

2016

LoRa Sensoren



Smart Sensor Systems



Thomas Schinkel
Koninklijke KPN NV
15-12-2016



Voorwoord

Mijn naam is Thomas Schinkel. Ik heb deze scriptie geschreven naar aanleiding voor mijn afstuderen op de opleiding Elektrotechniek aan de Haagse hogeschool in Delft. Mijn afstudeerstage heb ik gelopen bij KPN. Op de afdeling Technology Labs, meer over deze afdeling volgt later in deze scriptie. Ik heb bij mijn sollicitatie voor een afstudeerstage aangegeven dat ik graag iets zou willen doen met sensoren. Omdat ik erg geïnteresseerd ben in hoe sensoren werken en wat sensoren zouden kunnen aansturen. Het toeval bleek dat KPN net een landelijk LoRa netwerk had uitgerold. Dit is een techniek om sensor data te verzenden. De techniek LoRa is nog volledig nieuw en daarom extra leuk om hieraan te werken. Zo heb ik het voorstel om een onderzoek te doen over het LoRa netwerk met beide handen aangepakt.

Dankwoord

Binnen deze stage heb ik in een leuk team gewerkt. Dit is het team van Technology Labs. Technology Labs is een klein team maar heel veel kennis en expertise. Uit dit team zou ik graag een aantal mensen willen bedanken. Als eerste Rolf Biesbroek, dit was mijn begeleider tijdens mijn afstudeerstage. Ik kon bijna letterlijk 24/7 bij hem terecht met vragen. Hierdoor kon ik altijd verder als ik een probleem had. Hij heeft mij ook erg geholpen met het opzetten van een Windows server die ik nodig had tijdens het project. Voor al deze hulp wil ik hem hartelijk bedanken! Verder wil ik nog Frank Mertz bedanken die de rol van Rolf overnam toen Rolf op vakantie was. Frank heeft mij geholpen met het nakijken van mijn plan van aanpak. Dit heeft me erg geholpen om het begin van het project goed op te zetten. Bedankt hiervoor Frank! Verder wil ik de rest van het team met Perry Jackson voorop als manager, bedanken voor een ontzettend leuke tijd bij Technology Labs. De tijd is voorbijgevlogen!

Bedankt allemaal dat ik de kans heb gekregen om af te studeren binnen KPN!

Gegevens

Naam: Thomas Schinkel

Studenten nummer: 12093483

E-mailadres: 12093483@student.hhs.nl

Begeleider: Rolf Biesbroek

E-mailadres: rolf.biesbroek@kpn.com

Samenvatting / Summary

Nederlands:

LoRa sensoren zijn sensoren waarbij de waarde die de sensor meet via LoRa verzonden wordt. LoRa is een techniek waarbij berichten met een laagvermogen over een lange afstand verzonden kunnen worden. LoRa staat dan ook voor 'Long Range Low Power'. Als een LoRa sensor data verstuurt zal deze worden ontvangen door het LoRa netwerk. Dit LoRa netwerk bestaat uit antennes (gateways) en de LoRaWAN Cloud (server). Zodra er een LoRa bericht wordt ontvangen door een gateway zal deze het bericht doorsturen naar de LoRaWAN Cloud. Deze LoRaWAN Cloud is, via een website, zo in te stellen dat het de LoRa berichten doorstuurt naar een eigen server.

Deze eigen server is in dit geval een WampServer. Een WampServer is een server die draait op Windows en is een combinatie van Apache, MySQL en PHP. Door deze combinatie kan er een database worden aangemaakt waarin tabellen met data komen te staan. De data in deze tabellen kan door een aanvraag verwijderd, vervangen, geplaatst of uitgelezen worden. De LoRaWAN Cloud zal na het ontvangen van een LoRa bericht een aanvraag doen bij de server om de data in een tabel plaatsen. Tijdens deze aanvraag zal de server een aantal dingen controleren, zoals of de juiste data wel is meegestuurd in de aanvraag. Na deze controle zal data opgeslagen worden in de juiste tabel. Ook zal er gekeken worden of de data geen kritische waardes bevatten. Kritische waardes zijn waardes die zich boven of onder al eerder opgegeven waardes bevinden (deze waardes komen uit een tabel van de database op de server). Als de data kritische waardes bevat zal de server een notificatie sturen naar een applicatie.

Deze applicatie is een Androidapplicatie. In deze applicatie kan de gebruiker alle info terugvinden over de verstuurd LoRa berichten, zoals de waardes van de sensoren op een bepaald tijdstip. Ook kunnen, in deze applicatie, de waardes opgegeven worden waarmee de sensor wordt vergeleken. Dit zijn de waardes waarmee wordt gekeken of de sensor waardes kritisch zijn, zoals eerder beschreven. De notificatie die de applicatie ontvangt bij een kritische waarde is een alarm voor een sensor. Zo wordt de gebruiker doormiddel van een notificatie in een applicatie gewaarschuwd of een bepaalde sensor waarde boven of onder een door de gebruiker ingestelde waarde zit.

De onderzoeksvraag die hierbij hoort is dan ook:

Hoe kan een server zo ingericht worden dat het informatie kan ontvangen van de LoRaWAN Cloud, deze informatie kan verwerken, notificaties naar een Androidapplicatie gestuurd worden en er data kan worden uitgewisseld tussen een Androidapplicatie en de server?

English:

LoRa sensors are sensors of which the value of the sensor is send by LoRa. LoRa is a technique in which messages are sent with low power over a long distance. Therefore, the acronym LoRa stands for 'Long Range Low Power'. When LoRa sensor sends data this will be received by the LoRa network. This LoRa network is the combination of antennas (gateways) and the LoRaWAN Cloud (server). Once a LoRa message is received a gateway, it will send this message to the LoRaWAN Cloud. This LoRaWAN Cloud can be set by website, to send the LoRa messages to a private server.

This private server in this case a WampServer. A WampServer is a server that runs Windows and is a combination of Apache, MySQL and PHP. This WampServer can create a database which contains

tables with data in it. The data in these tables can be removed, replaced, put or read by an application. After receiving LoRa message the LoRaWAN Cloud will make a request to the WampServer to set the data in a table. During this request the WampServer will check a number of things, including a check on completeness of the request. After this check the data will be stored in the right table. Also the values in the data will be checked if it no critical values. This critical values are values witch are higher or lower than previously stored values (this values are stored in a table in the database on the WampServer). If the data contains critical values the WampServer will send a notification to an application.

In this application the user can retrieve all the information from the sent LoRa messages, for example the values of the sensors at a specific date and time. Also in this application the user can set the values witch will be compared to the values in a LoRa message. This are the values as mentioned previously, the values that checks if values are critical or not. The notification, that the application receives, is an alarm that a value is a critical value. So if the application received a notification the user knows that one of the sensors has a critical value that is above or below a value that the user has set.

In this case the research question is:

How can different sensors send their data via the LoRa network in a way an application can read this data and receive notifications that the sensor values are below or above the threshold values set by the application?

Inhoudsopgaven

LIJST MET FIGUREN.....	9
LIJST MET TABELLEN	10
1 INLEIDING.....	12
2 ACHTERGRONDEN	14
2.1 Het bedrijf	14
2.2 Producten	15
2.3 Aanleiding voor het project	15
3 OMSCHRIJVING VAN DE OPDRACHT.....	16
3.1 Huidige situatie.....	16
3.2 Doelstelling.....	17
3.3 Uitwerking opdracht omschrijving	17
3.4 Onderzoeksvraag	19
3.5 Deelvragen	19
4 LORA SENSOREN.....	21
4.1 LoRa demo Marvin (Arduino).....	21
4.1.1 Hardware	21
4.1.2 Software	23
4.2 Verzenden van een LoRa bericht doormiddel van een RN2483	25
4.2.1 Initialisatie	25
4.2.2 Verzending.....	26
4.3 LoRa demo beweging sensor en temperatuur weertand (NTC)	27
4.3.1 Hardware	28
4.3.2 Software	30
4.4 LoRa demo Rookmelder.....	33
4.4.1 Hardware	33
4.4.2 Software	34

4.5	Sensoren, sensor data en sensor kalibratie	35
4.5.1	Gebruikte sensoren	35
4.5.2	Formule Temperatuursensor ((ND03R00104J, 100kΩ NTC)	36
4.5.3	Formule Grove – Temperatuursensor V1.2	38
4.5.4	Kalibratie sensoren	38
4.5.4.1	Temperatuursensor (ND03R00104J, 100kΩ NTC)	38
4.5.4.2	Temperatuursensor (Grove – Temperatuursensor V1.2)	39
4.5.4.3	Lichtintensiteit sensor (Grove – Light sensor V1.1)	40
4.5.5	Ongebruikte sensoren	41
5	LORAWAN CLOUD SERVER	42
5.1	KPN LoRa Developer Portal	42
5.1.1	AS_KEY (Application Server Key)	42
5.1.2	Nwkskey (Netwrok Session Key)	42
5.1.3	Devaddr (Device Adress)	42
5.1.4	Appskey (Application Session Key)	42
5.1.5	Device EUI	42
6	SERVER	43
6.1	WampServer installeren	43
6.2	WampServer configuratie	44
6.3	DNS en Virtual Machine	44
6.4	Instellen poort WampServer	45
6.5	SSL Certificaat	45
6.6	Waarom DNS en SSL-Certificaat gebruikt worden	48
6.7	Bestanden om WampServer te kunnen gebruiken	48
6.8	Tabellen	50
6.9	User Configuration (Gebruikersconfiguratie)	52
6.10	Sensors	53
6.10.1	SQL-tabel Sensors	53
6.10.2	GET Sensors	53
6.10.3	DELETE Sensors	54
6.11	Sensor Data	55
6.11.1	SQL-tabel Sensor Data	55

6.11.2	POST Sensor Data	55
7	ANDROIDAPPLICATIE	59
7.1	Programma	59
7.2	Opbouw applicatie	60
7.2.1	Nieuw project	60
7.3	Request	61
7.4	Lijst met data	63
7.5	Verschillende onderdelen applicatie	64
7.5.1	Login (LoginActivity)	64
7.5.2	Register (RegisterActivity)	64
7.5.3	Sensors (SensorsActivity)	65
7.5.4	Data (DataActivity)	65
7.5.5	Triggers (TriggersActivity)	66
7.5.6	Alarms (AlarmsActivity)	67
7.5.7	Notifications (NotificationsActivity)	68
7.5.8	Sensor Data (SensorsShowData)	68
7.5.9	Sensor grafiek (SensorsShowGraph)	69
7.5.10	Belangrijke code uitleg, gebruiker configuratie	70
8	VOORDELEN EN BEPERKINGEN VAN LORA.....	71
8.1	Voordelen	71
8.2	Beperkingen	71
8.3	LoRa sensoren in Smart Buildings	72
9	CONCLUSIE	73
9.1	Onderzoeksvragen	73
9.2	Verbeteringen	75
10	AANBEVELING	77
11	BIJLAGE.....	79
11.1	Bronvermelding	79
11.1.1	Literatuur	79

11.1.2	Internet.....	79
11.2	Componentenlijst met datasheets	81
11.3	Sensor Triggers	82
11.3.1	SQL-tabel Sensor Triggers.....	82
11.3.2	GET Sensor Triggers.....	82
11.3.3	POST Sensor Triggers.....	82
11.3.4	DELETE Sensor Triggers	83
11.4	Sensor Data	83
11.4.1	GET Sensor Data	83
11.4.2	DELETE Sensor Data.....	83
11.5	Sensor Alarms.....	84
11.5.1	SQL tabel Sensor Alarms.....	84
11.5.2	GET Sensor Alarms	84
11.5.3	DELETE Sensor Alarms	84
11.6	User Register	85
11.6.1	POST User Register	85
11.7	User Login.....	85
11.7.1	SQL-tabel User Login	85
11.7.2	POST User Login	85
11.8	Sensor Names.....	86
11.8.1	SQL-tabel Sensor Names	86
11.8.2	GET Sensor Names	86
11.8.3	POST Sensor Names	86
11.8.4	DELETE Sensor Names.....	86
11.9	Notification Tokens.....	87
11.9.1	SQL-tabel Notification tokens	87
11.9.2	GET Notification Tokens	87
11.9.3	POST Notification Tokens	87
11.10	Het frame van de applicatie.....	88
11.11	LoRa kosten	90
11.12	httpd.conf	91
11.13	http.v-host.conf.....	94
11.14	Volledige meting voor Kalibratie ND03R00104J	94
11.15	Volledige meting voor Kalibratie Grove – Temperatuursensor V1.2	100

11.16	Formule Temperatuur (NTCLE100E3334JB0, NTC 330kΩ) afgekeurde sensor	108
-------	---	-----

Lijst met figuren

Figuur 1	Organogram KPN (vanuit Technology Labs)	14
Figuur 2	Notificatie voorbeeld	18
Figuur 3	Overzicht van LoRa sensor tot applicatie.	18
Figuur 4	Marvin bord [7]	21
Figuur 5	DHT11 en DHT22 sensor aan een Grove connector	22
Figuur 6	Software LoRa demo Marvin (Arduino)	24
Figuur 7	Verzenden van een LoRa bericht met een RN2483	25
Figuur 8	Low Power Modes van de MSP430 [4]	27
Figuur 9	Hardware van LoRa beweging sensor en temperatuur weerstand (NTC)	28
Figuur 10	Interrupt Service Routine en slaap modus (software)	30
Figuur 11	Flowchart software LoRa beweging sensor en temperatuur weerstand (NTC)	32
Figuur 12	Hardware LoRa demo Rookmelder	33
Figuur 13	Flowchart software LoRa demo Rookmelder	34
Figuur 14	Spanningsdeler	36
Figuur 15	Spanningsdeler	40
Figuur 16	Grafiek lichtsterkte (in lux) tegenover de LDR waarde (in kΩ) uit de datasheet van de GL5528	41
Figuur 17	Edit Device venster met nwkskey, devaddr en appskey	42
Figuur 18	Configuratie bestanden WampServer	44
Figuur 19	WampServer benaderen	44
Figuur 20	WampServer op de virtuele machine	45
Figuur 21	SSL-Certificaat ketting	46
Figuur 22	Include map bestanden	48
Figuur 23	lora_demo_server map	48
Figuur 24	Server werking	49
Figuur 25	Structuur (structuur van de SQL-tabel)	51
Figuur 26	Verkennen (data in SQL-tabel)	51
Figuur 27	POST sensor_data token check	56
Figuur 28	Flowchart van een POST request naar sensor_data tabel	58
Figuur 29	Navigation Drawer	60
Figuur 30	Server werking	61
Figuur 31	POST request code	62
Figuur 32	Voorbeeld van een lijst (foto AndroidHive.info)	63
Figuur 33	Login scherm	64
Figuur 34	Register scherm	64
Figuur 35	Sensors scherm	65
Figuur 36	Data scherm	66

Figuur 37 Trigger scherm.....	66
Figuur 38 Alarms scherm.....	67
Figuur 39 Notificatie voorbeeld.....	68
Figuur 40 Notifications scherm	68
Figuur 41 Sensor Data scherm.....	68
Figuur 42 Sensor grafiek scherm (gedraaid).....	69
Figuur 43 MainActivity na openen nieuw project.....	88
Figuur 44 LoRa kosten	90
Figuur 45 Spanningsdeler	108
Figuur 46 Formule datasheet van de NTCLE100E3334JB0.....	108
Figuur 47 Constante voor formule met een Bèta van 4570K (datasheet NTCLE100E3334JB0).....	109

Lijst met tabellen

Tabel 1 Gekozen sensoren.	17
Tabel 2 Grove connectoren aansluitingen sensoren.....	22
Tabel 3 Modi met functies ervan	27
Tabel 4 Gebruikte sensoren	36
Tabel 5 DHT22 en ND03R00104J zes verschillende temperatuurmetingen	39
Tabel 6 Vijf metingen van de temperatuur in graden DHT22 sensor en Grove – Temperatuursensor V1.2.....	40
Tabel 7 Ongebruikte sensoren	41
Tabel 8 CSR-aanvraag opdracht prompt vragen	46
Tabel 9 Config.php code.....	48
Tabel 10 Uitleg SQL-tabel	52
Tabel 11 SQL-tabel Sensors	53
Tabel 12 GET-sensors	54
Tabel 13 DELETE sensors	55
Tabel 14 SQL-tabel Sensor Data	55
Tabel 15 Payload data van één sensor	57
Tabel 16 Literatuur	79
Tabel 17 Internet.....	80
Tabel 18 Gebruikte componenten met bijbehorende datasheet	81
Tabel 19 SQL-tabel Sensor Triggers.....	82
Tabel 20 GET Sensor Triggers	82
Tabel 21 POST Sensor Triggers	83
Tabel 22 DELETE Sensor Triggers.....	83
Tabel 23 GET Sensor Data	83
Tabel 24 DELETE Sensor Data	84
Tabel 25 SQL tabel Sensor Alarms.....	84
Tabel 26 GET Sensor Alarms.....	84

Tabel 27 DELETE Sensor Alarms	85
Tabel 28 POST User Register	85
Tabel 29 SQL-tabel User Login.....	85
Tabel 30 POST User Login.....	86
Tabel 31 SQL-tabel Sensor Names.....	86
Tabel 32 GET Sensor Names	86
Tabel 33 POST Sensor Names.....	86
Tabel 34 DELETE Sensor Names	87
Tabel 35 SQL-tabel Notification tokens.....	87
Tabel 36 GET Notification Tokens	87
Tabel 37 POST Notifications Tokens.....	87
Tabel 38 Volledige meting voor Kalibratie ND03R00104J.....	99
Tabel 39 Volledige meting voor Kalibratie Grove – Temperatuursensor V1.2	107

1 Inleiding

LoRa sensoren zijn sensoren waarbij het signaal een grote afstand kan bereiken met een laag verbruik. In Nederland heeft KPN een landelijk dekkend netwerk gemaakt om LoRa berichten te kunnen ontvangen. Dit zorgt ervoor dat LoRa sensoren in Nederland overal, waar (gereflecteerd) daglicht te zien is, gegevens van de sensoren kan verzenden. Ook gebeurt het verzenden met een laag stroom verbruik. Deze combinatie maakt het mogelijk om sensoren te bouwen die, gekoppeld aan een volle batterij, een hele lange tijd mee gaat. Hierbij moet gedacht worden aan 10 jaar of langer!^[11] Het voordeel hiervan is dat de sensoren niet afhankelijk zijn van een voedingsbron, zoals een stopcontact. Ook hoeft er voor een hele lange tijd niet naar de sensor omgekeken te worden omdat de batterij zo lang mee gaat. Dit maakt het mogelijk om sensoren te plaatsen op lastige plekken. Bijvoorbeeld plekken waar geen stopcontact in de buurt is. Zo kan een boer, die de condities van zijn land wilt meten, overal sensoren op zijn land plaatsen waar die maar wilt. Want op zijn land kunnen de LoRa sensoren immers altijd data versturen en werken de sensoren voor een hele lange tijd een batterij.

Zo is ProRail ook een proef begonnen met LoRa^[8]. ProRail gaat de temperaturen van de wissels monitoren doormiddel van LoRa sensoren. Dit om te zien of de verwarmingen die in de wissels zitten wel correct werken. De LoRa sensoren zullen om de zoveel tijd een LoRa bericht versturen met daarin de temperatuur van de wissels. Zo krijgt ProRail een goed beeld van de temperaturen van de wissels over een langere tijd. Hieruit kan ProRail opmaken of een wissel niet te koud wordt (en kan bevriezen). Als een wissel namelijk te koud wordt is er waarschijnlijk iets mis met de verwarming. Als reactie daarop zal ProRail iemand langs sturen om de verwarming in de wissel te laten maken. Bij het tijdig laten maken van de verwarming in een wissel voorkomt ProRail dat de wissel beschadigd door bevriezing of dat er treinen niet meer langs kunnen omdat de wissel niet meer kan omschakelen door bevriezing. Zo heeft ProRail baat bij het gebruik van de LoRa sensoren.

LoRa sensoren kunnen ook gebruikt worden in huizen of in gebouwen. Omdat de LoRa sensoren op batterij lang mee gaan, kunnen de sensoren overal geplaatst worden waar mogelijk. De enige voorwaarde is dat er daglicht (reflectie) te zien moet zijn. Het maakt daarom ook niet uit of de sensoren verplaatst worden, dit omdat de sensoren niet gebonden zijn aan bijvoorbeeld een stopcontract.

LoRa berichten versturen de sensor data naar het LoRa netwerk. Dit heeft als voordeel dat de sensoren niet verbonden te hoeven zijn met bijvoorbeeld de Wifi in huis of in een gebouw. De LoRa sensoren kunnen zich daarom ook overal rond het huis of gebouw kunnen bevinden. Zo kan bijvoorbeeld de lichtsterkte buiten gemeten worden en aan de hand daarvan rolluiken worden aangestuurd.

In deze scriptie zal het gebruik van LoRa sensoren bestudeerd worden. Dit om uit te zoeken of LoRa sensoren gebruikt kunnen worden in een technisch gebouw van KPN. De opdracht wordt uitgevoerd op de afdeling Technology Labs van KPN.

De aanleiding van het project is het bouwen van een demo die het gebruik van een sensornetwerk toepassing en de bruikbaarheid van het LoRa netwerk demonstreert. Dit in het bijzonder voor de toepassing van een (technische)ruimte monitoring platform. Het resultaat van deze demo is om interne en externe klanten die bij Technology Labs komen enthousiast te maken om het LoRa netwerk te gaan gebruiken.

In deze scriptie worden de volgende hoofdstukken uitgewerkt:

- **Achtergronden**, in dit hoofdstuk worden de achtergrond van het project beschreven met daarin het bedrijf gezien vanuit Technology Labs, de producten die Technology Labs oplevert en de aanleiding van het project.
- **Omschrijving van de opdracht**, in dit hoofdstuk wordt de opdracht beschreven, de huidige situatie geschetst, de doelstelling opgesteld en aan de hand daarvan onderzoeksvragen en deelvragen opgesteld.
- **LoRa sensoren**, in dit hoofdstuk worden de LoRa sensoren uitgelegd, het gaat hier om drie demo's waarbij de hardware en software beschreven wordt. Ook wordt er uitgelicht welke sensoren er gebruikt worden, welke data deze sensoren afgeven en hoe de sensoren gekalibreerd worden.
- **LoRaWAN Cloud server**, in dit hoofdstuk wordt de LoRaWAN Cloud server uitgelegd. Hierbij wordt verteld hoe deze server is in te stellen.
- **Server**, in dit hoofdstuk wordt er uitgelegd hoe de server is opgebouwd en wat er allemaal voor nodig is om de server te laten draaien. Ook worden de functies die op de server gebruikt worden uitgebreid uitgelegd. Dit alles om een compleet beeld te krijgen over hoe de server werkt en in elkaar zit.
- **Androidapplicatie**, in dit hoofdstuk wordt de Androidapplicatie uitgelicht met daarin alle schermen die in de applicatie tegen zijn te komen. Ook wordt er uitgelegd welk programma er voor het programmeren van de Androidapplicatie wordt gebruikt, hoe de applicatie is opgebouwd, wat een request is en hoe de verschillende data in de applicatie worden weergegeven.
- **Voordelen en beperkingen van LoRa**, in dit hoofdstuk worden de voordelen en beperkingen van het gebruik van het LoRa netwerk uitgelicht.
- **Conclusie**, in dit hoofdstuk staat de conclusie van de opdracht deze conclusie wordt gedaan aan de hand van de onderzoeksvragen. Ook wordt in de conclusie besproken wat er nog verbeterd zou kunnen worden aan het project.
- **Aanbeveling**, in dit hoofdstuk wordt een aanbeveling gedaan voor het gebruik van LoRa.
- **Bijlage**, in dit hoofdstuk staat de bronvermelding, de componentenlijst met bijbehorende datasheets en alle extra informatie.

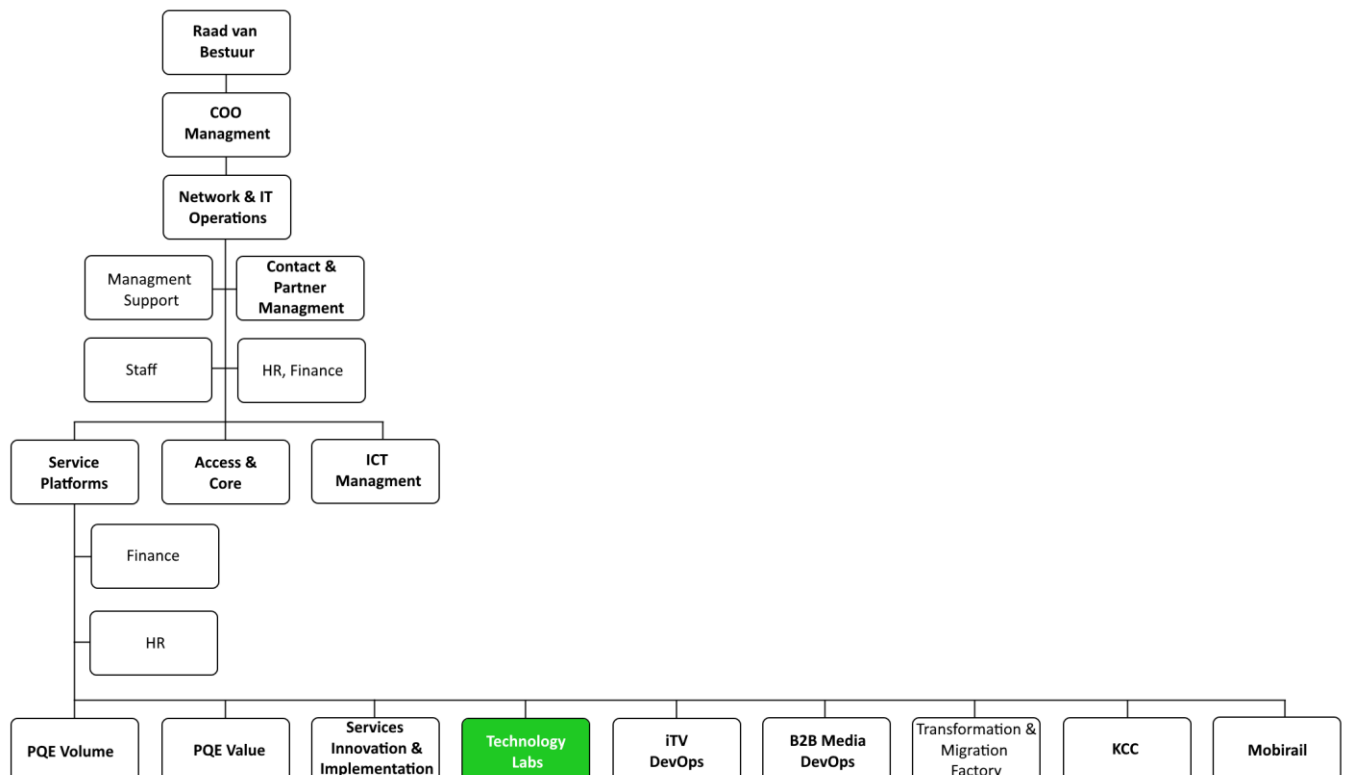
2 Achtergronden

2.1 Het bedrijf

Het bedrijf KPN uit het oog van Technology Labs is als volgt opgebouwd, bovenin het organogram staat het Raad van Bestuur. Deze Raad stuurt het COO (Chief Operations Officer) Management aan. Dit management stuurt op haar beurt de Network & IT Operations afdeling aan. Network & IT Operations sturen drie afdelingen aan: Service Platforms, Access & Core Networks en ICT Management. Hierbij wordt Network & IT Operations ondersteunt door het Management Support, het Contact & partner Management, HR (Human Resources), Finance en verdere Staff functies. De Service Platforms zorgen voor innovatie /ontwikkeling van nieuwe diensten. De Access & Core Networks zorgt voor innovatie van het acces en core netwerken. Het ICT Management zorgt voor het beheer van de netwerken en diensten.

Technology Labs valt onder de afdeling Service Platforms. Deze afdeling stuurt nog acht andere afdelingen aan ondersteund door HR en Finance.

Het organogram van KPN is te zien in Figuur 1, deze is opgebouwd uit het oog van Technology Labs, daardoor staan de niet relevante afdelingen er niet op.



Figuur 1 Organogram KPN (vanuit Technology Labs)

2.2 Producten

Het product dat Technology Lab oplevert zijn verschillende Proof of Concepts (PoC's). In deze PoC's wordt aangetoond of een idee levensvatbaar is. Dit wordt aan de hand van een aantal stappen gedaan. Eerst komen er verschillende (technologische) ideeën in een zogenoemde ideeën box. Uit deze ideeën box worden, op basis van ervaring en inzicht, de belangrijke ideeën uitgekozen. Deze ideeën worden uitgewerkt in een kort plan van aanpak met hierin de doelstelling, grenzen, wie het nodig heeft en wie er betrokken zijn. Deze plan van aanpakken worden getoetst bij een commissie van mensen uit het hoger management. Deze commissie zal beslissen of de ideeën geschikt zijn om door te onderzoeken. Deze ideeën zullen dat moeten voldoen aan een of meerder van de volgende pilaren:

- Technology leadership, hiermee wordt bedoeld dat het idee nog niet bestaat (in Nederland) en KPN zich daardoor leider in de markt kan maken.
- Kosten verlagend
- NPS verhogend, hiermee wordt bedoeld dat het klanten tevredenheid verhoogd wordt door bepaalde producten of diensten te verbeteren. Hiermee hoopt KPN haar klanten langer aan zich te binden.
- KPN Strategie, hiermee wordt bedoeld dat het idee de KPN Strategie gaat verbeteren of ondersteunen. Zoals in dit geval dat KPN het LoRa netwerk landelijk heeft uitgerold, dit is een nieuwe strategie van KPN.
- New business/new services, hiermee wordt het uitbreiden van het portfolio bedoeld. Een bestaande service wordt uitgebreid.

Als de ideeën door de commissie goed gekeurd zijn zullen de ideeën uitgewerkt worden tot een PoC. In dit PoC zal worden uitgezocht of het idee levensvatbaar is. Dit kan ook worden aangetoond met een demonstratie van het idee.

2.3 Aanleiding voor het project

De aanleiding van het project is het bouwen van een demo die het gebruik van een sensornetwerk toepassing en de bruikbaarheid van het LoRa netwerk demonstreert. Dit in het bijzonder voor de toepassing van een (technische)ruimte monitoring platform. Het resultaat van deze demo is om interne en externe klanten die bij Technology Labs komen enthousiast te maken om het LoRa netwerk te gaan gebruiken.

3 Omschrijving van de opdracht

De opdracht omschrijving zoals die geformuleerd is door KPN.

Doel van de opdracht:

Ontwikkelen, implementeren en analyseren van een Smart Building systeem bestaande uit LoRa sensoren in een technisch gebouw van KPN.

Ontwikkelen & implementeren:

Door de sensoren te registreren op het LoRa netwerk voor IoT (Internet of Things) van KPN kan er een actueel status overzicht (User Interface) worden verkregen van het technische gebouw. Hiervoor kunnen bestaande of zelf te ontwikkelen sensoren worden gebruikt. Denk aan sensoren zoals temperatuur, luchtvochtigheid, luchtdruk, geluidsniveau, knoppen en schakelaars (deur open/dicht, receptiebel). Een belangrijke eigenschap voor het Smart Building systeem is de mogelijkheid om alarmeringen en notificaties te ontvangen.

Analyse:

Met het eindverslag verwachten wij inzicht te verkrijgen in de voordelen en beperkingen van de LoRa sensortechnologie voor smart building concepten.

Vorbereiding:

Van de student wordt verwacht vooraf een planning te maken van de uit te voeren werkzaamheden en daarbij het kader van het te ontwikkelen Smart Building systeem vast te leggen.

Oplevering:

Ter afsluiting van de opdracht wordt een onderzoeksrapport opgeleverd over LoRa sensoren voor Smart Building systemen. Daarnaast zal een werkend prototype bestaande uit verschillende type LoRa sensoren met bijbehorende User Interface inclusief de implementatiedocumentatie van dit prototype onderdeel zijn van de oplevering.

3.1 Huidige situatie

Voor de huidige situatie is gekeken naar het technische gebouw PB400 van KPN. In dit gebouw zijn een al een aantal sensor systemen aanwezig:

- Deursensoren die meten of een deur open of dicht is.
- Rookmelders die meten of er rook aanwezig is.
- Lichtintensiteit sensor die meet wat de lichtsterkte is.

De eerste twee sensoren zijn voor alarmering. Zo zal er een alarm in de meldkamer afgaan als de deursensor meet dat er een deur open staat die niet open hoort te staan. Ook zal er een alarm afgaan als er ergens rook wordt gedetecteerd. Dit alarm zal ook alle alarmen in het gebouw activeren zodat er een evacuatie in gang kan worden gezet.

De derde sensor, de lichtintensiteit sensor, meet de lichtintensiteit aan de hand daarvan zullen de rolluiken bediend worden. De rolluiken zullen bij weinig licht omhoog, en bij veel licht omlaag, gedaan worden.

Het grote nadeel van deze sensoren is dat ze allemaal via kabel verbonden zijn met een centrale computer die daarna iets met deze gegevens zal doen, bijvoorbeeld het waarschuwen van de meldkamer.

3.2 Doelstelling

De doelstelling voor dit project is om te bekijken of het LoRa sensoren geschikt zijn om te gebruiken in een technisch gebouw van KPN. Dit zal gebeuren aan de hand van het maken van een prototype. Dit prototype zal de gebruiker moeten waarschuwen als een LoRa sensor een bepaalde waarde heeft, dit is te vergelijken met bijvoorbeeld het rookalarm in de huidige situatie. Het hele project zal worden uitgewerkt in een onderzoeksrapport en implementatiedocumentatie van het prototype van de LoRa sensor.

3.3 Uitwerking opdracht omschrijving

Om een Smart Building systeem zo goed mogelijk te kunnen nabootsen in een prototype is er gekozen voor de onderstaande sensoren. Dit omdat deze sensoren allemaal in een Smart Building systeem voor kunnen komen en daardoor ook zo toegepast kunnen worden binnen een Smart Building systeem.

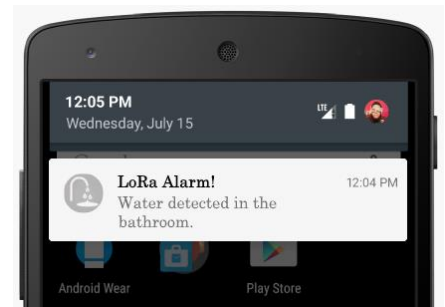
Logo	Soort Sensor	Sensor Meeting	Sensor Toepassing(en)	Reactietijd
	Bewegingssensor	Detecteert beweging.	<ul style="list-style-type: none"> - Als er geen beweging wordt gedetecteerd kan het licht uit gezet worden, omdat er niemand aanwezig is. - Geef een waarschuwing als er beweging is waar niemand aanwezig mag zijn. 	Direct (binnen 1 seconde data versturen)
	CO- en Rookmelder	Detecteert koolstofmonoxide en rook.	Geef een waarschuwing als de CO- en Rookmelder afgaat.	Direct (binnen 1 seconde data versturen)
	Luchtvochtigheidssensor	Meet de luchtvochtigheid (in %).	Geef een waarschuwing of stuur een regelaar aan bij een bepaalde luchtvochtigheid.	Om het half uur
	Temperatuursensor	Meet de temperatuur (in °C).	Geef een waarschuwing of stuur een regelaar aan bij een bepaalde temperatuur.	Om het half uur
	Glasbreuksensor	Detecteert of de ruit, waar de sensor op aangesloten is, gebroken is.	Geef een waarschuwing als de ruit gebroken is.	Direct (binnen 1 seconde data versturen)
	Deur- en Raamsensor	Detecteert of de deur of het raam open of dicht is.	Geef waarschuwing als is ingesteld dat de deur of het raam niet open mag zijn en dit wel is.	Direct (binnen 1 seconde data versturen)
	Waterdetector	Detecteert water.	Geef een waarschuwing als er water aanwezig is.	Direct (binnen 1 seconde data versturen)
	Lichtintensiteit sensor	Meet de lichtintensiteit (in lx).	<ul style="list-style-type: none"> - Zet het licht aan of uit bij een bepaalde lichtintensiteit. - Zet het zonnescherm naar beneden bij een bepaalde lichtintensiteit. 	Om het half uur

Tabel 1 Gekozen sensoren.

Naast deze sensoren moet er ook een User Interface komen. Deze User Interface zal drie dingen moeten kunnen:

- Sensor waardes weergeven
- Instellingen invoeren (bijvoorbeeld, stuur een waarschuwing als de temperatuur boven de 25 graden wordt)
- Waarschuwingen ontvangen

Voor deze User Interface is gekozen om een Androidapplicatie te maken. Deze Androidapplicatie zal de data van de sensoren weergeven in overzicht. In dit overzicht kunnen alle sensor instellingen apart van elkaar gewijzigd worden. Daarnaast zal de Androidapplicatie een notificatie ontvangen als waarschuwing voor bepaalde voorwaardes. Een voorbeeld van hoe een notificatie eruit ziet is te zien in Figuur 2.

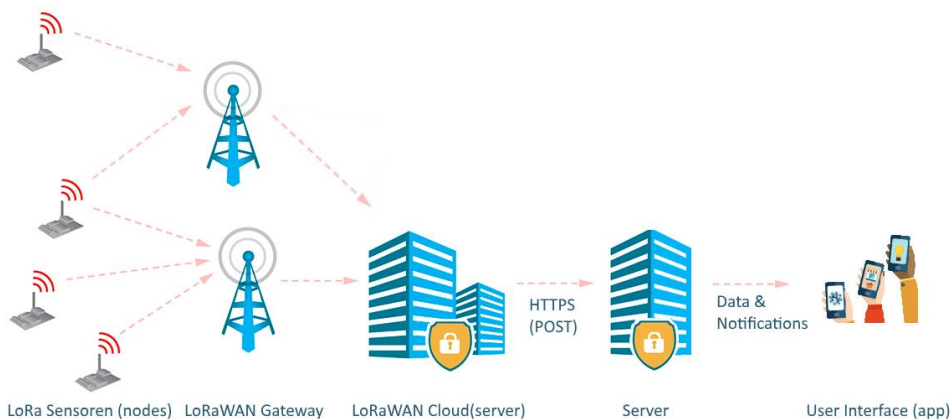


Figuur 2 Notificatie voorbeeld

Om deze sensoren uiteindelijk weer te kunnen geven in de Androidapplicatie zijn een aantal onderdelen nodig (zie Figuur 3):

- LoRa sensoren, dit is een samenvoeging van sensoren met een LoRa zender. Deze LoRa sensoren zullen de sensor gegevens doorsturen naar alle LoRaWAN gateways die binnen het bereik van de LoRa sensor staan.
- LoRaWAN gateways, deze gateways sturen de LoRa sensor gegevens door naar de LoRaWAN Cloud.
- LoRaWAN Cloud, de LoRaWAN Cloud is zo in te stellen dat de LoRa sensor gegevens die worden ontvangen worden doorgestuurd naar de eigen server. Als hetzelfde bericht door meerdere gateways wordt doorgestuurd zal de LoRaWAN Cloud het bericht maar één keer naar de eigen server sturen.
- Server, de (eigen) server zal de gegevens vergelijken met de instellingen die de gebruiker (via de app) ingesteld heeft. Als deze gegevens een kritische waarde hebben en er een waarschuwing gestuurd moet worden, zal de server een notificatie naar de Androidapplicatie sturen

De Androidapplicatie zal uiteindelijk notificaties moeten kunnen ontvangen, sensor gegevens moeten kunnen opvragen van de server en instellingen naar de server moeten kunnen sturen.



Figuur 3 Overzicht van LoRa sensor tot applicatie.

De volgende punten zullen daarom uitgewerkt moeten worden:

- De LoRa sensoren
- De LoRaWAN Cloud instellen dat het de sensor informatie naar de eigen server stuurt.
- De eigen server.
- De Androidapplicatie.
- Verslag over de bovengenoemde punten

3.4 Onderzoeksvraag

Hoe kunnen verschillende sensoren hun gegevens via het LoRa netwerk versturen zodat een applicatie deze gegevens kan uitlezen en notificaties (waarschuwingen) kan ontvangen als sensor gegevens een onder of boven een door de applicatie ingestelde waarde komen?

