

Documentatie - Proof of Concept & Applicatie

IDP Waterkering

Gemaakt door:
Bart Hiemstra
Sijmen Jaarsma
Vince van Noort

Inhoudsopgave

Documentatie -	
Proof of Concept & Applicatie	1
Inhoudsopgave	2
Proof of Concept	3
Functioneel Ontwerp	4
Technisch Ontwerp	6
Sensor	6
Motoren	6
Applicatie	6
Websockets	7
Programmeertaal	7
Threads	7
Failover	7
Fritzing model	8
Algemene werking	9
Motor thread	9
Sensor thread	10
Testen van het systeem	11
Doel	11
Meten of de ultrasone sensor correct werkt	11
Controleren of de poort open/sluit op basis van de waterstand	11
Testen of het failover systeem correct werkt	12
Testen van de werking van de motoren	12
Bronvermelding	12

Proof of Concept

Het Proof of Concept van de waterkering meet door middel van een ultrasonische sensor de waterstand. Indien het water te hoog staat worden sluisdeuren gesloten. Wanneer het water daarna weer op normaal niveau is, worden de sluisdeuren weer geopend. Door een failover systeem wordt gegarandeerd dat de waterkering altijd zal werken.

In het Proof of Concept zijn de onderstaande onderdelen gebruikt:

Hardware:

- HC-SR04 ultrasonische sensor - 1x
- 28byj-48 stappenmotor - 2x
- Raspberry Pi - 2x
- Jumper Wires (MM, MF, FF)
- Resistors (330ohm & 470ohm)
- Breadboard
- UTP netwerkkabel

Software:

- Django
- Django Channels
- Vue.js & Chart.js

Functioneel Ontwerp

In het Plan van Aanpak is er beschreven wat de applicatie moet kunnen. De beschrijving die wij hebben gegeven luidt:

“Aan het einde van het project, op 3 februari 2017, wil ons team een prototype opleveren dat de maeslantkering nabootst. Dit prototype moet kunnen meten hoe hoog het water staat, de gemeten data opslaan en reageren op de stand van het water. De documentatie die wordt geschreven bij het prototype is duidelijk voor een student HBO-ICT eerstejaars aan de Hogeschool Utrecht. Het schaalmodel zal op zijn minst sensoren voor het meten van de waterstand bevatten, met daarnaast een sluisdeur die bediend wordt door twee stappenmotoren. Indien haalbaar willen we dit systeem werkzaam maken in een bak met water. Aan het einde van het project zullen wij een posterpresentatie houden die voldoet aan de eisen van het iPass project.”

Wat moet de het proof of concept concreet kunnen?

1. **De applicatie moet kunnen draaien op een Raspberry Pi 3 Model B+**

De Raspberry Pi 3 Model B+ is een minicomputer, ontworpen om applicaties te draaien. Ons team wil graag een Python programma laten draaien op zo'n minicomputer. Hiervoor moeten we rekening houden met wat de Raspberry Pi qua computerkracht aankan.

2. **De waterstand moet per seconde kunnen worden gemeten door een ultrasonische sensor**

De Raspberry Pi moet een ultrasonische sensor kunnen aansturen om de waterstand te meten. Een ultrasonische sensor is een sensor die een geluidsgolf afschiet met een erg hoge frequentie (onhoorbaar voor mensen). Deze geluidsgolf weerkaatst op een object. De tijd die de geluidsgolf erover doet om terug te komen naar de sensor, kan worden omgerekend naar een afstand op basis van de snelheid van geluid (343 meter per seconde).

3. **De waterstanden moeten worden opgeslagen in een SQLite database**

De gemeten afstand moet kunnen worden opgeslagen. Dit willen wij bereiken door dit in een SQLite database te doen. Een SQLite database is een enkel bestand die fungeert als database. In deze database kun je makkelijk data ophalen en opslaan.

4. **De twee motoren moeten op basis van de gemeten waterstand worden aangestuurd**

De motoren moeten worden aangestuurd door de applicatie op basis van de laatste paar metingen. Wanneer de gemeten waterstand over een langere tijd een ingestelde grens overschrijdt, zal de applicatie de motoren moeten aansturen om open of dicht te gaan.

5. De twee motoren moeten deuren kunnen dragen en een nauwkeurige aantal graden kunnen draaien

De door de Raspberry Pi gestuurde motoren moeten een nauwkeurig aantal graden kunnen draaien terwijl deze de deuren van de waterkering draagt. Het aantal nauwkeurige graden moet kunnen worden beheerd in de applicatie.

6. Meldingen van het aansturen van de deuren moeten worden opgeslagen in een database

De meldingen die worden gegeven door de applicatie wanneer de deuren worden aangestuurd, moeten kunnen worden opgeslagen in een database (hiervoor benoemd).

