

WasteBin

Generated by Doxygen 1.9.4

1 Bureaubladonderzoek	1
1.1 Hypotheses en feiten die baat hebben bij de ontwikkeling van de Waste Bin	1
1.1.1 Inhoud	1
1.2 Tabellen	1
1.3 Achtergrond informatie	1
1.4 Feiten	2
1.5 Hypotheses	2
1.6 Aanpak onderzoek en bronanalyse	3
1.6.1 Uitwerking vragen	3
1.7 Onderzoek naar koelmogelijkheden	4
1.8 Onderzoek naar koelmogelijkheden	5
1.9 Wat weerhoudt veel hoogbouwwoonbewoners ervan om hun GFE-afval te scheiden?	6
2 WasteBin Project: Het creëren van een innovatieve oplossing voor geurvrije en milieuvriendelijke afvalverwerking	7
2.0.1 Inleiding	7
2.0.2 Belangrijkste doelstellingen	7
2.0.3 Kenmerken en functionaliteit van het apparaat	7
2.0.4 Voordelen en impact	8
2.0.5 Conclusie	8
3 Het Gebruik van Meerdere ESP32-Microcontrollers met Node-Red	9
3.1 Inleiding	9
3.2 Doelen	9
3.2.1 Doel #1	9
3.3 Onderbouwing keuze ESP32 C3	9
3.4 Waarom geen Arduino of Raspberry Pi als computer voor de WasteBin?	10
3.5 Data forwarding	10
3.6 De Data	10
3.7 Meldingen voor de gebruiker	11
3.8 Verstuur interval	11
3.8.1 In geval van fouten	11
3.9 Node-Red	11
3.9.1 Node-Red flow	11
3.10 Database	12
4 Ontwerpkeuzes	13
5 Testplan	15
5.1 Inhoud	15
5.2 Peltier modules	15
5.3 Doelen	15
5.4 Deksel positie	16
5.5 Temperatuur en luchtvochtigheid	16

5.5.1 Doel #1	16
5.5.2 Doel #2	16
5.5.3 Doel #3	16
5.6 Testopstelling	16
5.7 Vereiste componenten	17
5.8 Meetapparatuur	17
5.9 Methodes	17
5.10 Testresultaten	17
6 File Index	19
6.1 File List	19
7 File Documentation	21
7.1 bureaubladonderzoek.md File Reference	21
7.2 introductie.md File Reference	21
7.3 IoT oplossing.md File Reference	21
7.4 Ontwerpkeuzes.md File Reference	21
7.5 testplan.md File Reference	21
7.6 waste_bin_controller.h File Reference	21
7.6.1 Function Documentation	22
7.6.1.1 controlPeltierModule()	22
7.6.1.2 handleIncomingMessage()	23
7.6.1.3 loop()	23
7.6.1.4 publishMessage()	23
7.6.1.5 publishSensorData()	23
7.6.1.6 readSensorData()	23
7.6.1.7 setup()	23
7.6.1.8 setupMQTTClient()	23
7.6.1.9 setupPeltierAndFan()	24
7.6.1.10 setupSHT4x()	24
7.6.1.11 setupWiFi()	24
7.6.1.12 updateLidPosition()	24
7.6.2 Variable Documentation	24
7.6.2.1 BUTTON_PIN	24
7.6.2.2 CLIENT_NAME	24
7.6.2.3 humidity_value	24
7.6.2.4 interval	25
7.6.2.5 last_msg_sent	25
7.6.2.6 lid_position	25
7.6.2.7 mqtt_client	25
7.6.2.8 MQTT_PASSWORD	25
7.6.2.9 MQTT_PORT	25
7.6.2.10 MQTT_SERVER	25

7.6.2.11 MQTT_USERNAME	25
7.6.2.12 msg_interval	26
7.6.2.13 PASSWORD	26
7.6.2.14 peltierandfanpin	26
7.6.2.15 peltierandfanstate	26
7.6.2.16 previousMillis	26
7.6.2.17 pwmFreq	26
7.6.2.18 SENSOR_DATA_TOPIC	26
7.6.2.19 setPoint	26
7.6.2.20 sht4	27
7.6.2.21 SSID	27
7.6.2.22 temperature_value	27
7.6.2.23 tolerance	27
7.6.2.24 wifi_client	27
7.7 waste_bin_controller.h	27
Index	29

Chapter 1

Bureaubladonderzoek

1.1 Hypotheses en feiten die baat hebben bij de ontwikkeling van de Waste Bin

1.1.1 Inhoud

- Tabellen
- Achtergrond informatie
- Feiten
- Hypotheses
- Hoofd- en deelvragen
- Aanpak onderzoek en bron analyse
- Uitwerking vragen
- Bibliografie

1.2 Tabellen

Tabel 1: Hypotheses

1.3 Achtergrond informatie

Voor het WasteBin project zijn we gevraagd om een GFE-bak te ontwerpen die GFE-afval vrijwel geurloos en hygiënisch opslaat. Om het geurloze resultaat te bereiken, moeten we het rottingsproces zoveel mogelijk vertragen. Hierbij moeten we een balans vinden tussen energieverbruik, kosten en tijd. We moeten onderzoeken hoe deze factoren invloed hebben op de werking van het product. Daarnaast moeten we vragen beantwoorden zoals: is een ozon-generator überhaupt nodig als we het rottingsproces vertragen? In dit document stellen we tests op, voeren we ze uit en bespreken we de resultaten, om vervolgens onze conclusies te implementeren in ons (technisch) ontwerp.

1.4 Feiten

- Een hoge luchtvochtigheid en temperatuur zorgen voor een sneller rottingsproces bij GFE-afval.
- Een te hoge concentratie ozon is gevaarlijk voor de gezondheid van de mens.
- Koude lucht kan een hogere luchtvochtigheid hebben dan warme lucht.

"Wanneer uurgemiddelde ozonconcentraties hoger zijn dan 180 microgram per kubieke meter lucht, is de luchtkwaliteit 'slecht'. Als deze waarde overschreden dreigt te worden, kunnen gevoelige mensen klachten krijgen en waarschuwt het RIVM. De luchtkwaliteit is 'zeer slecht' wanneer de concentraties drie uur lang hoger zijn dan 240 microgram per kubieke meter lucht. Wanneer deze waarde overschreden wordt, kan iedereen klachten krijgen. Als dit dreigt te gebeuren, zet het RIVM de waarschuwing om in een alarm." (RIVM, sd)

1.5 Hypotheses

Hypothese	Uitwerking/opmerkingen
Een gegeven materiaal houdt zijn temperatuur beter vast dan lucht.	Thermoskan idee: hoe voller de thermoskan is gevuld met hete drank, hoe langer het duurt voor de inhoud is afgekoeld. Dus lucht ontnemen of de bak vullen met water om de optimale temperatuur makkelijk te behouden.
Water geleid temperatuur beter dan lucht.	Door de vuilnisbak met water te vullen en het GFE-afval in een zak hierin te koelen, zal dit een efficiëntere thermische overdracht hebben dan via de lucht. De verspreiding van de geur van het rottingsproces zal door het koelen van het GFE-afval minder snel plaatsvinden. Hierdoor is de ozon-generator overbodig.
Het invriezen van het GFE-afval stopt het rottingsproces volledig.	Het invriezen van het GFE-afval zal ervoor zorgen dat de micro-organismen die zich in het afval bevinden niet verder kunnen groeien. Sommige organismen kunnen dit alsnog doen, maar dit gebeurt langzamer en zal in de prullenbak nog niet plaatsvinden.
Het ontnemen van (een deel van) de zuurstof in combinatie met een ozon-generator remt het rottingsproces voldoende af.	De combinatie van ozon en minder zuurstof zal zorgen dat het afval dusdanig lang geremd wordt dat koeling niet nodig is.