De onderzoeksvraag omvat het begin van de keten (zoals te zien is in Figuur 3): de sensoren die op het LoRa netwerk worden aangesloten. En het einde van de keten: de applicatie. Wat daar tussen zit is al automatisch omvatten door het begin en einde. Want de data van de sensoren zal via het LoRa netwerk bij de applicatie moeten uitkomen. Dit proces zal niet gemaakt kunnen worden zonder het midden stuk: het LoRa netwerk en de server. Deze vraag resulteert daarom in een aantal deelvragen.

3.5 Deelvragen

Hoe werken de verschillende sensoren en wat is de output van de sensoren?

Eerst zal er onderzocht moeten worden hoe de gekozen sensoren werken en wat de output (uitgang) van deze sensoren is. Dit moet gebeuren om te weten te komen hoe de sensoren werken, hoe er data uit de sensoren gehaald kan worden en wat deze data voorstelt. Dit om uiteindelijk aan de applicatie kant deze data te kunnen vertalen naar de echte waarde.

Hoe kan de sensor data naar het LoRa netwerk verstuurd worden?

Als er bekend is welke data de sensoren generen is het de vraag hoe deze data verstuurd kan worden naar het LoRa netwerk. Dit zal moeten gebeuren met een LoRa zender die de data zal doorsturen naar het LoRa netwerk. Hoe dit moet en hoe dit gaat zal moeten worden onderzocht.

Hoe kan de LoRaWAN Cloud ingesteld worden dat het de sensor gegevens naar de eigen server stuurt?

Als de LoRa sensoren eenmaal data versturen naar het LoRa netwerk zal deze data uit het netwerk naar de eigen server gestuurd moeten worden. Dit gebeurt doormiddel van een HTTPS-POST bericht. Dit is een bericht die de server ontvangt en de server aanzet tot een actie. In dit geval zal het betekenen dat de server de informatie zal opslaan en zal kijken of er een actie ondernomen moeten worden. Dit houdt in, als de LoRaWAN Cloud bijvoorbeeld de temperatuur doorstuurt van de temperatuursensor en de server is zo ingesteld (door de applicatie) dat de server boven de 25 graden een notificatie (waarschuwing) moet versturen, dat deze notificatie ook verstuurd wordt.

Hoe kan een server zo ingericht worden dat het informatie kan ontvangen van de LoRaWAN Cloud, deze informatie kan verwerken, notificaties naar een Androidapplicatie gestuurd worden en er data kan worden uitgewisseld tussen een Androidapplicatie en de server?

Deze vraag omvat alles wat de server moet kunnen, de server moet informatie kunnen ontvangen, informatie verwerken en informatie doorsturen. Met deze informatie wordt de data van de sensor of de data uit de applicatie bedoeld.

Hoe kan een Androidapplicatie notificaties ontvangen, data van de server weergeven en data naar de server sturen?

Deze vraag omvat alles wat de Androidapplicatie moet kunnen. Hoe dit er precies uit komt te zien is iets voor later. Het belangrijkste is dat deze drie functies erin verwerkt worden.

Wat zijn de voordelen van het LoRa netwerk?

Deze vraag heeft vooral betrekking op de ervaringen die tijdens deze opdracht worden opgedaan. In het eindverslag zal duidelijk naar voren komen wat de voordelen van het LoRa netwerk zijn. Dit heeft betrekking tot de LoRa sensoren in een Smart Building.

Wat zijn de beperkingen van het LoRa netwerk?

Net als de vraag wat de voordelen van het LoRa netwerk zijn, kan ook de vraag gesteld worden wat de beperkingen zijn. Deze beperkingen zullen ook in het eindverslag naar voren komen. Dit heeft betrekking tot de LoRa sensoren in een Smart Building.

4 LoRa sensoren

Voor de LoRa sensoren zijn er drie verschillende varianten gekozen. De eerste een LoRa demo doormiddel van een Marvin bord. Deze demo is gemaakt met een al bestaand bord die uitgewerkt is tot een LoRa demo die maar liefst 8 sensoren uitleest en deze naar het LoRa netwerk stuurt. De tweede demo is een LoRa demo met een bewegingssensor en een temperatuur weerstand. Hierbij is de hardware zelf bedacht en tot demo gemaakt. Deze demo zal laten zien hoe twee verschillende sensoren uitgelezen kunnen worden en deze data verstuurd wordt naar het LoRa netwerk. De laatste demo is een demo waarbij een rookmelder wordt gebruikt om via een microcontroller en een LoRa module een bericht naar het LoRa netwerk te sturen als er rook wordt gedetecteerd. Het doel van deze berichten is dat ze later via een server kunnen worden uitgelezen door de gebruiker (in een Androidapplicatie). Zo is de gebruiker altijd op de hoogte van de data die uit de sensoren wordt gehaald.

4.1 LoRa demo Marvin (Arduino)

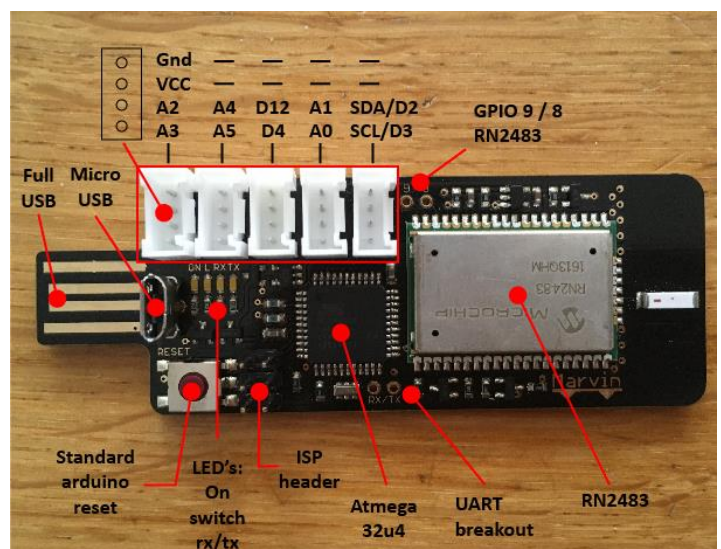
Het Marvin bord zoals te zien in Figuur 4 is gemaakt door de IoT Academy. Deze IoT Academy is een stichting die workshops geeft over het gebruik van LoRa. Iedereen die aan een workshop van de IoT Academy meedoet ontvangt een Marvin bord. Op dit Marvin bord zit een Arduino (ATmega32u4) microcontroller die direct via de USB is aan te sturen. Deze Arduino is verbonden met een RN2483 module. Dit is een module die LoRa berichten kan versturen.

4.1.1 Hardware

De hardware bestaat uit een Marvin bord zoals te zien in Figuur 4. Op die bord zit een USB-aansluiting en een micro USB-aansluiting. Deze aansluitingen kunnen gebruikt worden om de Arduino op het Marvin bord te programmeren. De Arduino is met RX en TX pinnen (UART-pinnen) verbonden met de RN2483. Over deze pinnen kunnen, door de Arduino, commando's verzonden en ontvangen worden van de RN2483. Dit is voor het verzenden van een LoRa bericht. Hoe dit verzenden in zijn werk gaat is te lezen in hoofdstuk 4.2 *Verzenden van een LoRa bericht doormiddel van een RN2483*.

Dit bord heeft ook een aantal leds zitten. Een led die aangeven dat het bord aanstaat, een led die via de Arduino aan te sturen is, een led aan de TX en een led aan de RX. Deze laatste twee leds zullen gaan knipperen als er een bericht over de RX of TX lijn verstuurd wordt. Dit geeft het verzenden en ontvangen van berichten naar en van de RN2483 module aan.

Verder heeft de Marvin ook een reset knop om de Arduino te kunnen resetten en een antenne die aangesloten is aan de RN2483. Deze antenne is ervoor bedoeld om de LoRa berichten van de RN2483 naar het LoRa netwerk te kunnen versturen. Ook zitten aan de Arduino vijf connectoren verbonden (zie rode rechthoek) deze connectoren heten Grove connectoren. In de onderstaande tabel (zie Tabel 2) is te zien welke sensoren erop welke connectoren aangesloten zitten.



Figuur 4 Marvin bord [7]

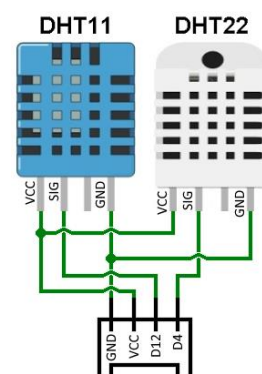
Connector nummer	Aangesloten sensor
1 (verbonden met Arduino pin A2 en A3)	Grove - Lichtsterktesensor (LDR-weerstand) (A3)
2 (verbonden met Arduino pin A4 en A5)	Grove - Watersensor (pin A5)
3 (verbonden met Arduino pin D12 en D5)	DHT11 (pin D12) en DHT22 (pin D5)
4 (verbonden met Arduino pin A1 en A0)	Grove – Temperatuursensor (NTC-weerstand) (pin A0)
5 (verbonden met Arduino pin D2 en D3)	Grove - Deursensor (magnetische switch) (pin D3)

Tabel 2 Grove connectoren aansluitingen sensoren

De Grove connectoren zijn een standaard connectoren die gebruikt worden om Grove modules op aan te sluiten. Deze Grove modules worden gemaakt door het bedrijf Seeed en zijn (online) verkrijgbaar. Het voordeel van deze Grove modules is dat deze altijd zullen passen op de Grove connectoren. Dit omdat de volgorde van pinnen van de connectoren altijd hetzelfde is. Er zijn voor deze demo vier van deze standaard Grove modules gebruikt. Op connector drie is er gekozen om een combinatie van een DHT11 en DHT22 sensor aan een Grove connector te maken zodat deze ook makkelijk op de Marvin aangesloten kan worden. Dit kan dan zonder dat er een hardware verandering moet worden gedaan aan de Marvin.

In Figuur 5 is weergegeven hoe deze DHT11 en DHT22 sensor zijn aangesloten op een Grove connector. DHT11 en DHT22 zijn temperatuur en luchtvochtigheid sensoren. Met een digitaal signaal worden deze sensoren uitgelezen. Eerst wordt er een start signaal gestuurd door de Arduino waarna de DHT-sensor de temperatuur en luchtvochtigheid terugstuurt.

De DHT-sensoren hebben, volgens de datasheet een pull-up weerstand nodig tussen de VCC en het signaal (SIG). Deze pull up weerstanden zijn, zoals te zien in Figuur 5, niet gebruikt. Dit omdat in de Arduino interne pull up weerstanden heeft. Deze zijn via de software te activeren, zo kunnen deze DHT-sensoren zonder hardware pull up weerstanden toch de gegevens verzenden. Deze software pull up wordt in de bibliotheek van de DHT geactiveerd, dit bestand zal aangeroepen worden in de software (in de initialisatie).



Figuur 5 DHT11 en DHT22 sensor aan een Grove connector

De andere sensoren die aangesloten zijn op de Grove connectoren zijn, de Grove – Lichtsterktesensor deze meet of er veel of weinig licht op de LDR-weerstand van de sensor valt. De Grove – Watersensor deze meet of er wel of geen water op de sensor ligt. De watersensor bestaat uit een aantal pcb-lijnen die verbonden zullen worden zodra er water op de sensor valt. De Grove – Temperatuursensor, deze geeft de waarde van de NTC-weerstand terug. Deze NTC weerstandwaarde verandert door de temperatuur. En als laatste de Grove – Deursensor, deze sensor is een magnetische schakelaar. Deze schakelaar is open als er geen magneet in de buurt is en is gesloten als er wel een magneet in de buurt is. Zo kan er, als er een magneet op een deur wordt geplaatst en de sensor op de deurpost, gemeten worden of een deur open of dicht is. De deursensor zou daarom ook gebruikt kunnen worden op een raam die open kan. De naam van de sensor zegt daarom niet dat het alleen voor een deur geschikt is.

4.1.2 Software

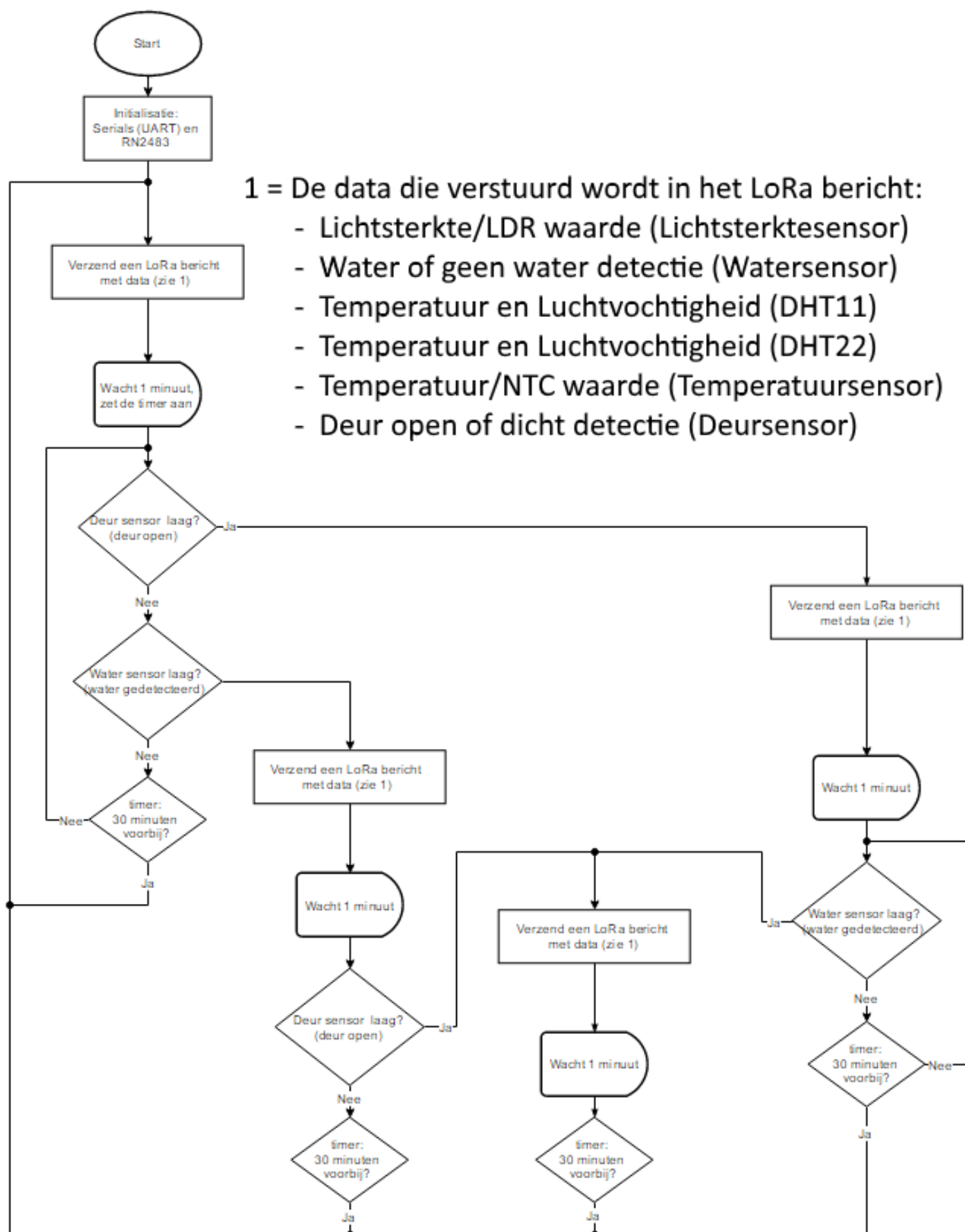
Voor de software is het programma Arduino IDE gebruikt. Dit is een programma om code voor de Arduino te kunnen schrijven. Met dit programma kan deze code ook op de Arduino geüpload worden. De gebruikte code voor deze demo zal bijgeleverd worden op een USB-stick bijbehorend bij deze scriptie.

Een flowchart van de code is te zien op de volgende bladzijde in Figuur 6. De Arduino zal bij het starten van de uitvoer van de code beginnen bij de initialisatie van de UART en de RN2483. De initialisatie van de RN2483 betekent dat de RN2483 een aantal commando's toegezonden krijgt om de RN2483 in te stellen voor verzending van LoRa berichten. Deze initialisatie is te zien in hoofdstuk 4.2.1 Initialisatie.

Na deze initialisatie zal de Arduino de RN2483 aansturen om een LoRa bericht te sturen. In Figuur 6 (achter 1=) is te zien wat dit bericht voor een inhoud heeft, naast deze waardes zal er ook informatie over de sensor zelf verstuurd worden. Deze informatie is het type sensor en zijn id. Dit alles is een bericht van 12 cijfers lang. Vier cijfers voor de sensor id, vier voor het type sensor en vier voor de data van de sensor. Ook zal het bericht aangeven hoeveel sensoren er verstuurd worden zodat de ontvanger (de server) weet hoeveel sensoren er verwerkt moeten worden.

Na het verzenden zal de Arduino een minuut gaan wachten en tegelijkertijd een timer van een half uur starten. Deze minuut is bedoeld om ervoor te zorgen dat er geen berichten te snel achter elkaar worden verstuurd. Na deze minuut zal de Arduino kijken of de Deursensor of de Watersensor laag zijn, dit houdt in dat de deur open is of dat er water gedetecteerd is. Als één van deze twee sensoren laag is zal er weer een LoRa bericht worden verstuurd waarna er weer een minuut gewacht wordt. Ook zal er een variabele worden gezet die ervoor zorgt dat de Arduino voor het aflopen van de timer van een half uur niet nog een keer zal kijken of de sensor die laag was opnieuw laag is. Na deze timer van een half uur zullen deze variabele gereset worden en zal er weer een LoRa bericht verstuurd worden. Als gevolg hiervan wordt er minimaal 1 en maximaal 3 LoRa berichten per half uur verstuurd. Als er 3 berichten per half uur zijn verstuurd is er een situatie ontstaan waarin de Deursensor en de Water sensor laag zijn geweest.

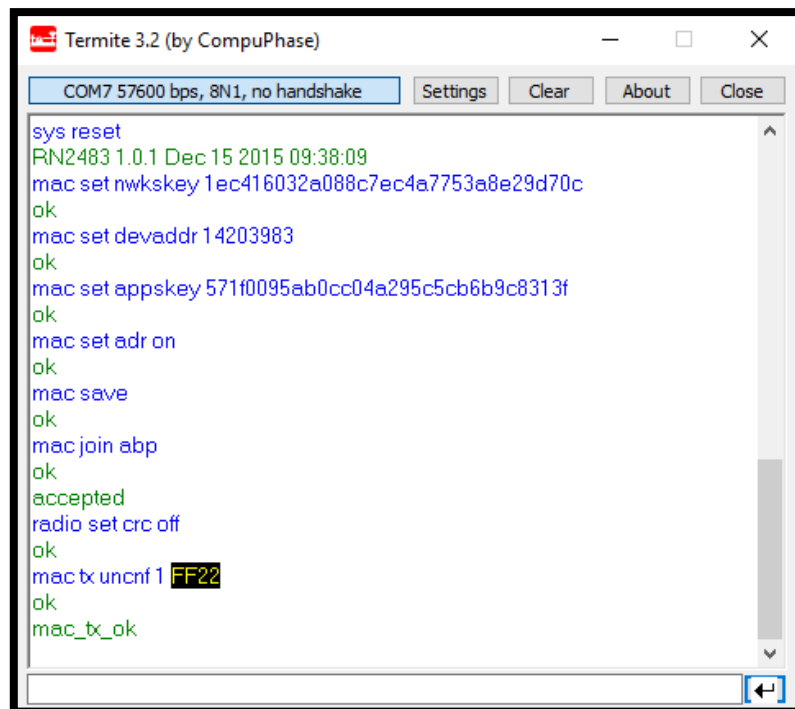
Omdat er na het verzenden een minuut gewacht wordt door Arduino kan het zo zijn dat er een sensor verandering gemist wordt. Bijvoorbeeld de Watersensor wordt binnen deze minuut eerst laag en daarna weer hoog. Dan zal de Arduino hier niet op reageren door een LoRa bericht te sturen. Voor deze demo is dat niet erg, maar als de sensoren als alarmering moeten werken zullen de sensoren altijd gecheckt moeten worden!



Figuur 6 Software LoRa demo Marvin (Arduino)

4.2 Verzenden van een LoRa bericht doormiddel van een RN2483

Voor het verzenden van een LoRa bericht is hier gebruikt gemaakt van de RN2483 module. Deze module is gemaakt door Microchip en kan gebruikt worden door er (via UART) berichten heen te sturen. Deze berichten worden verstuurd op een snelheid van 57600 bits per second. Dit is de standaard baud rate van de RN2483. De RN2483 zal het bericht versturen op 868 MHz, dit is de frequentie van LoRa in Nederland. Om de berichten te kunnen versturen heeft de RN2483 bepaalde sleutels nodig. Met deze sleutels kan de RN2483 zich aanmelden bij het LoRa netwerk. De sleutels worden ingesteld in de initialisatie. Daarna zal er data verzonden kunnen worden. Alle commando's die hiervoor gebruikt worden zijn te zien in Figuur 7. In het blauw zijn de commando's te zien die naar de RN2483 worden gestuurd en in het groen het antwoord van de RN2483 op deze commando's.



```
Termit 3.2 (by CompuPhase)
COM7 57600 bps, 8N1, no handshake
sys reset
RN2483 1.0.1 Dec 15 2015 09:38:09
mac set nwkskey 1ec416032a088c7ec4a7753a8e29d70c
ok
mac set devaddr 14203983
ok
mac set appskey 571f0095ab0cc04a295c5cb6b9c8313f
ok
mac set adr on
ok
mac save
ok
mac join abp
ok
accepted
radio set crc off
ok
mac tx uncnf 1 FF22
ok
mac_tx_ok
```

Figuur 7 Verzenden van een LoRa bericht met een RN2483

4.2.1 Initialisatie

sys reset

Met dit commando zal de RN2483 zich resetten en opnieuw opstarten.

Het antwoord van de RN2483 hierop is: de naam van de module, de firmware versie en de datum wanneer de firmware uitgebracht is.

mac set nwkskey [nwkskey]

Met dit commando zal de nwkskey (network session key) ingesteld worden. De nwkskey is een 16 bytes hexadecimale getal (een byte is 2 karakters lang en daarom is de nwkskey $2 * 16 = 32$ karakters lang). Het antwoord van de RN2483 hierop is 'ok' of 'invalid_param'. Dit geeft aan of de invoer juist is of niet, bij 'ok' is de invoer juist.

mac set devaddr [devaddr]

Met dit commando wordt het devaddr (Unique network device adres) ingesteld, dit is een 4 bytes adres. Het antwoord van de RN2483 hierop is 'ok' of 'invalid_param'. Dit geeft aan of de invoer juist is of niet, bij 'ok' is de invoer juist.

mac set appskey [appskey]

Met dit commando wordt de appskey (applicatie session key) ingesteld dit is een 16 bytes hexadecimale getal. Het antwoord van de RN2483 hierop is 'ok' of 'invalid_param'. Dit geeft aan of de invoer juist is of niet, bij 'ok' is de invoer juist.

Wat er precies ingevuld moet worden voor de nwkskey, devaddr en appskey wordt later in hoofdstuk 5 LoRaWAN Cloud server besproken. Hier is te lezen wat de sleutels zijn die ingevoerd moeten worden.

mac set adr on

Zet het ADR-mechanisme aan. Dit mechanisme zorgt voor de optimalisatie van de datasnelheid en het zendvermogen gebaseerd op de informatie verkregen van het netwerk.

mac save

Dit commando slaat alle (voorgaande) instellingen op. Dit gebeurt op de EEPROM van de RN2483 en daarom zullen de instellingen na een herstart nog steeds ingesteld zijn.

4.2.2 Verzending

mac join abp

Met dit commando zal de RN2483 proberen zich aan te melden bij het netwerk, de abp (activation by personalization) geeft aan dat het om activatie doormiddel van personalisatie gaat.

Het antwoord zal uit twee delen bestaan, het eerste deel is weer 'ok' of 'invalid_param' dit geeft aan of het commando juist of onjuist is. Het tweede deel geeft aan of het aanmelden gelukt is. Als het antwoord 'accepted' is de aanmeld procedure goed gegaan.

radio set crc off

Dit commando zet de crc modus uit. Als de crc modus aan staat zal de RN2483 een extra stukje extra data meesturen waarmee errors opgespoord kunnen worden.

Het antwoord hierop is net als bij de vorige commando's weer 'ok' of 'invalid_param'.

mac tx uncnf [poort nummer] [data]

Dit commando zal de data verzenden, de uncnf staat voor unconfirmed. Dit betekent dat het bericht geen acknowledgment (ack) van het netwerk verwacht. Een ack is een bevestiging van het netwerk dat het bericht ontvangen is. Als het commando cnf bevat in plaats van uncnf zal de RN2483 het bericht een bepaald aantal keer hersturen tot het wel een ack van het netwerk terugkrijgt. Dit aantal kan ingesteld worden door het commando mac set retex [aantal].

Het poort nummer is een getal tussen de 1 en de 223 dit is de poort waarop het bericht verstuurd zal worden. Na het poortnummer komt de data die verstuurd moet worden. Na het poortnummer komt de data die verstuurd moet worden. Dit is een hexadecimaal getal van een aantal bytes, hierin wordt alle data die doorgegeven moet worden ingezet. In Figuur 7 is dit FF22.

Het antwoord van de RN2483 is eerst een bevestiging of de bericht invoer goed is gegaan, als er iets fout gaat kunnen er, niet zoals bij de andere commando's, ook andere foutmeldingen zijn dan 'invalid_param'.

Het tweede antwoord is een bevestiging dat het bericht ontvangen is, dit kunnen de volgende vier berichten zijn:

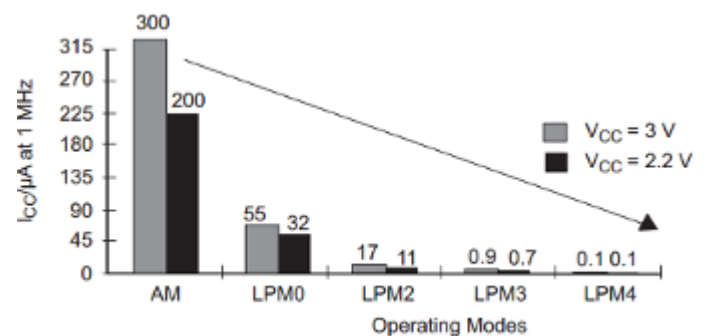
- mac_tx_ok, dit betekent dat het bericht goed ontvangen is (er is een ack ontvangen is van het netwerk)
- mac_rx [poort nummer] [data], dit betekent dat het bericht goed ontvangen is en dat er een downlink bericht (bericht van het netwerk naar de RN2483) verstuurd is met het een bepaald poort nummer en data.
- mac_err, dit betekent dat er is geen ack ontvangen van het netwerk, het verzenden kan dan mislukt zijn.

- invalid_data_len, dit betekent dat de lengte van de data te groot is.

LET OP! Er zit een limiet aan het aantal berichten dat via het LoRa netwerk verstuurd kan worden, de zogenoemde uplink berichten (van de RN2483 naar het LoRa netwerk toe), zie 11.11 LoRa kosten. Ook mag er maar 1% van de tijd (in een uur) verzonden worden. Dit betekent dat er niet constant berichten verzonden kunnen worden (als bijvoorbeeld de ontvangende antenne verder weg staat is de verzend tijd groter). Zie hoofdstuk 8.2 wat de beperkingen hierbij zijn.

4.3 LoRa demo beweging sensor en temperatuur weertand (NTC)

Omdat een deel van het woord LoRa staat voor low power (laagvermogen) is er gekozen om een demo te maken die weinig verbruikt en daardoor lang mee kan op 1 batterij. In Figuur 8 zijn de Low Power Modes (laagvermogen modi) van de MSP430 te zien. Hierin is te zien dat de laagste modus LMP4 maar een verbruik heeft van 1µA(!) en in de LMP3 een verbruik van 0.9 µA bij 3 volt. Dit is zeer weinig en daardoor geschikt om voor deze demo te gebruiken.



Figuur 8 Low Power Modes van de MSP430 [4]

De microcontroller zal in LMP4 modus gaan als het alleen wakker moet worden van een interrupt (onderbreking) op een bepaalde pin. In LPM4 zullen alle belangrijke functies uitgeschakeld worden zie Tabel 3. In LPM3 mode zal de microcontroller alleen uit slaap modus gewekt worden door een interrupt of een timer overflow (overloop). Dit betekent dat er een timer kan worden ingesteld die aan de hand van een kristal na een bepaalde tijd overloopt en daardoor de microcontroller wekt. Hiervoor is de ACLK klok nodig, daarom dat het LMP3 mode is (zie Tabel 3). De timer wordt gebruikt om de microcontroller een tijdje niks te laten doen. De MSP430 heeft zelfs in de Active Modus (AM) maar een verbruik van 300 µA bij 3 volt. De microcontroller voor deze demo is de MSP430G2553 uit de MSP430 familie. In de onderstaande Tabel 3:

	CPU	MCLK	DCO	SMCLK	ACLK	DC generator
AM (Actieve modus)	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld
LPM0 (Low Power Mode 0)	Uitgeschakeld	Uitgeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld
LPM1 (Low Power Mode 1)	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld *	Ingeschakeld	Ingeschakeld	Uitgeschakeld *
LPM2 (Low Power Mode 2)	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Ingeschakeld	Ingeschakeld
LPM3 (Low Power Mode 3)	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Ingeschakeld	Uitgeschakeld
LPM4 (Low Power Mode 4)	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld	Uitgeschakeld

Tabel 3 Modi met functies ervan

*uitgeschakeld als de DCO niet gebruikt wordt voor de SMCLK.

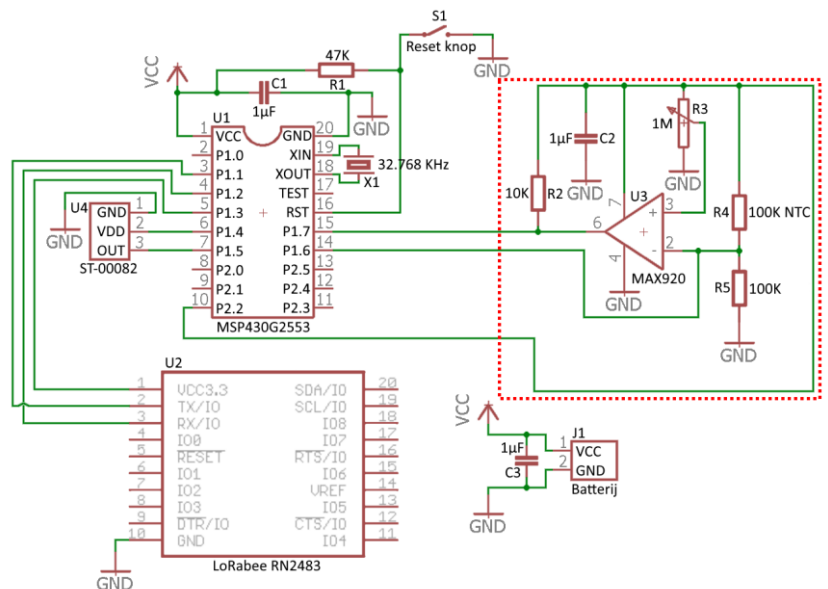
De afkorting die in Tabel 3 staan hebben de volgende betekenissen:

- CPU (Central Processing Unit), dit is de centrale processor in de microcontroller
- MCLK (Main System Clock), dit is de klok voor de CPU en het systeem
- DCO (Digitally Controlled Oscillator), interne oscillator om de klokken SMCLK en MCLK op te wekken.
- DC generator (Direct Current generator), voeding voor de interne oscillator (DCO)
- SMCLK (Sub-System Clock), dit is de klok voor de verschillende modules die niet tot de CPU behoren. De SMCLK kan opgewekt worden door een extern kristal of de DCO.
- ACLK (Auxilairt Clock), dit is de klok die wordt opgewekt door een extern kristal (externe oscillator)

4.3.1 Hardware

De hardware bestaat uit 4 delen, de MSP430G2553 microcontroller, de LoRabee RN2483, de ST-00082 PIR Motion (beweging sensor) sensor en temperatuur sensor (NTC-weerstand) met comparator.

De MSP430G2553 is een microcontroller die de gehele schakeling aanstuurt. Deze microcontroller heeft een ontkoppelcondensator C1 tussen de plus voeding (VCC) en de ground (GND) om ruis weg te filteren. Verder is er een pull up weerstand (R1) aangesloten aan de VCC en de reset (RST) pin. Deze zorgt ervoor dat de microcontroller niet gereset wordt. De reset pin zit ook verbonden aan de reset knop (S1). Als er op deze knop gedrukt wordt zal de reset reset met de ground (GND) verbonden en wordt de microcontroller gereset.



Figuur 9 Hardware van LoRa beweging sensor en temperatuur weerstand (NTC)

Tussen de pinnen XIN en XOUT zit een kristal van 32.768 kHz. Dit kristal heeft een load van 12.5uF en heeft daarom geen extra condensatoren nodig. Het kristal zal worden gebruikt om de interne timer op een zo laag mogelijk vermogen te laten werken. Zo kan de microcontroller bijna helemaal uitgeschakeld worden terwijl er wel een timer loop die de microcontroller kan wekken als dit nodig is. De microcontroller zit verbonden aan de LoRabee RN2483. Hierop zit de RN2483 module met een antenne. De RN2483 module kan berichten naar het LoRa netwerk sturen. De microcontroller schakelt dan eerst de module in pin P1.3 hoog te maken, deze pin is verbonden met de VCC van de LoraBee. Waarna er via de UART TX commando's worden verzonden naar de RN2483 module (pin 2 van LoRabee). De RN2483 module verwerkt deze commando's en antwoord terug via de UART RX (pin 3 van de LoRabee). Zodra het bericht verzonden is zal de microcontroller de LoRabee weer uitschakelen door pin P1.3 laag te maken.

Aan pin P1.4 en P1.5 van de microcontroller zit de ST-00082 verbonden. Dit is een PIR Motion (beweging) sensor. Deze sensor zal als pin P1.4, de VDD van de ST-00082, hoog is een hoog signaal aan pin P1.5 (de OUT van de ST-00082) aanbieden zodra er beweging wordt gedetecteerd. De pin OUT van de ST-00082 zal 3.5 seconden hoog blijven na de laatste detectie van beweging. Let wel op! De ST-00082 zal bij het activeren eerst 3.5 seconden een hoog signaal afgeven op de OUT, zelfs bij geen beweging. De microcontroller zal daarom eerst meer dan 3.5 seconden moeten wachten na het activeren van de ST-00082, voor de ST-00082 beweging kan detecteren.

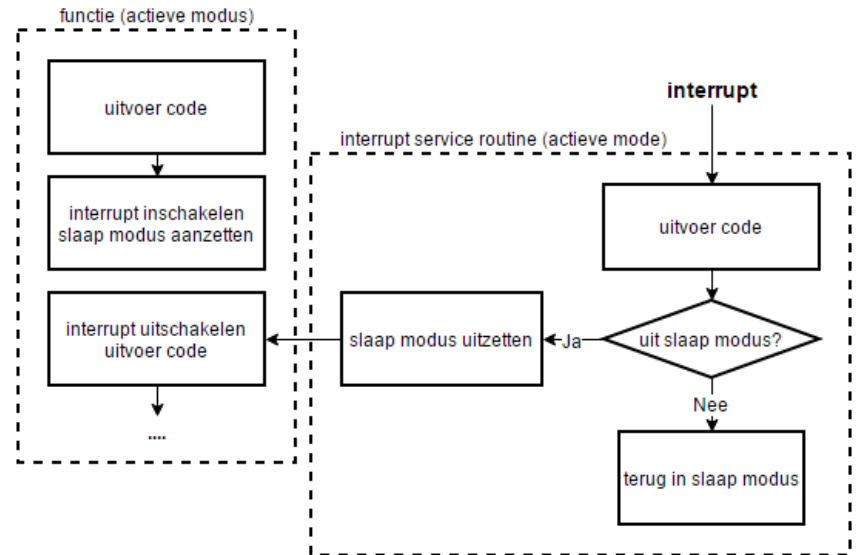
In het rode omlijnde vierkant is de temperatuur sensor (R4, 100K NTC) te zien. Deze vormt samen met de weerstand R5 een spanningsdeling die via pin P1.6 uitgelezen kan worden. Deze spanningsdeling gaat ook de comparator MAX920 in. Deze comparator heeft als referentie spanning een spanningsdeling die de potmeter R3 maakt. Zo zal de referentie spanning op pin 3 (uit R3) vergeleken worden met de spanning op pin 4 (tussen R5 en de temperatuur sensor R4). Zodra de spanning op pin 4 hoger is dan op pin 3 zal de comparator uitgang (pin 6) hoog worden. Deze zal dan even hoog als de voedingspanning van de comparator zijn. Dit omdat er een pull up weerstand tussen de uitgang en de voedingspanning zit. Het idee van deze comparator is om een interrupt op pin P1.7 van de microcontroller te generen zodra de temperatuur (de weerstandswaarde van de NTC) boven een door R3 ingestelde waarden komt. Deze interrupt kan ook veroorzaakt worden als de temperatuur onder de door R3 instelde waarde komt, hiervoor moeten de weerstanden R4 (de temperatuur NTC) en R5 omgedraaid worden. Let op! Als deze weerstanden omgedraaid worden verandert wel de formule om de temperatuur mee te berekenen. De formule voor de NTC komt later in dit hoofdstuk aan bod.

Er is gekozen voor een 10 kilo ohm pull up weerstand (R2) omdat de interne pull down weerstand van de microcontroller 47 kilo ohm is. Als de weerstand R2 te dicht bij de 47 kilo ohm komt zal de uitgang van de comparator niet volledig hoog worden. Omdat de R2 dan een spanningsdelers vormt met de interne pull down. Als dit gebeurt zal de spanning niet hoog genoeg zijn om als hoog signaal te dienen. En zal de microcontroller nooit een interrupt krijgen. De condensator C2 heeft dezelfde werking als de condensator C1 alleen filtert deze de ruis voor de comparator weg.

Verder zal er een batterij van 3.3 volt aan de batterij connector aangesloten moeten worden om de het geheel te voeden.

4.3.2 Software

De code van de software is geschreven in Code Composer Studio 6.2.0. Dit is een programma van Texas Instruments. In combinatie met de MSP430 launchpad kan er in de MSP430G2553 microcontroller een code geladen worden die de microcontroller na opstart zal uitvoeren. Deze code kan ook via de launchpad gedebugd worden. Hiermee kunnen makkelijk fouten uit de code worden gehaald. *Let wel op de launchpad kan UART berichten maar uitlezen tot en met een baud rate van 9600, de communicatie tussen de RN2483 en de microcontroller gaat met een baud rate van 57600 bits per seconde en kan met de launchpad niet gelezen worden!* De code van de software is bijgeleverd op een USB-stick bij de scriptie.



Figuur 10 Interrupt Service Routine en slaap modus (software)

Een flowchart van de totale code is te zien in Figuur 11. Wat de software kortgezegd doet is initialiseren, een LoRa bericht versturen, een minuut slapen. Daarna zal de microcontroller slapen tot er een interrupt wordt ontvangen door de beweging sensor of de comparator aangesloten op de temperatuur sensor. Als een van deze interrupts binnen komen zal de microcontroller wakker worden en een nieuw LoRa bericht verzenden. Na het verzenden van dit LoRa bericht zal de microcontroller weer een minuut gaan slapen. Ook zorgt de microcontroller doormiddel van een timer dat de microcontroller 15 minuten lang niet meer zal reageren op dezelfde interrupt. Dit komt erop neer dat tussen dezelfde interrupt (van de beweging sensor of de comparator) altijd minimaal 15 minuten zit. Ook zit er tussen twee LoRa berichten altijd een minuut.