7. De gemeten waterstanden en meldingen moeten in een webapplicatie worden getoond, hierin moet je een grafiek kunnen zien waarin je de waterstand van afgelopen minuut kunt bekijken

De waterstanden moeten worden weergegeven in een grafiek via de webapplicatie. Een webapplicatie is een applicatie die je via een browser kunt bekijken en hiermee interacteren. De grafiek biedt de optie om de geschiedenis van de waterstand te kunnen zien.

8. De sensoren en motoren moeten kunnen worden aangestuurd door een Raspberry Pi en indien deze uitvalt door een backup Raspberry Pi (ook wel Failover genoemd)

De waterkering moet met een failover systeem kunnen draaien. Dat betekent dat wanneer de Raspberry Pi (die de sensoren en motoren aanstuurt) uitvalt, een andere Raspberry Pi dit kan overnemen. Dit zorgt ervoor dat wanneer er geen stroom meer is of de applicatie crasht, het proces niet komt stil te staan.

Technisch Ontwerp

In de onderstaande afbeelding is een representatie van het technisch ontwerp te zien, waarin de vereiste onderdelen zijn verwerkt. Hierin zijn de sluisdeuren weggelaten om alleen op het technisch gedeelte in te gaan.

Sensor

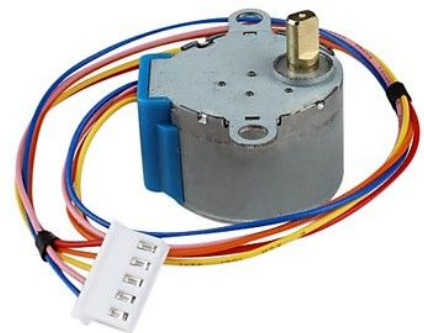
De HC-SR04 ultrasonische sensor wordt gebruikt in het PoC. Er is voor deze sensor gekozen omdat het een goedkope maar zeer nauwkeurige sensor is. De sensor werkt op basis van geluid, zoals de naam al omschrijft. De sensor kan op een afstand van 2 - 500 cm op 0.3cm nauwkeurig meten, na deze afstand werkt de sensor niet meer naar behoren. De sensor zal de afstand meten door de ultrasonische golf op het water te weerkaatsen en deze terug te ontvangen in de sensor. De sensor heeft vier pinnen waaronder: stroom, trigger, echo en aarde. Op bladzijde [BLADZIJDE] is gedocumenteerd hoe de ultrasonische sensor in het PoC is verwerkt.



Verantwoording: nauwkeurig en goedkoop

Motoren

De 28byj-48 stappenmotor wordt gebruikt om de sluisdeuren aan te sturen. Deze stappenmotoren zijn gekozen omdat ze aan kunt sturen in een aantal graden en omdat ze heel goedkoop beschikbaar zijn. Wanneer je een 28byj-48 stappenmotor online bestelt, krijg je een motordriver bord erbij. De motoren werken door het aanzetten van verschillende magnetische velden. Door dit snel achter elkaar te doen gaat de motor draaien. Op bladzijde [BLADZIJDE] is gedocumenteerd hoe de stappenmotoren in het PoC zijn verwerkt.



Verantwoording: aantal graden aanstuurbaar en goedkoop

Applicatie

Om de applicatie is gekozen voor het Python Framework Django. De keuze voor Django is gebaseerd op dat Django betrouwbaar is en het toelaat om een applicatie snel te programmeren. Pinterest & Disqus zijn goede voorbeelden voor de betrouwbaarheid van Django, applicaties die dagelijks door miljoenen mensen worden gebruikt.

Verantwoording: betrouwbaar

django

Websockets

Om vanaf meerdere browsers de webapplicatie synchroon te kunnen bekijken hebben we gekozen voor Django Channels. Een implementatie van websockets in Django. De websockets zorgen ervoor dat elke openstaande browser dezelfde updates door krijgt van de server. Zo zouden er meerdere schermen in een gebouw kunnen staan die de waterstanden door krijgen.

Verantwoording: synchroon monitoring en dichte integratie met Django

Programmeertaal

Om de applicatie te programmeren hebben we samen met het framework Django gekozen voor Python. We hebben deze keuze gemaakt omdat Django is gebouwd op Python en om deze reden is de applicatie relatief erg snel in elkaar te zetten. Python is een programmeertaal die makkelijk te begrijpen is maar weinig grenzen heeft naast snelheid vergeleken met talen als C/C# en Java.

Verantwoording: prototype snel op te stellen

Threads

We hebben er voor gekozen om met threads te werken voor de Applicatie. De reden hiervoor is dat we kunnen garanderen dat een lopende functie de andere functie niet blokkeert. In de applicatie zijn er naast de standaard Django threads, 2 threads die constant zichzelf herhalen. De implementatie van deze threads staat verderop in dit document beschreven.

