar behoren werkt, hierdoor hoef je niet zelf te testen of een class werkt.
  + Door niet zelf elke class te hoeven testen, bespaar je tijd en heb je meer ruimte om andere functie te implementeren in het project.

Een library kan vergeleken worden met een fysieke boekenkast in een bibliotheek. Binnen deze bibliotheek zijn meerdere schappen met boeken te vinden die onderverdeeld zijn in verschillende categorieën. Een Java-library is hierin vergelijkbaar, libraries zijn verdeeld in zogenaamde “packages”, deze kunnen gezien worden als de verschillende planken met categorieën in een bibliotheek, in deze packages staan geteste classes de klaar voor gebruik zijn.

Om dit beter te illustreren hebben we een voorbeeld van vertex-academy gebruikt om dit uit te beelden:



Aan de linkerkant worden de verschillende libraries weergegeven, waarvan elke library zijn eigen classes heeft waar geteste code in staat. In dit voorbeeld wordt de java.lang library gebruikt, uit deze library word de math class gebruikt om verschillende functie van de gebruikten. Wanneer je de wortel van een getal wilt berekenen zou je veel code moeten schrijven als je geen library zou gebruiken, maar wanneer je wel gebruik zou maken van de java.lang library, kun je de math class gebruiken, deze bevat de sqrt() methode die de wortel berekent, om dit binnen enkele regels uit te voeren.

## Welke libraries maken communicatie tussen systemen mogelijk?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Java Comm API** | **jSerialcomm** | **jRxTx** | **PureJavaComm\*** |
| Arduino pc Comm | X | X | X | X |
| Arduino Pi Comm | X | X |  | X\* |
| Pi pc Comm | X | X |  | X |

### 4.3.1 Java Communications API

De Java Communication API is een standaard extensie die data kan verzenden en ontvangen via RS-232 serial poorten en via IEEE 1284-parallel poorten. Hiermee kunnen Javaprogramma’s met in principe elk apparaat communiceren dat is verbonden via één van deze poorten. De API opereert op een laag niveau binnen de applicatie, het begrijpt alleen hoe bytes naar deze poorten moeten worden verzonden en ontvangen. Het begrijpt niets van wat deze bytes betekenen.

### 4.3.2 jSerialcomm

jSerialcomm is een Java-library die ontworpen is om een platformonafhankelijke manier te bieden om toegang te krijgen tot standaard serial ports zonder dat externe libraries, native code of andere tools nodig zijn. Het is bedoeld als alternatief voor RxTx en de (verouderde) Java Communications API, met meer gebruiksgemak, verbeterde ondersteuning voor time-outs en de mogelijkheid om meerdere poorten tegelijk te openen.

### 4.3.3 jRxTx

jRxTx pakt de populaire RXTX-bibliotheek waarvoor de ontwikkeling is stopgezet op.   
jRxTx biedt twee soorten API's:

1. De originele RXTX API
2. Een nieuwe verbeterde API.

### 4.3.4 PureJavaComm

PureJavaComm is een API met toegang tot serial ports van Java, dus deze library is gefocust aan de kant van de programmeur niet de end users.  
PureJavaComm is een gemakkelijkere op te zetten alternatief voor de RXTX. Het is volledig in Java gecodeerd zodat het gemakkelijk is voor programmeurs om dingen te ontwikkelen en debuggen.

## Welke libraries komen het meest overeen met de eisen van het project?

Om te bepalen welke libraries het meest overeenkomen met de eisen van het project moet er gekeken worden naar de eisen zoals die beschreven staan in 3.2 prototyping. Wanneer een library volledig voldoet aan deze eisen, of meer bied, word deze library meegenomen in de opties om te gebruiken voor het project.

### Java Comm API

Metingen verrichten en doorsturen – **ja**/ ~~nee~~

De Java Communication API is instaat om berechten te verzenden en ontvangen via een RS-232 poort en een IEEE 1284-parallel poort. Dit geeft de mogelijkheid om met elk apparaat te communiceren dat verbonden is met een van deze poorten.

Actuatoren besturen en uitlezen vanuit het centrale systeem – **ja**/ ~~nee~~

De API begrijpt hoe bytes naar deze poorten moeten worden verzonden en ontvangen. Door middel van deze bytes kunnen deze actuatoren worden geïnformeerd en uitgevoerd ook kunnen deze bytes gebruik worden om de huidige status van de actuatoren uit te lezen.

Verschillende invoer en selecties kunnen doorvoeren – **ja**/ ~~nee~~

Met behulp van datastreams is het mogelijk om verschillende soorten data uit te lezen en weg te schrijven. Met deze functie kunnen we verschillende invoer en selecties doorvoeren.