Dit slapen doet de microcontroller om stroom verbruik te verminderen. Dit werkt op de volgende manier, in Figuur 10 begint de code in een functie. In deze functie voert de microcontroller een code uit. Deze code kan bijvoorbeeld het versturen zijn van een LoRa bericht. Als de microcontroller hiermee klaar is zal deze de interrupt en slaap modus aanzetten. Nu zal de microcontroller worden in één van de Low Power Modi gaan (zie Tabel 3). De microcontroller zal alleen wakker worden door een interrupt. Deze interrupt kan een Watch Dog Timer interrupt zijn. Dit is een interrupt die de timer op zal hogen. Dit gebeurt door de Watch Dog Timer om de seconde een interrupt te laten genereren. Deze Watch Dog Timer werkt doormiddel van een kristal van 32768 hertz, te zien in Figuur 9. Na 32768 klok flanken (gegenereerd door de kristal) is er een seconde voorbij en geeft de Watch Dog Timer een interrupt. Zodra deze interrupt veroorzaakt wordt zal de code in de 'Interrupt Service Routine' (naam van de interrupt functie) uitgevoerd. In deze functie (zie Figuur 10) is de microcontroller weer in actieve modus en wordt er een stuk code uitgevoerd. In het geval van de Watch Dog Timer interrupt is dit een teller (timer) ophogen. Ook zal er beslist worden of de microcontroller weer wakker gemaakt moet worden of niet. In het geval van de Watch Dog timer interrupt is dit bijvoorbeeld als de timer tot 15 minuten heeft geteld. Nu zal de 'Interrupt Service Routine' de microcontroller wekken. Na dit wekken zal de microcontroller de code uitvoeren waar

deze gebleven was. In Figuur 10 betekent dit dat de microcontroller de interrupt uitschakelt en code gaat uitvoeren. Als de 'Interrupt Service Routine' de microcontroller niet wakker maakt zal deze weer gaan slapen tot de volgende interrupt.

Bij het verzenden van een LoRa bericht zal de microcontroller de beide sensoren aanzetten, dit wordt gedaan door pin P1.4 (beweging sensor) en pin P2.2 (temperatuur NTC) hoog te maken (zie Figuur 9). Hierna zullen (na 4 seconde) de beweging sensor (pin P1.5) en de ADC-waarde van temperatuur NTC (pin 1.6) uitgelezen worden. Deze data zal via een LoRa bericht verstuurd worden. Dit zal worden gedaan door de RN2483 aan te zetten (pin 1.3 hoog maken) en daarna de juiste commando's naar de RN2483 te sturen zie 4.2 Verzenden van een LoRa bericht doormiddel van een RN2483.

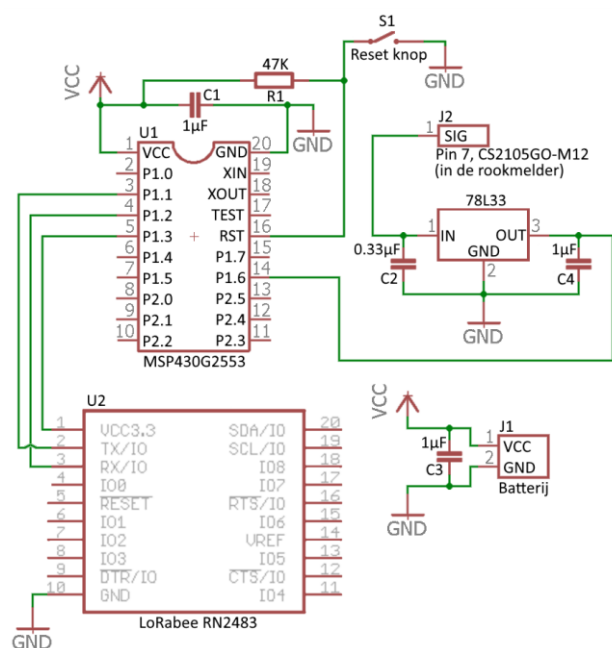
32

4.4 LoRa demo Rookmelder

Deze LoRa demo koppelt een rookmelder aan een microcontroller die ervoor zorgt dat er een LoRa bericht verstuurd wordt door de RN2483. Deze demo lijkt heel erg op de LoRa demo met de bewegingssensor en temperatuur weerstand. Alleen verschil tussen deze twee demo's is dat deze direct zal reageren op een alarm van de rookmelder. Bij de andere twee demo's zal de er na een bericht gewacht of geslapen worden als er net een LoRa bericht verstuurd is. Deze demo zal direct een bericht sturen bij een alarm. Dit omdat het om alarmering gaan en daarom zal er direct actie ondernomen moeten worden.

4.4.1 Hardware

Een schema van de hardware is te zien in Figuur 12. Het schema lijkt erg op het schema uit Figuur 9, de hardware uit hoofdstuk 4.3 LoRa demo beweging sensor en temperatuur weertand (NTC). Het grote verschil tussen de beide schema's is dat in dit schema geen beweging sensor en temperatuur weerstand heeft maar in plaats daarvan een rookmelder uitgang. Dit gebeurt door in de rookmelder pin 7 van het component CS2105GO-M12 te verbinden aan de ingang (IN) van de 78L33. Deze pin 7 wordt hoog (ongeveer 11 volt) zodra de rookmelder rook detecteert. Omdat de microcontroller maar door 3.3 volt wordt gevoed door de batterij zal deze geen 11 volt als ingang kunnen verwerken. Daarom wordt het 78L33 component gebruikt. Deze zorgt ervoor dat de ingang (IN) als er 11 volt op staat wordt teruggebracht tot 3.3 volt aan de uitgang (OUT). Deze uitgang is weer verbonden met pin P1.6 van de microcontroller. Zo kan de microcontroller het signaal uit de rookmelder uitlezen zonder dat het een hoge spanning op pin P1.6 staat. De condensatoren C2 en C4 hebben als functie om de binnenkomende en uitgaande signaal van de 78L33 te filteren van ruis. De rest van de schakeling heeft dezelfde werking als besproken in het hardware in hoofdstuk 4.3 LoRa demo beweging sensor en temperatuur weertand (NTC).



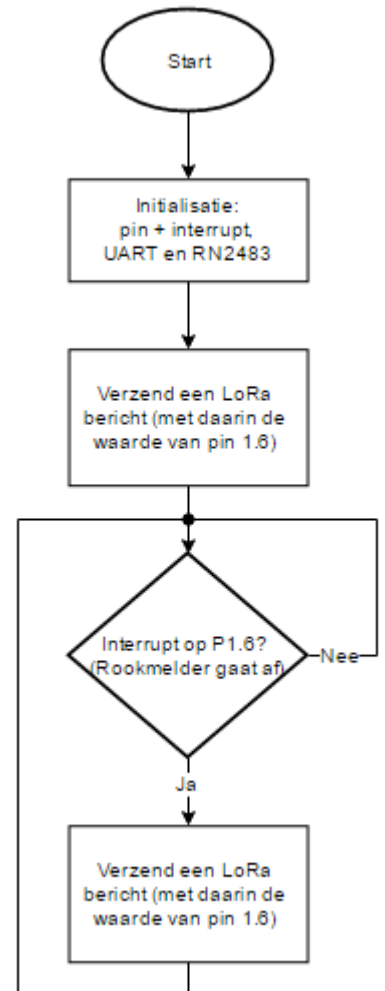
Figuur 12 Hardware LoRa demo Rookmelder

4.4.2 Software

De software wordt net als bij de software uit in hoofdstuk 4.3 LoRa demo beweging sensor en temperatuur weertand (NTC) geprogrammeerd met de lauchpad. De code van de software is bijgeleverd op een USB-stick bij de scriptie.

Een flowchart van de code is te zien in Figuur 13. De code begint met de initialisatie van de pin waar de rookmelder (indirect) op aangesloten zit, pin P1.6. Deze pin wordt via de software via een interne pull-down aan de ground gezet. Dit is gedaan zodat de pin normaal laag is en bij een alarm van de rookmelder de pin hoog wordt. Ook wordt de interrupt voor deze pin P1.6 gezet, zodat de pinverandering van laag naar hoog de microcontroller wakker maakt nadat het in slaap modus is gegaan. Na de initialisatie van de pin worden ook de UART en de RN2483 geïnitieerd. Als dat gebeurd is zal de microcontroller commando's naar de RN2483 sturen waarmee er een LoRa bericht verstuurd wordt naar het netwerk. Dit bericht kan gezien worden als het opstart bericht.

Daarna zal de microcontroller gaan slapen in LPM4 en alleen wakker worden bij een interrupt op pin P1.6. Een interrupt op P1.6 betekent dat de Rookmelder af gaat. De microcontroller zal nu onmiddellijk de RN2483 aansturen om een LoRa bericht naar het netwerk sturen. Als het bericht verstuurd is zal de microcontroller weer gaan slapen en wachten tot de volgende interrupt. Deze interrupt wordt alleen gegenereerd als de waarde op pin P1.6 van laag naar hoog gaat. Dit betekent dat er geen interrupt volgt zodra de rookmelder zal stoppen met alarmeren.



Figuur 13 Flowchart software LoRa demo Rookmelder

4.5 Sensoren, sensor data en sensor kalibratie

4.5.1 Gebruikte sensoren

In Tabel 4 zijn alle gebruikte sensoren te zien met welke data er uit de sensoren komt, of er kalibratie nodig is, wat den sensor id en sensor type is en met welke formule de data omgerekend kan worden naar de werkelijke sensor waarden. De data zal altijd zo worden weergegeven dat de minimale data altijd 0000 is. Ook zal de lengte van de data altijd 4 getallen (2 bytes) zijn. Dit is zo gedaan zodat er meerdere sensors met data tegelijkertijd verstuurd kunnen worden. Zo zal er uiteindelijk 6 bytes per sensor worden gebruikt, 2 bytes voor de sensor_id, 2 bytes voor het sensor_type en 2 bytes voor de sensor data. Door deze vaste lengte kunnen de sensoren uit elkaar gehaald worden.

Soort Sensor	Gebruikte sensor	Sensor signaal (data)	Sensor kalibratie	Sensor id, sensor type	Formule
Beweging sensor	ST-00082 PIR Motion sensor	(Na 3.5 seconden na aanzetten) Hoog bij beweging (data = 0001) Laag bij geen beweging (data = 0000)	In de fabriek al gekalibreerd.	Sensor_id = 0001 Sensor_type = 0002	Sensor waarden [geen eenheid] = ontvangen data
Rookmelder	Elro rookmelder RM144F (Detecteert geen CO!)	(Op pin 7 van het component CS2105GO-M12) Hoog bij rookdetectie (data = 0001) Laag bij geen rookdetectie (data = 0000)	In de fabriek al gekalibreerd.	Sensor_id = 0001 Sensor_type = 0021	Sensor waarden [geen eenheid] = ontvangen data
Luchtvochtigheidssensor	DHT11	Luchtvochtigheid tussen de 20 en 80 % (data van 0000 tot en met 0060)	In de fabriek al gekalibreerd.	Sensor_id = 0001 Sensor_type = 0016	Sensor waarden [in %] = ontvangen data + 20
	DHT22	Luchtvochtigheid tussen de 0 en 100.0 % (data van 0000 tot en met 1000)	In de fabriek al gekalibreerd.	Sensor_id = 0001 Sensor_type = 0018	Sensor waarden [in %] = ontvangen data / 10
Temperatuur-sensor	ND03R00104J 100kΩ NTC	Temperatuur tussen de -40 en 125 graden (data, ADC-waarden, van 0000 tot en met 1023)	Nodig, zie hoofdstuk 4.5.4.1	Sensor_id = 0001 Sensor_type = 0001	Zie hoofdstuk 4.5.2 Formule Temperatuursensor ((ND03R00104J, 100kΩ NTC).
	Grove – Temperature sensor V1.2	Temperatuur tussen de -40 en 125 graden (data, ADC-waarden,	Nodig, zie hoofdstuk 4.5.4.2	Sensor_id = 0001	Zie hoofdstuk 4.5.3 Formule Grove – Temperatuursensor

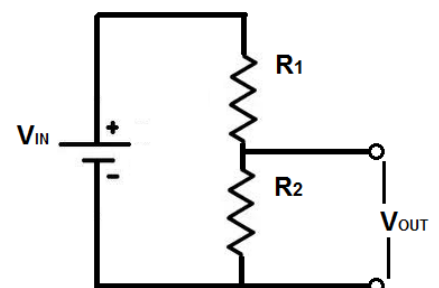
		van 0000 tot en met 1023)		Sensor_type = 0011	
	DHT11	Temperatuur tussen de 0 en 50 graden (data van 0000 tot en met 0050)	In de fabriek al gekalibreerd.	Sensor_id = 0001 Sensor_type = 0015	Sensor waarden [in °C] = ontvangen data
	DHT22	Temperatuur tussen de -40.0 en 125.0 graden (data van 0000 tot 1650)	In de fabriek al gekalibreerd.	Sensor_id = 0001 Sensor_type = 0017	Sensor waarden [in °C] = (ontvangen data / 10) - 40
Deur- en Raamsensor	Grove – Magnetic Switch	Hoog bij deur gesloten (sensor tegen de magneet) (data = 0001) Laag bij deur open (data = 0000)	Kan niet gekalibreerd worden.	Sensor_id = 0001 Sensor_type = 0014	Sensor waarden [geen eenheid] = ontvangen data
Water-detector	Grove – Water sensor	Hoog bij geen water detectie (data = 0001) Laag bij water detectie (data = 0000)	Kan niet gekalibreerd worden.	Sensor_id = 0001 Sensor_type = 0013	Sensor waarden [geen eenheid] = ontvangen data
Licht-intensiteit sensor	Grove – Light sensor V1.1	Hoog bij licht (data = 0001) Laag geen licht (data = 0000)	Nodig, zie hoofdstuk 4.5.4.3	Sensor_id = 0001 Sensor_type = 0012	Sensor waarden [geen eenheid] = ontvangen data

Tabel 4 Gebruikte sensoren

4.5.2 Formule Temperatuursensor ((ND03R00104J, 100kΩ NTC)

De temperatuursensor is een ND03R00104J 100kΩ NTC-weerstand. Deze weerstand vormt samen met een 100kΩ weerstand een spanningsdeler zoals te zien in Figuur 14. In dit figuur is de ND03R00104J 100kΩ NTC-weerstand uitgebeeld als R1 en de 100kΩ weerstand is R2. De spanning tussen deze weerstanden is uitgebeeld als V_{out}. Omdat met de waarde van de NTC-weerstand (R1) de temperatuur uitgerekend zal worden. Wordt de formule aan de hand van R1 geformuleerd:

$$R_1 = R_2 * \left(\frac{V_{in}}{V_{out}} - 1 \right) \quad [\text{formule 1}]$$



Figuur 14 Spanningsdeler

In deze formule is de waarde R2 de 100kΩ (100000Ω) weerstand, de V_{in} is de maximale ADC-waarde van 1023 en V_{out} is de ADC-waarde van de V_{out} spanning. Deze ADC-waarde zal door de microcontroller uitgelezen worden.

Voor het uitreken van de temperatuur aan de hand van de waarde van de NTC-weerstand wordt de volgende formule gebruikt, dit is een standaard formule voor een NTC-weerstand (de β -vergelijking):

$$T = \left(\frac{\log\left(\frac{R_1}{R_0}\right)}{\beta} + \frac{1}{T_0} \right)^{-1} - K$$

[formule 2]

In deze formule is T de temperatuur in graden, R1 de waarde van de NTC-weerstand uitgerekend in *formule 1*, R0 is de waarde van de NTC op 25 graden in dit geval is dat 100k Ω (100000), β is de bèta van de NTC-weerstand in dit geval 4400 dit staat in de datasheet zie hoofdstuk 11.2 [D5], T0 is de waarde van 25 graden omgerekend naar Kelvin dat is 298.15 en K de omreken factor om van Kelvin naar graden te gaan heeft daarom de waarde van 273.15. Als *formule 1* en *formule 2* gecombineerd worden, en de waardes ingevuld, vormt dat de volgende formule:

$$T = \left(\frac{\log\left(\frac{100000 * \left(\frac{1023}{V_{out}} - 1\right)}{100000}\right)}{4400} + \frac{1}{298.15} \right)^{-1} - 273.15$$

[formule 3]

Deze formule is korter te schrijven als:

$$T = \left(\frac{\log\left(\frac{1023}{V_{out}} - 1\right)}{4400} + \frac{1}{298.15} \right)^{-1} - 273.15$$

[formule 4]

Om terug te verwijzen naar Tabel 4 is T de sensor waarde met als eenheid $^{\circ}\text{C}$ en Vout de ADC-waarde die wordt ontvangen (de ontvangen data).

4.5.3 Formule Grove – Temperatuursensor V1.2

De formule voor de deze sensor staat op de wiki pagina van Seeed (zie hoofdstuk 11.2. [D6]). En is de volgende formule:

$$T = \left(\frac{\log\left(\frac{1023}{V_{OUT}} - 1\right)}{4275} + \frac{1}{298.15} \right)^{-1} - 273.15$$

[formule 5]

Hierin is T de temperatuur in °C en Vout de ADC-waarde gemeten door de Arduino. Om terug te referent naar Tabel 4. De ontvangen waarde is de ADC-waarde en de daarbij horende sensor waarden is de waarde van T.

4.5.4 Kalibratie sensoren

Bij het kalibreren van de sensoren wordt er nagegaan of de formules die gevormd zijn voor de sensoren wel kloppen danwel de eigenschappen van de sensor aangepast zodat deze past bij de gestelde eisen. Dit wordt gedaan aan de hand van een referentie sensor. Hierbij wordt ervan uit gegaan dat deze referentie sensor waarde klopt. De waardes die uit de formules komen worden vergeleken met de waarde van de referentie sensor. Komen deze waardes (enigszins) overeen dan is de formule juist. Wijkt de formule af dan zal er een nieuwe formule gemaakt moeten worden voor de sensor. Zodra de formule klopt geeft de sensor de werkelijke waarde aan en is de sensor goed gekalibreerd. Voor de kalibratie van de temperatuur sensoren wordt verwacht dat de waardes minder dan 1 graden zullen afwijken van de referentie sensor.

4.5.4.1 Temperatuursensor (ND03R00104J, 100kΩ NTC)

Voor het kalibreren van de ND03R00104J temperatuur sensor is als referentie de DHT22 sensor gebruikt. Deze sensor is al gekalibreerd in de fabriek en zal daardoor een goede indicatie van de werkelijke temperatuur geven. Beide sensoren worden aangesloten de Arduino die aangesloten zit op het Marvin bordje. De Arduino zal om de 5 seconde de waardes van de sensoren ophalen en in het geval van de NTC de waarde uitrekenen met de formule. De resultaten van de metingen zijn te zien in de onderstaande Tabel 5. De code van de Arduino waarmee de kalibratie gedaan is staat op een USB-stick bij de scriptie.

De metingen geven de laatste 20 metingen van de DHT22 en NTC-weerstand aan. Dit omdat daar de DHT22 sensor een stabiele waarde had. Onderaan de tabel staan de gemiddeldes van de verschillende sensoren.

Uit de metingen blijkt dat er verschil zit in de metingen van de NTC en de DHT22. Maar omdat dit verschil minder is dan 0.5 graden Celsius is valt dit nog binnen de vooraf bepaalde afwijking van 1 graden. Dit verschil is te verklaren door de standaard afwijking van de ND03R00104J van -5% tot 5%. Zo kan weerstand waarde van de ND03R00104J dus afwijken van de werkelijke waarde.

DHT22	NTC	DHT22	NTC	DHT22	NTC	DHT22	NTC	DHT22	NTC	DHT22	NTC
20	19,81	8,7	8,9	16,7	16,86	10,7	11,46	17,5	17,63	21,9	21,44
20	19,81	8,6	8,99	16,7	16,93	10,7	11,46	17,5	17,63	21,9	21,44
20	19,81	8,6	8,99	16,7	17,01	10,7	11,38	17,5	17,63	21,9	21,44
20	19,81	8,6	8,9	16,7	17,01	10,7	11,38	17,5	17,63	21,9	21,44
20	19,81	8,6	8,82	16,7	16,93	10,8	11,14	17,5	17,63	21,9	21,44
20	19,81	8,6	8,99	16,7	16,86	10,7	11,14	17,6	17,63	21,9	21,44
20	19,81	8,6	8,99	16,7	16,86	10,7	11,14	17,6	17,63	21,9	21,44
20	19,89	8,6	8,99	16,7	16,93	10,7	11,14	17,6	17,63	21,9	21,44
20	19,89	8,6	8,9	16,7	16,93	10,8	11,06	17,6	17,56	21,9	21,44
20	19,89	8,6	8,82	16,7	16,93	10,7	11,06	17,6	17,63	21,9	21,52
20	19,89	8,6	8,99	16,7	16,86	10,7	11,06	17,6	17,63	21,9	21,44
20,1	19,89	8,6	8,9	16,8	16,78	10,7	10,97	17,6	17,63	21,9	21,44
20	19,89	8,6	8,82	16,7	16,86	10,7	10,89	17,6	17,71	21,9	21,44
20,1	19,89	8,6	8,82	16,7	16,86	10,7	11,22	17,6	17,63	21,9	21,44
20	19,89	8,6	8,9	16,8	16,86	10,7	10,97	17,6	17,63	21,9	21,44
20	19,89	8,6	8,82	16,7	16,86	10,7	10,97	17,6	17,63	21,9	21,44
20,1	19,89	8,6	8,9	16,7	16,86	10,7	11,06	17,6	17,63	21,9	21,44
20,1	19,89	8,6	8,9	16,7	16,93	10,6	10,97	17,6	17,71	21,9	21,44
20,1	19,89	8,6	8,82	16,8	16,86	10,7	10,97	17,6	17,63	21,9	21,44
20,1	19,89	8,6	8,9	16,7	16,78	10,6	10,97	17,6	17,63	21,9	21,44
20,03	19,86	8,61	8,90	16,72	16,89	10,70	11,12	17,58	17,63	21,90	21,44
Vershil 0.17		Vershil -0.29		Vershil -0.17		Vershil -0.42		Vershil -0.05		Vershil 0.46	

Tabel 5 DHT22 en ND03R00104J zes verschillende temperatuurmetingen

Voor alle metingen zie hoofdstuk 11.14 Volledige meting voor Kalibratie ND03R00104J.

4.5.4.2 Temperatuursensor (Grove – Temperatuursensor V1.2)

In Tabel 6 zijn de metingen te zien die gemaakt zijn doormiddel van het van een DHT22 sensor en de Grove – Temperatuursensor V1.2 op het Marvin bordje (Arduino) aan te sluiten. Om de om de vijf seconde werd er een meting gedaan. De DHT22 sensor werd dan digitaal uitgelezen en de Grove – Temperatuursensor V1.2 via een ADC-waarde. Deze ADC-waarde werd via de *formule 5* zoals te zien in hoofdstuk 4.5.3 om gerekend naar de temperatuur waarde in graden.

Zoals te zien in de metingen zijn de waardes van de DHT22 en de Grove – Temperatuursensor V1.2 in de eerste vier metingen exact of ongeveer gelijk. Pas als de temperatuur heel laag wordt zoals in de vierde meting wordt het verschil tussen de DHT22 en de Grove – Temperatuursensor V1.2 groter. Omdat de temperatuur sensor bedoeld is om te gebruiken binnen gebouwen, en ervan uit gegaan kan worden dat het daar geen min 21 graden wordt, is deze grove sensor geschikt om te gebruiken met de bijbehorende formule. Deze komt namelijk overeen met wat de werkelijke temperatuur is.

De Grove – Temperatuursensor V1.2 is met de *formule 5* zoals te zien in hoofdstuk 4.5.3 dus goed gekalibreerd.

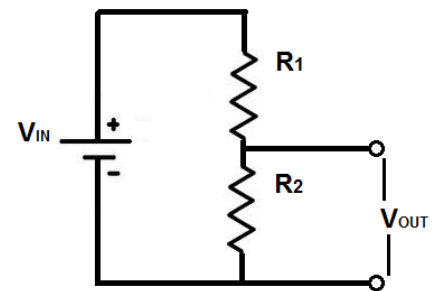
DHT22	Grove	DHT22	Grove	DHT22	Grove	DHT22	Grove	DHT22	Grove
22.50	22.46	21.10	21.10	8.80	8.89	5.90	6.18	-21.30	-19.82
22.50	22.46	21.10	21.10	8.80	8.89	5.90	6.18	-21.30	-19.61
22.50	22.46	21.10	21.10	8.80	8.89	5.90	6.09	-21.30	-19.61
22.50	22.46	21.10	21.10	8.80	8.89	5.90	6.18	-21.30	-19.61
22.50	22.46	21.10	21.10	8.80	8.97	5.90	6.18	-21.30	-19.82
22.50	22.46	21.10	21.10	8.80	8.89	5.90	6.18	-21.30	-19.61

Tabel 6 Vijf metingen van de temperatuur in graden DHT22 sensor en Grove – Temperatuursensor V1.2

Voor de volledige meting zie bijlage hoofdstuk 11.15 uit deze tabellen zijn de bovenstaande tabellen gevormd, hierbij is gekeken naar waar de DHT22 waarde stabiel was en niet meer daalde of steeg.

4.5.4.3 Lichtintensiteit sensor (Grove – Light sensor V1.1)

De NTC-weerstand die op de Grove – Light sensor v1.1 module zit is de GL5528. Deze NTC-weerstand vormt samen met een 10kΩ weerstand een spanningsdeler. Als de spanningsdeler uit Figuur 15 gebruikt wordt dan is R1 de NTC-weerstand en R2 de 10kΩ weerstand. De Vout van deze spanningsdeling gaat naar een opamp. Deze opamp is zo verbonden dat het een buffer vormt. De opamp heeft daarom geen invloed op het circuit maar splitst de impedantie van de spanningsdeler met de impedantie van de uitgang van de opamp. De uitgang van de opamp is verbonden aan de uitgang, het signaal, van de module. Uit metingen blijkt dat bij volledige verduistering van de omgeving van de uitgang van de sensor een ADC-waarde van 0 heeft en dat bij een volledig lichte omgeving de ADC-waarde 776 is. Dit is de maximale ADC-waarde want, als er een zaklamp tegen de LDR gehouden wordt zal de ADC-waarden niet toenemen. Als deze ADC-waarde in de formule worden ingevuld om de NTC-weerstandswaarde uit te rekenen komt daar de volgende uitkomst uit:



Figuur 15 Spanningsdeler

$$R_1 = \frac{R_2 * V_{in}}{V_{out}} - R_2 \text{ waardes invullen} \rightarrow \frac{10000 * 1023}{776} - 10000 = 3183\Omega$$

formule 6

Als er bij de waarde van de NTC weerstand (R1) in de grafiek van de datasheet de bijbehorende lichtsterkte wordt opgezocht komt er een waarde van ongeveer 33 lux tot meer dan 100 lux uit (zie Figuur 16 de rode lijn met de blauwe stippen). Verder staat in de datasheet dat de waarde waarbij licht gedetecteerd wordt bij 10 lux is. De waardes die bij deze 10 lux horen zijn 8kΩ en 20kΩ, zie de blauwen lijn met de rode stippen in zie Figuur 16.

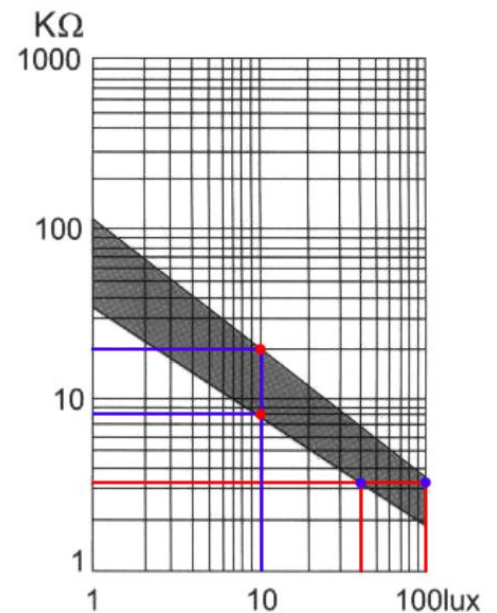
Als de uiterste waarde voor 10 lux wordt gepakt, de 20kΩ zal daar een ADC-waarde bijhoren van:

$$V_{OUT} = \frac{R_2 * V_{in}}{R_1 + R_2} \text{ waardes invullen} \rightarrow \frac{10000 * 1023}{20000 + 10000} = 341$$

formule 7

Als de NTC weerstand een waarde meet van onder de 10 lux (onder de ADC-waarde 341) betekend dit dat het 'donker' is en is de waarde gelijk of boven de 10 lux (gelijk of boven de ADC-waarde 341) dan betekend dit dat het 'licht' is. Wat dan de precieze lichtsterkte is niet na te gaan. Dit omdat de grafiek maar loopt tot 100 lux en dit is ook de minimale weerstandwaarde is van de LDR (3183Ω). Dit terwijl een bewolkte dag al een waarde heeft van ~1000 lux en een kantoor ruimte een waarde van ~500 lux [10]. De echte waarde bij licht zal daarom boven de 100 lux liggen. Deze sensor is daarom niet geschikt om waardes te kunnen meten boven de 100 lux. Om deze reden is kan er dus alleen onscheid gemaakt worden tussen 'donker', een waarde onder de 10 lux en 'licht', een waarde gelijk of hoger dan 10 lux.

Deze waarde van 341 zal in de code gezet worden, alle ADC-waardes van onder de 341 geven als data '0000' terug (donker) en alle waardes gelijk en hoger geven de data '0001' terug (licht).



Figuur 16 Grafiek lichtsterkte (in lux) tegenover de LDR waarde (in kΩ) uit de datasheet van de GL5528

4.5.5 Ongebruikte sensoren

Naast alle gebruikte sensoren zijn er ook een aantal sensoren niet gebruikt, zie Tabel 7 voor de reden.

Soort Sensor	Reden
CO-melder	De CO-melder is niet opgenomen in de demo's omdat de sensor relatief duur ten opzichte van een rookmelder en werkt ongeveer hetzelfde als de rookmelder. Daarom heeft de CO-melder opzich geen toegevoegde waarde om te gebruiken in de demo.
Glasbreuksensor	Het idee was om de ABUS FU7301W, te gebruiken. Deze sensor werkt als een switch. De sensor wordt op een raam geplak (dubbel glas) waarna het switch gedeelte open zal gaan als er een glasbreuk wordt veroorzaakt in een straal van 2 meter rond de sensor. De reden dat de sensor niet opgenomen is in de demo is dat het lastig demonstreren is en de werking verder gelijk is aan andere sensoren in die wel in de demo voorkomen.
Temperatuursensor NTCLE100E3334JB0	In eerste instantie was het idee om de NTCLE100E3334JB0 NTC-weerstand te gebruiken als temperatuursensoren. Alleen bij het meten bleek dat de sensor ongeschikt was omdat de temperatuur die berekend met de formule uit de datasheet niet de juiste temperatuur teruggaf. De formule is wel uitgewerkt in de bijlage hoofdstuk11.16.

Tabel 7 Ongebruikte sensoren

5 LoRaWAN Cloud server

De LoRaWAN Cloud server is op twee sites in te stellen:

KPN LoRa Developer Portal (<https://loradeveloper.mendixcloud.com/>)

KPN LoRa deviceManager (<https://www.kpn-lora.com/deviceManager/>)

Om de KPN LoRa deviceManager (Thingpark) te kunnen gebruiken moet er een account aangevraagd worden. Ook zal voor elke LoRa sensor sleutels aangevraagd moeten worden. Hiervoor is moet een abonnement voor moeten afgesloten. De kosten hiervan hangen af van hoeveel berichten er worden verzonden per dag of per jaar. Zie bijlage hoofdstuk 11.11 LoRa kosten. Daarom is in dit geval gekozen om de gratis KPN LoRa Developer Portal te gebruiken.

5.1 KPN LoRa Developer Portal

Met de KPN LoRa Developer Portal kan er met een nieuw account voor 10 weken 10 LoRa sensoren worden getest. Hierdoor kunnen er 10 sets van sleutels worden aangemaakt. Ga hiervoor naar de site en registreer een account en log met het account in. Nu kan er via de knop 'Add Device' een LoRa sensor worden geregistreerd. Klik op de knop en vul de volgende velden in, onder 'Device name' komt de naam van de sensor te staan en onder 'Desination URL' kom de link naar de server te staan. In dit geval is die link https://lora-techlab.kpn.com/lora_demo_server/v1/sensor_data.

5.1.1 AS_KEY (Application Server Key)

De AS_KEY staat in het venster 'My Account' dit venster komt op als er geklikt wordt op icoon van een poppetje rechtsboven in het scherm. Dit icoontje staat naast de tekst 'Hello [naam]'.

5.1.2 Nwkskey (Netwrok Session Key)

De nwkskey (NwkSKey) staat in het venster 'Edit Device' dit venster komt op als er geklikt wordt op de tekst 'Edit' die helemaal recht staat achter de sensor die toegevoegd is via de knop 'Add Device'. Zie Figuur 17 voor het venster.

5.1.3 Devaddr (Device Address)

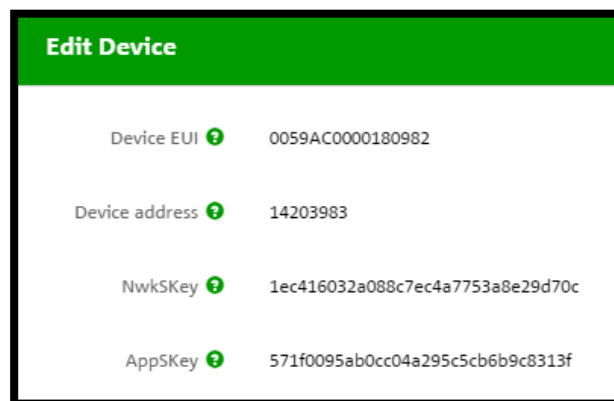
De devaddr (Device adress) is in hetzelfde venster te vinden als de nwkskey zie Figuur 17.

5.1.4 Appskey (Application Session Key)

De appskey (AppSKey) is in hetzelfde venster te vinden als de nwkskey zie Figuur 17.

5.1.5 Device EUI

De Device EUI is in hetzelfde venster te vinden als de nwkskey zie Figuur 17, maar zal nergens ingevoerd hoeven te worden. De Device EUI is een uniek nummer dat de LoRa sensor onderscheid. Dit unieke nummer wordt via de LoRaWAN Cloud mee gestuurd naar de server. Deze zal ook worden weergegeven in de applicatie. Zo kan er altijd worden gezien van welke LoRa sensor het bericht komt.



Edit Device	
Device EUI	0059AC0000180982
Device address	14203983
NwkSKey	1ec416032a088c7ec4a7753a8e29d70c
AppSKey	571f0095ab0cc04a295c5cb6b9c8313f

Figuur 17 Edit Device venster met nwkskey, devaddr en appskey

6 Server

Voor de server is gekozen om een WampServer te gebruiken. Wamp staat voor Windows, Apache, MySQL en PHP. Deze combinatie maakt het mogelijk om onder Windows doormiddel van Apache een webpagina te laten draaien waaraan een (MySQL) database verbonden zit. Dit alles wordt in de scripttaal PHP geschreven.

6.1 WampServer installeren

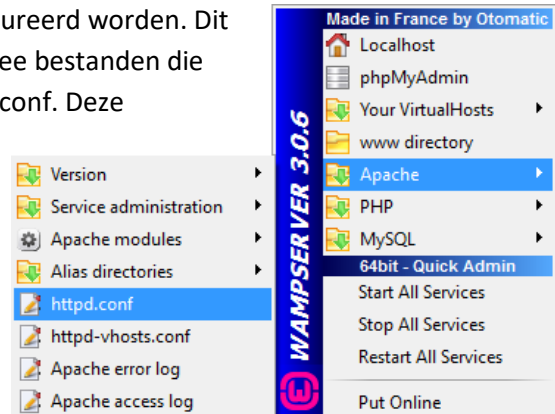
Om de WampServer te kunnen gebruiken zullen de volgende stappen moeten worden ondernomen:

- Download de WampServer via <http://www.wampserver.com/en/> dit kan de 32 bits of 64 bits versie zijn, qua werking zijn ze hetzelfde.
- Ga na het downloaden naar de map C:\wamp (32 bits versie) of C:\wamp64 (64 bits versie) en klik op wampmanager.exe. Sta deze toepassing toe en de WampServer zal opstarten. Als alles goed gaat zal in de Windows balk naast de tijd en datum een W-icoontje verschijnen (klik op de uitbouw pijl wanneer het icoon niet in de balk staat), deze kleurt groen als alle services gestart zijn. *Mocht het W-icoontje rood of oranje blijven zijn niet alle services gestart. Als dit gebeurt klik dan rechtermuisknop op het icoontje ga naar Tools en klik op Test Port 80. Nu zal poort 80 getest worden en er opent zich een scherm. Als dit scherm aangeeft dat de poort gebruikt wordt door een andere applicatie zal deze afgesloten moeten worden. Dit kan doormiddel van het PID-nummer van de applicatie die de poort gebruikt. Dit PID-nummer staat in datzelfde scherm. Het afsluiten van de applicatie gaat als volgt, ga naar taakbeheer (knoppen combinatie ctrl + alt + delete en open taakbeheer). Open de tab Details (komt in beeld zal er op 'meer details' geklikt wordt), zoek de applicatie met PID-nummer en klik op beëindigen. Klik nu op het W-icoontje en klik Restart All Services. Nu zal het icoontje groen worden omdat poort 80 nu wel vrij is.*
- Ga nu naar een browser en typ in de adres balk: <http://localhost/phpmyadmin/>, nu komt het phpmyadmin login scherm tevoorschijn. De eerste keer aanmelden is de gebruikersnaam root zonder wachtwoord. Vul root in en klik op Starten.
- Dit scherm is de administrator panel van de SQL-database. Het eerste wat er gedaan moet worden is het aanmaken van een wachtwoord. Klik op het tabblad Gebruikersaccounts, klik in de rij root op Rechten bewerken. Klik nu bovenin (onder het tabblad gebruikersaccounts) op Wachtwoord wijzigen. Vul nu een wachtwoord in het hokje Wachtwoord in, en doe dit nog een keer in het hokje Tik opnieuw in. Klik nu op Starten en het wachtwoord zal gewijzigd zijn. Onthoud dit wachtwoord goed!

Na het downloaden van de WampServer via en het installeren van de WampServer zal op de C:\ schijf de map van de WampServer staan.

6.2 WampServer configuratie

De WampServer moet na het installeren ook nog geconfigureerd worden. Dit zodat de server te benaderen is vanaf elk apparaat. De twee bestanden die aangepast moeten worden zijn `httpd.conf` en `http-vhosts.conf`. Deze bestanden staan in de Wamp map op de C:\ schijf. Maar de bestanden zijn makkelijker te vinden door het klikken op het Wamp-icoon (W). Hier staan de bestanden onder Apache zie Figuur 18. In deze config bestanden zullen er een aantal stukjes code veranderd moeten worden. De code van `httpd.conf` staat in de bijlage in hoofdstuk 11.12 `httpd.conf`. En de code van `http-vhost.conf` staat in de bijlage in hoofdstuk 11.13 `http.v-host.conf`.



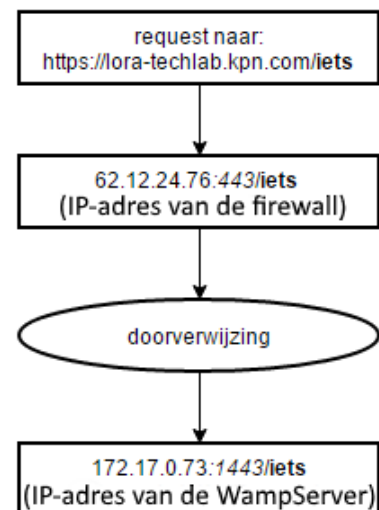
Figuur 18 Configuratie bestanden WampServer

6.3 DNS en Virtual Machine

De WampServer zal, om altijd bereikbaar te zijn, altijd moeten draaien. Dit kan op een vaste computer maar dan zal deze computer altijd aan moeten staan. Dit is een oplossing maar als de computer ook voor andere dingen gebruikt wordt is dit niet de handigste oplossing. Daarom is in dit geval gekozen om de WampServer te laten draaien op een virtuele Windows machine (Windows Server 2012).

De DNS (Domain Name System) linkt een naam aan een IP-adres. Zo correspondeert de naam *lora-techlab.kpn.com* nu aan het IP-adres 62.12.24.76. Dit IP-adres is het IP-adres van de firewall. De WampServer zelf heeft ook een IP-adres. In dit geval is dat 172.17.0.73. De virtuele machine zal, om de WampServer te kunnen bereiken, het IP-adres van de firewall moeten doorverwijzen naar het IP-adres van de WampServer.