Hoofd- en deelvragen:

1. Wat heeft de grootste invloed op het rottingsproces? a. Welk effect heeft de omgevingstemperatuur op het rottingsproces? i. Hoeveel invloed heeft een negatieve temperatuur op het rottingsproces? b. Welk effect heeft luchtvochtigheid op het rottingsproces? i. Hoeveel effect heeft een lagere luchtvochtigheid op het rottingsproces? c. Welk effect heeft zuurstof op het rottingsproces? i. Welk effect heeft de aanwezigheid van zuurstof in een ruimte op het rottingsproces? ii. Hoeveel effect heeft zuurstof op het rottingsproces? d. Wat is de maximale tijd waarop het rottingsproces kan worden uitgesteld of vertraagd rekening houdend met de drie eerder genoemde factoren? e. Is het mogelijk om het effect van ozon-generatie te compenseren met koeling (tot en op het vriespunt)?
2. Hoeveel liter GFE-afval moet de WasteBin kunnen opslaan? a. Hoeveel liter GFE-afval produceert een gemiddeld huishouden per dag? (stedelijk gebied) b. Wat is het maximale volume van de prullenbak en is dit haalbaar? c. Wat verstaan we onder een huishouden?

3. Wat weerhoudt veel bewoners van hoogbouwoningen ervan om hun GFE-afval te scheiden? a. Welke drempels kunnen we realistisch gezien verlagen om het scheidingsproces te bevorderen?

1.6 Aanpak onderzoek en bronanalyse

Ons onderzoek, testen en documentatie zijn gebaseerd op het prototype dat ESE ontwikkelt voor Insyte. We zullen de vragen voornamelijk proberen op te lossen door gericht te zoeken op internet. We kunnen hiervoor bronnen gebruiken van Google Scholar of andere bronnen die betrouwbaar lijken. We zullen ook de bronnen gebruiken die door Insyte aan ons zijn geleverd. Deze bronnen zijn onderzoeken die onder andere zijn uitgevoerd door de HVA. Het is echter discutabel of deze bronnen betrouwbaar zijn, dus we zullen kijken naar hun werkwijze.

1.6.1 Uitwerking vragen

1. Wat heeft de grootste invloed op het rottingsproces? De grootste invloed is bla bla bla.

a. **Welk effect heeft de omgevingstemperatuur op het rottingsproces?** Volgens onderstaande grafiek neemt de afbraak van planten toe wanneer de temperatuur stijgt. Dit kan worden verklaard doordat veel bacteriën beter functioneren bij hogere temperaturen. Bij lagere temperaturen wordt het voor deze bacteriën moeilijker om te functioneren, waardoor het rottingsproces van planten vertraagt. Bij 40 graden Fahrenheit (4 graden Celsius) hebben deze groenten ongeveer 400 uur (16 dagen) nodig om het rottingsproces goed te starten. Het is gunstig om de omgevingstemperatuur laag te houden om het rottingsproces te vertragen.

b. **Welk effect heeft luchtvochtigheid op het rottingsproces?** Bij een hoge luchtvochtigheid (vooral in combinatie met een hoge omgevingstemperatuur) kan er broei ontstaan in een bak, wat leidt tot de vorming van schimmels. Schimmels spelen ook een rol in het rottingsproces en kunnen onaangename geuren veroorzaken. Het is daarom belangrijk om de luchtvochtigheid zo laag mogelijk te houden.

c. **Welk effect heeft zuurstof op het rottingsproces?** Schimmels zijn aerobe organismen, maar zelfs bij zeer lage zuurstofconcentraties is groei mogelijk (Wösten, 2017). Dit betekent dat het gunstig is om een omgeving met weinig zuurstof te creëren, maar het is niet het enige middel om het rottingsproces te voorkomen.

d. **Wat is de maximale tijd waarop het rottingsproces kan worden uitgesteld of vertraagd rekening houdend met de drie eerder genoemde factoren?** De maximale tijd is bla bla bla. Dit is ook afhankelijk van de inhoud van de bak en het gewenste energielabel.

e. **Is het mogelijk om het effect van ozon-generatie te compenseren met koeling (tot en met het vriespunt)?** Op internet hebben we meerdere keren gelezen dat onjuist gebruik van ozongeneratoren kan leiden tot hoge concentraties ozon (O₃), wat schadelijk is voor de gezondheid van organismen. Sommige mensen zijn gevoeliger dan anderen en kunnen negatieve gezondheidseffecten ervaren. Mensen die bijzonder kwetsbaar zijn, zijn onder andere kinderen, ouderen en mensen met astma (Government, 2015).

2. Hoeveel liter GFE-afval moet de WasteBin kunnen opslaan? De WasteBin moet zoveel liter GFE-afval kunnen opslaan.

a. **Hoeveel liter GFE-afval produceert een gemiddeld huishouden per dag? (stedelijk)** Vanuit de gemeente Amsterdam is bekend dat er jaarlijks ongeveer tachtig kilo GFE-afval per persoon wordt weggegooid. Dit komt neer op ongeveer 200 gram per dag. Deze cijfers gelden specifiek voor stedelijke gebieden, waar het scheiden van dit afval moeilijker is vanwege de grote hoeveelheid hoogbouw.

Jaarlijks produceert een gemiddeld persoon ongeveer tachtig kilo GFE-afval, maar hiervan belandt ongeveer 70 kilo bij het restafval omdat het niet gescheiden wordt (Schoonvelde, sd).

b. **Wat is het maximale volume van de prullenbak en is dit realistisch?** Het optimale volume van de prullenbak moet afgestemd zijn op de grootte van de prullenbak die huishoudens al gebruiken. Op die manier zal de gebruiker,

samen met het andere afval, ook het GFE-afval scheiden en de prullenbak volledig benutten. Volgens verschillende websites (Fonq.nl, brabantia.nl, prullenbak-expert.nl) is een 30-liter prullenbak geschikt voor één persoon, dus we gaan hier in dit geval van uit.