### jSerialcomm

Metingen verrichten en doorsturen – **ja**/~~nee~~

jSerialcomm is bekwaam om raw data bytes te lezen en schrijven direct naar een serial port, ook is het een mogelijkheid om dit te doen via de input- en outputsStream interfaces.

jSerialcomm bied de mogelijkheid om evenementen uit te lezen en uit te voeren via callbacks. Voor deze callbacks kunnen er een aantal notificaties worden weergegeven, deze notificaties zijn:

* Nieuwe data beschikbaar om uit te lezen
* Data succesvol weggeschreven
* Een complete vooraf afgesproken lengte data packet is aangekomen
* Een ongelimiteerd string gebaseerd bericht is aangekomen

Actuatoren besturen en uitlezen vanuit het centrale systeem – **ja**/ ~~nee~~

jSerialcomm bevat een modus genaamd ‘Event-Driven Callback Mode’. Binnen deze modus is het mogelijk om een eigen callback te registreren die de events van de actuatoren uitleest.

Verschillende invoer en selecties doorvoeren – **ja**/~~nee~~

Om de invoer en selecties door te voeren kan de event-driven callback modus gebruikt worden. Standaard invoer en selecties zullen voor gedefinieerd zijn, maar enkele die niet gedefinieerd zijn kan de gebruiker zelf aanmaken om de acties toch te kunnen doorvoeren door middel van de library.

### jRxTx

jRxTx is de drop in replacement voor RXTX. Hierbij heeft jRxTx dezelfde functionaliteiten als dat RXTX heeft. jRxTx wrapped de RXTX API. Er zijn 2 opties om de RXTX API te gebruiken binnen jRxTx:

* De originele RXTX API.
* Een nieuwe verbeterde API gebaseerd op de originele RXTX API.

RXTX is een direct fork van de javacomm api en bevat alle functionaliteiten van de API tot op dat punt. Gezien de vereisten vanuit het project daaronder vallen is jRxTx in staat om alle functionaliteiten te bieden binnen het project.

### PureJavaComm

PureJavaComm wil een drop in replacement voor de niet verder ondersteunde javacomm zijn en een makkelijker alternatief voor RXTX. Hierom is PureJavaComm in staat de functionaliteiten van javacomm te reproduceren. Door deze mogelijkheid is PureJavaComm een mogelijke library om te gebruiken gezien dezelfde functionele eisen als javacomm bezit voor dit project.

## 4.5 Welke library gaan we in het project gebruiken?

Van de 4 opties die wij onderzocht hebben, hebben wij gekozen om serial communicatie te gebruiken voor het project. Deze keuze is gemaakt omdat serial communicatie in de Arduino gebouwd zit en hierom direct bruikbaar is zonder bijvoorbeeld extra onderdelen of stappen die gezet moeten worden om de communicatiemethode te kunnen gebruiken. Serial communicatie biedt ook de mogelijkheid om uitgebreid te worden door het draadloos te maken. Een draadloze serial verbinding zou het project gebruiksvriendelijker maken gezien de behoefte van draden niet meer noodzakelijk is.

Na onderzoek gedaan te hebben en vergelijkingen hebben gemaakt met andere libraries kwam jSerialcomm op nummer 1 uit, de reden hiervoor is dat jSerialcomm het meest up-to-date is van de libraries en ook alle functionaliteiten bezit wat wij nodig gaan hebben met dit project.  
Java Comm API leek in het eerste opzicht op een hele goede keuze als we keken naar de functionaliteiten, maar toen kwamen we erachter dat de library niet meer ondersteunt wordt. Hierdoor viel Java Comm API weg.

jRxTx en PureJavaComm worden niet meer ondersteunt net als Java Comm API en verder geven deze mindere functionaliteiten vergeleken met jSerialcomm en deze libraries zijn minder geschikt voor ons type project.

Tijdens de realisatiefase van het project hebben we besloten om voor de communicatie tussen de applicatie en de raspberrypi een TCP-socket verbinding te gebruiken. Het gebruik van jSerialcomm is een mogelijkheid maar gezien het gebruik van de raspberrypi is het makkelijker om gebruik te maken van TCP-sockets en de pi als een server te zien. Op deze manier kunnen we het uitlezen van de sensoren veel makkelijker maken en weergeven in de applicatie.