Als er een verzoek gestuurd wordt naar *lora-techlab.kpn.com* is dit verzoek gelijk aan een verzoek naar het IP-adres 62.12.24.76. Dit verzoek wordt door de firewall doorverwezen naar het IP-adres 172.17.0.73. Met andere woorden een verzoek naar *lora-techlab.kpn.com* is eigenlijk een verzoek naar het IP-adres 172.17.0.73 (zie Figuur 19). Deze techniek heet Network Address Translation (NAT). In Figuur 19 is ook te zien dat het om een `https://` verzoek gaat. Dit verzoek is een verzoek naar poort 443 van het gelinkte IP-adres. Zo wordt er eigenlijk een verzoek gedaan naar poort 443 van het IP-adres 62.12.24.76. De doorverwijzing hiervan is naar poort 1443, dit omdat de WampServer wel op poort 1443 kan luisteren maar niet op poort 443. De techniek die gebruikt wordt voor het doorverwijzen heet Port Address Translation (PAT).



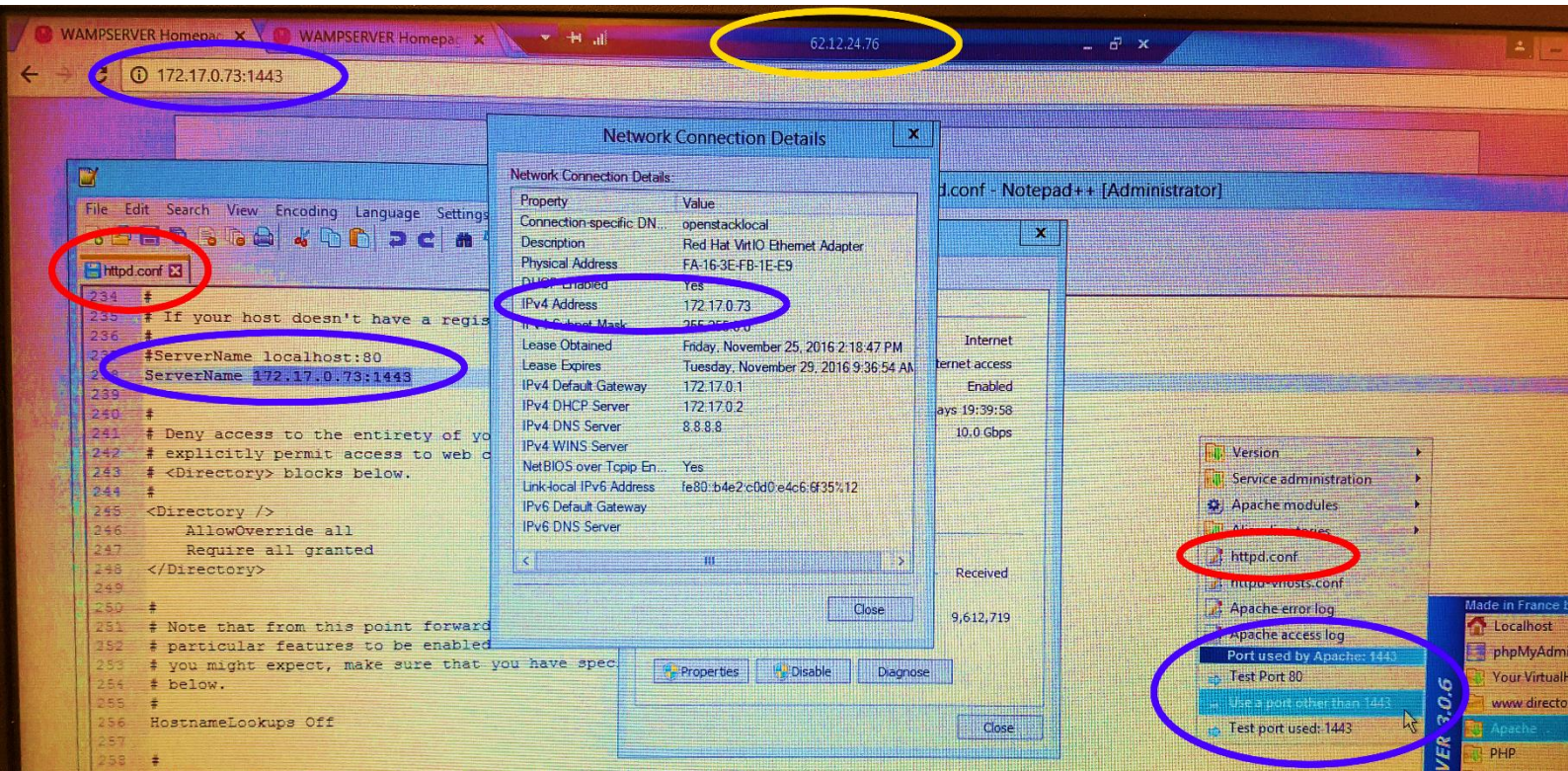
Figuur 19 WampServer benaderen

Dus als er naar het sub domein *iets* een request wordt gedaan via `https://lora-techlab.kpn.com/iets` wordt er eigenlijk een request gedaan naar `172.17.0.73/iets` op poort 1443. De reden dat het een `https://` verzoek is, is omdat het LoRa netwerk de binnengekomen informatie alleen naar een 443 poort stuurt (met verzoek met SSL Certificaat).

LET OP, zorg hierbij wel dat er op virtuele machine poort 1443 door de Windows firewall laat!

6.4 Instellen poort WampServer

Zoals in Figuur 19 te zien zal de WampServer luisteren op poort 1443. In Figuur 20 is een afbeelding te zien van het scherm in de virtuele machine. In de oranje ovaal is het IP van deze virtuele machine te zien. Via de link 172.17.0.73:1443 (IP 172.17.0.73 poort 1443) is de homepage van de WampServer te bereiken. Dit IP wordt bepaald door het netwerk, zoals te zien in de blauwe ovaal in het midden. In de blauwe ovaal helemaal recht in het scherm is te zien waar de poort 1443 ingesteld moet worden. Dit is onder Apache die door een klik op het Wamp-icoontje tevoorschijn komt in te stellen. In het httpd.conf bestand moeten dit IP-adres en de bijbehorende poort achter ServerName ingevuld moeten worden. Dit bestand is op dezelfde plaats te vinden als waar de poort ingesteld moet worden zie rode ovaal linksonder. Als deze instellingen goed staan is de WampServer op de virtuele machine te bereiken op 172.17.0.73:1443 en via alle andere apparaten via <https://loratechlab.kpn.com>.



Figuur 20 WampServer op de virtuele machine

6.5 SSL Certificaat

Om informatie te kunnen ontvangen van het LoRa netwerk is een SSL-certificaat vereist. Een SSL-certificaat is nodig om aan te tonen dat de server echt is wie die is. Zo zal de informatie uit het LoRa netwerk altijd naar de juiste server gestuurd worden, want zodra het SSL-certificaat niet klopt zal het LoRa netwerk de informatie ook niet versturen.

Het SSL Certificaat zal aangevraagd moeten worden met een Certificate Signing Request-file. Deze CSR-file kan gemaakt worden met OpenSSL dit is een programma die in bestanden van de WampServer zijn meegenomen. Dit zijn alleen de verkeerde bestanden, omdat er een verkeerde versie van OpenSSL in de WampServer is meegenomen. Deze OpenSSL versie werkt niet samen met

de nieuwste WampServer. Volg daarom de volgende instructies:

<http://stackoverflow.com/questions/17354788/enabling-openssl-in-wamp> [5] en vervang de oude bestanden met nieuwe OpenSSL bestanden. Zodra dit is gedaan kan er via de Opdrachtprompt (voer deze uit als Administrator) een CSR-file en geheime sleutel aangemaakt worden. Dit kan op de volgende manier, voer in de Opdrachtprompt de volgende regel in (alles tussen vierkante haken is optioneel of is een waarde die zelf ingevuld moet worden):

```
cd C:\wamp[64]\bin\apache\apache[apache versie, bijvoorbeeld 2.4.23]\bin
```

```
openssl req -out [CSR bestand naam].csr -new -newkey rsa:2048 -nodes -keyout [geheime sleutel naam].key
```

Nu komt er in de opdracht prompt een vragen te staan, vul de antwoorden in en druk op enter voor de volgende vraag:

Country name	Vul hier het land in
State or Province Name	Vul hier de provincie in
Localisty Name	Vul hier de locatie in
Organization Name	Vul hier de bedrijfsnaam in
Organization Unit Name	Vul hier de afdeling naam in
Common Name	Vul hier de domeinnaam in
Email Address	Vul hier het e-mailadres in
A challenge password	Vul hier niks in (alleen enter drukken)
An optional company name	Vul hier niks in (alleen enter drukken)

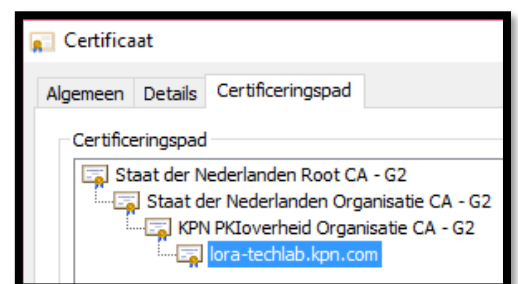
Tabel 8 CSR-aanvraag opdracht prompt vragen

Na het invullen van deze vragen zal het CSR-bestand en geheime sleutel aangemaakt zijn. Deze zullen in map C:\wamp[64]\bin\apache\apache[apache versie, bijvoorbeeld 2.4.23]\bin staan. Nu kan met dit CSR-bestand een SSL-certificaat aangevraagd worden.

Als het SSL-certificaat is aangevraagd zal de reactie op deze aanvraag het servercertificaat zijn. Dit is een file die er zo uit ziet:

```
-----BEGIN CERTIFICATE-----
MIIDrTCCAxagAwIBA.....
-----END CERTIFICATE-----
```

Alleen is in de echte file de middelste code een stuk langer. Dit certificaat is opzichzelf geen vertrouwt certificaat, omdat dit geen openbaar certificaat is, wel kan dit certificaat vertrouwt worden maar dan zal de aanvragende partij dit certificaat als vertrouwt moeten zien. Maar omdat die niet zal gebeuren moet het certificaat in een certificaat ketting worden gezet zoals te zien in Figuur 21. De bovenste drie certificaten zijn al bestaande certificaten die moeten worden gebruikt in combinatie met het aangevraagde certificaat (de onderste certificaat). Om dit voor elkaar te krijgen zal de file `httpd-ssl.conf` aangepast moeten worden. In deze file staan de configuraties voor de SSL-verbinding. In deze file moet een verwijzing komen te staan naar de eerder gemaakte geheime sleutel en de certificaat ketting bestanden. Het certificaat ketting bestand wordt als volgt gemaakt: eerst wordt de hele serversertificaat in een file



Figuur 21 SSL-Certificaat ketting

gezet, daarna het KPN PKloverheid Organisatie certificaat, daarna het Staat der Nederlanden Organisatie certificaat en als laatste de Staat der Nederlanden Root certificaat.

De file ziet er dan kort gezegd zo uit:

```
-----BEGIN CERTIFICATE-----  
Code van de aangevraagde ssl certificaat  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Code van KPN PKloverheid Organisatie certificaat  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Code van Staat der Nederlanden Organisatie certificaat  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Code van Staat der Nederlanden Root certificaat  
-----END CERTIFICATE-----
```

Het bestand met daarin de codes van alle certificaten zal bij het antwoord van de aanvraag van het ssl-certificaat zitten.

Om de rest van de bestanden te verkrijgen die nodig zijn is het nodig om weer de opdrachtenprompt te openen en weer deze opdracht in te voeren:

```
cd C:\wamp[64]\bin\apache\apache[apache versie, bijvoorbeeld 2.4.23]\bin
```

Na deze opdracht zal het server certificaat aangemaakt moeten worden, dit gaat doormiddel van de volgende opdracht:

```
openssl x509 -in [CSR bestand naam, eerder ingevuld].csr -out [naam voor de servercertificaat].cert -  
req -signkey [geheime sleutel naam, eerder ingevuld].key -days 3650
```

De days in deze opdracht zijn het aantal dagen dat het certificaat geldig is. Zie het https-ssl.conf bestand in de bijlage hoe het bestand nu ingevuld moeten worden en waar de sleutels moeten komen te staan.

LET OP, de server zal voor https altijd moeten luisteren op poort 443, pas deze daarom niet aan in de httpd-ssl.conf file!

Ook zal het bestand httpd-conf hier aangepast moeten worden, deze file is al eerder besproken en deze aanpassing is daar al in meegenomen zie comments in de code (hoofdstuk 11.12 httpd.conf). De request zoals te zien in Figuur 19 zal nu niet meer op poort 1443 zijn maar op poort 443. Dit omdat dit de standaard poort is voor een SSL verbinding.

LET OP, zorg hierbij wel dat er op virtuele machine poort 443 door de Windows firewall laat!

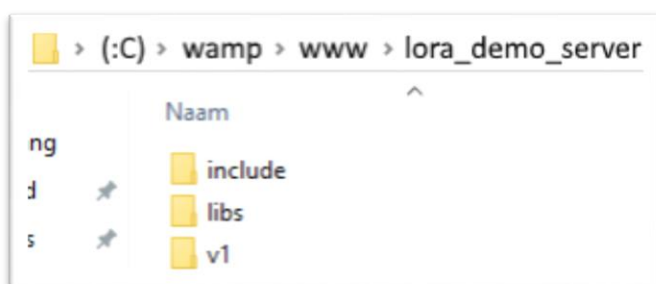
De http-ssl.conf file en alle andere files zijn te vinden op de USB-stick die bijgeleverd is met de scriptie. Hierin staan alle bestanden van de hele WampServer. Raadpleeg deze om de code van de http-ssl.conf file en andere bestanden uit de WampServer te bekijken.

6.6 Waarom DNS en SSL-Certificaat gebruikt worden

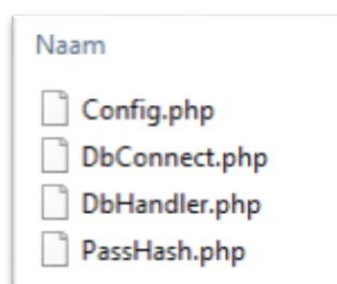
De reden waarom er een DNS en een SSL-Certificaat gebruikt worden is omdat het LoRa netwerk dit eist. Berichten worden alleen verzonden naar een DNS en niet naar een IP-adres. Ook wordt het bericht niet verzonden als de server geen geldig SSL-Certificaat bevat.

6.7 Bestanden om WampServer te kunnen gebruiken

In de map www map van de WampServer (pad: C:\wamp[64]\www) komt de map met de naam van de server in lora_demo_server. In de lora_demo_server map komen drie mappen te staan, include, libs en v1. Die ziet er dan uit zoals weergegeven in Figuur 23.



Figuur 23 lora_demo_server map



Figuur 22 Include map bestanden

In Figuur 22 is de zijn de bestanden te zien die in de map include horen te staan. Dit zijn Config.php, DbConnect.php, DbHandler.php en PassHash.php.

In de Config.php komt de configuratie van de database te staan zie Tabel 9.

```

1  <?php
2  /**
3   * Database configuration
4   */
5   define('DB_USERNAME', 'root');
6   define('DB_PASSWORD', 'password');
7   define('DB_HOST', 'localhost');
8   define('DB_NAME', 'lora_demo_server');
9
10  define('USER_CREATED_SUCCESSFULLY', 0);
11  define('USER_CREATE_FAILED', 1);
12  define('USER_ALREADY_EXISTED', 2);
13  ?>

```

Tabel 9 Config.php code

In dit bestand gaat het vooral om regel 5 tot en met 8. In regel 5 zal de gebruikersnaam ingevuld moeten worden. Deze gebruikersnaam is hetzelfde waarmee ingelogd is zoals besproken in het begin van dit hoofdstuk. Het bijbehoren wachtwoord wat in het begin van het hoofdstuk is gewijzigd zal hier op de plek van password moeten komen te staan (regel 6). Op regel 7 moet de host ingevuld worden, dit is localhost. Localhost zal namelijk wijzen naar het IP van de computer zelf, door hier localhost in te vullen is deze instelling altijd juist. Op regel 8 zal de naam van de database ingevuld moeten worden. In dit geval is dat lora_demo_server.

Het bestand DbConnect.php zorgt voor de database connectie. In deze file staat de functie Connect() die aangeroepen kan worden wanneer er een connectie met de database nodig is.

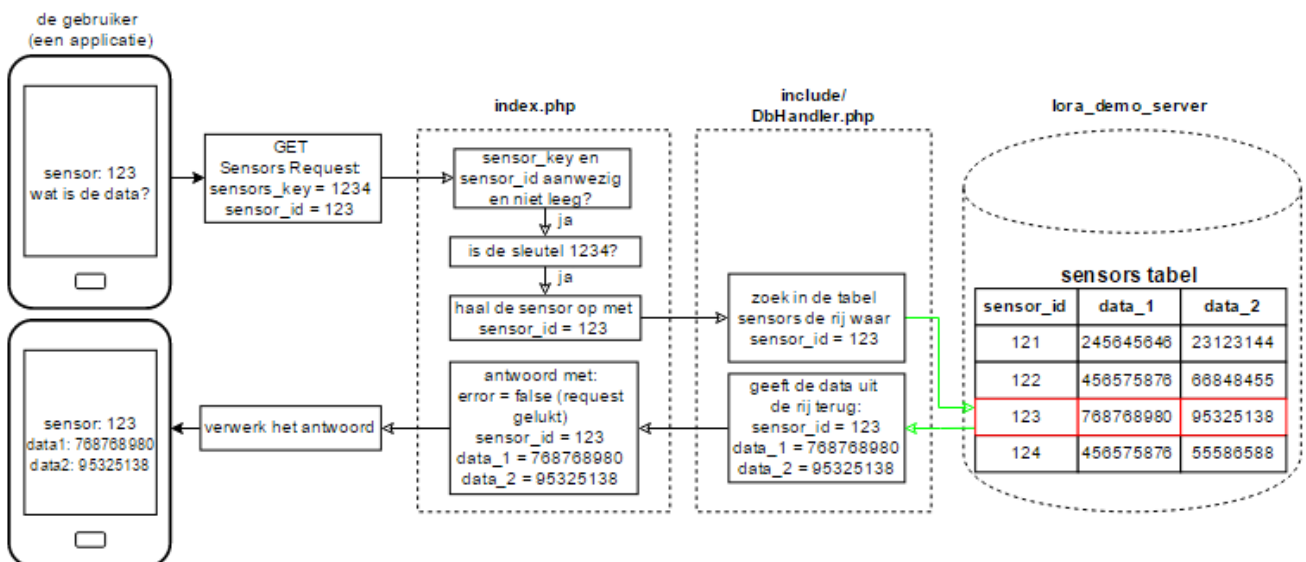
Dit bestand staat net als alle bestanden van WampServer op de USB-stick bijgeleverd bij de scriptie.

Het bestand DbHandler.php is het bestand waarmee de tabellen in de database benadert kunnen worden. Later meer over dit bestand.

Het bestand PassHash.php zal later worden gebruikt om een wachtwoord als een hash op te slaan. Dit houdt in dat het wachtwoord geëncrypt wordt, waardoor er niet na te gaan is wat het wachtwoord is. Wel kan de hash vergeleken worden met de hash van een ingevoerd wachtwoord, als deze overeen komen is het wachtwoord juist.

Verder staat er in de map zoals te zien is in Figuur 23 de map libs. In deze map staat de libery (bibliotheek) die gebruikt wordt om de PHP-bestanden te kunnen laten werken. De libery die gebruikt wordt is het Slim Framework[12]. Zoals de naam al zegt is dit het frame van de server.

In de map lora_demo_server (Figuur 23) is de laatste map v1, dit staat voor versie 1. Deze map kan ook een andere naam hebben. In deze map komen twee bestanden te staan .htaccess en index.php. Het bestand .htaccess is verder niet belangrijk en hoeft niet aangepast te worden. Het bestand index.php wel, in dit bestand worden alle requesten verwerkt. Een request is een verzoek voor bepaalde data. Een request kan verschillende vormen hebben zoals, GET, POST, PUT of DELETE. Deze hebben verschillende invloeden op de data, zo zal GET alleen de data opvragen, POST data in de database zetten, PUT de data in de database vervangen en DELETE de data uit de database verwijderen. Hoe dit precies werkt zal aan de hand van Figuur 24 uitgelegd worden.



Figuur 24 Server werking

In dit voorbeeld wil de gebruiker (in dit geval een applicatie op een mobiele telefoon) de data van sensor 123 weten. Dit betekent dat er data uit de database gehaald moet worden. Dit gebeurt doormiddel van een GET request. Dit request wil de data uit de tabel sensors halen, die in de lora_demo_server database staat. Hiervoor zijn twee parameters nodig namelijk de sensors_key (de

sleutel om de aanvraag te kunnen doen) en `sensor_id` (de identiteit van de sensor). In dit voorbeeld zal de request als link zo uitzien:

https://lora-techlab.kpn.com/lora_demo_server/v1/sensors?sensors_key=1234&sensor_id=123

De link begint met de host `https://lora-techlab.kpn.com`, met daarachter de database naam, mapnaam, tabel naam, en [\(query\) parameters](#). Deze parameters worden op de volgende manier aan de link toegevoegd. De eerste parameter begint met een vraagteken met daarna de parameter naam, een gelijkteken en dan de parameter waarde. De tweede parameter begint met een ampersand en daarna de parameter naam, het gelijkteken en de parameter waarde. Zou er een derde of meer parameters worden toegevoegd worden deze op dezelfde manier als de tweede parameter toegevoegd.

In dit voorbeeld wordt een GET request gemaakt voor de tabel `sensors`. Als dit gedaan wordt zal er in het bestand `index.php` gezocht worden naar de functie die de GET afhandelt voor de tabel `sensors`. In deze functie wordt er een aanroep gedaan naar een functie in het `Dbhandler.php` bestand de zogenaamde db-functies (database functies). Deze functies kunnen data in de tabel zetten, uit de tabel halen, uit de tabel verwijderen en in de tabel vervangen. Dit zijn eigenlijk de echte POST, GET, DELETE en PUT functies. In sommige gevallen zullen er ook meer db-functies gebruikt worden in één functie. Zo zal er bijvoorbeeld bij een POST request eerst gekeken moeten worden of er data in de tabel staat (GET) waarna er beslist wordt of de data vervangen moet worden (PUT) of er nieuwe data moet worden toegevoegd (POST).

Voor de opzet van de server is gebruik gemaakt van de Android Login and Registration with PHP, MySQL and SQLite tutorial van AndroidHive[3].

6.8 Tabellen

Om data op te slaan zullen er tabellen op de server gezet moeten worden. Deze tabellen heten SQL (Structured Query Language) tabellen. Deze zijn te zien op de localhost in phpMyAdmin. Deze phpMyAdmin is te bereiken via de browser (`localhost/phpmyadmin`).

Als er ingelogd is kan er links in het beeld onder 'New' een nieuwe database aangemaakt worden. In het aanmaak scherm van deze database kan een naam gegeven worden aan de database (in dit geval `lora_demo_server`, deze database naam is ook ingevoerd in het bestand `config.php` zie Tabel 9) waarna er op de knop 'Create' gedrukt wordt om de database aan te maken. Nu verschijnt de database naam links in het scherm.

Als er op het plusteken naast de database geklikt wordt verschijnt onder de database naam het kopje 'New'. Als er op dit kopje geklikt wordt zal er een nieuwe tabel aangemaakt kunnen worden. Bovenaan wordt de tabel naam ingevuld met daaronder de kolommen van de tabel. Wat er precies in de kolommen moet komen te staan is verderop in dit hoofdstuk te lezen.

In het voorbeeld in Figuur 24 wordt bijvoorbeeld de tabel `sensors` gebruikt om data uit te halen. De gegevens die nodig zijn voor deze tabel zijn te lezen in hoofdstuk 6.11.1 SQL-tabel Sensor Data. Naast deze tabel zijn er nog veel meer tabellen zoals bijvoorbeeld de tabel `notification_token`. De SQL Structuur tabel hiervan is te zien in het onderstaande figuur (zie Figuur 25).

Hoe de data die in deze tabel eruit ziet is te zien in Figuur 26. In deze tabel is maar één rij toegevoegd maar een tabel kan meer rijen hebben. Bij het voorbeeld uit Figuur 24 heeft de tabel bijvoorbeeld vier rijen waarbij er in één van de kolommen (in dit geval de sensor_id kolom) wordt gezocht naar een waarde (in dit geval 123) waarna de data uit de hele rij terug gegeven wordt.

#	Naam	Type	Collatie	Attributen	Leeg	Standaardwaarde	Opmerkingen	Extra	Actie
1	token	text	latin1_swedish_ci		Nee	Geen			Veranderen Verwijderen Primaire sleutel Meer
2	notification_on	text	latin1_swedish_ci		Nee	Geen			Veranderen Verwijderen Primaire sleutel Meer

Figuur 25 Structuur (structuur van de SQL-tabel)

token	notification_on
dGYxiVTcUDU:APA91bGNb9XQTrdZRIeSb_JU-wt_Au3Xx-sLw...	true

Figuur 26 Verkennen (data in SQL-tabel)

De kolommen die in Figuur 25 zijn niet allemaal relevant, dit omdat niet alle kolommen gebruikt worden. Hier een korte uitleg van wat de kolommen betekenen:

Naam	Uitleg
# (rij nummer)	Dit is het rij nummer, dit nummer heeft eigenlijk geen functie en is alleen om de tabel overzichtelijk te houden.
Naam	Dit is de naam van de kolom, deze is zeer belangrijk. Het heeft als functie dat de kolom gevonden kan worden en daardoor uitgelezen, bewerkt of verwijderd kan worden.
Type	Dit is het type data dat in de kolom kan worden gezet. Ook kan er bij het invoeren van het type een (maximum) lengte worden ingevuld. Zo kan het type bijvoorbeeld een 'int(11)' zijn. Dit betekent dat het type een integer (getal) van 11 cijfers lang is. In Figuur 25 is dit type 'text', dit kunnen alle karakters zijn.
Collatie	Dit is het lettertype dat wordt gebruikt. De collatie zoals te zien in Figuur 25 is 'latin1_swedish_ci' dit lettertype is niet gevoelig voor hoofdletters en daarom het default lettertype.
Attributen	Hier kan ingesteld worden of de variabele 'BINARY', 'UNSIGNED', 'UNSIGNED ZEROFILL' of 'on update CURRENT_TIMESTAMP'. Zo kan het bijvoorbeeld zijn dat het type een 'int(4)' (getal van 4 cijfers) is en het attribuut 'UNSIGNED ZEROFILL', dan wordt er bij een waarde van 4 de waarden 0004 in de tabel gezet.
Leeg	Dit geeft aan of de waarde in de tabel leeg kan zijn. De enige twee keuzes hierin zijn ja of nee.
Standaardwaarde	Standaardwaarde geeft de standaard beginwaarde van de variabele.
Opmerkingen	Hier kunnen opmerkingen worden ingevuld over de kolom, dit heeft verder geen invloed op de kolom.
Extra	In deze kolom wordt aangegeven of de tabel AUTO_INCREMENT (automatisch ophogen) heeft of niet. Dit is een extra die kan worden gebruikt om de tabel een automatische nummering mee te geven.
Actie	Hier kunnen bepaalde acties worden gedaan, zoals het veranderen of verwijderen van de kolom. Ook kan er een primaire sleutel worden toegevoegd. Deze primaire sleutel kan

	maar een keer vergeven worden en geeft aan dat alle waardes in de kolom uniek moeten zijn. Dit wordt vaak in combinatie met AUTO_INCREMENT gebruikt. Ook geeft deze primaire sleutel aan dat de waarde in de tabel niet NULL kan zijn.
--	--

Tabel 10 Uitleg SQL-tabel

Zoals te lezen is in Tabel 10 worden lang niet alle kolommen gebruikt. En daarom zullen alleen de gebruikte kolommen worden weergegeven in SQL-tabellen in dit verdere hoofdstuk.

6.9 User Configuration (Gebruikersconfiguratie)

Om te beginnen is de code zo opgebouwd dat aan het begin van de code een stuk User Configuration staat. In dit stuk zal de gebruiker de configuratie kunnen doen. De gebruiker zal hier de sleutels voor de verschillende functies kunnen aanpassen. Deze sleutels zijn ter beveiliging van de server. Zo wordt een request met een verkeerde sleutel geweigerd door de server. Dit betekent dat als bijvoorbeeld de triggers van een sensor opgehaald moeten worden maar de trigger sleutel komt niet overeen met de ingevoerde sleutel zal de server de triggers niet doorgeven. Dit betekent dat als de gebruiker deze sleutels verandert deze dat wel zal door moeten geven aan de applicatie. Dit zodat deze dezelfde sleutels blijft gebruiken waarmee de gegevens kunnen worden opgeslaan of opgehaald. Verder staat er ook nog de AS_KEY dit is de application server key. Deze sleutel is nodig om de token te maken waarmee een LoRa bericht zijn echtheid kan aantonen. Deze sleutel is bekend in het LoRa netwerk en op de server. Zie hoofdstuk 5.1.1 om te zien waar de AS_KEY te vinden is.

Er staan ook twee functies in de configuratie de `getNotificationBody` en `getNotificationIcon`. Deze functies hebben te maken met de notificaties die worden verstuurd zodra een sensor waarde een bepaalde trigger overschrijdt. Met `getNotificationBody` wordt de body van de notificatie opgehaald. Hierin wordt aan de hand van het sensor type de sensor eenheid ingevoerd en de formule voor het omrekenen van de data voor de triggers. Zo kan het bijvoorbeeld zijn dat een temperatuur sensor van type 0001 een waarden kan hebben van 0 tot 1000. Deze waarde stelt dan een waarde voor van 0 tot 50 graden. De eenheid hierbij is dan °C en de formule `trigger_high/200` of `trigger_low /200`. Zo wordt de code:

```
function getNotificationBody($sensor_id, $bool_trigger_low, $trigger_low, $trigger_high) {

    switch (substr($sensor_id, 4, 4)) { //sensor type is the last 4 digits of sensor_id, started at 4 and 4 digits long.
        case "0001":

            $sensor_unit = " °C";
            if ($bool_trigger_low){
                $sensor_warning = "The temperature is lower than ";
                return $sensor_warning.number_format(($trigger_low/200) , 2, '.', '').$sensor_unit;
            }else{
                $sensor_warning = "The temperature is higher than ";
                return $sensor_warning.number_format(($trigger_high/200) , 2, '.', '').$sensor_unit;
            }

            break;

        }

    }

}
```

De functie bestaat uit een switch die de laatste 4 cijfers van de sensor id pakt om het sensor type te krijgen. Dit sensor type is in dit voorbeeld 0001 zoals achter case te zien. In deze case staat als eerste

de `sensor_unit` hier moet in dit geval °C gezet worden. In de return komt dan de formule van de sensor te staan. In dit voorbeeld is dat de trigger / 200 (zie Tabel 4 uit hoofdstuk Gebruikte sensoren 4.5.1. Voor alle formules van de sensoren). Achter deze formule staat een twee, dit zijn het aantal getallen achter de komma ook zorgt dit getal ervoor dat bij meer getallen achter de komma het getal netjes afgerond wordt. De punt en komma tussen haakjes daar weer achter zorgen voor duizendtallen met een komma worden weergegeven en een komma getal met een punt bijvoorbeeld: 3,800,700.99. Dit kan ook aangepast worden naar wat de gebruiker wil. De totale return, en wat de notificatie zal laten zijn, zal in dit voorbeeld eerst de waarschuwing zijn zoals *The temperature is lower than*, daarna de waarden bijvoorbeeld 20 en daarna de sensor eenheid (°C). Het de totale zin wordt dan *The temperature is lower than 20 °C*. Als de trigger laag is ingesteld op 20 graden en de temperatuur ook daadwerkelijk lager dan 20 graden is.

In de `getNotificationIcon` functie kan de afbeelding worden ingesteld die bij een notificatie wordt weergegeven. Dit wordt weer aan de hand van het sensor type gedaan. De return waarden hier is de naam van de afbeelding. Deze naam moet wel overeenkomen met de afbeelding naam in de applicatie!

6.10 Sensors

In de SQL-tabel van Sensors staat alle laatste informatie en data van alle sensoren die bekend zijn op de server. Dit betekent dat alle sensoren maar een keer in deze tabel kunnen staan. Verder zal de data worden vervangen zodra er nieuwe data van een al bekende sensor op de server binnen komt.

6.10.1 SQL-tabel Sensors

In de onderstaande tabel is de SQL-tabel van Sensors te zien. Hierin zal, als de sensor nog niet in de tabel staat (nieuw `sensor_id`), alle kolommen worden ingevuld. Daarna zal bij het binnen komen van een sensor met hetzelfde `sensor_id` alleen de kolommen die beginnen met `last_` worden bijgewerkt.

Naam	Type	Collatie	Leeg	Standaardwaarde
<code>sensor_id</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>sensor_DevEUI</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>first_update</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>last_update</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>last_value</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>last_trigger_value</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>last_trigger_low</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>
<code>last_trigger_high</code>	text	latin1_swedish_ci	Nee	<i>Geen</i>

Tabel 11 SQL-tabel Sensors

6.10.2 GET Sensors

Als er een GET request wordt gedaan naar Sensors zullen de volgende twee (query) parameters mee moeten worden gestuurd: `sensors_key` en `sensor_id`. Deze zullen gecontroleerd worden of deze aanwezig zijn in de request. Zijn deze parameters niet aanwezig is de request niet in orde en zal er een error bericht als antwoord terug gestuurd worden. Dit error bericht vat twee variabelen de variabelen `error` en `message`. Als er een error bericht gestuurd wordt dan is de variabele `error` 'true' (waar) en in de `message` zal de bijbehorende foutmelding komen te staan. Kloppen de parameters wel dan zal de `sensors_key` worden vergeleken met de (bij user configuration) ingevulde `sensors` sleutel. Komt deze sleutel overeen zal er worden gekeken naar de parameter `sensor_id`. Hierbij zijn

er twee mogelijkheden als de `sensor_id` waarde gelijk is aan 'All' dan zullen alle sensoren met bijbehorende waardes worden teruggestuurd. Ook zit in dit bericht het aantal sensoren dat in de tabel zit meesturen. Dit zal worden gebruikt om een array (lijst) met sensoren te maken die op de server bekend zijn. En deze array zal daarna in samen met de variabele `number_of_sensors` (het aantal sensoren in de lijst) en een positief error bericht (`error = false` en `message` zegt dat het verwerken goed gegaan is).

De tweede mogelijkheid is dat `sensor_id` een tekst is met 8 getallen (bijvoorbeeld 00010003). Deze tekst geeft dan een bepaalde sensor aan. Nu wordt er een bericht teruggestuurd met alleen de waardes uit de rij waar `sensor_id` overeenkomt met die tekst. Dit antwoord zal dan ook de variabele `number_of_sensors` met als waarde '1' (omdat het maar om één sensor gaat) bevatten en een positief error bericht (weer met `error = false` en `message` zegt dat het verwerken goed gegaan is). Het kan ook voorkomen dat het `sensor_id` niet klopt of dat de tabel leeg is. Dan zal er een error bericht worden teruggestuurd met een `message` waarin staat wat er fout is gegaan.

In de onderstaande Tabel 12 is dit verhaal kort samengevat:

Parameters in request	Aangeropen db-functie	Verwerking	Antwoord
<code>sensor_id = 'All'</code> (<code>sensors_key</code>)	getAllSensors: geeft alle data uit de tabel terug.	Alle data wordt in een array gezet en de grootte van deze array wordt in <code>number_of_sensors</code> gezet (dit is het aantal verschillende sensors).	Array met de variabelen: <ul style="list-style-type: none"> - <code>sensor_id</code> - <code>sensor_DevEUI</code> - <code>first_update</code> - <code>last_update</code> - <code>last_value</code> - <code>last_trigger_value</code> - <code>last_trigger_low</code> - <code>last_trigger_high</code> Variabele: <ul style="list-style-type: none"> - <code>number_of_sensors</code> <code>error = false</code> <code>message = [message]</code>
<code>sensor_id = [sensor_id]</code> (<code>trigger_key</code>)	getSensors: geeft de data uit de rij terug waar <code>sensor_id</code> overeenkomt met de parameter <code>sensor_id</code> .	<code>number_of_sensors</code> wordt op '1' gezet, dit omdat er maar één sensor wordt opgehaald.	Variabelen: <ul style="list-style-type: none"> - <code>sensor_id</code> - <code>sensor_DevEUI</code> - <code>first_update</code> - <code>last_update</code> - <code>last_value</code> - <code>last_trigger_value</code> - <code>last_trigger_low</code> - <code>last_trigger_high</code> - <code>number_of_sensors</code> <code>error = false</code> <code>message = [message]</code>

Tabel 12 GET-sensors

6.10.3 DELETE Sensors

In Tabel 13 is de functie DELETE sensors uitgewerkt. Hierbij zijn `sensor_id` en `sensors_key` de parameter invoer. Daarbij zal eerst de `sensors_key` gecontroleerd worden of die klopt waarna de functie verder zal worden uitgevoerd. Daarna zijn er twee mogelijkheden voor de `sensor_id` waarde. Dit zijn 'All' of een gewone `sensor_id` waarde (8-cijferig getal) hierbij zijn er twee verschillende db-functies die kunnen worden aangeroepen. Het antwoord op de DELETE sensors request zal, als alles

goed is gegaan, een bericht zijn met daarin de variabele error die 'false' is en de variabele message met het bericht erin dat het verwijderen van de sensor of sensoren goed gegaan is.

Parameters in request	Aangeropen db-functie	Verwerking	Antwoord
sensor_id = 'All' (sensors_key)	deleteAllSensors: verwijdert alle rijen uit de tabel zodat de tabel helemaal leeg is.	<i>Geen speciale handelingen</i>	<i>error = false message = [message]</i>
sensor_id = [sensor_id] (sensors_key)	deleteSensors: verwijdert alle data aan de hand van de sensor_id, alle rijen waar de sensor_id overeenkomen zullen verwijderd worden.	<i>Geen speciale handelingen</i>	<i>error = false message = [message]</i>

Tabel 13 DELETE sensors

6.11 Sensor Data

6.11.1 SQL-tabel Sensor Data

In Tabel 14 is de SQL van de sensor data te zien. In deze tabel staat één bijzondere variabele namelijk de timestamp. De timestamp is een variabele die als type ook timestamp heeft (type tijd). De waarde die deze variabele krijgt is de tijd dat de rij met data wordt ingevoerd in de tabel. Er wordt bijvoorbeeld data gepost in de tabel dan worden alle waardes ingevuld behalve de timestamp de timestamp zal automatisch de tijd dat het dan heeft in de tabel zetten. Zo wordt de rij: [huidige tijd] [sensor_id waarde] [sensor_DevEUI waarde] etcetera.

Naam	Type	Collatie	Leeg	Standaardwaarde
timestamp	timestamp	-	Nee	CURRENT_TIMESTAMP
sensor_id	text	latin1_swedish_ci	Nee	Geen
sensor_DevEUI	text	latin1_swedish_ci	Nee	Geen
sensor_value	text	latin1_swedish_ci	Nee	Geen
time	text	latin1_swedish_ci	Nee	Geen
trigger_value	text	latin1_swedish_ci	Nee	Geen
trigger_low	text	latin1_swedish_ci	Nee	Geen
trigger_high	text	latin1_swedish_ci	Nee	Geen

Tabel 14 SQL-tabel Sensor Data

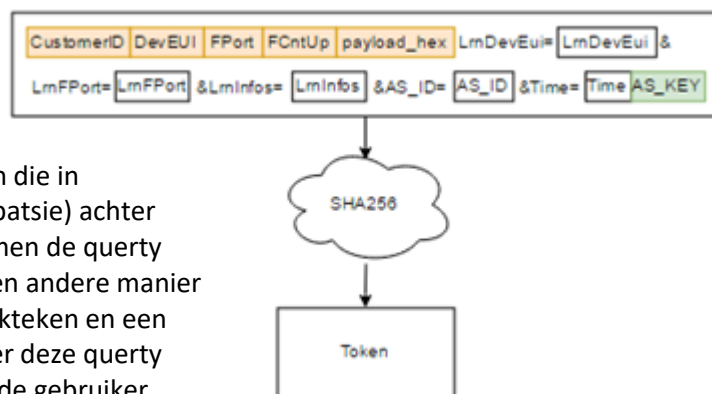
6.11.2 POST Sensor Data

De functie die een POST request naar de tabel sensor_data is de belangrijkste functie. Dit omdat in deze functie de LoRa berichten van het LoRa netwerk binnenkomen. De handelingen die de functie uit zal voeren zijn het beste uit te leggen in een flowchart (zie Figuur 28). Om te beginnen checkt de functie of alle query parameters in de request aanwezig zijn. Zodra er één mist of zodra er één leeg is zal de functie een error bericht terugsturen naar de verzender van de request ook zal de functie wordt dan worden afgesloten. Dit error bericht kan er dan als volgt uit zien:

```
{"error":true,"message":"Required field(s) LrnDevEui, LrnFPort, LrnInfos, AS_ID, Time, Token is missing or empty"}
```

Dit is een JSON-antwoord met twee variables, de variabele error die in dit geval 'true' is en de variabele message die in dit geval 'Required field(s) LrnDevEui, LrnFPort, LrnInfos, AS_ID, Time, Token is missing or empty' is. In dit request zijn, zoals te lezen, geen van alle query parameters ingevuld.