In 2020 produceerde één persoon gemiddeld 140 kilo afval per jaar (totaal). Als we rekening houden met het feit dat er jaarlijks 70 tot 80 kilo GFE-afval wordt geproduceerd (waarvan slechts een deel van ongeveer 10 kilo wordt gescheiden), kunnen we stellen dat het volume van de GFE-afvalbak maximaal de helft zou moeten zijn (Rijksoverheid, sd).

3. Wat weerhoudt veel bewoners van hoogbouwoningen ervan om hun GFE-afval te scheiden? Er zijn verschillende drempels die veel bewoners van hoogbouwoningen ervan weerhouden om hun GFE-afval te scheiden.

a. Welke drempels kunnen realistisch gezien verlaagd worden zodat het scheidingsproces wel wordt uitgevoerd? Er kunnen verschillende drempels worden verlaagd om het scheiden van GFE-afval te bevorderen. Enkele mogelijke drempels zijn de beschikbaarheid van aparte GFE-afvalbakken in hoogbouwoningen, onduidelijkheid over wat wel en niet bij GFE-afval hoort en het gebrek aan bewustzijn en educatie over het belang van het scheiden van GFE-afval.

Om deze drempels te verlagen en het scheidingsproces te bevorderen, kunnen de volgende maatregelen worden genomen:

1. **Beschikbaarheid van aparte GFE-afvalbakken:** Het plaatsen van speciale GFE-afvalbakken in hoogbouwoningen kan het scheiden van GFE-afval gemakkelijker maken. Deze bakken moeten voldoende capaciteit hebben en duidelijk gelabeld zijn.
2. **Duidelijke richtlijnen voor GFE-afval:** Het verstrekken van duidelijke informatie en richtlijnen over wat wel en niet bij GFE-afval hoort, kan verwarring verminderen en bewoners helpen bij het correct scheiden van hun afval. Dit kan worden gedaan door middel van educatief materiaal, zoals brochures, posters of online gidsen.
3. **Bewustmakingscampagnes:** Het uitvoeren van bewustmakingscampagnes gericht op hoogbouwoningen kan het bewustzijn vergroten over de voordelen van het scheiden van GFE-afval. Deze campagnes kunnen informatie bevatten over de milieueffecten van GFE-afval, tips voor het verminderen van voedselverspilling en de positieve impact van afvalscheiding op het milieu.
4. **Samenwerking met afvalverwerkingsbedrijven:** Samenwerking met afvalverwerkingsbedrijven kan helpen bij het opzetten van efficiënte systemen voor het ophalen en verwerken van GFE-afval uit hoogbouwoningen. Door het bieden van gemakkelijke en betrouwbare opties voor afvalverwijdering, kunnen bewoners worden aangemoedigd om actief deel te nemen aan het scheidingsproces.

Door deze maatregelen te implementeren, kunnen de drempels voor het scheiden van GFE-afval in hoogbouwoningen worden verlaagd en kunnen bewoners worden gestimuleerd om actief deel te nemen aan het verminderen van voedselverspilling en het bevorderen van duurzaam afvalbeheer.

1.7 Onderzoek naar koelmogelijkheden

Verschillende categorieën, uiteindelijk is alles een warmtepomp. Verschillende categorieën kunnen worden onderscheiden op basis van techniek, volume en temperatuur. Enkele voorbeelden van koeltechnieken zijn:

- Compressor
- Peltier-effect
- Magnetische koeling

Voor ons onderzoek hebben we besloten ons te richten op Peltier-modules. Deze modules hebben de volgende voordelen ten opzichte van andere koeloplossingen:

- Ze zijn verkrijgbaar in verschillende formaten, waaronder compacte formaten.
- Ze zijn relatief goedkoop in vergelijking met andere koeltechnieken.
- Ze werken geluidloos vanwege het solid-state cooling-principe.
- Er zijn geen extra gassen of waterleidingen nodig.

Het is echter belangrijk om het te koelen gebied goed te isoleren. Op deze manier kan de Peltier-module op een efficiënte manier koelen zonder constant op vol vermogen te werken, wat ook warmte genereert vanwege het feit dat het een warmtepomp is.

1.8 Onderzoek naar koelmogelijkheden

Verschillende categorieën, uiteindelijk is alles een warmtepomp. Verschillende categorieën kunnen worden onderscheiden op basis van techniek, volume en temperatuur. Enkele voorbeelden van koeltechnieken zijn:

- Compressor
- Peltier-effect
- Magnetische koeling

Opstelling met Peltier module

Om aan de binnenkant van de buitenbak de modules te monteren zo, dat de binnenbak tegen de koude kant raakt van de modules. We hebben besloten dat we het beste onze tijd en energie kunnen steken in Peltier modules omdat deze:

- Beschikbaar in veel (compacte) formaten is
- Relatief goedkoop zijn (kijkend naar andere koeloplossingen)
- Geluidloos functioneren (solid state cooling)
- Geen extra gassen, waterleidingen et cetera nodig

Wel is het zeer belangrijk om het te koelen gebied goed te isoleren. Op deze manier hoeft de Peltier module niet constant op vol vermogen te koelen (waarbij ook hitte ontstaat, aangezien het een warmtepomp is).

1.9 Wat weerhoudt veel hoogbouwwoningbewoners ervan om hun GFE-afval te scheiden?

Uit het onderzoek "Vuilnis in de flat" van de Design Innovation Group kunnen we zien hoe de meeste mensen (22 in totaal) hun keuzes maken met betrekking tot afvalscheiding.

Op basis van dit onderzoek kunnen we concluderen dat voornamelijk de geur de uiteindelijke oorzaak lijkt te zijn voor het niet scheiden van GFE-afval. Om ervoor te zorgen dat dit scheidingsproces wel wordt uitgevoerd, kunnen we realistisch gezien de volgende drempels verlagen:

- We kunnen proberen de geur van het rottingsproces zoveel mogelijk te verminderen of te stoppen. Dit kan worden bereikt door het vertragen of stoppen van het rottingsproces, of door het verwijderen van de geuren die vrijkomen bij het rottingsproces.
- De specifieke aanpak om dit te bereiken moet nog worden vastgesteld.