# **Discussie**

## 5.1 Conclusie

Uit het onderzoek hebben wij meerdere libraries gevonden die in aanmerking kwamen voor het project:

- Java comm API

- jSerialcomm

- jRxTx

- PureJavaComm

Door te onderzoeken wat de verschillende mogelijkheden zijn van de libraries en te kijken naar andere criteria waaronder de ondersteuning hebben we vrij snel gekozen voor jSerialcomm. Deze library kwam het beste overeen met de eisen van het project en is het meest recent wat betreft de ondersteuning.

Met de prototyping van de library is er bewezen dat er communicatie tussen de library en het systeem mogelijk is, hiermee hebben we aangetoond dat er binnen de development een goede workflow kan plaatsvinden.

## 5.2 Discussie

Om binnen het onderzoek selectief te kunnen kijken naar de verschillende libraries hebben we de eisen gepakt en hierbij een selectie gemaakt op basis van de populaire Java-libraries. We hebben het op deze manier aangepakt om het aantal te onderzoeken libraries klein te houden terwijl er een goede selectie kon worden gemaakt.

Bij deze aanpak zijn wel een aantal zwakke punten te vinden:

1. Door het op deze manier aan te pakken zijn er wellicht goede libraries buiten beschouwing gebleven die bij ons project goed gebruikt konden worden.
2. De kwaliteit van de libraries tegen over elkaar was erg te vergelijken gezien de libraries allemaal onder de populairste libraries vallen en daarom veel aandacht hebben gekregen bij het ontwikkelproces. Dit betekent voor ons dat dit een onderdeel was waarop een library eigenlijk niet kon worden afgerekend.
3. Omdat de selectie libraries relatief klein was, was de onderhoud en ondersteuning onderling erg verschillend. Dit toont aan dat of de library niet of weinig nog gebruikt wordt of dat er een beter alternatief is ontstaan waardoor development van die library gestopt is.

Bij het onderzoek waren een paar onverwachte uitkomsten:

1. De Java Comm API, ondanks dat deze library alles heeft wat betreft eisen vanuit het project, is dit niet de library geworden die we gaan gebruiken in het project. Voornamelijk omdat jSerialComm recenter ondersteuning gekregen heeft maar ook omdat de API nog andere alternatieven had die beter waren in vergelijking.
2. Over het algemeen was het onverwacht dat de helft van de selectie van libraries niet recentelijke ondersteuning of onderhoud hebben gehad om welke reden dan ook. Ook dit is te verklaren door betere alternatieven.

# **Literatuurlijst**

4.1 Welke communicatiemogelijkheden zijn er?

4.1.1 Communicatie door WiFi

Arduino (SM), (2019, 24 december). WiFi library. Geraadpleegd van <https://www.arduino.cc/en/Reference/WiFi>

* + 1. Communicatie door Serial-Ports

Arduino, (z.d.). Serial. Geraadpleegd van <https://www.arduino.cc/en/reference/serial&gt;>

Instructables (ehsmaes), (z.d.). Arduino Wireless Serial Communication. Geraadpleegd van <https://www.instructables.com/Arduino-Wireless-Serial-Communication/>

* 1. Wat is een Java-library?

Happycoding, (z.d.). Libraries. Geraadpleegd van <https://happycoding.io/tutorials/java/libraries>

Vertex Academy, (2016, 5 augustus). What is the Java Library? Geraadpleegd van <https://vertex-academy.com/tutorials/en/what-is-java-library/?doing_wp_cron=1613745242.5388228893280029296875>

* 1. Welke libraries maken communicatie tussen systemen mogelijk?
     1. Java Comm API

Oracle, (z.d.). Java Communications API. Geraadpleegd van <https://www.oracle.com/java/technologies/java-communications-api.html>

* + 1. jSerialComm

Fazecast, Inc., (z.d.). jSerialComm. Geraadpleegd van <https://fazecast.github.io/jSerialComm/>

* + 1. jRxTx

OpenMUC, (z.d.). jRxTx. Geraadpleegd van <https://github.com/openmuc/jrxtx>

OpenMUC, (z.d.). Index. Geraadpleegd van <https://openmuc.github.io/jrxtx/javadoc/index.html>

OPEN SOURCE ECOLOGY, (2012, 13 november). RXTX. Geraadpleegd van <https://wiki.opensourceecology.org/wiki/RXTX>

* + 1. PureJavaComm

nyholku, (z.d.). PureJavaComm. Geraadpleegd van <https://github.com/nyholku/purejavacomm>

Kustaa “Kusti” Nyholm, (2011, 5 mei). PureJavaComm with JTermios. Geraadpleegd van <http://www.sparetimelabs.com/purejavacomm/purejavacomm.php>