Na de query parameter check zal er worden gekeken of de body van het bericht aanwezig is. De body van het bericht is alle data die met het bericht wordt meegestuurd. Hierna volgt een check waarbij wordt gekeken of de body waar de data in staat, die later zal worden gebruikt, de body zelf is of in de variabele 'DevEUI_uplink' staat. Als dit laatste het geval is (data komt van thingspark) zal de data in 'DevEUI_uplink' worden benoemd als body data. Is dit niet het geval dan is de body ook de body data (data komt van developer portal) en hoeft er geen verdere actie ondernomen te worden. Het verschil tussen thingspark en de developer portal is te lezen in hoofdstuk 5 LoRaWAN Cloud server. Als bekend is waar de body data staat zal er worden gecontroleerd of alle variabelen die later nodig zijn aanwezig zijn in de body. Als er ergens in een controle wordt geconstateerd dat er een variabele leeg is of helemaal niet aanwezig is zal er een error bericht terug gestuurd worden. De volgende check is in Figuur 28 rook gemarkeerd omdat dit een cruciale check is. In deze check zal namelijk de token worden gecheckt. Met deze token kan er worden nagegaan of een bericht wel echt van het LoRa netwerk komt.



Figuur 27 POST sensor_data token check

In Figuur 27 is te zien hoe deze check uitgevoerd wordt. De oranje gekleurde variabelen zijn variabelen die in de body data staan. Deze worden allemaal (zonder spatie) achter elkaar geplakt. Achter deze body data variabelen komen de query parameter variabelen. Deze variabelen worden op een andere manier weergegeven namelijk met de naam ervoor met gelijkteken en een ampersand tussen de verschillende variabelen. Achter deze query parameters komt de AS_KEY. Dit is een sleutel die in de gebruiker configuratie ingevoerd moet worden. Deze moet overeenkomen met de sleutel van de LoRa sensor. Als al deze variabelen achter elkaar zijn geplakt (zonder spaties) wordt deze geencrypt door middel van SHA256. SHA256 is een encryptie methode die een 32 bytes hexadecimaal getal vormt uit de aan elkaar geplakte variabelen. Dit 32 bytes hexadecimaal getal zal de token weergeven. Deze token zal overeen moeten komen met de token die in de query parameters staat. Als deze overeenkomen is het bericht van het LoRa netwerk en om die reden een echt bericht.

Na deze check kan de payload uitgelezen worden. Hierin staat de data van het LoRa bericht. De eerste twee cijfers van de payload geven het aantal sensoren aan. De lus die daarna volgt zal even vaak worden uitgevoerd als er aantal sensoren zijn. Dit wordt er bedoelt met de vraag of het aantal sensoren gelijk is aan het aantal van i. Hierbij is i het aantal keer dat de lus doorlopen is. Zodra i wel gelijk is aan het aantal sensoren zal er een bericht worden teruggestuurd naar de verzender dat het versturen van de data is gelukt. Dit betekend een bericht met daarin de variabele error is 'false' en de variabele message met daarin 'The sensor(s) are posted'. Als i nog niet gelijk is aan het aantal sensoren zal de sensor id en sensor waarde uit de payload moeten worden gehaald (blauw omrand).

Dit is een getal van 12 cijfers lang. De eerste 8 cijfers zijn de sensor id en de laatste 4 zijn de sensor waarde. Het sensor id bevat de variabele sensor_id (eerste 4 getallen) en sensor_type (laatste 4 getallen). Dit is niet de sensor_id die gebruikt wordt bij de server maar de sensor_id die in de instellingen van de LoRa sensoren verandert kan worden. Deze sensor_id zorgt ervoor dat sensoren met dezelfde sensor_type uit elkaar gehaald kunnen worden. Een overzicht hiervan is te zien in Tabel 15:

sensor id								sensor waarde			
sensor_id (LoRa sensoren)				sensor_type				sensor_value			
1	2	3	4	5	6	7	8	9	10	11	12

Tabel 15 Payload data van één sensor

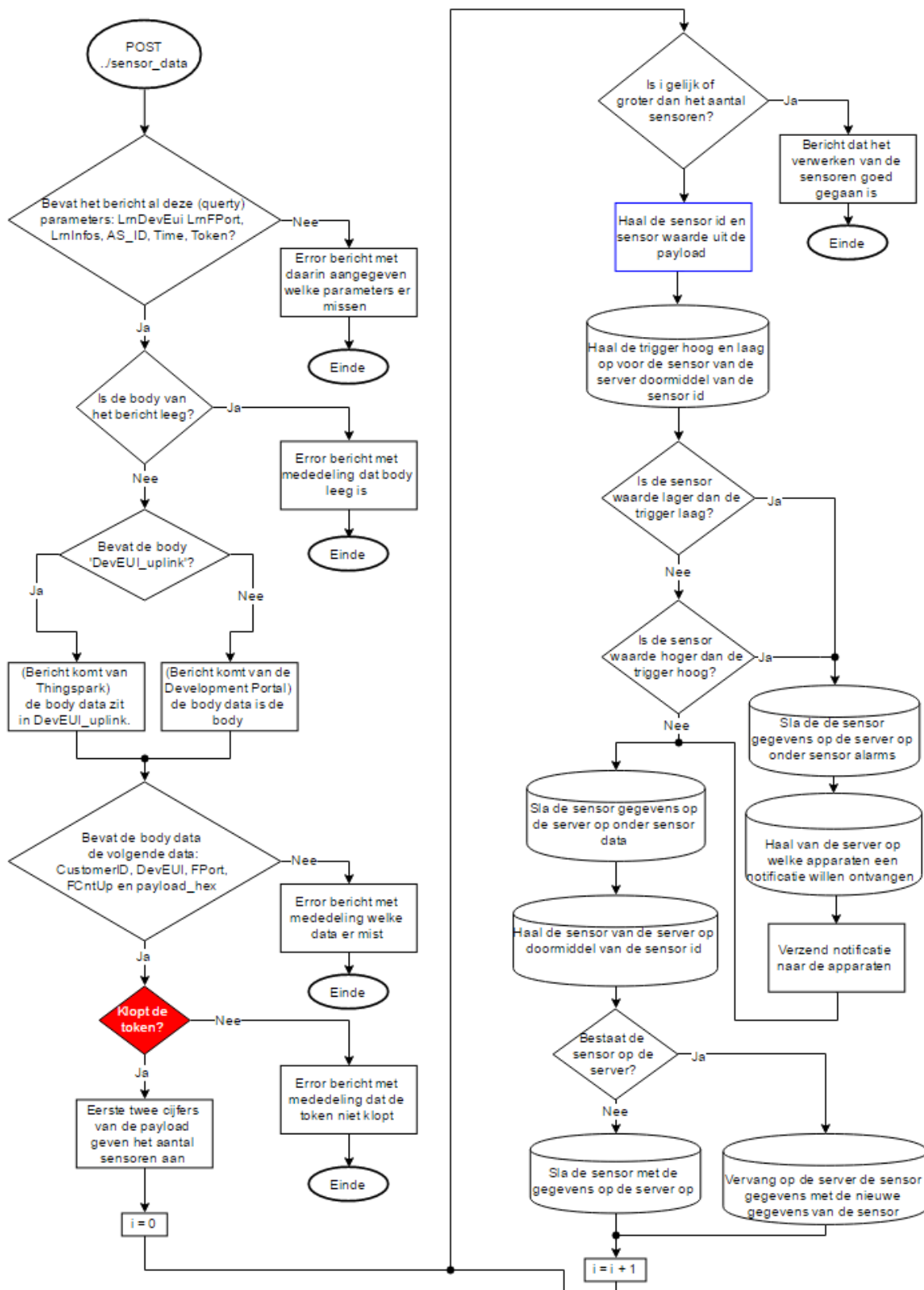
De payload begint met 2 getallen voor het aantal sensoren daarna 12 cijfers voor de sensor, daarna 12 cijfers voor de volgende sensor, (en ga zo maar door,) afhankelijk hoeveel sensoren er in de payload zijn mee gestuurd. Na het opsplitsen van de sensor id en sensor waarde wordt de db-functie (database functie uit het bestand Dbhandler.php) **getSensor_triggers** aangeroepen. Met deze functie wordt de sensor id mee gestuurd. De functie zal dan uit de sensor_triggers tabel de rij met data halen waar de sensor id overeenkomt met de meegestuurde sensor id.

De functie zal de trigger laag en trigger hoog van de sensor terugsturen. Deze twee triggers zullen vergeleken worden met de sensor waarde. Als de sensor waarde lager is dan de trigger laag of hoger is dan de trigger hoog zal de server een notificatie naar de gebruiker(s) sturen. Als de gebruiker(s) in de applicatie de notificatie functie hebben aangezet zal er in de tabel notification_tokens de token staan met daarbij of de notificaties aan of uit staan. Eerst zal de server het alarm, want dat is het als er een trigger overschreden wordt, opslaan in de tabel sensor_alarms dit wordt gedaan door de db-functie **postSensor_alarms** aan te roepen. Met deze aanroep worden de sensor id, de sensor waarde, de tijd, de trigger waarde, de trigger hoog en trigger laag meegestuurd. Nu zal de notificatie worden verzonden. Dit gaat als volgt in zijn werk, eerst wordt uit de tabel sensor_names de naam van de sensor opgehaald dit wordt gedaan door de db-functie **getSensor_name** hierbij wordt de sensor id gebruikt om de naam in de tabel op te zoeken. De titel van de notificatie wordt dan 'Alarm for ' met daarachter de sensor naam. Bestaat er geen naam voor deze sensor dan zal de titel 'LoRa Alarm' zijn. In de body van de notificatie (de zin onder de titel) zal dan de uitleg van het alarm staan.

Dit is de zin die in de gebruikersconfiguratie hoofdstuk 6.9 is ingesteld. In dat voorbeeld was het *The temperature is lower than 20 °C*. Verder worden nog het plaatje, het geluid, de tijd en de actie die het moet doen als er op de notificatie geklikt wordt ingesteld (openen alarm scherm). Samen met de token(s) die opgehaald wordt uit de db-functie **getAllNotification_tokens** zal de notificatie naar verschillende gebruikers verstuurd worden. Hierbij stuurt de server een zelf een POST request naar de Firebase Cloud Messaging [9]. Dit protocol zal aan de hand van de meegestuurde token en de data die het moet weergeven een notificatie naar het apparaat sturen waar de token aan gelinkt is. De token is namelijk uniek voor het apparaat en daardoor hoort één token bij één apparaat.

Na het verzenden zal de server de data opslaan in de sensor_data tabel. Deze zal worden gevuld door een aanroep naar de db-functie **postSensor_data**. Hier wordt dezelfde data meegestuurd als bij postSensor_alarms. Na het plaatsen van deze data zal de server kijken of de sensor al een keer data verstuurd heeft. Dit doet het door uit de tabel sensors de sensor met het sensor id te halen. Dit wordt gedaan met een aanroep naar **getSensors**. Geeft de tabel dan terug dat de sensor al bestaat op de server (in de tabel staat) zal de server de nieuwe sensor waardes (alle variabelen waar last_ voor staat zie Tabel 11 SQL-tabel Sensors) vervangen door de nieuwe waardes. Dit wordt gedaan door aanroep van de db-functie **putSensors**. Bestaat de sensor nog niet in de tabel zal de server alle variabelen in de tabel sensors (Tabel 11 SQL-tabel Sensors) vullen doormiddel van de db-functie **postSensors**. Na het update van de sensors tabel zal de i variabele opgehoogd worden en zal de lus nog een keer gestart worden. Net zo lang tot i gelijk is aan het aantal sensoren.

De requesten naar de andere functies en de andere SQL-tabellen zijn te zien in de bijlage. Zie hoofdstuk 11.3 Sensor Triggers en verder.



Figuur 28 Flowchart van een POST request naar sensor_data tabel

7 Androidapplicatie

7.1 Programma

Voor het maken van een Androidapplicatie is er een development programma nodig om de applicatie te kunnen programmeren. De officiële Android IDE (Integrated development environment) is Android Studio. In dit programma kan de software geschreven worden voor de applicatie. Deze software kan dan tot applicatie gebouwd worden waarna deze applicatie kan runnen op een Android apparaat naar keuze. Ook heeft Android Studio een debug mode, hiermee kunnen fouten in het programma makkelijk opgespoord en verbeterd worden.

Androidstudio is te downloaden op: <https://developer.android.com/studio/index.html>.

Open na het downloaden op de .exe file en de Android Studio installer zal vanzelf openen.

Let wel op, sommige computers hebben de (laatste) versie van Java SE nodig. Dit omdat de jdk die in deze Java-versie zit nodig is voor Android Studio. Deze zal als waarde ingesteld moeten worden in de omgevingsvariabele JAVA_HOME. Dit gebeurt op de volgende manier:

Ga naar Configuratiescherm\Systeem en beveiliging\Systeem, klik op Geavanceerde systeeminstellingen. Klik in de het tabblad Geavanceerd de knop Omgevingsvariabelen.... Klik nu onder Systeemvariabelen, Nieuw... en vul de volgende variabelen in:

Naam van variabele:	JAVA_HOME
Waarde van variabele:	[Pad waar jdk in staat]\[jdk versie]\nBijvoorbeeld: C:\Program Files\Java\jdk1.8.0_102\

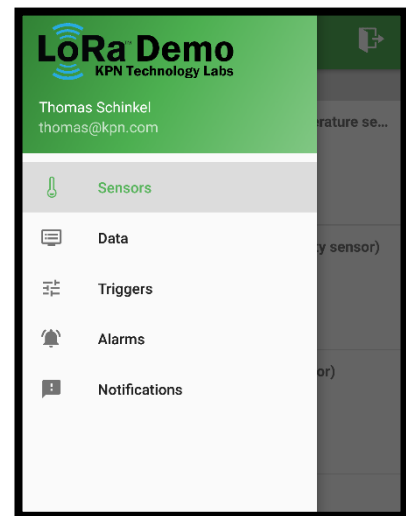
Klik na het invullen op OK, klik in het scherm Omgevingsvariabelen ook op OK en klik ook in het scherm Systeemeigenschappen op OK. Start hierna opnieuw de computer op en hierna zal Android Studio

7.2 Opbouw applicatie

7.2.1 Nieuw project

Om een applicatie te maken zal er een nieuw project gemaakt moeten worden. Dit wordt gedaan door middel van de volgende stappen:

1. Open Android Studio en klik op 'Start a new Android Studio Project' (als er al een Project open staat ga dan in de bovenste balk en klik op File -> New -> New Project...)
2. Dit is het menu om een nieuw project aan te maken ('Create New Project' menu). In het eerste scherm zal de applicatie naam, bedrijf domein en project locatie ('Application name', 'Company Domain' en 'Project location') ingevuld moeten worden. Vul deze gegevens in en klik op 'Next'.
3. In het tweede scherm wordt gevraagd waar de applicatie op gaat draaien, in dit geval is dat de telefoon. Vink daarom 'Phone and Tablet' aan. Ook wordt hier gevraagd wat de minimale SDK moet zijn (minimale API-level). Dit is de minimale softwareversie die ondersteunt moet worden door de applicatie. API 15 is bijvoorbeeld Android-versie 4.0.3, deze versie wordt door 97.4% van alle Android telefoons ondersteund. Kies hier een geschikte versie en klik weer op 'Next'.
4. In dit scherm 'Activity' worden toegevoegd aan het begin project. Voor dit project is gekozen voor de activity 'Navigation Drawer Activity'. Dit is een soort lade die tevoorschijn komt als in de applicatie van helemaal links van het scherm naar het midden wordt geschoven. Ook komt deze lade tevoorschijn als er op de 3 horizontale strepen wordt gedrukt. In deze lade kan er Zie Figuur 29 hoe dit er uiteindelijk uit zal komen te zien. Klik dus op 'Navigation Drawer Activity' en klik 'Next'. *Hoe de code van deze activity eruit ziet wordt uitgelegd in hoofdstuk 11.10 'Het frame van de applicatie'. Dit om een beetje een indruk te geven welke bestanden er nog meer nodig zijn om dit scherm er zo uit te laten zien.*
5. Dit is het laatste scherm, hierin kan de Activity naam ('Activity name'), lay-out naam ('lay-out name') en titel ('title') veranderd worden. Klik hierna op 'Finish'. Nu gaat Android Studio het nieuwe project aanmaken.

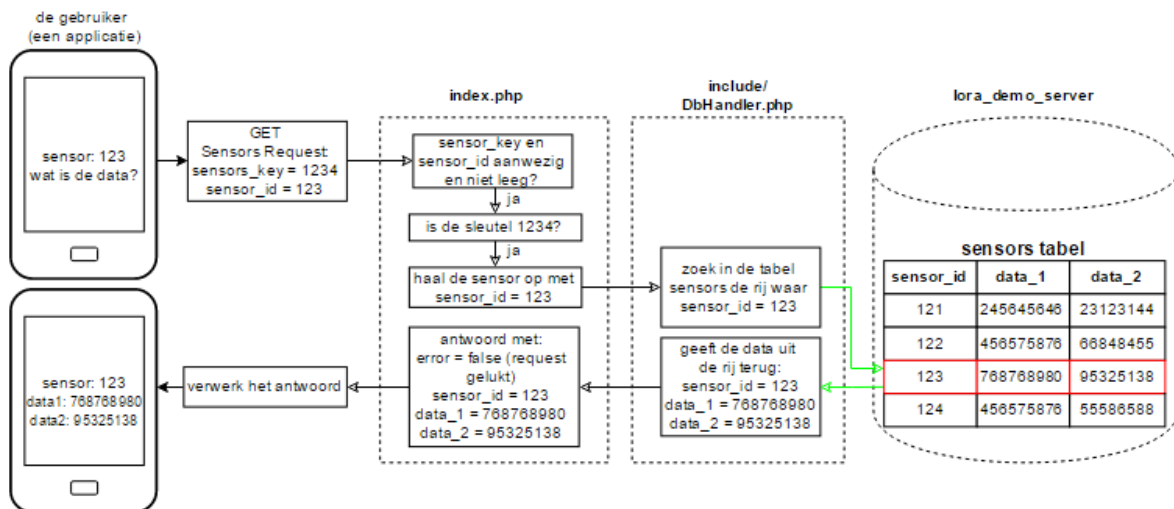


Figuur 29 Navigation Drawer

Al deze instellingen kunnen later nog gewijzigd worden.

7.3 Request

Figuur 30 is hetzelfde figuur te zien die voorkomt in het hoofdstuk server. In dit hoofdstuk is uitgelegd dat de gebruiker die in de applicatie zit de sensor data wil weten van sensor 123. Om terug te krijgen wat de sensor data is maakt de applicatie een GET request. Dit GET request wordt uiteindelijk beantwoord met de data van de sensor. Naast het GET request zijn er nog twee requesten die gebruikt worden. De POST request om data op de server te zetten en de DELETE request om data van de server te verwijderen.



Figuur 30 Server werking

In de onderstaande code staat een POST request functie:

```

1 private void request(final String param1, final String param2) {
2     // Tag used to cancel the request
3     String tag_string_req = "req";
4     String url= "https://google.com/info"+"?param1="+param1+"&param2="+param2;
5
6     StringRequest strReq = new StringRequest(Method.POST,
7         url, new Response.Listener<String>() {
8
9         @Override
10        public void onResponse(String response) {
11
12            try {
13                JSONObject jsonObj = new JSONObject(response);
14                boolean error = jsonObj.getBoolean("error");
15
16                // Check for error node in json
17                if (!error) {
18                    // Check for other nodes in json
19                    String node1 = jsonObj.getString("node1");

```

```
20         String node2 = jsonObj.getString("node2");
21     } else {
22         // Error in request. Get the error message
23         String errorMsg = jsonObj.getString("message");
24         Toast.makeText(getApplicationContext(),
25             errorMsg, Toast.LENGTH_LONG).show();
26     }
27 } catch (JSONException e) {
28     // JSON error
29     e.printStackTrace();
30     Toast.makeText(getApplicationContext(), "Json error: " +
31         e.getMessage(), Toast.LENGTH_LONG).show();
32 }
33
34 }
35 }, new Response.ErrorListener() {
36
37     @Override
38     public void onErrorResponse(VolleyError error) {
39         Log.e(TAG, "Request Error: " + error.getMessage());
40         Toast.makeText(getApplicationContext(),
41             error.getMessage(), Toast.LENGTH_LONG).show();
42     }
43 }) {
44     /*
45     @Override
46     protected Map<String, String> getParams() {
47         // Posting parameters to url
48         Map<String, String> params = new HashMap<String, String>();
49         params.put("param1", param1);
50         params.put("param2", param2);
51
52         return params;
53     }
54     */
55 };
56
57 // Adding request to request queue
58 ApplicationController.getInstance().addToRequestQueue(strReq, tag_string_req);
59 }
```

Figuur 31 POST request code

Deze functie kan aangeroepen worden doormiddel van de code: `request(param1, param2)`; Hierin zijn de beide parameters een string (een stukje tekst) met daarin de waarde van param1 en param2. Deze worden op de volgende manier aan een URL toegevoegd:

<https://google.com/info?param1=param1¶m2=param2>

Dit is een URL die ook gewoon in een browser zo gebruikt kan worden. Alleen in een browser is deze link altijd een GET request. En in de code komen ook nog de DELETE en POST request voor.

Als deze functie wordt aangeroepen zullen de twee parameters *param1* en *param2* naar de server worden verzonden. In dit voorbeeld is de server dan google.nl. Achter google.nl staat de plaats op de server die wordt aangeroepen. In dit geval staat er /info de plaats waar de request heen gaat is het *info* domein. Zo kunnen de verschillende parameters naar verschillende plekken op de server worden gestuurd. Zo kan het bijvoorbeeld zijn dat param3 en param4 naar plek *informatie* moeten dan wordt de link: <https://google.com/infomatie?param3=param3¶m3=param3> . Zo kan er een ander deel op de server google.com aangeroepen worden. Zo zal het ook werken met de requesten die in de code gedaan worden.

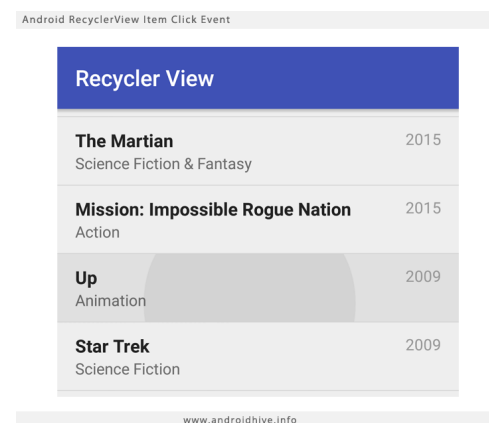
In de functie request kan op de volgende twee manieren deze parameters worden toegevoegd:

- Zoals op regel 4, hierin wordt de URL gedefinieerd, op deze manier wordt letterlijk de URL zoals hierboven gedefinieerd.
- Zoals op regel 44 tot en met 54, de code staat nu als commentaar (haal /* op regel 44 en */ op regel 54 weg om de code te gebruiken), maar als de code gebruikt wordt zullen op deze manier ook de parameters meegestuurd kunnen worden. Wel moet de URL dan verkort worden tot alleen https://google.nl/.

In de gemaakte applicatie is vaak gekozen voor manier een, omdat zo de hele URL bij elkaar staat.

7.4 Lijst met data

Om data mooi weer te geven is er gekozen voor het gebruik van lijsten. In Figuur 32 is zo een lijst te zien. Hierin is te zien dat een lijst uit verschillende stukjes bestaan die onder elkaar worden gezet en daardoor een lijst vormen. In dit voorbeeld zijn het films die worden weergegeven met titel, categorie en jaar van uitbrengen. Elk stukje kan (met een mooi effect) worden aangeklikt waarna er een actie aangehangen kan worden. In Figuur 32 wordt dit voorgedaan door op de film Up te klikken. Dit plaatje komt uit een tutorial van AndroidHive. In deze tutorial wordt stap voor stap uitgelegd hoe zo een lijst opgebouwd wordt[2]. Deze manier van een lijst opbouwen is ook in de code gebruikt. Dit omdat het een makkelijke en mooie manier van weergaven is.



Figuur 32 Voorbeeld van een lijst (foto AndroidHive.info)

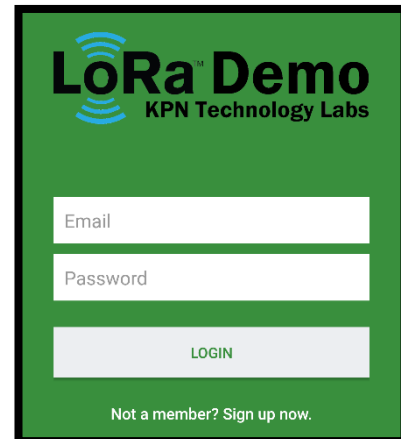
7.5 Verschillende onderdelen applicatie

De applicatie bestaat uit verschillende onderdelen ofwel ‘activities’. Elk van deze activiteiten hebben een scherm die getoond wordt zodra deze activity geopend wordt deze schermen zullen hieronder per activity behandeld worden.

7.5.1 Login (LoginActivity)

In Figuur 33 is het login scherm te zien. Dit is het eerste scherm dat verschijnt bij het voor de eerste keer installeren van de applicatie. Zodra er een keer ingelogd is, en er niet uitgelogd wordt zal de applicatie ingelogd blijven. Dit betekent dat dit scherm overgeslagen zal worden en het Sensors scherm gelijk zal openen. In het inlog scherm zal de gebruiker zijn e-mailadres en password in moeten vullen waarna het kan inloggen via de login knop. Zodra de knop wordt ingedrukt zal eerst worden gekeken de velden e-mailadres en password niet leeg zijn. Zo niet, dan wordt er gekeken of het e-mailadres en password overeenkomen met het e-mailadres en password die op de server opgeslagen staan.

Dit gaat doormiddel van een POST request naar *user_login*. Als het opgegeven e-mailadres en password overeenkomen met het e-mailadres en password in de tabel *user_login* zal de server de naam (ingevuld bij registratie) en e-mailadres als antwoord terugsturen. De naam en e-mailadres zullen opgeslagen worden dit opslaan gebeurt op het apparaat waar de applicatie op draait. Na het opslaan zal het scherm Sensors zal geopend worden en staat de applicatie als ingelogd. Als de gebruiker nog geen account heeft aangemeld kan er geklikt worden op ‘Sign up now’ waarna het Register scherm geopend wordt.

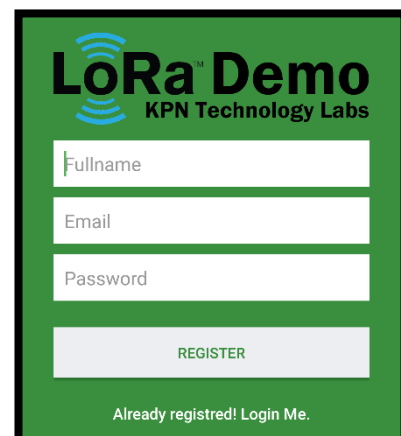


Figuur 33 Login scherm

Voor het Login en Register scherm is een code gebruikt uit de tutorial van AndroidHive[3].

7.5.2 Register (RegisterActivity)

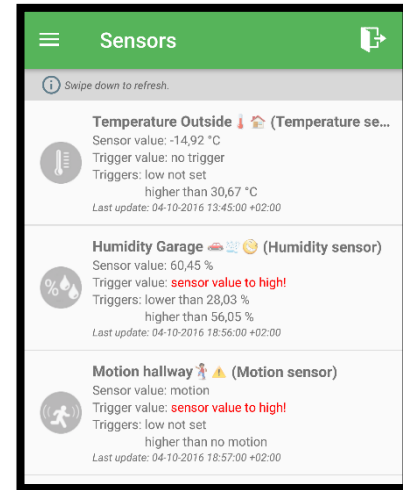
In Figuur 34 is het Register scherm te zien. Deze kan alleen benaderd worden via het Login scherm. Deze doorverwijzing vanaf het login scherm kan worden verwijderd als er wordt gekozen om alleen de al geregistreerde gebruikers toegang te geven tot de applicatie. Om te kunnen registreren zal er een naam, e-mailadres en password ingevuld moeten worden. Met het e-mailadres in combinatie met het password zal er daarna ingelogd kunnen worden. Als alle velden juist zijn ingevuld (het e-mailadres moet bijvoorbeeld een e-mailadres vorm hebben) en er wordt op de register knop geklikt maakt de applicatie een POST request naar *user_register*. Als het e-mailadres nog niet is gebruikt en de gegevens succesvol zijn opgeslagen in de tabel *user_login* zal het antwoord van de server de naam en het e-mailadres zijn. Er zal nu een tekst in het beeld verschijnen die verteld dat er ingelogd kan worden. Het inloggen kan nu door op de tekst ‘Login Me’ te klikken. Hierdoor zal het scherm terugkeren naar het Login scherm.



Figuur 34 Register scherm

7.5.3 Sensors (SensorsActivity)

In Figuur 35 is het Sensors scherm te zien. Deze zal verschijnen nadat de applicatie wordt ingelogd of als deze nog ingelogd staat. In dit scherm zijn, in een lijst, alle verschillende sensoren te zien die op de server opgeslagen staan. De server slaat een sensor op als deze ooit data heeft verstuurd naar de server (als de sensor niet verwijderd is door de gebruiker). Per sensor worden de volgende data getoond: de huidige sensor naam, deze wordt opgehaald door een GET request te sturen naar *sensor_names*, het antwoord op dit request is een array (lijst) met namen van alle sensoren gekoppeld aan de *sensor_id*. Door het als de *sensor_id* dan bekend is kan de naam weergegeven worden in het scherm. De rest van de data die weergegeven wordt is (tussen haakjes) het sensor type, de sensor waarde bij de laatste update, de trigger waarde bij de laatste update, de triggers bij de laatste update, de tijd van de laatste update en een plaatje van de desbetreffende sensor. Deze data wordt opgehaald door een GET request te doen naar *Sensors*. In dit request worden alle sensoren opgevraagd (*sensor_id = All*). Het antwoord op dit request is een array (lijst) van alle sensors met de bijbehorende data. Ook geeft de server terug hoeveel sensoren er in de tabel staan zodat de applicatie weet hoelang de lijst moet zijn. Als er op een van deze sensoren geklikt wordt zal het scherm Sensor Data geopend worden. De lijst kan vernieuwd worden door de lijst naar beneden te schuiven en los te laten (swipe down to refresh). Bij het vernieuwen zal er een niet GET request worden gedaan naar *Sensors*.



Figuur 35 Sensors scherm

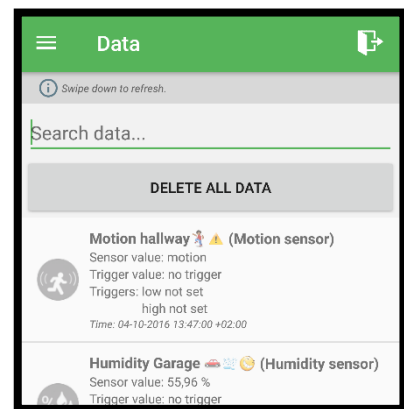
Ook kan er, net als in alle schermen na het inloggen, geklikt worden op het uitlog icoon. Als dit gebeurt zal de applicatie vragen of de gebruiker er zeker van is dat de applicatie uitgelogd moet worden.

Om uit dit scherm naar andere schermen te gaan zal het schuif menu geopend moeten worden. Dit kan doormiddel van klikken op de drie streepjes. Er kan ook gekozen worden om het scherm vanuit links het scherm in te schuiven. In dit schuifscherm kan er naar de schermen: Data, Triggers, Alarms en Notifications gegaan worden.

7.5.4 Data (DataActivity)

In Figuur 36 is het Data scherm te zien, hierin wordt alle data van alle sensoren weer gegeven. Dit weergegeven gebeurt in dezelfde vorm als in het Sensors scherm. Alleen de data ophalen gebeurt anders namelijk door een GET request naar *sensor_data* te sturen met daarin de request om alle data (*sensor_id = All*) als antwoord terug te sturen. De namen van de sensoren worden net als in het Sensors scherm opgehaald door een GET request naar *sensor_names* te sturen. Omdat het hier gaat om alle data van alle sensoren zal deze lijst een stuk groter zijn dan bij in het Sensors scherm. Daarom heeft dit scherm een zoek balk waarmee de lijst op sensor naam gefilterd zal worden zodra er iets getypt wordt in de 'Search data...' plek. Ook is er een knop om alle data te wissen. Als er op deze knop gedrukt wordt zal er een DELETE request worden gestuurd naar *sensor_data*. In dit

request zal gevraagd worden alle data (*sensor_id = all*) uit de tabel te verwijderen. Als dit gelukt is zal het antwoord een bevestiging zijn dat het verwijderen gelukt is. Nu zal het scherm zich verversen (en een nieuw GET request verstuurd worden). Dit zal resulteren dat de lijst met data nu leeg is. Het verwijderen van alle data kan bijvoorbeeld handig zijn als de sensoren worden verplaatst en de vorige data er daarom niet meer toe doet. Als deze knop ingedrukt wordt zal er nog wel om een bevestiging gevraagd worden, dit om te voorkomen dat per ongeluk alle data verwijderd wordt. Ook in dit scherm zal er bij een klik op een van de data het scherm Sensor Data geopend worden. Verder zijn de uitlog knop, het uitschuifbare menu en het vernieuwen op dezelfde manier gedaan als in het Sensors scherm.

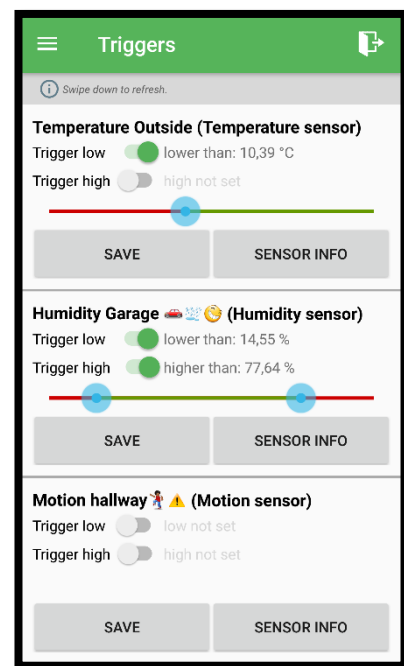


Figuur 36 Data scherm

7.5.5 Triggers (TriggersActivity)

In Figuur 37 is het Triggers scherm te zien. In dit scherm staat een lijst met alle triggers. De namen van de sensoren worden net als in het Sensors scherm opgehaald door een GET request naar *sensor_names* te sturen. De data in dit scherm wordt opgehaald door een GET request naar *sensor_triggers* met in de request de parameter *sensor_id = all* om alle triggers aan te vragen. Het antwoord van de server zal dan een lijst zijn met alle sensoren met de bijbehorende trigger data en uit hoeveel verschillende sensors deze lijst bestaat. Deze lijst bestaat uit de huidige sensor naam en sensor type. Achter 'Trigger low' en 'Trigger high' staan schuifknoppen waarmee de trigger laag en trigger hoog aan en uit gezet kunnen worden.

Als bijvoorbeeld de trigger laag aangezet is en de trigger hoog uit, zoals 'Temperature Outside' in Figuur 37, dan kan in schuifbalk eronder de trigger laag ingesteld worden. De tekst achter de schuifbalk van trigger laag geeft aan welke waarde is ingesteld op de schuifbalk. Hierbij zal het rode gebied, de waarde lager dan de trigger laag (in dit geval 10.39 °C) een trigger geven. Ook kunnen beide triggers aan gezet worden, zoals 'Humidity Garage' heeft, dan zal er een trigger gegeven worden als de sensor waarde boven of onder de trigger waardes uitkomen. Dat betekent in het rode gebied van de schuifbalk. Ook kan er natuurlijk voor gekozen worden om helemaal geen triggers in te stellen dit kan door beide triggers uit te zetten. Als een trigger waarde veranderd is door de gebruiken zal deze de nieuwe trigger waardes kunnen opslaan door op de save knop te drukken.



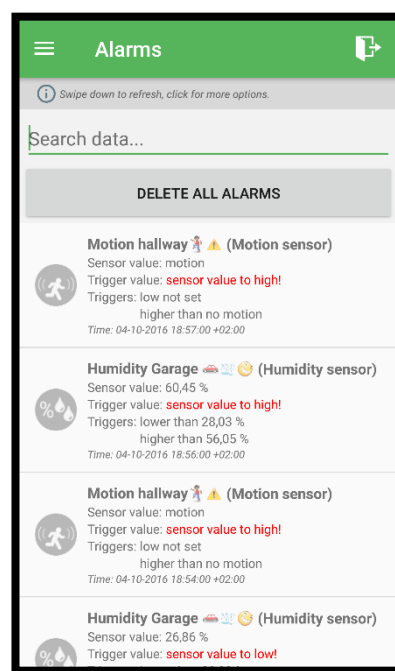
Figuur 37 Trigger scherm

De save knop zal alleen werken als er daadwerkelijk een andere trigger waarde is ingevoerd. De waarden wordt opgeslagen door een POST request te sturen naar *sensor_triggers*. In dit request zal worden aangegeven om welke sensor het gaat (*sensor_id*) en wat de nieuwe trigger waardes zijn (*trigger_low* en *trigger_high*). Na het opslaan zal het de data verversen (een nieuw GET request gedaan worden) en de nieuwe data in het scherm worden weergegeven. Om de sensor

informatie van de desbetreffende sensor te bekijken kan er op de knop 'Sensor info' gedrukt worden. Als dit gebeurt zal het scherm Sensor Data geopend worden. Verder zijn de uitlog knop, het uitschuifbare menu en het vernieuwen op dezelfde manier gedaan als in het Sensors scherm.

7.5.6 Alarms (AlarmsActivity)

In Figuur 38 is het Alarms scherm te zien. In dit scherm is een lijst weergegeven met alle alarmen. De namen van de sensoren worden net als in het Sensors scherm opgehaald door een GET request naar *sensor_names* te sturen. De data in de lijst wordt opgehaald door een GET request naar *sensor_alarms*. In dit request worden ook alle sensoren (*sensor_id* = All) gevraagd. De lijst met alarmen worden op dezelfde manier weergegeven als in het Data scherm. Dit betekent dat er ook gezocht kan worden op sensor naam en dat alle sensor alarmen met de knop 'Delete All Alarms' verwijderd kunnen worden. Voor het verwijderen van alle alarms zal er een DELETE request gedaan worden naar *sensor_alarms*. De rest van de request is gelijk aan die in het Data scherm. Het verschil ten opzichte van het scherm Data is dat in dit scherm alleen de alarmen staan, dit betekend dat de sensor data alleen in dit scherm komen als de sensor waarden hoger of lager dan de ingestelde triggers komt. Ook is het klikken op de sensor anders. Als er nu in dit scherm op een sensor geklikt wordt zal er een vraag komen of het alarm waarop geklikt is verwijderd moet worden, alle alarmen van deze sensor verwijderd moeten worden of dat er naar sensor info gegaan moet worden.



Figuur 38 Alarms scherm

Hier kunne daarom twee verschillende DELETE requesten worden verstuurd, dit licht aan welke keuze er wordt gemaakt. Bij het verwijderen van het alarm van alleen de geklikte sensor zal het DELETE request het *update_number* mee sturen waarna alleen het alarm met dat *update_number* verwijderd wordt. Dit *update_number* wordt niet weergegeven in het scherm maar staat wel in de lijst die via de GET request verkregen is. Als ervoor verwijderen van alle alarmen van de sensor gekozen wordt zal het DELETE request aan de hand van de *sensor_id* de sensor alarmen verwijderd worden. Na een van deze verwijder acties zal het scherm vernieuwd worden met de data van een nieuw GET request. Als er als optie wordt gekozen voor de sensor info zal het scherm naar Sensor Data over gaan. De reden dat in dit scherm ook een enkel alarm verwijderd kan worden is, als de gebruiker bijvoorbeeld weet dat die op dat moment zelf een trigger veroorzaakt heeft en deze wil verwijderen uit de lijst. Verder zijn de uitlog knop, het uitschuifbare menu en het vernieuwen op dezelfde manier gedaan als in het Sensors scherm.