"In diverse wijken in Utrecht wordt het GFT opgehaald. Eén geïnterviewde was in haar straat de enige die de GFT-bak buiten zet. Vaak werd die bak dan op de route vergeten. Twee geïnterviewden scheidden GFT maar zijn afgehaakt; de stank, vliegjes en natte zooi in huis werden als redenen gegeven." (designinnovationgroup, 2015)

Bibliografie

- Platenius, H. (1939). Effect of temperature on the rate of deterioration of fresh vegetables. Washinton, D.C.: Journal of Agricultural Research.
- Rijksoverheid. (sd). Opgehaald van Rijksoverheid.nl: <https://www.rijksoverheid.nl/onderwerpen/afval/huishoudelijk-afval>
- RIVM. (sd). Smog Door Ozon. Opgehaald van RIVM.nl: <https://www.rivm.nl/smog/smog-door-ozon>
- Schoonvelde, G. (sd). Infomil.nl. Opgehaald van Infomil.nl: [<https://www.infomil.nl/actueel/nieuws-perspectief-1/verbetering-afvalscheiding-hoogbouw-mits/>](<https://www.infomil.nl/actueel/n>

Chapter 2

WasteBin Project: Het creëren van een innovatieve oplossing voor geurvrije en milieuvriendelijke afvalverwerking

2.0.1 Inleiding

Het WasteBin Project heeft tot doel het probleem van onaangename geuren in kleine leefruimtes, zoals appartementen, aan te pakken, terwijl het efficiënte afvalscheiding en milieubehoud bevordert. Door de ontwikkeling van een uniek apparaat voorkomt dit project niet alleen onaangename geuren, maar stimuleert het ook gebruikers om verschillende soorten afval te scheiden, wat leidt tot effectievere recycling en een verminderde milieubelasting.

2.0.2 Belangrijkste doelstellingen

1. Geurpreventie: Het primaire doel van het WasteBin Project is het creëren van een apparaat dat onaangename geuren die uit afvalbakken komen effectief opvangt en neutraliseert. Dit zal de leefomgeving aanzienlijk verbeteren voor mensen die in kleine ruimtes wonen.
2. Afvalscheiding: Door indirect het weggooien van algemeen voedselafval (GFE-afval) in reguliere afvalbakken te ontmoedigen, stimuleert het project de scheiding van verschillende soorten afval. Dit moedigt gebruikers aan om milieuvriendelijke praktijken te omarmen, wat resulteert in gemakkelijkere sortering van recyclebaar materiaal en een vermindering van afval op de vuilnisbelt.
3. Milieueffect: Het WasteBin Project draagt bij aan milieubehoud door de hoeveelheid afval die op de vuilnisbelt terechtkomt te minimaliseren. Door een betere scheiding van afval kunnen waardevolle recyclebare materialen gemakkelijker worden teruggewonnen, wat de winning van ruwe materialen vermindert en natuurlijke hulpbronnen beschermt.

2.0.3 Kenmerken en functionaliteit van het apparaat

Het WasteBin-apparaat heeft verschillende innovatieve kenmerken om zijn doelstellingen te bereiken:

1. Mechanisme voor geurbeheersing: Het apparaat maakt gebruik van verschillende technologieën voor geurbeheersing, zoals temperatuurregeling, geactiveerde koolstoffilters en deodorizers, om onaangename geuren die uit afvalbakken komen vast te leggen en te neutraliseren. Dit zorgt voor een frisse en geurvrije leefomgeving voor gebruikers.

2. Hulp bij afvalscheiding: Het apparaat bevat een intuïtieve interface die gebruikers informeert over afvalscheidingspraktijken. Het biedt visuele indicatoren en herinneringen voor het scheiden van recyclebaar materiaal, organisch afval en andere categorieën, waardoor het proces gemakkelijk en gebruiksvriendelijk wordt.
3. Gebruiksvriendelijk ontwerp: Het WasteBin-apparaat is ontworpen om compact, esthetisch aantrekkelijk en gemakkelijk te gebruiken te zijn. Het past naadloos in kleine leefruimtes en vormt een aanvulling op het algehele interieurontwerp. De verwijderbare compartiment van het apparaat vereenvoudigt het afvalbeheer en de schoonmaakproces.

2.0.4 Voordelen en impact

Het WasteBin Project biedt talrijke voordelen en positieve effecten:

1. Verbeterde levenskwaliteit: Door onaangename geuren te elimineren, verbetert het apparaat aanzienlijk de leefomgeving voor mensen die in kleine ruimtes wonen. Het bevordert comfort, welzijn en een aangename sfeer.
2. Verbeterde afvalscheiding: Het project moedigt gebruikers effectief aan om verschillende soorten afval te scheiden, wat leidt tot een hoger recyclingpercentage en een verminderde milieubelasting. Dit draagt bij aan een duurzamer afvalbeheersysteem.
3. Bewustwording en educatie: Het WasteBin-apparaat fungeert als een educatief hulpmiddel en vergroot het bewustzijn over het belang van afvalscheiding en de positieve milieu-impact ervan. Door verantwoord afvalbeheer te bevorderen, geeft het project gebruikers de mogelijkheid om milieuvriendelijke keuzes te maken.
4. Langetermijnduurzaamheid: Door het WasteBin Project wordt de adoptie van duurzame afvalbeheerpraktijken mainstream, wat een cultuur van milieubewustzijn bevordert en blijvende positieve veranderingen in afvalverwerkingsgewoonten teweegbrengt.

2.0.5 Conclusie

Het WasteBin Project streeft ernaar een innovatief apparaat te creëren dat niet alleen onaangename geuren in kleine leefruimtes elimineert, maar ook afvalscheiding en milieuvriendelijkheid bevordert. Door geavanceerde mechanismen voor geurbeheersing, hulp bij afvalscheiding en gebruiksvriendelijke kenmerken te integreren, stelt het apparaat individuen in staat om actief een rol te spelen in het behoud van het milieu en tegelijkertijd hun levenskwaliteit te verbeteren. Met dit project streven we ernaar bij te dragen aan een schonere, groenere en meer duurzame toekomst voor iedereen.

Chapter 3

Het Gebruik van Meerdere ESP32-Microcontrollers met Node-Red

3.1 Inleiding

Meerdere ESP32-microcontrollers kunnen in combinatie met Node-Red worden gebruikt om een visuele interface te bouwen op basis van de gegevens die zijn gelogd door de ESP32. Elke ESP32 kan worden geconfigureerd om gegevens van een specifieke sensor of apparaat te loggen en deze via Wi-Fi naar Node-Red te verzenden. Node-Red kan vervolgens de gegevens ontvangen en verwerken met behulp van verschillende nodes.

3.2 Doelen

3.2.1 Doel #1

Het opzetten van een systeem waarbij meerdere ESP32-microcontrollers worden gebruikt om gegevens te loggen en deze naar Node-Red te verzenden voor visualisatie en verwerking.

3.3 Onderbouwing keuze ESP32 C3

Wij kiezen voor een ESP32 omdat deze de volgende voordelen biedt:

- Relatief goedkoop (€2 per stuk).
- Beschikbaarheid (1300 op voorraad bij Mouser, afhankelijk van het model).
- Geschikt voor de taken die moeten worden uitgevoerd.
- Compact formaat.

3.4 Waarom geen Arduino of Raspberry Pi als computer voor de WasteBin?

Raspberry Pi's zijn momenteel moeilijk verkrijgbaar, duur in aanschaf voor een testplatform en vereisen meer inspanning om op te zetten. Bovendien zijn ze vaak te krachtig voor onze doeleinden en daardoor overbodig qua rekenkracht. Arduino's zijn redelijke alternatieven, maar duurder dan losse ESP32-modules. Desalniettemin kunnen ESP32-modules via de Arduino IDE worden geprogrammeerd met behulp van de Arduino-codebibliotheken, indien gewenst.

3.5 Data forwarding

Data forwarding is de plaats waar alle data van de ESP32's (of vuilnisbakken) wordt verzameld. Een mogelijke optie is het gebruik van specifieke hardware, zoals een Raspberry Pi met een broker (zoals Mosquitto) en een Node-Red-dashboard. Wij kiezen ervoor om de broker en het dashboard op onze lokale laptop te installeren in plaats van een Raspberry Pi te gebruiken. We hebben hiervoor meerdere redenen:

1. We vermijden de noodzaak van een extra apparaat (Raspberry Pi). Alles op één apparaat maakt het eenvoudiger. Bovendien zijn Raspberry Pi's moeilijk verkrijgbaar en vormen ze een extra bron van potentiële hardwarefouten (zoals SD-kaartcorruptie) en softwareproblemen (Linux-problemen) enzovoort.
2. Als we fysiek ergens anders willen werken, zouden we steeds de Raspberry Pi moeten verbinden met het netwerk op die locatie via een beeldscherm en toetsenbord.
3. De Raspberry Pi kan een beveiligingsrisico vormen.

Stappen:

1. ESP32-gegevens laten versturen (verbinding maken met internet).
2. Opzetten van een MQTT-broker.
3. De MQTT-broker accepteert gegevens van de ESP32.
4. De MQTT-broker stuurt gegevens door naar Node-Red (lokaal).
5. Inkomende gegevens van de MQTT-broker worden omgezet naar een dashboardweergave.
6. Data van Node-Red doorsturen naar een database.

3.6 De Data

De volgende gegevens willen we versturen vanaf de ESP32:

- Temperatuur (dubbel)
- Luchtvochtigheid (dubbel)
- Klepstand (boolean)
- Foutcodes (integer)
- WasteBin-ID (macadres)

Deze gegevens worden via MQTT naar Node-Red verzonden. Van daaruit worden de gegevens doorgestuurd naar een SQLite-database. We hebben gekozen voor SQLite vanwege de handige SQLiteStudio-omgeving waarin we de database gemakkelijk kunnen ontwerpen, opbouwen en testen.

3.7 Meldingen voor de gebruiker

Voor onze opstelling willen we een luidspreker gebruiken als indicator die verschillende waarschuwingen kan afgeven wanneer dat nodig is. Deze waarschuwingen zijn onder andere:

- De klep staat te lang open (mogelijk)
- De binnenkant van de bak is te warm (mogelijk klep open of slechte koeling)
- Algemene fout (ander probleem)

We hebben ervoor gekozen om deze waarschuwingen ook naar ons dashboard te sturen als ze zich voordoen. Deze waarschuwingen worden weergegeven als foutcodes.

3.8 Verstuur interval

In principe worden alle gegevens elke 15 minuten verstuurd. We denken dat dit voldoende informatie is voor elke vuilnisbak om eventuele problemen te detecteren (bijvoorbeeld te lage temperatuur). Bovendien voorkomt dit dat de database overvol raakt met gegevens, vooral wanneer er 30 vuilnisbakken zijn.

3.8.1 In geval van fouten

In geval van een fout wordt dit interval genegeerd. Het interval wordt dan 2 minuten, indien er zich één of meerdere fouten blijven voordoen.

3.9 Node-Red

We zullen Node-Red gebruiken om alle gegevens van MQTT te ontvangen, te verwerken en weer te geven. Alle ontvangen informatie wordt weergegeven via een Node-Red UI-tabblad. Dit UI-tabblad, ook wel "dashboard" genoemd in Node-Red, is uitbreidbaar indien gewenst. Alle prototype vuilnisbakken kunnen in real-time worden weergegeven op het dashboard als dat gewenst is.

De Node-Red UI ziet er als volgt uit voor twee vuilnisbakken:

3.9.1 Node-Red flow

Om dit alles te laten werken, hebben we een Node-Red flow gemaakt. De Node-Red flow ziet er als volgt uit:

In deze flow ontvangt de eerste node (paars) de MQTT-gegevens. Vervolgens wordt er een tijdstempel aan het bericht toegevoegd en worden alle inkomende berichten gescheiden met behulp van een schakelaar (switch). De gegevens worden gescheiden op basis van het macadres (uniek) van de vuilnisbak.

Nadat de berichten zijn gescheiden, komen ze in een grote functieblok (oranje) dat het bericht analyseert en verdeelt over de uit puts van het functieblok. Ik heb hiervoor gekozen, zodat het functieblok modulair blijft en de uitvoer flexibel is. De uitvoer wordt gestuurd naar dashboardblokken (blauw) en naar nieuwe functieblokken die extra gegevens toevoegen (zoals sensor-ID) en deze als query naar een SQLite-database sturen.

3.10 Database

Alle informatie wordt via Node-Red doorgestuurd naar een SQLite-database. Het doel van de database is om een doorzoekbare geschiedenis van alle ontvangen gegevens van de vuilnisbakken te creëren en een overzicht te bieden van alle gebruikers (testpersonen) en hun gegevens.

De database is als volgt opgebouwd:

Hierbij zijn alle onderstreepte woorden sleutelwaarden. Helaas toont het Database Entity Relationship-diagram (DBER) niet alle details, maar het geeft wel een goed overzicht van de databasestructuur.

De tabellen en relaties in de database zijn als volgt:

Tabelnaam	Veldnaam	Type	Relatie
Users	UserID	INTEGER	PRIMARY KEY
	Name	TEXT	
	Email	TEXT	
	Phone	TEXT	
WasteBins	WasteBinID	INTEGER	PRIMARY KEY
	Location	TEXT	
	SensorID	INTEGER	
	UserID	INTEGER	FOREIGN KEY
SensorData	SensorDataID	INTEGER	PRIMARY KEY
	WasteBinID	INTEGER	FOREIGN KEY
	Timestamp	TIMESTAMP	
	Temperature	REAL	
	Humidity	REAL	
	ValveStatus	INTEGER	
SensorStatus	ErrorCode	INTEGER	
	SensorID	INTEGER	PRIMARY KEY
	Status	TEXT	
	LastUpdated	TIMESTAMP	

Dit is de tabelstructuur voor de database, waarbij elke tabel de bijbehorende velden en datatypes bevat. De tabelrelaties zijn aangegeven met "PRIMARY KEY" en "FOREIGN KEY" om de verbanden tussen de tabellen weer te geven.

De database kan worden doorzocht met behulp van "views" in SQLiteStudio. Deze kunnen door de gebruiker worden gemaakt en aangepast.

Dit is een overzicht van de opzet voor het gebruik van meerdere ESP32-microcontrollers met Node-Red. Met deze configuratie kunnen we gegevens loggen van verschillende sensoren of apparaten, deze visualiseren op een dashboard en ze opslaan in een database voor verdere analyse en beheer.