7.5.7 Notifications (NotificationsActivity)

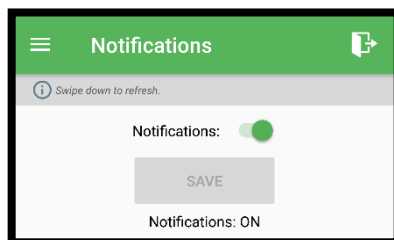
In Figuur 36 is het Notifications scherm te zien. In dit scherm kan de gebruiker ervoor kiezen om notificaties te ontvangen zodra er sensor waarde lager of hoger is dan de trigger waardes (en de sensor om die reden ook in het scherm Alarms terecht komt).

De huidige status zal worden opgehaald door een GET request te sturen met daarin de token die gegenereerd wordt met behulp

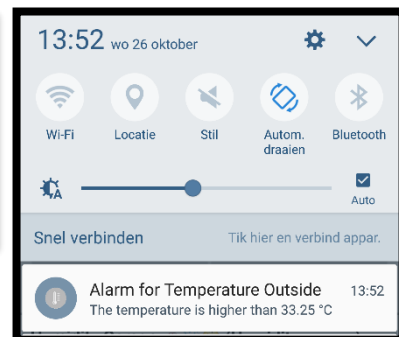
van de firebase libery(bibliotheek, zie *documentatie Firebase Cloud Messaging* [9]), in Android Studio (de token is uniek).

Na het aan of uitzetten van de notificaties (de schuif knop) zal er op de save knop gedrukt moeten worden.

Zodra dit gebeurt zal er een POST request naar *notification_tokens* verstuurd worden met in de parameter token en de parameter waarin staat of de notificaties aan of uit moeten staan. Als er een alarm ontstaat en de notificaties staat aan zal de server een notificatie sturen als de notificatie optie op aan staat. Een voorbeeld van een notificatie is te zien in Figuur 39. Zodra deze notificatie wordt aangeklikt zal het scherm Alarms geopend worden en zal de notificatie verwijderd worden. Verder zijn de uitlog knop, het uitschuifbare menu en het vernieuwen op dezelfde manier gedaan als in het Sensors scherm.



Figuur 40 Notifications scherm



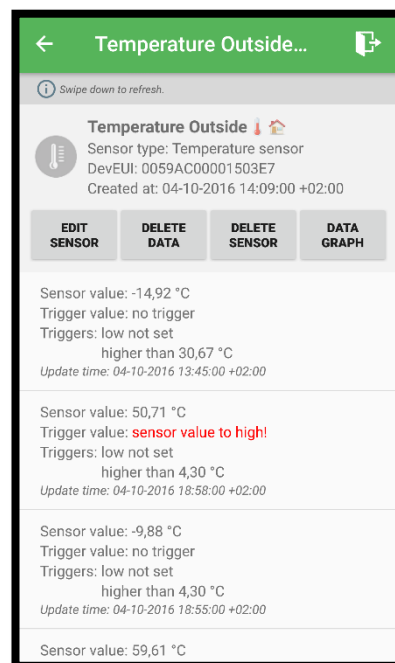
Figuur 39 Notificatie voorbeeld

7.5.8 Sensor Data (SensorsShowData)

In Figuur 41 is het Sensor Data scherm te zien. Dit scherm geeft alle informatie over een bepaalde sensor. Het scherm is alleen te bereiken via een ander scherm en daarom heeft het links bovenin een terug knop. Met deze knop wordt er teruggekeerd naar het vorige scherm. In het vorige scherm wordt een sensor aangeklikt en informatie, zoals het sensor_id, de naam, het Sensor type, de DevEUI en aanmaak tijd (Created at) doorgegeven aan dit scherm. Met sensor_id als parameter zal er een GET request gedaan worden naar *sensor_data*. Hierdoor zal de server alleen de data van de specifieke sensor teruggeven. Deze data zal worden weergegeven in een lijst onder de knoppen.

Met de knop Edit Sensor kan de naam van de sensor aangepast worden. Als de nieuwe naam opgeslagen wordt zal er een POST naar *sensor_names* gedaan worden waarna de nieuwe sensor naam opgeslagen wordt. Na het antwoord dat het opslaan gelukt is zal het scherm de sensor naam aanpassen naar de nieuwe sensor naam. Met de knop Delete Data kan de sensor data verwijderd worden. Dit wordt gedaan door een DELETE request te versturen naar *sensor_data* met als parameter sensor_id.

Na het verwijderen zal de lijst ook vernieuwd worden met een nieuw GET request en de lijst zal daarna leeg zijn omdat er geen data meer voor deze sensor is. De derde knop is Delete Sensor, deze knop zal de gehele sensor verwijderen. Dit wordt gedaan door een DELETE request te versturen naar *sensors* met als parameter sensor_id, nu zal de server alles verwijderen wat met deze sensor te maken heeft. Daarom zal er niks meer te zien zijn in dit scherm en wordt er daarom ook



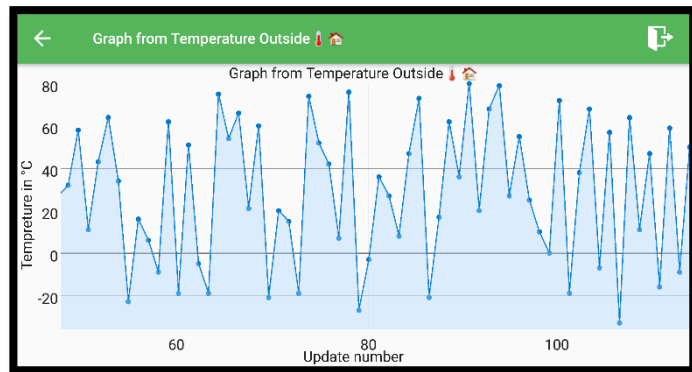
Figuur 41 Sensor Data scherm

terug gekeerd naar het vorige scherm. De laatste knop is Data Graph deze knop opent het scherm Sensor Grafiek.

Dit scherm kan net als de vorige ook de lijst met data vernieuwen. Ook via dit scherm kan er uitgelogd worden.

7.5.9 Sensor grafiek (SensorsShowGraph)

In Figuur 42 is het Sensor grafiek scherm te zien. Dit scherm geeft alle data van een bepaalde sensor weer in een grafiek. Welke sensor dit is en welke data daarbij hoort krijgt dit scherm door van het Sensor Data scherm. Hier kan ook naar teruggekeerd worden door de terug knop. De grafiek in en uit te zoomen aan de hand van de X-as (Update number as). Dit scherm is ter extra informatie over de sensor. Ook via dit scherm kan er uitgelogd worden.



Figuur 42 Sensor grafiek scherm (gedraaid)

7.5.10 Belangrijke code uitleg, gebruiker configuratie

Het bestand met de code waarin de gebruiker de configuraties kan doet staat onder (com.example.thomas.lorademo ->) App -> AppConfig. In dit bestand kan de gebruiker de volgende variabele aanpassen:

- Host naam (nu <https://lora-techlab.kpn.com/>)
- Server naam (nu [lora_demo_server/](#))
- Versie van de server (nu [v1/](#))
- De verschillende links naar de verschillende tabellen in de server met de bijbehorende sleutel naam (bijvoorbeeld [sensor_triggers?trigger_key=](#))
- De verschillende sleutels (bijvoorbeeld [ffffff-9bfc-4549-b5f3-191980a05ef6B](#) als trigger sleutel)

Deze variabele zijn ervoor om de URL's te maken. Als er bijvoorbeeld een request gemaakt moet worden naar trigger functie op de server. Dan wordt de URL alle bovenstaande instellingen achter elkaar: https://lora-techlab.kpn.com/lora_demo_server/v1/sensor_triggers?trigger_key=ffffff-9bfc-4549-b5f3-191980a05ef6B. Achter deze URL zal nog wel een (query) parameter voor de sensor_id moeten komen om de request compleet te maken. Maar de basis is altijd hetzelfde en dat is de URL van alle bovenstaande variabele achter elkaar. **LET OP! als de sleutels veranderd worden moeten de sleutels op de server ook veranderd worden, zodat ze weer overeenkomen!** Dit geldt natuurlijk ook voor de andere variabele. Want zodra de URL niet goed is zal de request niet aankomen.

Onder deze variabele komt de formule configuratie. Hier worden de instellingen van de verschillende sensoren ingevoerd deze zijn uitgewerkt in Tabel 4 Gebruikte sensoren. Als eerste komt de configuratie om de sensor types een naam te geven. Dit omdat de sensor types op de server opgeslagen staan als nummer. Maar voor de gebruiker zegt een nummer niet zoveel, vandaar dat hier een naam kan worden gegeven aan een nummer. De tweede configuratie is een plaatje aan een sensor type linken. Dit zodat bijvoorbeeld een temperatuur sensor een plaatje van een thermometer heeft. Bij de derde configuratie zal de gebruiker de formules van de verschillende sensoren ingevoerd moeten worden. Dit zodat de applicatie aan de hand van het sensor type de bijbehorende sensor waarde kan worden uitgerekend. In de volgende configuratie wordt de maximale waarde van sensor gezet (dit is maximale waarde van de data). Dit is voor de sensor grafieken zodat de y-as van de grafieken van 0 tot het maximale waarden lopen.

In de vijfde configuratie wordt de eenheid van de sensor aan de hand van het sensor type geconfigureerd. De configuratie daarna is voor de tekst voor de eenheden op de y-as deze eenheden zullen, in combinatie met de sensor waarde op de y-as komen te staan. Ook deze eenheden worden gelinkt aan het sensor type. De configuratie daaronder is voor de tekst op de y-as. Dit om aan te geven wat de data op de y-as inhoud. De configuratie daar weer onder is om aan te geven of er op de y-as eenheden moeten worden laten zien of alleen tekst. Zo kan het zijn bij bijvoorbeeld de beweging sensor dat de er alleen tekst moet staan zoals motion (beweging) of no motion (geen beweging). Deze tekst zal geconfigureerd worden bij de configuratie voor de y-as eenheden. Deze twee configuraties werken daarom nauw samen!

Zoals te zien is deze configuratie een stuk groter dan de configuratie op de server. Dit komt door de configuraties aan de grafiek hierin kan bijna alles per sensor verschillend zijn en daarom zijn er zoveel configuraties nodig.

8 Voordelen en beperkingen van LoRa

Door het gebruik van LoRa kunnen, op basis van ervaring, de volgende voordelen en beperkingen worden gesteld.

8.1 Voordelen

Een heel groot voordeel van LoRa is de 'Long Range' (lange afstand). Eén antenne (gateway) kan (in open veld) LoRa berichten ontvangen tot op 14 kilometer afstand. Deze afstand wordt natuurlijk wel minder als de LoRa sensoren binnen een gebouw staan. Dan zal het signaal verzwakt worden door de muren en andere gebouwen. Het voordeel dat er zo een lange afstand kan worden overbrugd is dat er relatief weinig antennes nodig zijn om in een groot gebied LoRa berichten te kunnen ontvangen. Hierdoor kunnen LoRa sensoren overal in Nederland berichten versturen. Dit geldt voor plekken in een gebouw waar het (gereflecteerde) daglicht te zien is.

Een ander groot voordeel is dat LoRa voor het verzenden van een bericht maar een laagvermogen ('Low Power') nodig heeft. Hierdoor is LoRa uitermate geschikt om te laten werken op batterijen. Zo kunnen er LoRa sensoren gemaakt worden die op een batterij meer dan 10 jaar mee kunnen gaan. Het voordeel hierbij is dat de LoRa sensoren overal geplaatst kunnen worden en er voor een lange tijd geen vervanging van batterijen nodig is. De techniek achter dit lage stroomverbruik is dat het signaal van het LoRa bericht wordt verspreid op een grote bandbreedte. Zo kan het bericht verzonden worden met een vermogen die lager is dan het ruis niveau.

De veiligheid is ook een voordeel van het LoRa netwerk. Zo moet de server die de berichten ontvangt een SSL-certificaat hebben. En wordt er met het bericht een token mee gestuurd. Deze token zal dan moeten worden vergeleken met een token die wordt gegenereerd op de server. De tokens zullen alleen overeenkomen als het bericht verstuurd is door het LoRa netwerk. Dit om te voorkomen dat er nep berichten door de server zal worden geaccepteerd als ware het een echt bericht is van het LoRa netwerk.

8.2 Beperkingen

Een beperking van het LoRa kan zijn dat het versturen van LoRa berichten niet onbeperkt kan. Er zijn twee factoren die een limiet geven aan het versturen van een LoRa bericht.

De eerste limiet is de tijd die de LoRa sensor nodig heeft om het bericht naar de antenne te sturen. Zo kan er het zijn dat de LoRa sensor zich ver van een antenne bevindt en/of in een gebouw waar het signaal sterk wordt afgezwakt. Als dit het geval is duurt het verzenden van het bericht langer en het signaal heeft meer vermogen nodig bij het verzenden. Wat de snelheid van het verzenden van het bericht moet zijn en wat het vermogen daarbij moet zijn wordt bepaald door de spreading factor (SF). De SF wordt hoger naarmate de afstand tot de antenne groter wordt (of het signaal door muren zwakker wordt). Zo geldt voor SF7 (minimum) dat de LoRa sensor binnen 2 kilometer van de antenne staat en voor SF12(maximum) dat de LoRa sensor binnen 14 kilometer staat. Het LoRa netwerk is zo ontworpen dat er overal in Nederland buiten een maximaal een SF van 10 geldt. Muren van gebouwen zorgen ervoor dat de SF met ongeveer 2 oploopt. Zo is er in elk gebouw waar (een reflectie van) daglicht te zien is een maximaal een SF van 12. Voor deze maximale SF van 12 geldt dat er met een snelheid van 290 bits per seconde wordt verzonden. Zo wordt de tijd van verzending

ongeveer 1600 milliseconden. Omdat er voor LoRa geldt dat LoRa sensoren maar 1% van de tijd in een uur berichten mag verzenden moet de LoRa sensor daarom, (als het constant berichten wil versturen) na één bericht met SF12 ongeveer 160000 milliseconde wachten. Dit is gelijk aan 160 seconde, ongeveer 2 minuten en 40 seconde.

De tweede limiet is het aantal berichten dat er per dag verzonden mag worden. Het verzenden van een LoRa bericht is niet gratis, om LoRa berichten te mogen verzenden zijn bepaalde abonnementen nodig. Zie 11.11 LoRa kosten voor de verschillende abonnementen, hier is te zien dat er met het grootste abonnement 300 berichten per dag verzonden kunnen worden. Dit is daarom ook een limiet waar rekening mee gehouden moet worden. Wel kan deze limiet overschreden worden, maar hier zijn dan wel extra kosten aan verbonden.

Een andere beperking is dat er in gebouwen soms folie voor de ramen wordt gebruikt om signalen te blokkeren. Dit zodat iemand die buiten het gebouw is bijvoorbeeld niet de wifi vanbinnen het gebouw kan gebruiken. Het nadeel is dat als dit folie gebruikt wordt de LoRa signalen er ook door geblokkeerd worden. Zo kunnen in zulke gebouwen geen LoRa sensoren worden gebruikt.

Ook een beperking is dat bij het ontvangen van LoRa berichten op de server de berichten moeten binnen komen op poort 443 (om het SSL-certificaat te kunnen opvragen). Zo kan deze poort niet meer worden gebruikt voor andere toepassingen. Door port forwarding zou er ook een andere poort gebruikt kunnen worden om de berichten te laten binnenkomen. Deze poort zal dan wel moeten verwijzen naar poort 443 van de server!

8.3 LoRa sensoren in Smart Buildings

Als de voordelen worden afgewogen tegen de beperkingen zal er nog steeds moeten worden gekeken naar wat de situatie binnen een gebouw is. Zo heeft het al geen zin om LoRa sensoren te gebruiken in gebouwen waar signaal blokkerend folie voor de ramen hangt. Ook kan het zijn dat het stroomverbruik geen issue is en de sensoren altijd op een vaste plek gehangen worden. Dan zijn er andere opties die gebruikt kunnen worden in plaats van LoRa sensoren. Zo zouden de sensoren bijvoorbeeld direct op een computer aangesloten kunnen worden.

Het gebruik van LoRa sensoren geeft de gebruiker vooral het gemak dat het niet afhankelijk is van de plek waar de sensoren hangen (als er maar daglicht te zien is), de sensoren weinig stroom verbruiken en de sensoren geen verdere instellingen nodig hebben.

9 Conclusie

9.1 Onderzoeksvragen

Hoe werken de verschillende sensoren en wat is de output van de sensoren?

De sensoren zijn in drie groepen te verdelen. De sensoren die alleen hoog en laag signaal (digitaal signaal) als uitgang (output) hebben. De analoge sensoren die een analoge waarde teruggeven (een ADC-waarde tussen de 0 en de 1023). En sensoren die een digitaal signaal teruggeven die de waarde van de sensor bevat. De werking van deze drie groepen is ook verschillend. Zo zal de laatste groep sensoren, die een digitaal signaal teruggeven, zullen eerst een start signaal moeten hebben voor ze de data terugsturen. De analoge sensoren zullen via een ADC-conversie uitgelezen moeten worden om achter de waarde te kunnen komen. De sensoren die alleen een hoog of laag signaal terug geven hebben geen aparte uitlees methode nodig.

Hoe kan de sensor data naar het LoRa netwerk verstuurd worden?

Dit kan doormiddel van het versturen van commando's via UART naar een RN2483 module waar een antenne op aangesloten is. Doormiddel van deze commando's worden een aantal sleutels ingesteld. De module zal aan de hand van deze sleutels zich kunnen aansluiten bij het LoRa netwerk om er berichten naar toe te kunnen sturen. Zodra het commando voor het verzenden voor een bericht verstuurd wordt, met daarin de data die verstuurd moet worden, zal de module het bericht naar het LoRa netwerk versturen. Het versturen van deze commando's wordt gedaan doormiddel van een microcontroller die de aangesloten sensoren uitleest en de data hiervan in het commando (voor het versturen van een bericht) zet.

Hoe kan de LoRaWAN Cloud ingesteld worden dat het de sensor gegevens naar de eigen server stuurt?

Hiervoor zijn twee mogelijkheden de KPN LoRa Developer portal of KPN LoRa deviceManager. Dit zijn beide websites waarop de LoRa sensor naam en de link voor de doorverwijzing naar de server ingevoerd kunnen worden. Als de LoRa Developer portal wordt gebruikt zullen er per sensor een aantal sleutels gegenereerd worden. Deze sleutels zullen worden gebruikt bij het versturen van LoRa berichten. Zo heeft de RN2483 module deze sleutels nodig om zich bij het LoRa netwerk (LoRaWAN Cloud) te kunnen aanmelden om daarna LoRa berichten te kunnen versturen. Als de KPN LoRa deviceManager gebruikt wordt moeten deze sleutels apart aangevraagd worden. Om de KPN LoRa deviceManager te kunnen gebruiken zal er een account aangevraagd worden. Ook in dit opzicht is de KPN LoRa Developer portal anders omdat iedereen hier een gratis account kan aanmaken.

Hoe kan een server zo ingericht worden dat het informatie kan ontvangen van de LoRaWAN Cloud, deze informatie kan verwerken, notificaties naar een Androidapplicatie gestuurd worden en er data kan worden uitgewisseld tussen een Androidapplicatie en de server?

De server zal requesten (aanvragen) kunnen ontvangen en verwerken. Op een request zal de server altijd een antwoord terugsturen. Het antwoord op de request hangt af van het request. Als er informatie opgeslagen moet worden in database van de server zal er een POST request gestuurd worden. Met deze POST request wordt de data mee gestuurd die in de database opgeslagen zal worden. Deze database bestaat uit verschillende tabellen waarin de verschillende data opgeslagen kan worden. Zo wordt de data uit een POST request van het LoRa netwerk (LoRaWAN Cloud) opgeslagen in de tabel sensor_data. Het verwerken van dit request gebeurt in een stuk code die aangeroepen wordt door de request. In dit voorbeeld is dat een POST naar sensor_data. In de code worden daarop weer functies aangeroepen die de data in de tabel zetten. Ook kan het zo zijn dat er

eerst data uit de tabel uitgelezen wordt waarna er met deze informatie een andere functie aangeroepen wordt. Zo haalt de server uit de tabel `notification_tokens` de token waarmee daarna een POST request wordt gedaan naar het Firebase Cloud Messaging [9]. Dit protocol zorgt ervoor dat er een notificatie wordt gestuurd naar het apparaat waar de token bij hoort.

De Androidapplicatie kan data uit de tabellen in de database op de server halen doormiddel van een GET request. Hierbij stuurt de server als antwoord op dit request de aangevraagde data. Ook kan de Androidapplicatie data in de tabellen zetten (POST request) of data uit de tabellen verwijderen (DELETE request).

Hoe kan een Androidapplicatie notificaties ontvangen, data van de server weergeven en data naar de server sturen?

De Androidapplicatie kan door middel van het versturen van requesten naar de server data ontvangen van de server. Deze data bevat het antwoord van het verstuurd request. Dit kan het opvragen van data uit een tabel op de database zijn. Dit is een GET request, hierbij zal de server als antwoord de aangevraagde data uit de aangevraagde tabel terugsturen. Ook kan de Androidapplicatie data in de tabel op de database op de server zetten. Dit gebeurt doormiddel van een POST request. In dit POST request wordt er data mee gestuurd die in een tabel gezet moet worden. Het antwoord van de server is hierbij een bevestiging dat het toevoegen in de tabel gelukt is. Ook kan de Androidapplicatie data verwijderen uit een tabel dit gebeurt doormiddel van een DELETE request.

Om de Androidapplicatie notificaties te kunnen laten ontvangen zal de firebase libery(bibliotheek) gebruikt worden. Deze bibliotheek bevat de functies op notificaties te kunnen ontvangen. Dit ontvangen gebeurt aan de hand van een token. Deze token zal doormiddel van de firebase bibliotheek gegenereerd worden. Deze token zal in de database van de server gezet moeten worden zodat de server met deze token notificaties naar de Androidapplicatie kan sturen. Dit gebeurt door een POST request te sturen naar de server om in tabel `notification_tokens` de token toe te voegen.

Wat zijn de voordelen van het LoRa netwerk?

De twee grootste voordelen van het LoRa netwerk zijn, de lange afstand die een LoRa bericht kan overbruggen tot het bij de ontvangende antenne is en het lage vermogen dat er bij dit verzenden verbruikt wordt. Zo kan een LoRa sensor, in de buitenlucht op open veld, zich maar liefst 14 kilometer van de antenne bevinden en zal het bericht nog ontvangen worden. Door het lage zendvermogen is het mogelijk om LoRa sensoren te kunnen maken die 10 jaar lang berichten kunnen verzenden om één batterij.

Een ander groot voordeel is de veiligheid van het netwerk. De server die de LoRa berichten ontvangt heeft een SSL-certificaat nodig. Ook wordt er met de berichten een token mee gestuurd waarmee het bericht op echtheid kan worden gecontroleerd.

Wat zijn de beperkingen van het LoRa netwerk?

De beperkingen die het LoRa netwerk heeft liggen vooral bij het aantal berichten dat er verstuurd kan en mag worden naar het netwerk. Zo mag de tijd dat een LoRa bericht aan het verzenden is maar 1% van de tijd in een uur zijn. Ook zijn er voor het versturen van LoRa berichten bepaalde abonnementen. Met het grootste abonnement kunnen er 300 berichten per dag verstuurd worden. Elk bericht dat er meer verzonden wordt zal extra kosten met zich meebrengen.

Hoe kunnen verschillende sensoren hun gegevens via het LoRa netwerk versturen zodat een applicatie deze gegevens kan uitlezen en notificaties (waarschuwingen) kan ontvangen als sensor gegevens een onder of boven een door de applicatie ingestelde waarde komen?

De verschillende sensoren zullen uitgelezen worden door een microcontroller. Dit uitlezen gebeurt of analoog (een ADC-waarde) of digitaal (een hoog/laag signaal of een signaal met daarin de waarde van de sensor). De microcontroller zal daarna een LoRa module (de RN2483) doormiddel van commando's initialiseren. Daarna zal de microcontroller via commando's de RN2483 de opdracht geven de uitgelezen data via een LoRa bericht naar het LoRa netwerk te sturen. Dit kan de RN2483 omdat er bij de initialisatie sleutels zijn ingevoerd door de microcontroller. Met deze sleutels kan er bij het LoRa netwerk worden aangemeld. Na deze aanmelding kan de RN2483 berichten naar het netwerk sturen.

Het netwerk weet nu, omdat het de sleutels kent, wie het LoRa bericht verstuurd heeft. Nu zal het LoRa netwerk de data doorsturen naar de server. Het LoRa netwerk weet naar welke server het de data moet sturen omdat via een website de link naar de server is ingesteld. Het doorsturen van het LoRa bericht door het LoRa netwerk zal een POST request naar de server zijn. Dit POST request, met daarin de data van de LoRa sensor, is een aanvraag om de data te kunnen opslaan in een tabel in de database van de server. In deze tabel komt dan de data en alle info van de sensor te staan.

Eén LoRa sensor kan meerdere sensoren bevatten en daardoor kan er in één POST request ook meerdere sensoren met de bijbehorende data ontvangen worden door de server. De server zal dan één voor één de data in de juiste tabellen zetten en controleren of de sensor data zich niet boven of onder een bepaalde ingestelde trigger bevindt. Als dit gebeurt zal de server een notificatie naar de Androidapplicatie moeten sturen. Dit verzenden van een Androidapplicatie kan worden gedaan door een POST request naar de 'Firebase Cloud Messaging' te sturen met daarin de token en het bericht dat de Androidapplicatie als notificatie moet ontvangen. De token geeft aan naar welk apparaat de notificatie verstuurd moet worden. Deze token is door de Androidapplicatie in een tabel op de server gezet. De server zal dus de token moeten ophalen uit de tabel zodra het de notificatie wil versturen. Zodra er meerdere tokens in deze tabel staat zal de server ook meerdere requesten sturen om notificaties te versturen.

De Androidapplicatie kan alle gegevens van de sensoren weergeven door de data uit de juiste tabellen via een GET request op te vragen van de server. Het antwoord van de server zal dan de opgevraagde data zijn. Ook kan de Androidapplicatie data uit de tabellen verwijderen (DELETE request) en data in de tabellen zetten (POST request). Zo kan in de Androidapplicatie alle sensor data bekeken worden en er bewerkingen aan worden gedaan.

9.2 Verbeteringen

Voor een verbetering van het geheel zijn er nog een aantal (kleine) punten die verbeterd kunnen worden:

- De formules voor de verschillende sensoren alleen op de server zetten. Op dit moment staan de formules van de sensoren op twee plekken in configuratie bestand van de Androidapplicatie en in de gebruikers configuratie op de server. Als er sensoren bijkomen moet er nu op twee plekken een formule worden aangepast. Dit kan efficiënter.
- Een LoRa abonnement aanschaffen en de KPN LoRa deviceManager gebruiken. Dit zodat de sleutels die voor de sensoren gebruikt worden voor de loop van het abonnement geldig blijven.

- De LoRa sensor met rookmelder zo maken zodat er een bevestiging komt dat het bericht ontvangen is. Dit is nodig omdat het voor kan komen dat een LoRa bericht de server niet bereikt om welke reden dan ook. Dan is het wel handig dat de LoRa sensor die aangesloten zit aan de rookmelder weet of het bericht is doorgekomen of dat de LoRa sensor het bericht nog eens moet sturen.
- Batterij leeg detectie, hiermee wordt bedoeld dat de LoRa sensor ook een bericht stuurt als er gedetecteerd wordt dat de batterij bijna leeg is. Dit zodat de gebruiker weet wanneer de batterij aan vervanging toe is.
- In het trigger scherm kunnen twee dingen verbeterd worden:
 - Het verversen van de gegevens kan worden gedaan door het scherm naar beneden te schuiven. Alleen deze optie staat te gevoelig, daardoor gaat omhoogschuiven van de lijst niet meer.
 - Bij twee opties (aan/uit) is een schuifbar niet de beste oplossing om de triggers in te stellen. Dit omdat er verwarring kan ontstaan over de waarde waar boven/onder er een alarm zal worden gegeven.

10 Aanbeveling

De doelstelling aan het begin van het project was het ontwikkelen, implementeren en analyseren van een Smart Building systeem bestaande uit LoRa sensoren in een technisch gebouw van KPN. Maar heeft zin om een Smart Building (een slim gebouw) uit te rusten met LoRa sensoren of het beter om hiervoor andere sensoren te gebruiken?

Het antwoord van deze vraag hangt af van de toepassing van de sensoren. Allereerst moet er bekeken worden of een gebouw wel geschikt is om LoRa sensoren in te gebruiken. Zo kan het voorkomen dat er weinig ramen in een gebouw zitten of dat er folie op de ramen zit die signalen kan blokkeren. Zo kan er een situatie ontstaan dat LoRa sensoren geen bereik hebben binnen een gebouw. In dit geval heeft het dus ook geen enkele zin om LoRa sensoren te gebruiken. Komt deze situatie voor in een gebouw moeten er andere oplossingen worden gezocht om sensoren te kunnen uitlezen, zoals bijvoorbeeld via Wifi, Zigbee of bluetooth.

Als tweede moet er nagedacht worden over de hoeveelheid berichten die een sensor moet versturen. Dit omdat het aantal berichten dat per dag verzonden kan worden een limiet heeft. Dit limiet hangt af van welk abonnement er aangeschaft is om LoRa berichten te verzenden. In het grootste abonnement kunnen er 300 berichten per dag worden verzonden. Zodra dit aantal van 300 in een dag wordt overschreden moet er per bericht bij betaald worden.

Ten derde moet er nagedacht worden over de snelheid dat de berichten achter elkaar verzonden moeten worden. Dit omdat de verzendtijd maar 1% van de tijd in een uur mag zijn. Zo kan het voorkomen dat de LoRa sensor ver van een antenne af staat en daardoor na één bericht minimaal 2 minuten en 40 seconden moet wachten. Als er toch een bericht wordt verstuurd wordt dan kan het zijn dat het bericht niet ontvangen of dat het bericht geblokkeerd wordt.

Door de limiet van de 300 berichten en de wachttijd samen te nemen kunnen er een aantal toepassingen niet gebruikt worden. Zo kunnen sensoren niet real-time (om de seconde) uitgelezen worden. Dit omdat dit meer dan 300 berichten per dag zou kosten (maar dit kan als er maar betaald wordt) maar vooral om het feit dat de LoRa sensor dan meer dan 1% van de tijd in een uur aan het verzenden kan zijn. Daardoor is real-time meten er kostbaar en daarbij is het niet zeker of alle berichten wel ontvangen worden door het LoRa netwerk en dus uiteindelijk bij de gebruiker. Zijn andere toepassingen waarbij de tijd van de metingen dicht bij elkaar liggen niet geschikt om LoRa bij te gebruiken. Dit omdat dan de limiet van 300 berichten per dag dan snel overschreden wordt en er daardoor veel buiten het abonnement om betaald moet worden.

De toepassingen waarbij LoRa uitermate geschikt zijn is bijvoorbeeld het voorbeeld uit de inleiding, waarbij ProRail de temperatuur van de wissels bijhoudt aan de hand van LoRa. Deze temperatuurmetingen zullen namelijk niet real-time hoeven maar een meting om de 10 minuten zou al een heel goed beeld geven over hoe de temperatuur in de wissels verloopt. Verder is het een groot voordeel dat overal buiten (in Nederland) bereik is met LoRa zo hoeft NS dus geen aparte ontvangers neer te zetten om de sensor waardes te kunnen ontvangen. Voor deze toepassing is LoRa dus een ideale oplossing.

Als er gekeken wordt naar sensoren binnen een gebouw en de beperkingen zoals eerder besproken geen obstakel bij het gebruik van LoRa sensoren dan heeft LoRa vooral het gemak dat het niet afhankelijk is van de plek waar de sensoren hangen (als er maar daglicht te zien is), de sensoren

weinig stroom verbruiken en de sensoren geen verdere instellingen nodig hebben. Zo kunnen sensoren bij het aanzetten direct berichten verzenden. Dit geeft de gebruiker van de LoRa sensoren het voordeel dat er geen extra werk is aan de sensoren bij het gebruik van LoRa. Het maakt dus ook geen verschil of een sensor rond het gebouw hangt of in het gebouw. Zo kunnen de sensoren dus makkelijk verplaatst worden als dit nodig is.

Het is dus belangrijk om eerst te bekijken of de beperkingen van het LoRa netwerk een obstakel zijn bij het gebruiken van de LoRa sensoren. Zodra dit niet het geval zijn LoRa sensoren ideaal voor het gebruik rond en in een gebouw.

11 Bijlage

11.1 Bronvermelding

11.1.1 Literatuur

[1]	Grit, R. (2015) <i>Projectmanagement</i> . Groningen: Noordhoff Uitgevers B.V.
-----	--

Tabel 16 Literatuur

11.1.2 Internet

[2]	Tamada, R. (2016, 3 september). Android working with RecyclerView. Geraadpleegd op 6 oktober 2016, van http://www.androidhive.info/2016/01/android-working-with-recycler-view/
[3]	Tamada, R. (2016, 20 februari). Android Login and Registration with PHP, MySQL and SQLite. Geraadpleegd op 18 oktober 2016, van http://www.androidhive.info/2012/01/android-login-and-registration-with-php-mysql-and-sqlite/
[4]	Auteur onbekend. (Datum onbekend). MSP430 interrupts and Low Power. Geraadpleegd op 15 november 2016 van http://www.argenox.com/library/msp430/msp430-interrupts-and-low-power-chapter-7/
[5]	Gsziszi. (2013, 18 juli). Antwoord op: Enabling OpenSSL in WAMP. Geraadpleegd op 1 december 2016 van http://stackoverflow.com/questions/17354788/enabling-openssl-in-wamp
[6]	Algemene vragen over Java en PHP. Geraadpleegd in oktober/november/december 2016 van http://stackoverflow.com/
[7]	Iotacademy. (Meerdere datums). Marvin + LoRaPayloadSimulator + NodeRed_KPN_LoRa. Geraadpleegd in november van https://github.com/iotacademy
[8]	Nieuwsredactie. (2016, 12 mei). ProRail aan de slag met 'internet of things'. Geraadpleegd op 11 december 2016 van https://www.rtvutrecht.nl/nieuws/1476809/prorail-aan-de-slag-met-internet-of-things.html
[9]	Auteur onbekend. (Datum onbekend). Firebase Cloud Messaging. Geraadpleegd op 2 november 2016 van https://firebase.google.com/docs/cloud-messaging/
[10]	Auteur onbekend. (Datum onbekend). Recommended Light Levels. Geraadpleegd op 2 december 2016 van https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf

[11]	Auteur onbekend. (Datum onbekend). LoRaWAN Sensors Transceiver. Geraadpleegd op 2 december 2016 van http://www.mcs-nl.com/index.php?option=com_jdownloads&Itemid=0&view=finish&catid=4&cid=1890
[12]	Auteur onbekend. (Datum onbekend). Slim Framework. Geraadpleegd op <i>niet van toepassing</i> van https://github.com/slimphp/Slim

Tabel 17 Internet

11.2 Componentenlijst met datasheets

Nummer	Naam	Soort	Datasheet
[D1]	DHT11	Temperatuur en luchtvochtigheidssensor	http://www.micropik.com/PDF/dht11.pdf
[D2]	DHT22	Temperatuur en luchtvochtigheidssensor	https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf
[D3]	ST-00082	Beweging sensor	(Informatiepagina) http://simplytronics.com/products/ST-00082
[D4]	RM144F	Rookmelder, Elro	(Datasheet van het CS2103GO-M12 component) http://pdf1.alldatasheet.com/datasheet-pdf/view/170359/FREESCALE/MC145010.html
[D5]	ND03R00104J	Temperatuursensor, 100kΩ NTC-weerstand	http://datasheets.avx.com/NTC_DiscThermistors.pdf
[D6]	Grove – Temperature V1.2	Temperatuursensor	(Informatiepagina) http://wiki.seeed.cc/Grove-Temperature_Sensor_V1.2/
[D7]	Grove – Magnetic switch	Deur- en raam sensor	(Informatiepagina) http://wiki.seeed.cc/Grove-Magnetic_Switch/
[D8]	Grove – Water sensor	Waterdetector	(Informatiepagina) http://wiki.seeed.cc/Grove-Water_Sensor/
[D9]	Grove – Light sensor V1.1	Lichtintensiteit sensor	(Informatiepagina) http://wiki.seeed.cc/Grove-Light_Sensor/
[D10]	RN2483	LoRa module	http://ww1.microchip.com/downloads/en/DeviceDoc/50002346A.pdf (Commando's document) http://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf
[D11]	MSP-EXP430G2 LaunchPad	LaunchPad om de MSP430G2553 te programmeren	http://www.ti.com/lit/ug/slau318g/slau318g.pdf
[D12]	MSP430G2553	Microcontroller MSP430G2553	http://www.ti.com/lit/ds/symlink/msp430g2453.pdf
[D13]	Marvin bord	Marvin bord met Arduino en RN2483	(Arduino datasheet) http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf
[D14]	MAX920EUK+T	Analoog comparator	http://www.farnell.com/datasheets/1143214.pdf?_ga=1.181846375.1780905501.1477391927
[D15]	78L33	Spanningsregelaar	http://www.st.com/content/ccc/resource/technical/document/datasheet/15/55/e5/aa/23/5b/43/fd/CD00000446.pdf/files/CD00000446.pdf/jcr:content/translations/en.CD00000446.pdf
[D16]	NTCLE100E3334JB0	Temperatuursensor, 100kΩ NTC-weerstand uiteindelijk niet gebruikt	http://www.vishay.com/docs/29049/ntcle100.pdf
[D17]	FU7301W	Glasbreukmelder, uiteindelijk niet gebruikt	https://www.abus.com/silver.pdf/abusproductsheet/37641/ez/dut-NL

Tabel 18 Gebruikte componenten met bijbehorende datasheet

11.3 Sensor Triggers

In de Sensor Triggers SQL-tabel staan de triggers laag en hoog van de sensoren. Elke sensor zal maar een keer in deze tabel voorkomen. Als er triggers voor de eerste keer worden opgeslagen zal de sensor in de tabel worden toegevoegd. Daarna zullen alleen de triggers laag en hoog worden aangepast zodra er voor dezelfde sensor trigger worden opgeslagen.

11.3.1 SQL-tabel Sensor Triggers

In de onderstaande tabel is de SQL-tabel van Sensor Triggers te zien. Hierin zal een bepaalde sensor_id nooit twee keer voorkomen. Dit omdat alleen de laatst ingestelde waarde van de triggers van belang is. De waarde van trigger_low en trigger_high zullen vervangen worden zodra er nieuwe waarden voor trigger_low en trigger_high binnenkomen met een overeenkomende sensor_id.