Chapter 4

Ontwerpkeuzes

technische specificaties:

#	MoSCoW	Descriptie
T		Het koelen wordt gedaan door middel van een Peltier-module.
T		We tonen met led lampjes aan of de temperatuur te warm of te koud is en of de bak leeg of vol is.
T		We geven met een Piezo speaker het signaal dat de klep te lang open staat.
T		We meten de klepstand met een hall effect sensor of een knop.
T		Voor de testopstelling gebruiken wij een Raspberry Pi.

functionele specificaties:

#	MoSCoW	Descriptie
F1	M	Het product moet de verspreiding van de GFE-afval geur voorkomen.
F1	M	Het GFE-rottingsproces moet vier dagen door het product worden geremd.
F1.1	M	Het GFE-afval moet gekoeld worden.
F1.↔ 1.1	S	Het product moet zo efficiënt mogelijk koelen.
F2	M	Prijs moet onder de €50 blijven voor de consument bij afname van 100 stuks.
F3	S	Het product moet indicatoren hebben voor de gebruiker wanneer deze te lang open blijft staan.
F3.1	S	Het product moet een temperatuurindicator (te warm/te koud) hebben.
F3.2	S	Het product moet een inhoudshoeveelheidindicator (vol/leeg) hebben.
F4	M	Het product heeft draadloze connectiviteit voor testdoeleinden.
F5	C	Het product moet feedback aan de gebruiker geven.
F5.1	C	Het product moet feedback geven d.m.v. audio.
F5.2	C	Het product moet feedback geven d.m.v. licht/beeld.
F6	W	Het product moet beschikken over een "off the grid" functie.
F7	M	Het apparaat mag niet meer dan €50 kosten.
F8	M	Prullenbakinhoud moet 6 liter zijn, afgestemd op het gemiddelde GFE-afval van een 4-persoonshuishouden.
F9	S	Het apparaat moet zo stil mogelijk zijn tijdens operatie.
F10	C	Het product moet weten of het geopend of gesloten is.
F11	S	Energie Terugwinning
F12	C	Klep detectie

#	MoSCoW	Descriptie
F13	C	Logfunctie

Chapter 5

Testplan

5.1 Inhoud

- Peltier modules
- Doelen
- Deksel positie
- Temperatuur en luchtvochtigheid
- Testopstelling
- Vereiste componenten
- Meetapparatuur
- Methodes
- Testresultaten

5.2 Peltier modules

De Peltier modules worden verder bestudeerd en getest om te kijken of deze geschikt zijn om te gebruiken voor onze koeling. We gaan gebruik maken van een 12V module. Dit omdat het de standaard is voor Peltier-modules en omdat de spanning afhankelijk is van het formaat vermoeden wij dat een 5V module niet genoeg warmte kan verplaatsen voor onze bak.

5.3 Doelen

Met dit onderzoek willen wij de volgende kwesties benaderen en uitwerken:

- Kunnen we het volledige systeem op 12V laten werken door een buck-converter te gebruiken voor componenten die 5V nodig hebben?
- Wat is de spanning die de ventilatoren vereisen en kunnen we dit regelen om zo een stilleren functionaliteit te krijgen? Hoeveel effect heeft dit op het koelen van de Peltier-module(s)?

5.4 Deksel positie

Het meten van de dekselpositie zullen wij doen d.m.v. een drukknopje of een hall effect sensor in combinatie met een magneet. De werking ervan hoeft naar onze mening niet getest te worden. Een knop is te simpel om een serieuze test voor te verzinnen en daarom gaan we voor de aanpak: krijgen we een waarde te zien wanneer we de knop indrukken? Dan werkt de knop.

5.5 Temperatuur en luchtvochtigheid

Omdat wij in de testfase informatie willen verzamelen van onze testopstelling, willen wij onder andere temperatuurmetingen doen om inzicht te krijgen in de werking en stabiliteit van de testopstelling.

5.5.1 Doel #1

Wij willen voor metingen weten wat de temperatuur en de luchtvochtigheid is in de ruimte waarin het GFE-afval zich bevindt. Hiervoor moeten wij:

- Een 2-in-1 sensor (temperatuur en luchtvochtigheid) kiezen en deze verifiëren op werking en accuraatheid (sensoren los testen).
- De sensor installeren in een gecontroleerde opstelling en de resultaten noteren.

5.5.2 Doel #2

Verder zijn wij benieuwd naar de verdeling van de koelmodules over de bak. De temperatuur kunnen we meten door middel van een thermische camera. Met zo'n camera kunnen wij de temperatuurgeleiding en -verspreiding door de bak zien. Het doel hiervan is om te bepalen waar we de Peltier-module het beste kunnen plaatsen.

5.5.3 Doel #3

Kijken of we met behulp van PWM een ventilator op een deel van zijn toeren kunnen

laat draaien om de temperatuur te regelen, maar vooral om te kijken of we de ventilator stiller kunnen krijgen zonder al te veel invloed op de temperatuur te hebben.

5.6 Testopstelling

Er wordt een uitlijning in het deksel gesneden voor de Peltier-module. Deze wordt met de "hete" kant boven hierin gelegd, zodat de heatsink boven zit. Met een thermometer meten wij de temperatuur in de bak, hiervoor wordt een gat gemaakt. De thermometer wordt aangesloten op een Raspberry Pi en hierin wordt de data gelogd en weergegeven. We meten de temperatuur in de tijd en vergelijken dit onder de 12V en 5V modules.

5.7 Vereiste componenten

- Voeding 12V 6A (of labvoeding)
- 2x Peltier modules 12V
- Heatsinks (voor Peltier) met ventilator
- Thermal paste (voor heatsinks)
- Temperatuursensor
- Vochtsensor
- Draden 10AWG, jumper draden (male/female)

5.8 Meetapparatuur

- Multimeter
- Thermometer
- Thermische camera
- Raspberry Pi

5.9 Methodes

Eis	Methode
De temperatuur- en luchtvochtigheidsmeting moet niet meer afwijken dan de aangegeven afwijking.	Meet de temperatuur en luchtvochtigheid met een gekalibreerde meter en vergelijk deze waarden. De eis is voldaan wanneer deze waarden minder afwijken dan de aangegeven afwijking.
De Peltier-modules moeten op hun meest optimale plaats gemonteerd worden.	Door de temperatuurspreiding te meten kunnen wij achterhalen welke plek het beste is voor deze modules.
Aangetoond moet worden hoeveel effect de ventilatoren hebben op de koeling wanneer dit door middel van PWM wordt geregeld.	De koeling wordt gemeten bij een duty cycle van 100%, 87.5%, 75%, 50% en 25%. Hierdoor is duidelijk hoeveel invloed het afvoeren van lucht heeft op de te behalen koelingstemperatuur.