Naam	Type	Collatie	Leeg	Standaardwaarde
sensor_id	text	latin1_swedish_ci	Nee	Geen
trigger_low	text	latin1_swedish_ci	Nee	Geen
trigger_high	text	latin1_swedish_ci	Nee	Geen

Tabel 19 SQL-tabel Sensor Triggers

11.3.2 GET Sensor Triggers

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' (trigger_key)	getAllSensor_triggers : geeft alle data uit de tabel terug.	Alle data wordt in een array gezet en de grootte van deze array wordt in <i>number_of_triggers</i> gezet (dit is het aantal verschillende triggers).	Array met de variable: - sensor_id - trigger_low - trigger_high Variabele: - number_of_triggers <i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] (trigger_key)	getSensor_triggers : geeft de data uit de rij terug waar sensor_id overeenkomt met de parameter sensor_id.	<i>number_of_triggers</i> wordt op '1' gezet, dit omdat er maar één trigger wordt opgehaald.	Variabele: - sensor_id - trigger_low - trigger_high - number_of_triggers <i>error = false</i> <i>message = [message]</i>

Tabel 20 GET Sensor Triggers

11.3.3 POST Sensor Triggers

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = [sensor_id] trigger_low = [trigger_low] trigger_high = [trigger_high] (trigger_key)	getSensor_trigger : geeft de data uit de rij terug waar de sensor_id overkomt met de parameter sensor_id. putSensor_triggers : vervangt trigger_low en trigger_high door de parameters trigger_low en trigger_high in de rij waar de sensor_id overeenkomt met de parameter sensor_id. <i>Of</i> postSensor_triggers : maakt een nieuwe rij met daarin de parameters sensor_id, trigger_low en trigger_high.	Doormiddel van de db-functie getSensor_trigger wordt er gecontroleerd of de sensor_id in de tabel staat. De db-functie geeft data terug als de sensor_id in de tabel staat en is leeg ('null') als de sensor_id niet in de tabel voorkomt. Als de sensor_id wel in de tabel staat wordt de db-functie putSensor_triggers aangeroepen. Bestaat de sensor_id niet in de tabel zal postSensor_triggers aangeroepen worden.	<i>error = false</i> <i>message = [message]</i>

Tabel 21 POST Sensor Triggers

11.3.4 DELETE Sensor Triggers

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' (trigger_key)	deleteAllSensor_triggers : verwijdert alle rijen uit de tabel zodat de tabel helemaal leeg is.	<i>Geen speciale handelingen</i>	<i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] (trigger_key)	deleteSensor_triggers : verwijdert alle data aan de hand van sensor_id, alle rijen waar de sensor_id overeenkomen zullen verwijderd worden.	<i>Geen speciale handelingen</i>	<i>error = false</i> <i>message = [message]</i>

Tabel 22 DELETE Sensor Triggers

11.4 Sensor Data

11.4.1 GET Sensor Data

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' (data_key)	getAllSensor_data : geeft alle data op volgorde van de timestamp terug. Zo komt de data op volgorde van tijd te staan.	Alle data (op de timestamp na) wordt in een array gezet en de grootte van deze array wordt in <i>number_of_data</i> gezet (dit is het aantal verschillende data).	Array met de variabele: - sensor_id - sensor_DevEUI - sensor_value - time - trigger_value - trigger_low - trigger_high Variabele: - number_of_data <i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] (data_key)	getSensor_data : geeft aan de hand van de sensor_id de data uit alle rijen in de tabel terug waarin de sensor_id overeenkomt, ook dit wordt gedaan op volgorde van de timestamp.	Alle data (op de timestamp na) wordt in een array gezet en de grootte van deze array wordt in <i>number_of_data</i> gezet (dit is het aantal verschillende data).	Array met de variabele: - sensor_id - sensor_DevEUI - sensor_value - time - trigger_value - trigger_low - trigger_high Variabele: - number_of_data <i>error = false</i> <i>message = [message]</i>

Tabel 23 GET Sensor Data

11.4.2 DELETE Sensor Data

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' (data_key)	deleteAllSensor_data : verwijdert alle rijen uit de tabel zodat de tabel helemaal leeg is. deleteAllSensorsLatestData : zet in alle rijen de variabele last_value en last_trigger_value uit de tabel sensors op 'null' (leeg) dit omdat de laatste waarde verwijderd is en daardoor ook verwijderd moet worden uit de sensors tabel, daardoor heeft de laatste trigger waarde ook geen betekenis meer en wordt daarom mee verwijderd.	<i>Geen speciale handelingen</i>	<i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] (data_key)	deleteSensor_data : verwijdert alle data aan de hand van sensor_id, alle rijen waar de sensor_id overeenkomen zullen verwijderd worden. deleteSensorsLatestData : zet in de rij waar de sensor_id overeenkomt met de sensor_id in de tabel de variabele last_value en	<i>Geen speciale handelingen</i>	<i>error = false</i> <i>message = [message]</i>

	last_trigger_value uit de tabel sensors op 'null' (leeg) dit omdat de laatste waarde verwijderd is en daardoor ook verwijderd moet worden uit de sensors tabel, daardoor heeft de laatste trigger waarde ook geen betekenis meer en wordt daarom mee verwijderd.		
--	--	--	--

Tabel 24 DELETE Sensor Data

11.5 Sensor Alarms

11.5.1 SQL tabel Sensor Alarms

Naam	Type	Collatie	Leeg	Standaardwaarde	Extra	Sleutel
update_number	int(11)		Nee	Geen	AUTO_INCREMENT	Primaire Sleutel
sensor_id	text	latin1_swedish_ci	Nee	Geen		
sensor_DevEUI	text	latin1_swedish_ci	Nee	Geen		
sensor_value	text	latin1_swedish_ci	Nee	Geen		
time	text	latin1_swedish_ci	Nee	Geen		
trigger_value	text	latin1_swedish_ci	Nee	Geen		
trigger_low	text	latin1_swedish_ci	Nee	Geen		
trigger_high	text	latin1_swedish_ci	Nee	Geen		

Tabel 25 SQL tabel Sensor Alarms

11.5.2 GET Sensor Alarms

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
(alarm_key)	getAllSensor_alarms: geeft alle data uit de tabel terug.	Alle data wordt in een array gezet en de grootte van deze array wordt in <i>number_of_data</i> gezet (dit is het aantal verschillende data).	Array met de variabelen: <ul style="list-style-type: none"> - update_number - sensor_id - sensor_DevEUI - sensor_value - time - trigger_value - trigger_low - trigger_high Variabele: <ul style="list-style-type: none"> - number_of_data <i>error = false</i> <i>message = [message]</i>

Tabel 26 GET Sensor Alarms

11.5.3 DELETE Sensor Alarms

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' update_number = [update_number] (alarm_key)	deleteAllSensor_alarms: verwijdert alle rijen uit de tabel zodat de tabel helemaal leeg is.	Eerst wordt er gekeken of sensor_id geen 'null' is waarna er wordt gekeken of sensor id 'all' is. Is dit beide waar dan zal de db-functie aangeroepen worden.	<i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] update_number = [update_number] (data_key)	deleteSensor_alarms: verwijdert alle rijen waar sensor_id overeenkomt met de parameter sensor_id.	Eerst wordt er gekeken of sensor_id geen 'null' is waarna er wordt gekeken of sensor id geen 'all' is. Is dit beide waar dan zal de db-functie aangeroepen worden.	<i>error = false</i> <i>message = [message]</i>
sensor_id = 'null'	deleteSensor_alarms_update_number: verwijdert alle rijen waar update_number	Als sensor_id 'null' is zal de db-functie aangeroepen worden.	<i>error = false</i>

update_number = [update_number] (alarm_key)	overeenkomt met de parameter update_number.		message = [message]
--	---	--	---------------------

Tabel 27 DELETE Sensor Alarms

11.6 User Register

Bij de user register wordt de data opgeslagen in de tabel user_login zie Tabel 29 SQL-tabel User Login.

11.6.1 POST User Register

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
name = [name] email = [email] password = [password]	createUserByEmail1 : maakt een nieuwe rij in de tabel aan en voegt daarin name, email en password_hash. Deze password_hash wordt verkregen door de functie hash uit het bestand PassHash.php aan te roepen en de parameter password mee te sturen. In deze hash functie wordt het password geëncrypt en teruggestuurd.	Controleert met de functie validateEmail of de parameter email een geldig e-mailadres is, is het een geldig email dan wordt de db-functie createUserByEmail1 aangeroepen. Het antwoord van deze db-functie kan drie dingen zijn: Aanmaken gelukt, dan is het antwoord positief, dan zal het antwoord naar de request verstuurd het antwoord hiernaast zijn. Het antwoord kan ook zijn aanmaken mislukt of email is al in gebruik. In beide gevallen is het antwoord <i>error = true</i> met de daarbijhorende <i>message</i> .	Variabelen: - name - email <i>error = false</i> <i>message = [message]</i>

Tabel 28 POST User Register

11.7 User Login

11.7.1 SQL-tabel User Login

Naam	Type	Collatie	Leeg	Standaardwaarde	Extra	Sleutel
id	int(11)	latin1_swedish_ci	Nee	Geen	AUTO_INCREMENT	Primaire Sleutel
name	varchar(250)	latin1_swedish_ci	Nee	NULL		
email	varchar(255)	latin1_swedish_ci	Nee	Geen		
password_hash	text	latin1_swedish_ci	Nee	Geen		

Tabel 29 SQL-tabel User Login

11.7.2 POST User Login

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
email = [email] password = [password]	checkLogin1 : haalt de password_hash uit de rij waar de parameter email overeenkomt met email in de tabel. Als er een rij is waar deze overeenkomen en de password_hash is op gehaald wordt de functie check_password uit het bestand PassHash.php aangeroepen. Met deze aanroep worden de ophaalde password_hash en de parameter password meegestuurd. Hierin wordt de parameter password geëncrypt en vergeleken met de password_hash. Als deze dan overeenkomen zal de checkLogin1 een positief antwoord geven (true) en klopt het password	Geeft de checkLogin1 een positief antwoord het antwoord naar de request verstuurd het antwoord hiernaast zijn. Is het antwoord negatief zal het antwoord zijn <i>error = true</i> met de daar bijhorende <i>message</i> .	Variabelen: - name - email <i>error = false</i> <i>message = [message]</i>

	niet of bestaat email niet zal de db-functie een negatief antwoord teruggeven(false).		
--	---	--	--

Tabel 30 POST User Login

11.8 Sensor Names

11.8.1 SQL-tabel Sensor Names

Naam	Type	Collatie	Leeg	Standaardwaarde
sensor_id	text	latin1_swedish_ci	Nee	<i>Geen</i>
sensor_name	text	latin1_swedish_ci	Nee	<i>Geen</i>

Tabel 31 SQL-tabel Sensor Names

11.8.2 GET Sensor Names

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' (name_key)	getAllSensor_names : geeft alle data in de tabel terug.	Alle data wordt in een array gezet en de grootte van deze array wordt in <i>number_of_names</i> gezet (dit is het aantal verschillende namen).	Array met de variabelen: - sensor_id - sensor_name Variabele: - number_of_data <i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] (name_key)	getSensor_name : geeft de data uit de rij terug waar de sensor_id overkomt met de parameter sensor_id.	<i>number_of_names = '1'</i> omdat er maar om 1 naam gevraagd wordt.	Variabelen: - sensor_id - sensor_name - number_of_data <i>error = false</i> <i>message = [message]</i>

Tabel 32 GET Sensor Names

11.8.3 POST Sensor Names

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = [sensor_id] sensor_name = [sensor_name] (name_key)	getSensor_name : geeft de data uit de rij terug waar de sensor_id overkomt met de parameter sensor_id. putSensor_name : vervangt sensor_name door de parameter sensor_name in de rij waar de sensor_id overeenkomt met de parameter sensor_id. <i>Of</i> postSensor_name : maakt een nieuwe rij met daarin de parameters sensor_id en sensor_name.	Doormiddel van de db-functie getSensor_name wordt er gecontroleerd of de sensor_id in de tabel staat. De db-functie geeft data terug als de sensor_id in de tabel staat en is leeg ('null') als de sensor_id niet in de tabel voorkomt. Als de sensor_id wel in de tabel staat wordt de db-functie putSensor_name aangeroepen. Bestaat de sensor_id niet in de tabel zal postSensor_name aangeroepen worden.	<i>error = false</i> <i>message = [message]</i>

Tabel 33 POST Sensor Names

11.8.4 DELETE Sensor Names

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = 'All' (name_key)	deleteSensor_names : verwijdert alle rijen uit de tabel zodat de tabel helemaal leeg is.	<i>Geen speciale handelingen</i>	<i>error = false</i> <i>message = [message]</i>
sensor_id = [sensor_id] (name_key)	deleteSensor_name : verwijdert de rij uit de tabel waar sensor_id	<i>Geen speciale handelingen</i>	<i>error = false</i> <i>message = [message]</i>

	overeenkomt met de parameter sensor_id.		
--	---	--	--

Tabel 34 DELETE Sensor Names

11.9 Notification Tokens

11.9.1 SQL-tabel Notification tokens

Naam	Type	Collatie	Leeg	Standaardwaarde
token	text	latin1_swedish_ci	Nee	<i>Geen</i>
notification_on	text	latin1_swedish_ci	Nee	<i>Geen</i>

Tabel 35 SQL-tabel Notification tokens

11.9.2 GET Notification Tokens

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
token = [token] (token_key)	getNotification_token : geeft de data uit de rij terug waar de token overkomt met de parameter token.	<i>Geen speciale handelingen</i>	Variabelen: - token - notification_on <i>error = false</i> <i>message = [message]</i>

Tabel 36 GET Notification Tokens

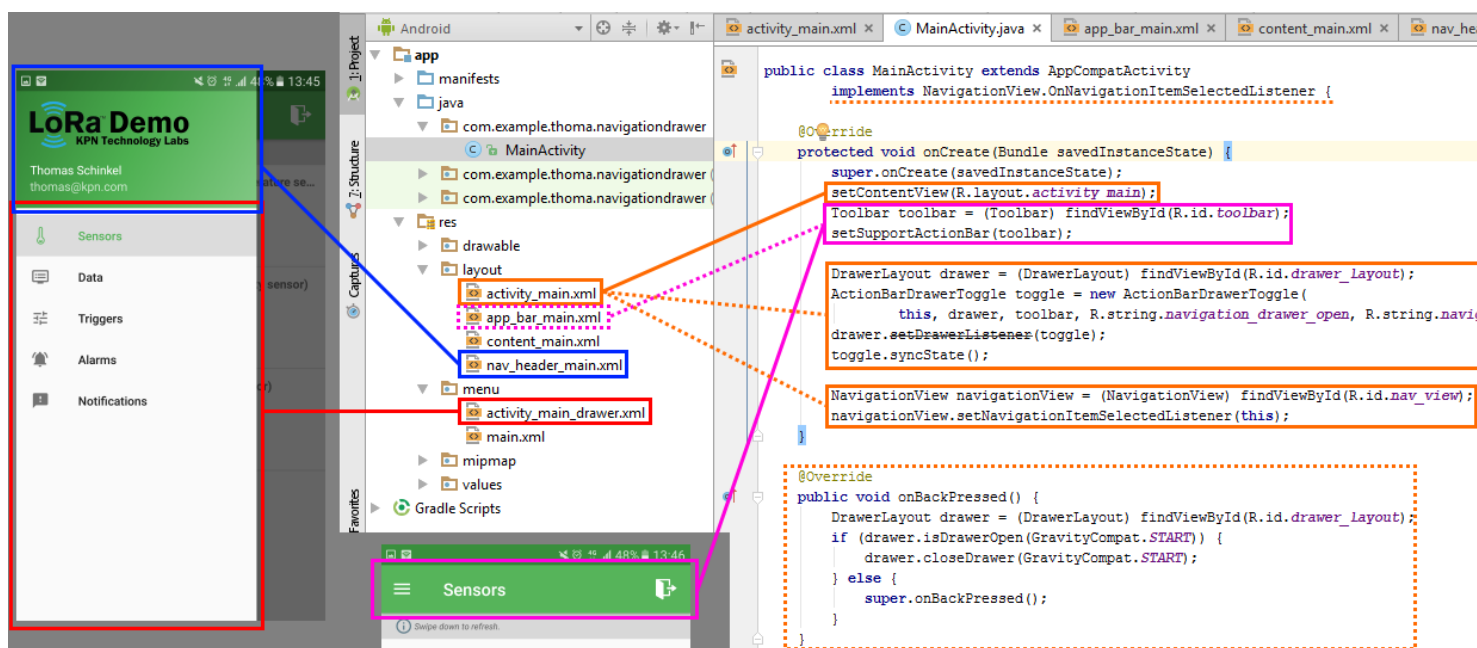
11.9.3 POST Notification Tokens

Parameters in request	Aangeroepen db-functie	Verwerking	Antwoord
sensor_id = [sensor_id] sensor_name = [sensor_name] (name_key)	getNotification_token : geeft de data uit de rij terug waar de token overkomt met de parameter token. putNotification_token vervangt notification_on door de parameter notification_on in de rij waar de token overeenkomt met de parameter token. <i>Of</i> postNotification_token : maakt een nieuwe rij met daarin de parameters sensor_id en sensor_name.	Doormiddel van de db-functie getNotification_token wordt er gecontroleerd of de token in de tabel staat. De db-functie geeft data terug als de token in de tabel staat en is leeg ('null') als de token niet in de tabel voorkomt. Als de token wel in de tabel staat wordt de db-functie putNotification_token aangeroepen. Bestaat de token niet in de tabel zal postNotification_token aangeroepen worden.	<i>error = false</i> <i>message = [message]</i>

Tabel 37 POST Notifications Tokens

11.10 Het frame van de applicatie

In het hoofdstuk 6.2.1 Nieuw project staat beschreven hoe een nieuw project wordt geopend. Als dat op dezelfde manier gedaan wordt komt de volgende MainActivity Code in beeld te staan:



Figuur 43 MainActivity na openen nieuw project

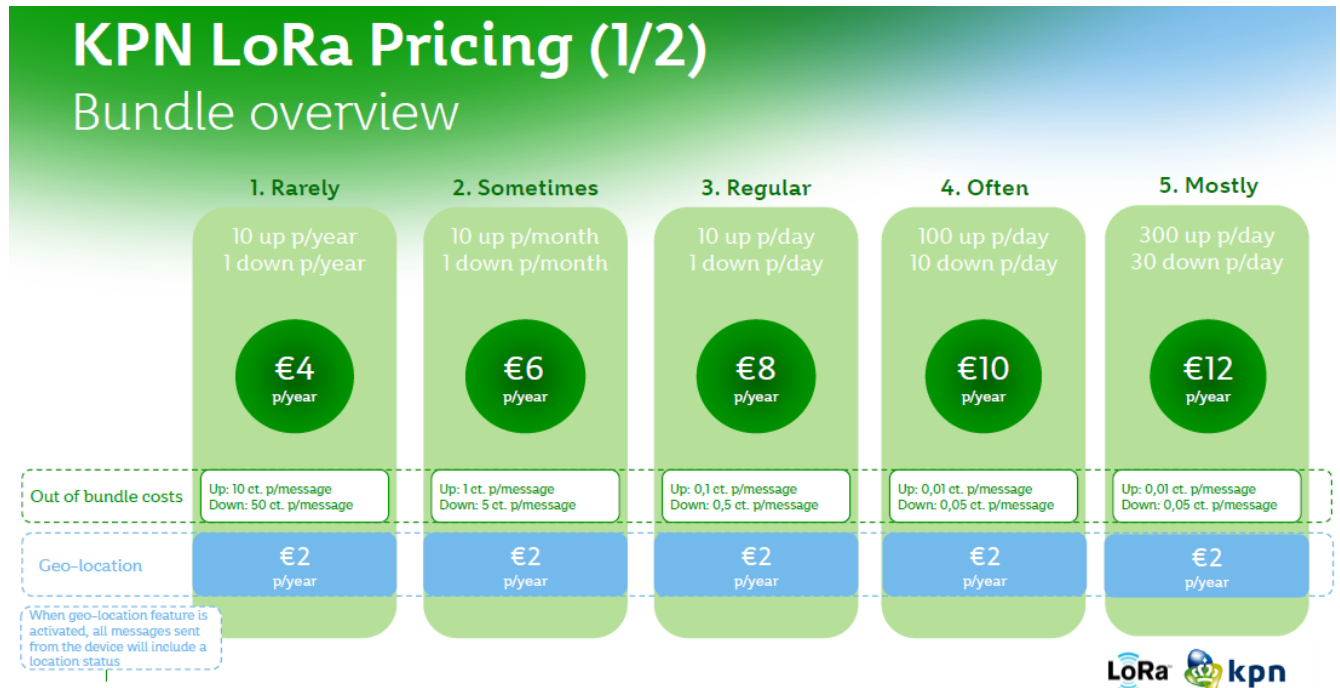
De functie 'protected void onCreate' wordt aangeroepen als de activity voor het eerst aangeroepen wordt. In deze functie worden aan aantal lay-outs aangeroepen, zo wordt als eerste (in het **oranje**) de lay-out activity_main als content gezet. Dit houdt in dat deze lay-out wordt weergegeven zodra de MainActivity aangeroepen wordt. In de file van deze activity_main worden de xml files nav_header_main (in het **blauw**) en activitiy_main_drawer (in het **rood**) samengevoegd zodat deze weergegeven worden in de applicatie zoals links te zien in Figuur 43. Dit is het schuifscherm zoals uitlegt in 6.2.1 Nieuw project. Dit schuifscherm wordt ingesteld zoals in het tweede **oranje** rechthoek te zien (DrawerLayout drawer). In de derde **oranje** rechthoek wordt ingesteld dat de code gaat bijhouden op welke menu item er wordt geklikt van de activity_main_drawer, dat is het menu weergegeven in de **rode** rechthoek, in de functie *onNavigationItemSelectedListener* wordt er vervolgens een actie aan gehangen. Verder wordt in de activity_main de xml file app_bar_main toegevoegd. Deze app_bar_main bevat de 'toolbar' ofwel de bovenste balk in het scherm (zie **roze** rechthoek).

Verder bevat de MainActivity nog de volgende functies:

- *onBackPressed* (**oranje** stippelrechthoek), deze functie wordt aangeroepen na het indrukken van de terug knop op de Android telefoon. Zodra deze functie wordt aangeroepen wordt er gekeken of het schuifmenu nog open staat, zo ja dan wordt deze dicht geschoven. Als het schuif menu al dicht is wordt gaat de activity naar de vorige activity, als deze er niet is (omdat het de eerste activity is of de vorige activity afgesloten is, het inlog scherm doet dit bijvoorbeeld zodat hier niet naar teruggekeerd kan worden door de terug knop) zal de applicatie worden afgesloten.

- *onCreateOptionsMenu*, ook deze functie wordt aangeroepen als de activity voor het eerst aangeroepen wordt. Deze functie zorgt ervoor dat er menu items in de bovenste balk gezet kunnen worden. In het geval van de LoRaDemo is dat de uitlog knop (zie **roze** rechthoek). Deze uitlog knop wordt in de main.xml toegevoegd.
- *onOptionsItemSelected*, deze functie zorgt ervoor dat als er op de uitlog knop gedrukt wordt dat er dan iets gebeurt, in dit geval wordt er een bevestiging gevraagd of het uitloggen echt moet gebeuren, waarna er bij een bevestiging wordt uitgelogd.
- *onNavigationItemSelected*, deze functie zorgt ervoor dat als er bijvoorbeeld op het menu item 'Data' (zie **rode** rechthoek) wordt gedrukt, dat de DataActivity dan geopend wordt.

11.11 LoRa kosten



Figuur 44 LoRa kosten

11.12 httpd.conf

In de onderstaande code staan alleen de stukken waarin code veranderd moet worden:

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 0.0.0.0:1443
Listen [::0]:1443

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule allowmethods_module modules/mod_allowmethods.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule auth_form_module modules/mod_auth_form.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
LoadModule authn_core_module modules/mod_authn_core.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authn_socache_module modules/mod_authn_socache.so
#LoadModule authnz_fcgi_module modules/mod_authnz_fcgi.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
LoadModule authz_core_module modules/mod_authz_core.so
#LoadModule authz_dbd_module modules/mod_authz_dbd.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
#LoadModule authz_owner_module modules/mod_authz_owner.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule buffer_module modules/mod_buffer.so
LoadModule cache_module modules/mod_cache.so
LoadModule cache_disk_module modules/mod_cache_disk.so
#LoadModule cache_socache_module modules/mod_cache_socache.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule charset_lite_module modules/mod_charset_lite.so
```

```
#LoadModule data_module modules/mod_data.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule dav_lock_module modules/mod_dav_lock.so
#LoadModule dbd_module modules/mod_dbd.so
#LoadModule deflate_module modules/mod_deflate.so
LoadModule dir_module modules/mod_dir.so
#LoadModule dumpio_module modules/mod_dumpio.so
LoadModule env_module modules/mod_env.so
#LoadModule expires_module modules/mod_expires.so
#LoadModule ext_filter_module modules/mod_ext_filter.so
LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule filter_module modules/mod_filter.so
#LoadModule http2_module modules/mod_http2.so
#LoadModule headers_module modules/mod_headers.so
#LoadModule heartbeat_module modules/mod_heartbeat.so
#LoadModule heartmonitor_module modules/mod_heartmonitor.so
#LoadModule ident_module modules/mod_ident.so
#LoadModule imagemap_module modules/mod_imagemap.so
LoadModule include_module modules/mod_include.so
#LoadModule info_module modules/mod_info.so
LoadModule isapi_module modules/mod_isapi.so
#LoadModule lbmethod_bybusyness_module modules/mod_lbmethod_bybusyness.so
#LoadModule lbmethod_byrequests_module modules/mod_lbmethod_byrequests.so
#LoadModule lbmethod_bytraffic_module modules/mod_lbmethod_bytraffic.so
#LoadModule lbmethod_heartbeat_module modules/mod_lbmethod_heartbeat.so
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule logio_module modules/mod_logio.so
LoadModule log_config_module modules/mod_log_config.so
#LoadModule log_debug_module modules/mod_log_debug.so
#LoadModule log_forensic_module modules/mod_log_forensic.so
#LoadModule lua_module modules/mod_lua.so
#LoadModule macro_module modules/mod_macro.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule negotiation_module modules/mod_negotiation.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_express_module modules/mod_proxy_express.so
#LoadModule proxy_fcgi_module modules/mod_proxy_fcgi.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_html_module modules/mod_proxy_html.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_http2_module modules/mod_proxy_http2.so
#LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
#LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
#LoadModule ratelimit_module modules/mod_ratelimit.so
#LoadModule reflector_module modules/mod_reflector.so
#LoadModule remoteip_module modules/mod_remoteip.so
#LoadModule request_module modules/mod_request.so
#LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule sed_module modules/mod_sed.so
#LoadModule session_module modules/mod_session.so
#LoadModule session_cookie_module modules/mod_session_cookie.so
#LoadModule session_crypto_module modules/mod_session_crypto.so
```

```
#LoadModule session_dbd_module modules/mod_session_dbd.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule slotmem_plain_module modules/mod_slotmem_plain.so
#LoadModule slotmem_shm_module modules/mod_slotmem_shm.so
#LoadModule socache_dbm_module modules/mod_socache_dbm.so
#LoadModule socache_memcache_module modules/mod_socache_memcache.so
##### FOR SSL config:#####
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
####COMMENT IT IF IT IS NOT NESSESARY##
#LoadModule speling_module modules/mod_speling.so
##### FOR SSL config:#####
LoadModule ssl_module modules/mod_ssl.so
####COMMENT IT IF IT IS NOT NESSESARY##
#LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule watchdog_module modules/mod_watchdog.so
#LoadModule xml2enc_module modules/mod_xml2enc.so

LoadModule php5_module "${INSTALL_DIR}/bin/php/php5.6.25/php5apache2_4.dll"

# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName localhost:80
ServerName 172.17.0.73:1443

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride all
    Require all granted
</Directory>

###Options +Indexes +FollowSymLinks +Multiviews

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   AllowOverride FileInfo AuthConfig Limit
#
AllowOverride all
Options all

#
# Controls who can get stuff from this server.
```



```
#

# onlineoffline tag - don't remove
Require all granted
</Directory>

#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ".ht*">
    Require all denied
</Files>

#

#
# "c:/Apache24/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "${INSTALL_DIR}/cgi-bin">
    AllowOverride all
    Options all
    Require all granted
</Directory>

# Virtual hosts
Include conf/extra/httpd-vhosts.conf

# Secure (SSL/TLS) connections
##### FOR SSL config:#####
Include conf/extra/httpd-ssl.conf
####COMMENT IT IF IT IS NOT NESSESARY###
#
# Note: The following must must be present to support
#   starting without SSL on platforms with no /dev/random equivalent
#   but a statically compiled-in mod_ssl.
#
```

11.13 http.v-host.conf

```
<VirtualHost *:1443>
```

```
    ServerName localhost:1443
    DocumentRoot c:/wamp64/www
```

```
</VirtualHost>
```

11.14 Volledige meting voor Kalibratie ND03R00104J

DHT22	NTC		DHT22	NTC		DHT22	NTC		DHT22	NTC		DHT22	NTC		DHT22	NTC
19.00	19.65		9.60	10.89		16.70	16.86		15.80	11.38		14.60	19.27		22.20	21.75
19.00	19.58		9.60	10.73		16.70	16.86		15.60	11.14		14.70	18.96		22.20	21.75

19.00	19.58		9.60	10.56		16.70	16.86		15.30	11.22		14.70	18.80		22.20	21.67
19.00	19.65		9.60	10.48		16.70	16.86		15.10	11.30		14.80	18.72		22.20	21.67
19.00	19.65		9.60	10.48		---	16.86		14.90	11.38		14.80	18.49		22.20	21.67
19.10	19.65		9.60	10.40		16.70	16.86		14.70	11.22		14.90	18.33		22.20	21.75
19.10	19.58		9.70	10.40		16.60	16.93		14.60	11.38		14.90	18.18		22.20	21.59
19.10	19.65		9.60	10.32		16.70	16.86		14.40	11.22		15.00	18.18		22.20	21.59
19.20	19.58		9.70	10.32		16.70	16.70		14.20	11.38		15.00	18.02		22.20	22.30
19.20	19.58		9.60	10.24		16.70	16.78		14.00	11.22		15.00	17.87		22.20	21.75
19.20	19.58		9.60	10.24		16.70	16.70		13.80	10.97		15.10	17.79		22.20	21.91
19.20	19.58		9.70	10.32		16.70	16.70		13.60	10.97		15.10	17.71		22.20	21.91
19.20	19.58		9.60	10.07		16.70	16.62		13.40	11.06		15.10	17.63		22.20	21.91
19.20	19.58		9.60	10.15		16.70	16.70		13.30	11.06		15.20	17.63		22.20	21.91
19.30	19.58		9.60	10.15		16.70	16.78		13.20	11.06		15.20	17.56		22.20	21.91
19.30	19.65		9.60	10.15		16.70	16.78		13.00	11.22		15.30	17.63		22.20	21.91
19.30	19.65		9.60	10.15		16.70	16.78		12.90	11.14		15.20	17.48		22.20	21.83
19.30	19.73		9.60	10.07		16.70	16.78		12.70	11.14		15.30	17.48		22.20	21.91
19.30	19.96		9.60	10.15		16.70	16.78		12.70	11.30		15.30	17.48		22.20	21.83
19.40	19.96		9.60	10.15		16.70	16.70		12.60	11.06		15.30	17.48		22.20	21.83
19.40	19.96		9.60	10.07		16.70	16.78		12.50	11.14		15.30	17.40		22.20	21.75
19.40	19.89		9.60	10.07		16.70	16.86		12.30	10.97		15.30	17.40		22.20	21.83
19.40	19.81		9.60	10.07		---	16.78		12.30	10.97		15.40	17.40		22.20	21.75
19.40	19.81		9.60	10.07		16.70	16.78		12.10	11.38		15.40	17.40		22.20	21.75
19.50	19.73		9.60	10.07		16.70	16.70		12.10	11.14		15.40	17.32		22.20	21.75
19.40	19.73		9.60	10.07		16.70	16.86		12.00	11.30		15.40	17.32		22.20	21.67
19.50	19.73		9.50	10.07		16.70	16.86		11.90	11.46		15.50	17.40		22.20	21.67
19.50	19.73		9.50	10.07		16.70	16.93		11.90	11.14		15.40	17.40		22.10	21.75
19.50	19.73		9.50	10.07		16.70	16.93		11.80	10.97		15.40	17.40		22.10	21.75
19.50	19.73		9.50	9.99		16.70	16.78		11.70	11.06		15.50	17.32		22.10	21.75
19.50	19.73		9.50	9.90		16.70	16.78		11.70	10.89		15.50	17.32		22.10	21.75
19.50	19.73		9.50	9.99		16.70	16.78		11.60	10.89		15.50	17.32		22.10	21.75
19.50	19.65		9.50	9.90		16.70	16.86		11.50	11.22		15.50	17.32		22.10	21.59
19.60	19.65		9.50	9.99		16.70	16.86		11.50	11.14		15.60	17.32		22.10	21.67
19.60	19.65		9.50	9.90		16.70	16.78		11.40	10.97		15.60	17.40		22.10	21.67
19.60	19.65		9.50	9.99		16.70	16.78		11.40	10.97		15.60	17.32		22.10	21.67
19.60	19.65		9.50	9.99		16.70	16.78		11.30	10.89		15.60	17.32		22.10	21.67
19.60	19.65		9.50	9.99		16.70	16.70		11.30	10.97		15.60	17.32		22.00	21.59
19.60	19.65		9.50	9.99		16.70	16.78		11.30	11.14		15.70	17.32		22.10	21.67
19.60	19.65		9.50	9.99		16.70	16.78		11.20	11.14		15.70	17.32		22.10	21.59
19.60	19.65		9.50	9.90		16.70	16.78		11.10	10.97		15.70	17.32		22.00	21.59
19.60	19.65		9.40	9.90		16.70	16.70		11.10	10.89		15.70	17.32		22.00	21.59
19.60	19.73		9.50	9.99		16.70	16.78		11.10	10.89		15.70	17.32		22.00	21.59

19.70	19.81		9.50	10.24		16.70	16.70		11.10	10.89		15.80	17.40		22.00	21.52
19.60	19.89		9.50	10.97		16.70	16.78		11.00	11.22		15.80	17.40		22.00	21.52
19.70	19.81		9.50	10.65		16.70	16.70		11.00	10.97		15.80	17.40		22.00	21.52
19.70	19.73		9.50	10.97		16.70	16.78		11.00	10.97		15.90	17.40		22.00	21.52
19.70	19.73		9.50	11.14		16.70	16.78		10.90	10.89		16.00	17.40		22.00	21.52
19.70	19.73		9.50	11.30		16.70	16.78		10.90	10.89		16.00	17.40		22.00	21.52
19.70	19.73		9.60	11.38		16.70	16.70		10.90	10.81		16.00	17.32		22.00	21.44
19.70	19.73		9.60	11.14		16.70	16.70		10.90	11.14		16.00	17.40		22.00	21.52
19.70	19.73		9.60	10.97		16.70	16.70		10.80	10.97		16.10	17.40		22.00	21.52
19.70	19.73		9.60	10.89		16.70	16.70		10.80	11.14		16.10	17.40		22.00	21.59
19.70	19.73		9.60	10.81		16.70	16.78		10.80	11.06		16.10	17.40		22.00	21.52
19.70	19.73		9.60	10.65		16.70	16.70		10.80	11.06		16.20	17.40		22.00	21.59
19.70	19.73		9.60	10.56		16.70	16.70		10.80	11.30		16.20	17.40		21.90	21.59
19.80	19.73		9.60	10.56		16.70	16.78		10.80	11.14		16.20	17.40		22.00	21.52
19.80	19.73		9.60	10.56		16.70	16.78		10.80	11.06		16.30	17.40		22.00	21.59
19.80	19.81		9.60	10.32		16.70	16.78		10.80	11.06		16.30	17.40		22.00	21.52
19.80	19.73		9.60	10.24		16.60	16.70		10.70	10.97		16.30	17.32		22.00	21.52
19.80	19.73		9.60	10.24		16.70	16.70		10.70	10.97		16.40	17.32		22.00	21.59
19.80	19.73		9.60	10.15		16.70	16.70		10.70	11.06		16.40	17.32		22.00	21.52
19.80	19.73		9.60	10.07		16.70	16.70		10.70	10.89		16.40	17.40		22.00	21.52
19.80	19.73		9.60	10.07		16.60	16.78		10.70	10.89		16.40	17.40		21.90	21.52
19.80	19.73		9.60	9.99		16.70	16.78		10.70	10.97		16.50	17.40		21.90	21.59
19.80	19.73		9.60	9.99		16.70	16.78		10.70	10.89		16.50	17.40		21.90	21.52
19.80	19.73		9.50	9.99		16.70	16.86		10.70	10.89		16.50	17.40		21.90	21.52
19.80	19.73		9.50	9.90		---	16.86		10.60	10.89		16.50	17.40		21.90	21.44
19.80	19.73		9.50	9.90		16.60	16.86		10.60	10.89		16.60	17.48		21.90	21.44
19.80	19.73		9.50	9.90		16.70	16.78		10.60	10.89		16.60	17.48		21.90	21.44
19.80	19.73		9.50	9.82		16.70	16.70		10.60	10.89		16.60	17.56		21.90	21.44
19.80	19.73		9.50	9.74		16.60	16.78		10.60	11.22		16.60	17.40		21.90	21.44
19.80	19.73		9.50	9.74		16.60	16.86		10.60	11.06		16.60	17.40		21.90	21.44
19.80	19.73		9.50	9.74		16.70	16.78		10.60	10.97		16.70	17.48		21.90	21.44
19.80	19.73		9.50	10.07		16.70	16.86		10.60	10.89		16.70	17.48		21.90	21.44
19.80	19.73		9.40	10.15		16.70	16.86		10.60	10.89		16.70	17.40		21.90	21.44
19.80	19.73		9.40	10.24		16.70	16.78		10.60	10.89		16.70	17.40		21.90	21.44
19.90	19.73		9.40	10.24		16.70	16.93		10.60	10.97		16.70	17.40		21.90	21.44
19.90	19.73		9.40	10.15		16.70	16.93		10.60	10.97		16.70	17.40		21.90	21.44
19.90	19.73		9.40	10.07		16.70	16.86		10.50	10.89		16.70	17.48		21.90	21.44
19.90	19.73		9.40	10.40		16.70	16.93		10.60	11.14		16.80	17.40		21.90	21.44
19.90	19.73		9.40	10.24		16.70	16.86		10.60	10.97		16.80	17.40		21.90	21.44
19.90	19.73		9.40	10.07		16.70	16.78		10.60	10.97		16.80	17.40		21.90	21.52
19.90	19.73		9.40	9.99		16.70	16.78		10.60	10.97		16.80	17.40		21.90	21.44

19.90	19.73		9.40	9.90		16.70	16.78		10.60	10.89		16.80	17.48		21.90	21.44
19.90	19.73		9.40	9.90		16.70	16.70		10.50	10.89		16.80	17.40		21.90	21.44
19.90	19.73		9.40	9.82		16.70	16.70		10.50	10.97		16.80	17.48		21.90	21.44
19.90	19.73		9.40	9.82		16.70	16.78		10.50	10.97		16.90	17.40		21.90	21.44
19.90	19.73		9.30	9.66		16.70	16.78		10.50	10.89		16.90	17.40		21.90	21.44
19.90	19.73		9.40	9.66		16.70	16.78		10.60	10.81		16.90	17.48		21.90	21.44
19.90	19.73		9.30	9.57		16.70	16.70		10.50	11.06		16.90	17.48		21.90	21.44
19.90	19.73		9.40	9.57		16.70	16.78		10.50	11.22		16.90	17.48		21.90	21.44
19.90	19.73		9.30	9.57		16.70	16.78		10.60	11.06		16.90	17.48		21.90	21.44
19.90	19.73		9.30	9.82		16.70	16.70		10.60	11.14		16.90	17.48			
19.90	19.73		9.30	9.57		16.70	16.70		10.50	11.14		17.00	17.48			
19.90	19.81		9.30	9.49		16.70	16.70		10.60	11.06		16.90	17.48			
19.90	19.73		9.30	9.49		16.70	16.86		10.60	10.97		16.90	17.40			
19.90	19.73		9.30	9.49		16.70	16.86		10.50	11.62		17.00	17.40			
19.90	19.81		9.30	9.57		16.70	16.78		10.60	11.22		17.00	17.48			
19.90	19.73		9.30	9.49		16.70	16.86		10.60	10.97		17.00	17.48			
19.90	19.73		9.30	9.49		16.70	16.78		10.60	11.30		17.00	17.48			
19.90	19.81		9.30	9.41		16.70	16.78		10.60	11.06		17.00	17.40			
20.00	19.81		9.20	9.32		16.70	16.86		10.60	10.97		17.00	17.40			
19.90	19.81		9.20	9.41		16.70	16.86		10.60	10.97		17.00	17.40			
19.90	19.81		9.20	9.49		16.70	16.86		10.60	10.97		17.10	17.48			
19.90	19.89		9.20	9.49		16.70	16.86		10.60	10.89		17.10	17.48			
19.90	19.81		9.20	9.49		16.70	16.93		10.60	11.06		17.10	17.48			
20.00	19.81		9.20	9.41		16.70	16.86		10.60	10.97		17.10	17.48			
20.00	19.81		9.20	9.32		16.70	16.86		10.60	10.89		17.10	17.48			
20.00	19.81		9.20	9.32		16.70	16.93		10.50	10.97		17.10	17.48			
20.00	19.73		9.20	9.32		16.70	16.86		10.60	11.46		17.10	17.40			
20.00	19.73		9.20	9.41		16.70	16.86		10.60	11.30		17.10	17.40			
20.00	19.81		9.20	9.41		16.70	16.86		10.60	11.06		17.10	17.56			
20.00	19.81		9.10	9.32		16.70	16.78		10.60	11.46		17.10	17.48			
20.00	19.81		9.10	9.32		16.70	16.86		10.60	11.38		17.20	17.40			
20.00	19.81		9.10	9.32		16.70	16.86		10.60	11.38		17.20	17.48			
20.00	19.81		9.10	9.24		16.70	16.93		10.60	11.30		17.20	17.48			
20.00	19.81		9.10	9.24		16.70	16.86		10.60	11.14		17.20	17.48			
20.00	19.81		9.10	9.24		16.70	16.78		10.60	11.14		17.20	17.56			
20.00	19.81		9.10	9.32		16.70	16.86		10.60	10.97		17.20	17.48			
20.00	19.73		9.10	9.32		16.70	16.86		10.60	10.97		17.20	17.48			
20.00	19.73		9.10	9.32		16.70	16.93		10.60	11.30		17.20	17.56			
20.00	19.81		9.10	9.24		16.70	16.86		10.60	11.79		17.30	17.48			
20.00	19.89		9.10	9.24		16.70	16.78		10.70	11.38		17.30	17.56			
20.00	19.89		9.00	9.24		16.70	16.78		10.60	11.38		17.30	17.48			

20.00	19.89		9.00	9.16		16.70	16.86		10.70	11.30		17.30	17.48			
20.00	19.81		9.00	9.24		16.70	16.86		10.70	11.14		17.30	17.48			
20.00	19.81		9.00	9.07		16.70	16.93		10.70	11.06		17.30	17.48			
20.00	19.81		9.00	9.24		16.70	16.93		10.70	11.30		17.30	17.48			
20.00	19.81		9.00	9.24		16.70	16.93		10.70	11.14		17.30	17.48			
20.00	19.81		8.90	9.32		16.70	16.93		10.70	11.06		17.30	17.48			
20.00	19.81		9.00	9.16		16.70	16.86		10.70	11.06		17.30	17.48			
20.00	19.81		9.00	9.07		16.70	16.78		10.70	11.14		17.30	17.48			
20.00	19.81		8.90	9.16		16.70	16.78		10.70	10.97		17.30	17.40			
20.00	19.81		8.90	9.16		16.70	16.86		10.70	11.06		17.30	17.48			
20.00	19.81		8.90	9.24		16.70	16.86		10.70	11.22		17.40	17.48			
20.00	19.81		8.90	9.24		16.70	16.78		10.60	10.97		17.30	17.48			
20.00	19.81		8.90	9.24		16.70	16.86		10.70	10.97		17.40	17.56			
20.00	19.89		8.90	9.16		16.70	16.93		10.70	11.14		17.40	17.56			
20.00	19.89		8.90	9.16		16.70	16.86		10.70	11.06		17.40	17.56			
20.00	19.89		8.90	9.07		16.70	16.93		10.60	10.97		17.40	17.56			
20.00	19.89		8.90	9.07		16.70	16.86		---	10.97		17.40	17.48			
20.10	19.89		8.90	9.16		16.70	16.78		10.60	11.06		17.40	17.56			
20.00	19.89		8.80	9.16		16.70	16.78		10.60	11.30		17.40	17.48			
20.10	19.89		8.80	8.99		16.70	16.78		10.60	11.06		17.40	17.48			
20.00	19.89		8.80	8.99		16.70	16.70		10.60	11.14		17.40	17.56			
20.00	19.89		8.80	9.07		16.70	16.78		10.60	11.14		17.40	17.48			
20.10	19.89		8.80	9.16		16.70	16.78		10.60	11.54		17.40	17.56			
20.10	19.89		8.80	8.99		16.70	16.78		10.70	11.38		17.40	17.48			
20.10	19.89		8.80	8.99		16.70	16.86		10.70	11.30		17.40	17.48			
20.10	19.89		8.80	9.07		16.70	16.86		10.70	11.87		17.50	17.56			
			8.80	9.07		16.70	16.93		10.70	11.46		17.40	17.56			
			8.80	9.16		16.70	16.86		10.70	11.46		17.50	17.56			
			8.80	8.99		16.70	16.86		10.70	11.38		17.40	17.56			
			8.70	8.90		16.70	16.93		10.70	11.38		17.50	17.56			
			8.70	8.90		16.70	16.93		10.80	11.14		17.50	17.56			
			8.70	8.99		16.70	17.01		10.70	11.14		17.50	17.63			
			8.70	9.07		16.70	16.93		10.70	11.14		17.50	17.63			
			8.70	9.07		16.70	16.86		10.70	11.14		17.50	17.56			
			8.70	9.07		16.70	16.86		10.80	11.06		17.50	17.56			
			8.70	9.16		16.70	16.86		10.70	11.06		17.50	17.56			
			8.70	8.99		16.70	16.93		10.70	11.06		17.50	17.48			
			8.70	8.90		16.70	17.01		10.70	10.97		17.50	17.56			
			8.70	8.99		16.70	17.01		10.70	10.89		17.50	17.63			
			8.70	8.90		16.70	17.01		10.70	11.22		17.50	17.56			
			8.60	8.99		16.70	16.93		10.70	10.97		17.50	17.56			

		8.60	8.99	16.70	16.86	10.70	10.97	17.50	17.63		
		8.60	8.90	16.70	16.78	10.70	11.06	17.50	17.63		
		8.60	8.82	16.70	16.78	10.60	10.97	17.50	17.63		
		8.60	8.99	16.70	16.86	10.70	10.97	17.50	17.63		
		8.60	8.99	16.70	16.86	10.60	10.97	17.50	17.63		
		8.60	8.99	16.70	16.78			17.60	17.63		
		8.60	8.90	16.70	16.86			17.60	17.63		
		8.60	8.82	16.70	16.86			17.60	17.63		
		8.60	8.99	16.70	16.86			17.60	17.56		
		8.60	8.90	16.70	16.78			17.60	17.63		
		8.60	8.82	16.70	16.78			17.60	17.63		
		8.60	8.82	16.70	16.86			17.60	17.63		
		8.60	8.90	16.70	16.86			17.60	17.71		
		8.60	8.82	16.70	17.01			17.60	17.63		
		8.60	8.90	16.70	17.09			17.60	17.63		
		8.60	8.90	16.80	16.93			17.60	17.63		
		8.60	8.82	16.70	16.86			17.60	17.63		
		8.60	8.90	16.70	16.86			17.60	17.71		
				16.70	16.93			17.60	17.63		
				16.70	17.01			17.60	17.63		
				16.70	17.01						
				16.70	16.93						
				16.70	16.86						
				16.70	16.86						
				16.70	16.93						
				16.70	16.93						
				16.70	16.93						
				16.70	16.86						
				16.80	16.78						
				16.70	16.86						
				16.70	16.86						
				16.80	16.86						
				16.70	16.86						
				16.70	16.86						
				16.70	16.93						
				16.80	16.86						
				16.70	16.78						

Tabel 38 Volledige meting voor Kalibratie ND03R00104J

11.15 Volledige meting voor Kalibratie Grove – Temperatuursensor V1.2

DHT22	Grove		DHT22	Grove		DHT22	Grove		DHT22	Grove		DHT22	Grove
21.50	21.98		10.50	14.13		20.90	20.78		11.60	11.08		21.20	21.02
21.50	21.98		10.60	14.37		20.90	20.78		11.50	10.92		21.20	20.70
21.60	22.06		10.90	14.53		20.90	20.78		11.40	10.75		21.10	19.42
21.60	22.14		11.00	14.77		20.90	20.78		11.30	10.58		21.00	18.78
21.60	22.14		11.20	15.10		20.90	20.78		11.20	10.33		20.70	17.26
21.60	22.14		11.40	15.26		20.90	20.78		11.10	10.08		20.20	15.66
21.70	22.22		11.60	15.42		20.90	20.78		11.00	9.99		19.70	14.04
21.70	22.22		11.70	15.58		20.90	20.78		11.00	9.83		19.20	12.57
21.70	22.22		11.90	15.74		20.90	20.78		10.80	9.66		18.60	11.25
21.80	22.30		12.10	15.98		20.90	20.78		10.70	9.40		17.90	9.99
21.80	22.30		12.20	16.14		20.90	20.70		10.60	9.32		17.20	8.72
21.80	22.30		12.40	16.22		20.90	20.78		10.50	9.23		16.50	7.59
21.80	22.30		12.60	16.38		20.90	20.78		10.40	9.15		15.80	6.62
21.90	22.30		12.70	16.54		20.90	20.70		10.30	9.06		15.10	5.55
21.90	22.30		12.90	16.62		20.90	20.70		10.20	8.89		14.40	4.55
21.90	22.38		13.00	16.78		20.90	20.70		10.10	8.80		13.60	3.62
21.90	22.38		13.20	16.86		20.90	20.70		10.00	8.63		12.90	2.77
21.90	22.30		13.30	16.94		20.90	20.70		9.90	8.55		12.20	1.90
21.90	22.38		13.50	17.10		20.90	20.78		9.80	8.46		11.50	1.11
22.00	22.38		13.60	17.18		20.90	20.70		9.70	8.37		10.90	0.31
22.00	22.38		13.70	17.34		20.90	20.70		9.60	8.29		10.20	-0.31
22.00	22.38		13.90	17.34		20.90	20.70		9.50	8.20		9.50	-0.93
22.00	22.38		14.00	17.42		20.90	20.70		9.40	8.11		8.90	-1.67
22.00	22.38		14.20	17.58		20.90	20.70		9.30	8.11		8.30	-2.31
22.00	22.38		14.30	17.58		20.90	20.70		9.20	8.03		7.60	-2.97
22.00	22.38		14.50	17.66		20.90	20.70		9.10	7.94		7.00	-3.42
22.10	22.38		14.60	17.74		20.90	20.70		9.10	7.85		6.40	-3.98
22.10	22.38		14.70	17.90		20.90	20.78		9.00	7.85		5.90	-4.44
22.10	22.38		14.90	17.90		20.90	20.70		8.90	7.68		5.30	-5.03
22.10	22.38		15.00	17.98		20.90	20.70		8.80	7.68		4.70	-5.51
22.10	22.38		15.10	18.06		20.90	20.70		8.80	7.59		4.20	-5.87
22.20	22.38		15.20	18.14		20.90	20.70		8.70	7.59		3.60	-6.48
22.20	22.38		15.30	18.22		20.90	20.70		8.60	7.50		3.10	-6.73
22.20	22.38		15.50	18.30		20.90	20.70		8.50	7.50		2.60	-7.11
22.20	22.46		15.50	18.30		20.90	20.78		8.50	7.33		2.10	-7.49
22.20	22.46		15.60	18.38		20.90	20.70		8.40	7.33		1.60	-7.62
22.20	22.46		15.80	18.46		20.90	20.70		8.30	7.24		1.20	-8.01
22.20	22.46		15.90	18.46		20.90	20.70		8.30	7.24		0.70	-8.40
22.20	22.38		16.00	18.54		20.90	20.78		8.20	7.15		0.20	-8.80
22.30	22.46		16.00	18.62		20.90	20.78		8.10	7.15		-0.20	-8.80
22.30	22.46		16.20	18.62		20.90	20.78		8.10	7.07		-0.60	-9.48
22.30	22.46		16.20	18.70		21.00	20.38		8.00	7.07		-1.00	-9.48
22.30	22.46		16.40	18.78		20.90	20.14		8.00	6.98		-1.40	-9.90
22.30	22.46		16.40	18.86		20.90	19.82		7.90	6.98		-1.80	-10.04

22.30	22.38		16.50	18.86		20.80	19.66		7.80	6.98		-2.30	-10.47
22.30	22.38		16.60	18.86		20.80	19.42		7.80	6.89		-2.60	-10.61
22.30	22.38		16.70	18.94		20.70	19.18		7.70	6.80		-3.00	-10.90
22.30	22.38		16.80	19.02		20.50	18.94		7.70	6.80		-3.30	-11.05
22.30	22.38		16.90	19.02		20.40	18.70		7.70	6.80		-3.70	-11.19
22.30	22.38		17.00	19.10		20.30	18.54		7.60	6.62		-4.00	-11.64
22.30	22.38		17.00	19.18		20.10	18.22		7.50	6.71		-4.30	-11.49
22.30	22.38		17.10	19.18		20.00	18.06		7.50	6.71		-4.60	-11.94
22.30	22.38		17.20	19.18		19.80	17.90		7.50	6.62		-5.00	-11.94
22.30	22.38		17.30	19.26		19.70	17.66		7.40	6.71		-5.30	-12.25
22.30	22.38		17.30	19.26		19.50	17.42		7.40	6.62		-5.60	-12.25
22.40	22.38		17.40	19.34		19.30	17.18		7.30	6.71		-5.90	-12.55
22.30	22.38		17.50	19.34		19.20	17.02		7.30	6.62		-6.20	-12.40
22.40	22.38		17.50	19.42		19.00	16.78		7.20	6.62		-6.50	-12.71
22.40	22.38		17.60	19.42		18.80	16.62		7.20	6.62		-6.70	-13.19
22.40	22.38		17.70	19.42		18.70	16.46		7.20	6.54		-7.00	-13.19
22.40	22.38		17.80	19.50		18.50	16.30		7.20	6.54		-7.30	-13.19
22.40	22.38		17.80	19.50		18.40	16.06		7.10	6.54		-7.50	-13.35
22.40	22.38		17.90	19.58		18.20	15.90		7.00	6.54		-7.80	-13.35
22.40	22.38		17.90	19.58		18.00	15.82		7.00	6.45		-8.00	-13.51
22.40	22.38		18.00	19.66		17.90	15.58		7.00	6.45		-8.30	-14.00
22.40	22.38		18.10	19.66		17.70	15.42		6.90	6.45		-8.50	-13.83
22.40	22.38		18.10	19.66		17.60	15.34		---	6.45		-8.80	-13.67
22.40	22.38		18.20	19.74		17.40	15.18		6.90	6.45		-9.00	-14.00
22.40	22.38		18.30	19.82		17.30	15.02		6.90	6.36		-9.20	-14.00
22.40	22.38		18.30	19.82		17.10	14.85		6.80	6.36		-9.40	-14.00
22.40	22.38		18.40	19.82		17.00	14.77		6.80	6.36		-9.70	-14.33
22.40	22.38		18.50	19.90		16.80	14.61		6.80	6.36		-9.90	-14.33
22.40	22.38		18.50	19.90		16.70	14.45		6.70	6.36		-10.10	-14.33
22.40	22.38		18.50	19.90		16.50	14.29		6.70	6.36		---	-14.50
22.40	22.38		18.60	19.90		16.40	14.21		6.70	6.36		-10.40	-14.84
22.40	22.38		18.60	19.98		16.20	13.96		6.70	6.36		-10.60	-14.67
22.40	22.38		18.70	19.98		16.10	13.88		6.70	6.36		-10.80	-14.84
22.40	22.38		18.70	19.98		16.00	13.72		6.70	6.27		-11.00	-15.01
22.40	22.38		18.80	20.06		15.80	13.64		6.60	6.27		-11.20	-15.18
22.40	22.38		18.80	20.06		15.70	13.47		---	6.27		-11.40	-15.18
22.40	22.38		18.90	20.14		15.60	13.39		6.60	6.36		-11.50	-15.01
22.40	22.38		18.90	20.14		15.40	13.23		6.60	6.36		-11.70	-15.36
22.40	22.38		19.00	20.14		15.30	13.15		6.50	6.27		-11.90	-15.36
22.40	22.38		19.00	20.14		15.20	13.07		6.50	6.18		-12.00	-15.36
22.40	22.38		19.00	20.14		15.10	12.98		6.50	6.18		-12.20	-15.89
22.40	22.38		19.10	20.22		15.00	12.82		6.50	6.18		-12.30	-15.89
22.40	22.38		19.20	20.22		14.80	12.74		6.50	6.18		-12.50	-15.71
22.40	22.38		19.20	20.22		14.70	12.66		6.40	6.09		-12.70	-15.89
22.40	22.38		19.20	20.22		14.60	12.49		6.40	6.09		-12.80	-15.89
22.50	22.38		19.30	20.22		14.50	12.41		6.40	6.18		-12.90	-15.89

22.50	22.46		19.30	20.30		14.40	12.41		6.40	6.18		-13.10	-15.89
22.50	22.46		19.30	20.22		14.30	12.25		6.30	6.09		-13.20	-16.07
22.50	22.38		19.40	20.30		14.20	12.16		6.30	6.18		-13.30	-15.71
22.50	22.38		19.40	20.38		14.10	12.08		6.30	6.18		-13.50	-16.25
22.50	22.38		19.50	20.38		14.00	12.00		6.30	6.18		-13.60	-16.07
22.50	22.38		19.50	20.38		13.90	11.92		6.30	6.18		-13.80	-16.25
22.50	22.38		19.50	20.38		13.80	11.92		6.30	6.18		-13.90	-16.44
22.50	22.38		19.50	20.46		13.70	11.83		6.30	6.09		-14.00	-16.25
22.50	22.38		19.60	20.38		13.60	11.75		6.20	6.09		-14.10	-16.62
22.50	22.38		19.60	20.46		13.50	11.67		6.20	6.09		-14.20	-16.44
22.50	22.38		19.70	20.46		13.40	11.58		6.20	6.09		-14.30	-16.62
22.40	22.38		19.70	20.46		13.30	11.58		6.20	6.18		-14.50	-16.62
22.50	22.38		19.70	20.54		13.20	11.50		6.20	6.09		-14.60	-16.44
22.50	22.38		19.80	20.54		13.10	11.42		6.20	6.09		-14.70	-16.81
22.50	22.38		19.80	20.46		13.10	11.33		6.20	6.09		-14.80	-16.62
22.40	22.38		19.90	20.54		13.00	11.25		6.10	6.09		-14.90	-16.81
22.40	22.38		19.90	20.54		12.90	11.17		6.10	6.00		-15.00	-16.81
22.40	22.38		19.90	20.54		12.80	11.17		6.10	6.00		-15.10	-17.00
22.40	22.38		19.90	20.54		12.70	11.17		6.10	6.09		-15.20	-16.81
22.50	22.38		19.90	20.54		12.70	11.00		6.10	5.91		-15.30	-16.81
22.50	22.38		20.00	20.54		12.60	11.00		6.10	6.00		-15.40	-16.81
22.40	22.38		20.00	20.54		12.50	10.92		6.10	6.00		-15.50	-17.19
22.50	22.46		20.00	20.54		12.50	11.00		6.00	6.00		-15.60	-17.00
22.50	22.46		20.10	20.54		12.40	10.83		6.00	6.00		-15.70	-17.00
22.40	22.46		20.10	20.62		12.30	10.83		6.00	6.00		-15.80	-17.19
22.50	22.46		20.10	20.62		12.20	10.75		6.00	6.00		-15.90	-17.38
22.50	22.38		20.10	20.62		12.20	10.75		6.00	6.00		-16.00	-17.19
22.50	22.38		20.10	20.62		12.10	10.75		6.00	6.00		-16.10	-17.19
22.50	22.38		20.20	20.62		12.10	10.75		6.00	5.91		-16.10	-17.19
22.50	22.46		20.20	20.62		12.00	10.83		6.00	6.00		-16.20	-17.38
22.50	22.38		20.20	20.62		12.00	10.83		6.00	6.00		-16.30	-17.19
22.50	22.46		20.20	20.62		11.90	10.83		6.00	5.91		-16.40	-17.57
22.50	22.38		20.20	20.62		11.90	10.75		6.00	6.00		-16.50	-17.57
22.50	22.46		20.20	20.62		11.80	10.67		5.90	6.09		-16.60	-17.57
22.50	22.46		20.30	20.70		11.80	10.67		6.00	6.09		-16.60	-17.57
22.50	22.46		20.30	20.70		---	10.58		5.90	6.09		-16.70	-17.77
22.50	22.46		20.30	20.70		11.70	10.58		5.90	6.09		-16.80	-17.57
22.50	22.38		20.30	20.70		11.60	10.50		5.90	6.09		-16.80	-17.77
22.50	22.46		20.30	20.70		11.60	10.50		5.90	6.09		-16.90	-17.57
22.50	22.46		20.30	20.70		11.50	10.42		5.90	6.09		-17.00	-17.77
22.50	22.46		20.40	20.70		11.50	10.42		5.90	6.09		-17.10	-17.57
22.50	22.46		20.40	20.70		11.40	10.33		5.90	6.00		-17.10	-17.96
22.50	22.46		20.40	20.70		11.40	10.33		5.90	6.09		-17.20	-17.96
22.50	22.46		20.40	20.70		11.30	10.33		5.90	6.09		-17.30	-17.77
22.50	22.46		20.40	20.70		11.30	10.25		5.90	6.00		-17.30	-17.96
22.50	22.46		20.40	20.70		11.20	10.25		5.90	6.00		-17.40	-18.16

22.50	22.38		20.40	20.70		11.20	10.25		5.90	6.00		-17.40	-17.96
22.50	22.38		20.40	20.70		11.10	10.16		5.90	5.91		-17.50	-17.96
22.50	22.38		20.40	20.70		11.10	10.08		5.90	6.00		-17.60	-18.16
22.50	22.38		20.50	20.70		11.00	10.08		5.90	6.09		-17.60	-17.96
22.50	22.38		20.50	20.70		11.00	10.08		5.90	6.09		-17.70	-18.16
22.50	22.38		20.50	20.70		11.00	10.08		5.90	6.09		-17.80	-17.96
22.50	22.46		20.50	20.70		10.90	9.99		5.90	6.09		-17.80	-17.96
22.50	22.38		20.50	20.78		10.90	9.99		5.90	6.09		-17.90	-17.77
22.50	22.46		20.50	20.78		10.80	9.91		5.90	6.18		-18.00	-17.96
22.50	22.38		20.50	20.70		10.80	9.91		5.90	6.18		-18.00	-18.16
22.50	22.46		20.50	20.70		10.80	9.83		---	6.18		-18.10	-18.36
22.50	22.38		20.50	20.78		10.70	9.83		5.90	6.18		-18.10	-18.36
22.50	22.38		20.50	20.70		10.70	9.83		5.90	6.18		-18.20	-18.36
22.50	22.38		20.60	20.78		10.60	9.83		5.90	6.27		-18.20	-18.36
22.50	22.38		20.60	20.78		10.60	9.74		5.90	6.18		-18.30	-18.36
22.50	22.38		20.60	20.78		10.50	9.74		5.90	6.18		-18.30	-18.36
22.50	22.38		20.60	20.78		10.50	9.74		5.90	6.18		-18.40	-18.36
22.50	22.46		20.60	20.78		10.50	9.74		5.90	6.18		-18.40	-18.36
22.50	22.46		20.60	20.78		10.50	9.66		5.90	6.18		-18.50	-18.36
22.50	22.46		20.60	20.78		10.40	9.66		5.90	6.18		-18.50	-18.56
22.50	22.46		20.60	20.78		10.40	9.66		5.90	6.18		-18.60	-18.36
22.50	22.46		20.60	20.78		10.40	9.66		5.90	6.27		-18.60	-18.56
22.50	22.46		20.60	20.78		10.30	9.83		5.90	6.18		-18.70	-18.36
22.50	22.46		20.60	20.78		10.30	10.08		5.90	6.27		-18.70	-18.56
			20.60	20.78		10.30	10.33		6.00	6.27		-18.80	-18.56
			20.70	20.78		10.30	10.58		5.90	6.18		-18.80	-18.56
			20.70	20.78		10.40	10.75		5.90	6.18		-18.90	-18.97
			20.70	20.78		10.40	10.92		5.90	6.18		-18.90	-18.56
			20.70	20.78		10.50	11.00		5.90	6.18		-19.00	-18.77
			20.70	20.78		10.50	11.00		5.90	6.18		-19.00	-18.77
			20.70	20.78		10.50	11.00		5.90	6.18		-19.00	-18.97
			20.70	20.78		10.60	10.92		5.90	6.18		-19.10	-18.77
			20.70	20.78		10.60	10.92		5.90	6.18		-19.20	-18.97
			20.70	20.78		10.60	10.83		5.90	6.18		-19.20	-18.77
			20.70	20.86		10.60	10.75		5.90	6.18		-19.20	-18.77
			20.70	20.78		10.60	10.75		5.90	6.18		-19.30	-18.97
			20.70	20.86		10.60	10.67		5.90	6.09		-19.30	-18.77
			20.70	20.86		10.60	10.58		5.90	6.18		-19.30	-18.77
			20.80	20.86		10.60	10.58		5.90	6.18		-19.40	-18.77
			20.80	20.86		10.60	10.50		5.90	6.18		-19.40	-18.97
			20.80	20.86		10.60	10.42		5.90	6.18		-19.50	-18.77
			20.80	20.86		10.60	10.33		5.90	6.09		-19.50	-19.18
			20.80	20.86		10.60	10.33		5.90	6.18		-19.50	-19.18
			20.80	20.86		10.50	10.25		5.90	6.18		-19.60	-18.97
			20.80	20.94		10.50	10.25		5.90	6.18		-19.60	-19.18
			20.80	20.86		10.50	10.25		5.90	6.18		-19.60	-19.18

			20.80	20.86		10.50	10.16		5.90	6.18		-19.70	-18.97
			20.80	20.86		10.40	10.08		5.90	6.18		-19.70	-18.97
			20.80	20.86		10.40	10.08		5.90	6.09		-19.70	-18.97
			20.80	20.86		10.40	9.99		5.90	6.18		-19.70	-18.77
			20.80	20.86		10.40	9.91		5.90	6.18		-19.80	-18.97
			20.80	20.86		10.40	9.91		5.90	6.18		-19.80	-19.18
			20.80	20.86		10.30	9.83		5.90	6.18		-19.80	-19.18
			20.80	20.86		10.30	9.83		5.90	6.18		-19.90	-19.39
			20.80	20.86		10.30	9.74		5.90	6.27		-19.90	-19.39
			20.80	20.86		10.20	9.74		5.90	6.36		-19.90	-19.39
			20.90	20.86		10.20	9.74		5.90	6.27		-20.00	-19.39
			20.90	20.94		10.20	9.66		5.90	6.18		-20.00	-19.18
			20.90	20.94		10.20	9.66		5.90	6.27		-20.00	-19.18
			20.90	20.94		10.20	9.57		5.90	6.36		-20.00	-19.39
			20.90	21.02		10.10	9.57		5.90	6.18		-20.10	-19.18
			20.90	21.02		10.10	9.49		5.90	6.27		-20.10	-19.39
			20.90	21.02		10.00	9.49		5.90	6.27		-20.10	-19.39
			20.90	21.02		10.00	9.40					-20.20	-19.18
			20.90	21.02		10.00	9.40					-20.20	-19.39
			21.00	21.02		10.00	9.40					-20.20	-19.39
			20.90	21.02		10.00	9.32					-20.20	-19.39
			21.00	21.02		9.90	9.32					-20.20	-19.39
			21.00	21.02		9.90	9.32					-20.30	-19.39
			21.00	21.02		9.90	9.32					-20.30	-19.39
			21.00	21.10		9.90	9.23					-20.30	-19.39
			21.00	21.10		9.80	9.23					-20.40	-19.39
			21.00	21.10		9.80	9.15					-20.40	-19.39
			21.00	21.02		9.80	9.15					-20.40	-19.61
			21.00	21.10		9.80	9.15					-20.40	-19.82
			21.00	21.10		9.70	9.06					-20.50	-19.82
			21.00	21.10		9.70	9.06					-20.50	-19.61
			21.00	21.10		9.70	9.06					-20.50	-19.39
			21.00	21.10		9.60	9.06					-20.50	-19.39
			21.00	21.10		9.60	8.97					-20.50	-19.39
			21.00	21.10		9.60	8.97					-20.60	-19.82
			21.10	21.10		9.60	8.97					-20.60	-19.18
			21.10	21.10		9.50	8.97					-20.60	-19.82
			21.10	21.10		9.50	8.89					-20.70	-19.39
			21.10	21.10		9.50	8.89					-20.70	-19.82
			21.10	21.10		9.40	8.80					-20.70	-19.82
			21.10	21.10		9.40	8.89					-20.70	-19.61
			21.10	21.10		9.40	8.80					-20.70	-19.82
			21.10	21.10		9.40	8.80					-20.70	-19.39
			21.10	21.10		9.30	8.80					-20.80	-19.82
			21.10	21.10		9.30	8.72					-20.80	-19.61
			21.10	21.10		9.30	8.72					-20.80	-19.61

			21.10	21.10		9.30	8.72					-20.80	-20.04
			21.10	21.10		9.30	8.72					-20.80	-19.61
			21.10	21.10		9.20	8.63					-20.90	-19.61
			21.10	21.10		9.20	8.63					-20.90	-19.82
			21.10	21.10		9.20	8.63					-20.90	-19.39
			21.10	21.10		9.20	8.63					-20.90	-19.82
			21.10	21.10		9.10	8.63					-20.90	-19.39
			21.10	21.10		9.10	8.55					-20.90	-19.61
			21.10	21.10		9.10	8.63					-21.00	-19.61
			21.10	21.10		9.10	8.55					-21.00	-19.82
			21.10	21.10		9.00	8.46					-21.00	-19.82
			21.10	21.10		9.00	8.46					-21.00	-19.61
			21.10	21.18		9.00	8.46					-21.00	-19.61
			21.10	21.10		9.00	8.46					-21.00	-20.04
			21.10	21.10		9.00	8.46					-21.00	-19.61
			21.10	21.10		8.90	8.37					-21.00	-19.61
			21.10	21.18		8.90	8.37					---	-19.82
						8.90	8.37					-21.10	-19.82
						8.90	8.46					-21.10	-19.61
						8.80	8.37					-21.10	-20.04
						8.80	8.46					-21.10	-19.61
						8.80	8.46					-21.10	-19.39
						8.80	8.46					-21.10	-19.61
						8.80	8.46					-21.20	-19.39
						8.80	8.46					-21.20	-19.39
						8.80	8.55					---	-19.61
						8.80	8.46					-21.20	-19.82
						8.80	8.55					-21.20	-19.61
						8.80	8.63					-21.20	-19.82
						8.80	8.55					---	-19.61
						8.80	8.55					-21.20	-19.61
						8.80	8.63					-21.20	-19.61
						8.70	8.63					-21.20	-19.61
						8.80	8.63					-21.20	-19.61
						8.80	8.55					-21.20	-19.39
						8.80	8.63					-21.20	-19.82
						8.70	8.63					-21.30	-19.82
						---	8.63					-21.20	-19.82
						8.70	8.63					-21.20	-19.61
						8.70	8.63					-21.20	-20.04
						8.70	8.63					-21.30	-19.61
						8.70	8.63					-21.30	-19.61
						8.70	8.63					-21.30	-19.82
						8.70	8.72					-21.30	-19.82
						8.80	8.72					-21.30	-19.82
						8.70	8.72					---	-19.82

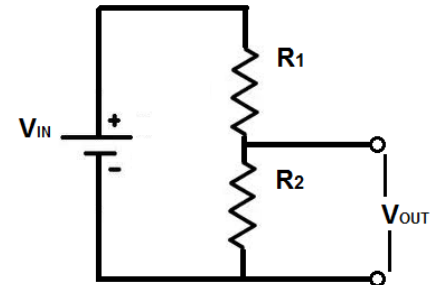
					8.70	8.63					---	-19.61
					8.70	8.72					---	-19.61
					8.70	8.72					-21.30	-19.82
					8.70	8.72					-21.30	-19.39
					8.70	8.72					-21.30	-19.82
					8.70	8.72					-21.30	-19.82
					8.70	8.80					-21.30	-19.39
					8.80	8.72					-21.30	-19.61
					8.70	8.72					-21.30	-19.61
					8.80	8.80					-21.30	-19.61
					8.80	8.72					-21.30	-19.61
					8.80	8.80					-21.30	-19.82
					8.70	8.80					-21.30	-19.61
					8.70	8.80					-21.30	-19.61
					8.70	8.80					-21.30	-19.61
					8.80	8.72					-21.30	-19.61
					8.70	8.80					-21.30	-19.61
					8.80	8.80					-21.30	-19.18
					8.70	8.89					-21.30	-19.61
					8.70	8.80					-21.30	-19.82
					8.80	8.80					-21.30	-19.61
					8.80	8.80					-21.30	-19.82
					8.70	8.80					-21.30	-19.82
					8.70	8.80					-21.30	-19.18
					8.80	8.80					-21.30	-19.61
					8.80	8.80					-21.30	-19.61
					8.70	8.89					-21.40	-19.82
					8.80	8.80					-21.40	-19.61
					8.70	8.80					-21.40	-19.61
					8.70	8.80					-21.40	-19.61
					8.80	8.80					-21.30	-19.61
					8.70	8.80					-21.40	-19.82
					8.80	8.80					-21.30	-19.61
					8.80	8.80					---	-19.82
					8.80	8.89					-21.40	-19.61
					8.80	8.89					-21.30	-19.61
					8.80	8.89					-21.30	-19.61
					8.80	8.89					-21.30	-19.82
					8.80	8.80					-21.30	-19.82
					8.80	8.89					-21.30	-19.61
					8.80	8.80					-21.30	-19.82
					8.80	8.89					-21.30	-20.04
					8.80	8.89					-21.30	-19.39
					8.80	8.89					-21.40	-19.39
					8.80	8.89					-21.40	-19.18

					8.80	8.89					-21.30	-19.39
					8.80	8.89					-21.40	-19.39
					8.80	8.80					-21.30	-19.61
					8.80	8.89					-21.30	-19.82
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.97						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.89						
					8.80	8.97						
					8.80	8.89						
					8.80	8.97						
					8.80	8.97						
					8.80	8.97						

Tabel 39 Volledige meting voor Kalibratie Grove – Temperatuursensor V1.2

11.16 Formule Temperatuur (NTCLE100E3334JB0, NTC 330kΩ) afgekeurde sensor

Om te beginnen is in Figuur 9 te zien dat de 330KΩ NTC een spanningsdeling vormt met een 330KΩ weerstand. Het midden van deze spanningsdeling gaat naar pin P1.6, deze meet de ADC-waarde van deze spanning in. Deze ADC-waarde is een waarde tussen de 0 en 1023 die staat voor een spanning tussen 0 en de 3,3 volt (maximale voedingspanning). Aan de hand van deze voedingspanning kan de weerstand van de NTC berekend worden. Deze weerstand waarde is nodig om de temperatuur mee uit te kunnen rekenen.



Figuur 45 Spanningsdeler

De formule voor een spanningsdeler is uit Figuur 45:

$$V_{OUT} = V_{IN} * \frac{R_2}{R_1 + R_2} \quad [\text{formule 8}]$$

Hierbij is de V_{IN} de voedingspanning, V_{OUT} de spanning uit de spanningsdeler (spanning op pin P1.6), R_1 de 330KΩ weerstand en R_2 de NTC-weerstand waarde. Omdat de waarde van R_2 nodig is om de temperatuur uit te rekenen wordt de formule omgeschreven zodat R_2 de uitkomst van de formule is:

$$R_2 = \frac{V_{OUT} * R_1}{V_{IN} - V_{OUT}} \quad [\text{formule 9}]$$

Met deze formule kan, zodra de ADC-waarde op pin P1.6 is uitgelezen, de waarde van R_2 berekend worden.

Voorbeeld, de ADC-waarde op pin P1.6 is 440. Nu zijn variabelen in de formule: $V_{OUT} = 440$, $V_{IN} = 1023$ (maximale ADC-waarden) en $R_1 = 330000$ (vaste weerstand van 330KΩ):

$$R_2 = \frac{440 * 330000}{1023 - 440} = 249056\Omega \quad [\text{formule 10}]$$

Met deze NTC-waarde van R_2 kan daarna de temperatuur uitgerekend worden. De formule die hierbij in de NTCLE100E3334JB0 datasheet (zie hoofdstuk 11.2 [D16]) staat is te zien in Figuur 46. Hierbij is de uitkomst van T (formule 9) de temperatuur in kelvin. De temperatuur in graden is dan het aantal kelvin – 273.15. De waarde R in de formule is de waarde van de NTC, en dit is de waarde die van R_2 in formule 9. De waarde voor R_{ref} is de R_{25} waarde van de NTC dat is in dit geval 330KΩ. Verder heeft deze NTC een B_{β} waarde van 4570K. De constante die in de formule gebruikt worden staan in Figuur 47.

FORMULAE TO DETERMINE NOMINAL RESISTANCE VALUES

The resistance values at intermediate temperatures, or the operating temperature values, can be calculated using the following interpolation laws (extended "Steinhart and Hart"):

$$R(T) = R_{ref} \times e^{(A + B/T + C/T^2 + D/T^3)} \quad (1)$$

$$T(R) = \left(A_1 + B_1 \ln \frac{R}{R_{ref}} + C_1 \ln^2 \frac{R}{R_{ref}} + D_1 \ln^3 \frac{R}{R_{ref}} \right)^{-1} \quad (2)$$

where:

$A, B, C, D, A_1, B_1, C_1$ and D_1 are constant values depending on the material concerned; see table below.

R_{ref} is the resistance value at a reference temperature (in this event 25 °C, $R_{ref} = R_{25}$).

T is the temperature in K.

Formulae numbered and are interchangeable with an error of max. 0.005 °C in the range 25 °C to 125 °C and max. 0.015 °C in the range - 40 °C to + 25 °C.

Figuur 46 Formule datasheet van de NTCLE100E3334JB0

PARAMETER FOR DETERMINING NOMINAL RESISTANCE VALUES											
NUMBER	B _{25/85} (K)	NAME	TOL. B (%)	A	B (K)	C (K ²)	D (K ³)	A ₁	B ₁ (K ⁻¹)	C ₁ (K ⁻²)	D ₁ (K ⁻³)
13	4570	Mat F. with Bn = 4570K	1.5	-17.6439	6022.726	-203157	-7.183526E+06	3.354016E-03	2.264097E-04	3.278184E-06	1.097628E-07

Figuur 47 Constante voor formule met een Bèta van 4570K (datasheet NTCLE100E3334JB0)

Om de formule uit Figuur 46 te kunnen gebruiken zal er nog aanpassing moeten gedaan worden aan deze formule. Dit omdat de LN-functie niet zo vaak gebruikt wordt omdat deze betekend dat het een log functie is met een natuurlijk grondtal. Zal in plaats daarvan de formule met een log functie geschreven worden:

$$\ln(x) = \log(x) \quad [\text{formule 11}]$$

De formule wordt dan, met R2 ingevoegd en met omrekening naar graden:

$$T = (A_1 + B_1 * \log\left(\frac{R_2}{R_{ref}}\right) + C_1 * \log\left(\frac{R_2}{R_{ref}}\right)^2 + D_1 * \log\left(\frac{R_2}{R_{ref}}\right)^3)^{-1} - 273.15$$

[formule 12]

Met daarin A₁= 3.354016*10⁻³, B₁= 2.264097*10⁻⁴, C₁= 3.278184*10⁻⁶, D₁= 1.097628*10⁻⁷, en R_{ref}= 330000. Dit maakt de formule:

$$T = \left(3.354016 * 10^{-3} + 2.264097 * 10^{-4} * \log\left(\frac{R_2}{R_{ref}}\right) + 3.278184 * 10^{-6} * \log\left(\frac{R_2}{R_{ref}}\right)^2 + 1.097628 * 10^{-7} * \log\left(\frac{R_2}{R_{ref}}\right)^3 \right)^{-1} - 273.15$$

[formule 13]

Dit maakt in combinatie met formule 8, met daarin Vout en R1 ingevuld (dit zijn vaste waardes) de hele formule:

$$T = \left(3.354016 * 10^{-3} + 2.264097 * 10^{-4} * \log\left(\frac{V_{OUT} * 330000}{1023 - V_{OUT}}\right) + 3.278184 * 10^{-6} * \log\left(\frac{V_{OUT} * 330000}{1023 - V_{OUT}}\right)^2 + 1.097628 * 10^{-7} * \log\left(\frac{V_{OUT} * 330000}{1023 - V_{OUT}}\right)^3 \right)^{-1} - 273.15$$

[formule 14]

Met daarin T de temperatuur in °C en Vout de ADC-waarde die gemeten wordt op pin P1.6 (zie Figuur 9).

Deze sensor is afgekeurd omdat na metingen van deze sensor er grote afwijkingen bleken te bestaan tussen de resultaten van de meting en de werkelijke temperatuur. Daarom is ervoor gekozen om een andere NTC-weerstand te gebruiken.