5.10 Testresultaten

Een tijdsopname van de data die door de vuilnisbak is opgeslagen, is in onderstaande tabel te vinden. Hierbij is sensoridnr 1 de temperatuursensor, 2 is de luchtvochtigheid en 3 is de klepstand van de vuilnisbak (knop). Aangezien de vuilnisbak hier nog maar net aanstaat, is de temperatuur nu op kamertemperatuur.

Tijd	Sensoridnr 1 (temperatuur)	Sensoridnr 2 (luchtvochtigheid)	Sensoridnr 3 (klepstand)
00:00:00	20 °C	50%	Gesloten
00:05:00	18 °C	48%	Open

Tijd	Sensoridnr 1 (temperatuur)	Sensoridnr 2 (luchtvochtigheid)	Sensoridnr 3 (klepstand)
00:10:00	16 °C	45%	Open
00:05:00	18 °C	48%	Open
00:10:00	16 °C	45%	Open
00:15:00	14 °C	42%	Open
00:20:00	13 °C	40%	Open
00:25:00	13 °C	39%	Open
00:30:00	13 °C	38%	Open
00:35:00	13 °C	37%	Open

Later neemt deze temperatuur zeer sterk af tot het ingestelde punt: 13 graden Celsius.

I.v.m. energiekosten en de slechte isolatie van de testopstelling kiezen wij ervoor om de temperatuur niet verder te laten dalen. Wat ons opviel is dat zelfs bij het ingestelde punt de heatsink niet erg warm wordt. Wij vermoeden daarom dat wij de temperatuur nog verder kunnen laten dalen als we dat willen. Helaas staat de ventilator wel voluit te blazen bij het koelen en is het geluidsniveau redelijk hoog. Naar onze mening kan dit worden verholpen op 2 manieren:

1. Verander de huidige ventilator naar een ander, hoger kwaliteitsmodel, met onder andere betere lagers.
2. Vergroot de heatsink of maak de heatsink zelfs een deel van de vuilnisbakbehuizing (metalen vuilnisbak).

Dit zijn de resultaten van het testplan voor de ESE op 8 februari 2023. Met deze resultaten kunnen we beoordelen of de Peltier-modules geschikt zijn voor onze koeling, hoe we de ventilatoren kunnen regelen om het geluidsniveau te verminderen en of we de temperatuur en luchtvochtigheid nauwkeurig kunnen meten in de testopstelling.

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

waste_bin_controller.h	21
--	----

Chapter 7

File Documentation

7.1 bureaubladonderzoek.md File Reference

7.2 introductie.md File Reference

7.3 IoT oplossing.md File Reference

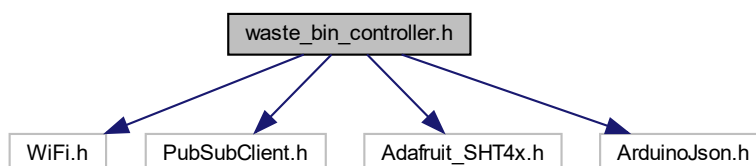
7.4 Ontwerpkeuzes.md File Reference

7.5 testplan.md File Reference

7.6 waste_bin_controller.h File Reference

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_SHT4x.h>
#include <ArduinoJson.h>
```

Include dependency graph for waste_bin_controller.h:



Functions

- void [setupWiFi](#) ()
- void [setupPeltierAndFan](#) ()
- void [setupSHT4x](#) ()
- void [setupMQTTClient](#) ()
- void [readSensorData](#) ()
- void [updateLidPosition](#) ()
- void [controlPeltierModule](#) ()
- void [publishSensorData](#) ()
- void [handleIncomingMessage](#) (char *topic, byte *payload, unsigned int length)
- void [publishMessage](#) (const char *topic, const float temperature, const float humidity, const bool [lid_position](#))
- void [setup](#) ()
- void [loop](#) ()

Variables

- const char * [SSID](#)
- const char * [PASSWORD](#)
- const char * [MQTT_SERVER](#)
- const char * [MQTT_USERNAME](#)
- const char * [MQTT_PASSWORD](#)
- const int [MQTT_PORT](#)
- const char * [CLIENT_NAME](#)
- const char * [SENSOR_DATA_TOPIC](#)
- const uint8_t [peltierandfanpin](#)
- const long [interval](#)
- int [peltierandfanstate](#)
- WiFiClient [wifi_client](#)
- PubSubClient [mqtt_client](#)
- unsigned long [last_msg_sent](#)
- const unsigned long [msg_interval](#)
- unsigned long [previousMillis](#)
- const int [setPoint](#)
- const int [tolerance](#)
- const int [pwmFreq](#)
- const int [BUTTON_PIN](#)
- bool [lid_position](#)
- float [humidity_value](#)
- float [temperature_value](#)
- Adafruit_SHT4x [sht4](#)

7.6.1 Function Documentation

7.6.1.1 [controlPeltierModule\(\)](#)

```
void controlPeltierModule ( )
```

7.6.1.2 handleIncomingMessage()

```
void handleIncomingMessage (
    char * topic,
    byte * payload,
    unsigned int length )
```

7.6.1.3 loop()

```
void loop ( )
```

7.6.1.4 publishMessage()

```
void publishMessage (
    const char * topic,
    const float temperature,
    const float humidity,
    const bool lid_position )
```

7.6.1.5 publishSensorData()

```
void publishSensorData ( )
```

7.6.1.6 readSensorData()

```
void readSensorData ( )
```

7.6.1.7 setup()

```
void setup ( )
```

7.6.1.8 setupMQTTClient()

```
void setupMQTTClient ( )
```

7.6.1.9 setupPeltierAndFan()

```
void setupPeltierAndFan ( )
```

7.6.1.10 setupSHT4x()

```
void setupSHT4x ( )
```

7.6.1.11 setupWiFi()

```
void setupWiFi ( )
```

7.6.1.12 updateLidPosition()

```
void updateLidPosition ( )
```

7.6.2 Variable Documentation

7.6.2.1 BUTTON_PIN

```
const int BUTTON_PIN [extern]
```

7.6.2.2 CLIENT_NAME

```
const char* CLIENT_NAME [extern]
```

7.6.2.3 humidity_value

```
float humidity_value [extern]
```

7.6.2.4 interval

```
const long interval [extern]
```

7.6.2.5 last_msg_sent

```
unsigned long last_msg_sent [extern]
```

7.6.2.6 lid_position

```
bool lid_position [extern]
```

7.6.2.7 mqtt_client

```
PubSubClient mqtt_client [extern]
```

7.6.2.8 MQTT_PASSWORD

```
const char* MQTT_PASSWORD [extern]
```

7.6.2.9 MQTT_PORT

```
const int MQTT_PORT [extern]
```

7.6.2.10 MQTT_SERVER

```
const char* MQTT_SERVER [extern]
```

7.6.2.11 MQTT_USERNAME

```
const char* MQTT_USERNAME [extern]
```

7.6.2.12 msg_interval

```
const unsigned long msg_interval [extern]
```

7.6.2.13 PASSWORD

```
const char* PASSWORD [extern]
```

7.6.2.14 peltierandfanpin

```
const uint8_t peltierandfanpin [extern]
```

7.6.2.15 peltierandfanstate

```
int peltierandfanstate [extern]
```

7.6.2.16 previousMillis

```
unsigned long previousMillis [extern]
```

7.6.2.17 pwmFreq

```
const int pwmFreq [extern]
```

7.6.2.18 SENSOR_DATA_TOPIC

```
const char* SENSOR_DATA_TOPIC [extern]
```

7.6.2.19 setPoint

```
const int setPoint [extern]
```


7.6.2.20 sht4

```
Adafruit_SHT4x sht4 [extern]
```

7.6.2.21 SSID

```
const char* SSID [extern]
```

7.6.2.22 temperature_value

```
float temperature_value [extern]
```

7.6.2.23 tolerance

```
const int tolerance [extern]
```

7.6.2.24 wifi_client

```
WiFiClient wifi_client [extern]
```

7.7 waste_bin_controller.h

[Go to the documentation of this file.](#)

```
1 #ifndef WASTE_BIN_CONTROLLER_H
2 #define WASTE_BIN_CONTROLLER_H
3
4 #include <WiFi.h>
5 #include <PubSubClient.h>
6 #include <Adafruit_SHT4x.h>
7 #include <ArduinoJson.h>
8
9 // WiFi settings
10 extern const char* SSID;
11 extern const char* PASSWORD;
12
13 // MQTT Broker settings
14 extern const char* MQTT_SERVER;
15 extern const char* MQTT_USERNAME;
16 extern const char* MQTT_PASSWORD;
17 extern const int MQTT_PORT;
18
19 extern const char* CLIENT_NAME;
20
21 // MQTT topics
22 extern const char* SENSOR_DATA_TOPIC;
23
24 // Device settings
25 extern const uint8_t peltierandfanpin;
26 extern const long interval;
```

```
27 extern int peltierandfanstate;
28
29 // PubSubClient setup
30 extern WiFiClient wifi_client;
31 extern PubSubClient mqtt_client;
32
33 // Other variables
34 extern unsigned long last_msg_sent;
35 extern const unsigned long msg_interval;
36 extern unsigned long previousMillis;
37
38 extern const int setPoint;
39 extern const int tolerance;
40 extern const int pwmFreq;
41
42 // Sensor pin
43 extern const int BUTTON_PIN;
44
45 // Sensor values
46 extern bool lid_position;
47 extern float humidity_value;
48 extern float temperature_value;
49
50 // SHT4x sensor setup
51 extern Adafruit_SHT4x sht4;
52
53 void setupWiFi();
54 void setupPeltierAndFan();
55 void setupSHT4x();
56 void setupMQTTClient();
57
58 void readSensorData();
59 void updateLidPosition();
60 void controlPeltierModule();
61 void publishSensorData();
62 void handleIncomingMessage(char* topic, byte* payload, unsigned int length);
63 void publishMessage(const char* topic, const float temperature, const float humidity, const bool
    lid_position);
64
65 void setup();
66 void loop();
67
68 #endif // WASTE_BIN_CONTROLLER_H
```

Index

bureaubladonderzoek.md, [21](#)
BUTTON_PIN
 waste_bin_controller.h, [24](#)

CLIENT_NAME
 waste_bin_controller.h, [24](#)
controlPeltierModule
 waste_bin_controller.h, [22](#)

handleIncomingMessage
 waste_bin_controller.h, [22](#)
humidity_value
 waste_bin_controller.h, [24](#)

interval
 waste_bin_controller.h, [24](#)
introductie.md, [21](#)
IoT oplossing.md, [21](#)

last_msg_sent
 waste_bin_controller.h, [25](#)
lid_position
 waste_bin_controller.h, [25](#)
loop
 waste_bin_controller.h, [23](#)

mqtt_client
 waste_bin_controller.h, [25](#)
MQTT_PASSWORD
 waste_bin_controller.h, [25](#)
MQTT_PORT
 waste_bin_controller.h, [25](#)
MQTT_SERVER
 waste_bin_controller.h, [25](#)
MQTT_USERNAME
 waste_bin_controller.h, [25](#)
msg_interval
 waste_bin_controller.h, [25](#)

Ontwerpkeuzes.md, [21](#)

PASSWORD
 waste_bin_controller.h, [26](#)
peltierandfanpin
 waste_bin_controller.h, [26](#)
peltierandfanstate
 waste_bin_controller.h, [26](#)
previousMillis
 waste_bin_controller.h, [26](#)
publishMessage
 waste_bin_controller.h, [23](#)

publishSensorData
 waste_bin_controller.h, [23](#)
pwmFreq
 waste_bin_controller.h, [26](#)

readSensorData
 waste_bin_controller.h, [23](#)

SENSOR_DATA_TOPIC
 waste_bin_controller.h, [26](#)
setPoint
 waste_bin_controller.h, [26](#)
setup
 waste_bin_controller.h, [23](#)
setupMQTTClient
 waste_bin_controller.h, [23](#)
setupPeltierAndFan
 waste_bin_controller.h, [23](#)
setupSHT4x
 waste_bin_controller.h, [24](#)
setupWiFi
 waste_bin_controller.h, [24](#)
sht4
 waste_bin_controller.h, [26](#)
SSID
 waste_bin_controller.h, [27](#)

temperature_value
 waste_bin_controller.h, [27](#)
testplan.md, [21](#)
tolerance
 waste_bin_controller.h, [27](#)

updateLidPosition
 waste_bin_controller.h, [24](#)

waste_bin_controller.h, [21](#), [27](#)
 BUTTON_PIN, [24](#)
 CLIENT_NAME, [24](#)
 controlPeltierModule, [22](#)
 handleIncomingMessage, [22](#)
 humidity_value, [24](#)
 interval, [24](#)
 last_msg_sent, [25](#)
 lid_position, [25](#)
 loop, [23](#)
 mqtt_client, [25](#)
 MQTT_PASSWORD, [25](#)
 MQTT_PORT, [25](#)
 MQTT_SERVER, [25](#)
 MQTT_USERNAME, [25](#)

- msg_interval, [25](#)
- PASSWORD, [26](#)
- peltierandfanpin, [26](#)
- peltierandfanstate, [26](#)
- previousMillis, [26](#)
- publishMessage, [23](#)
- publishSensorData, [23](#)
- pwmFreq, [26](#)
- readSensorData, [23](#)
- SENSOR_DATA_TOPIC, [26](#)
- setPoint, [26](#)
- setup, [23](#)
- setupMQTTClient, [23](#)
- setupPeltierAndFan, [23](#)
- setupSHT4x, [24](#)
- setupWiFi, [24](#)
- sht4, [26](#)
- SSID, [27](#)
- temperature_value, [27](#)
- tolerance, [27](#)
- updateLidPosition, [24](#)
- wifi_client, [27](#)
- wifi_client
 - waste_bin_controller.h, [27](#)