Verantwoording: parallelle processen

Failover

Voor de failover hebben we gekozen om twee Raspberry Pi's met een ethernet kabel met elkaar te verbinden. De verbinding gaat via een veilige SSH verbinding met encryptie. Er zijn 2 verschillende type Raspberry Pi's. De hoofd Raspberry Pi, en de failover Raspberry Pi. De hoofd Raspberry Pi kopieert elke 20 seconden de geupdate SQLite database file naar de Failover Raspberry Pi. De failover Raspberry Pi controleert elke 20 seconden of het databasebestand (dat vanaf de hoofd Raspberry Pi gekopieerd zou moeten zijn), niet ouder is dan 60 seconden. Indien het de hoofd Raspberry Pi, 3 keer niet is gelukt om het databasebestand te kopiëren, zal het databasebestand op de failover Raspberry Pi ouder zijn dan 60 seconden en zal de failover Raspberry Pi zijn eigen Django applicatie starten.

Kort uitgelegd:

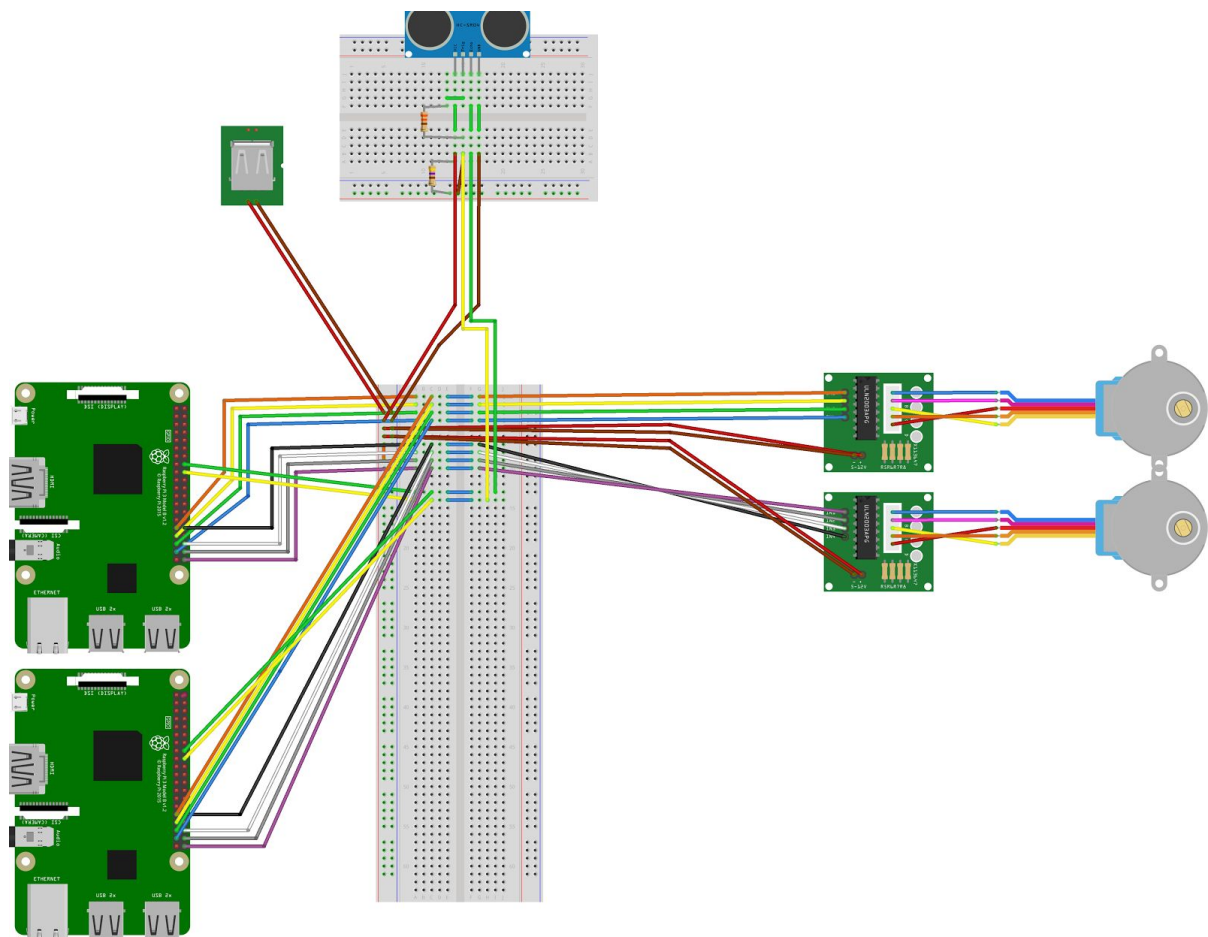
1. De hoofd Raspberry Pi stuurt elke 20 seconden de meest recente database file naar de failover Raspberry Pi
2. De failover Raspberry Pi kijkt elke 20 seconden of het databasebestand (op het eigen apparaat) niet ouder is dan 60 seconden (de hoofd Raspberry kan er 3 keer

niet in slagen om het bestand te kopiëren). Wanneer dit het geval is zal deze zelf de Applicatie starten.

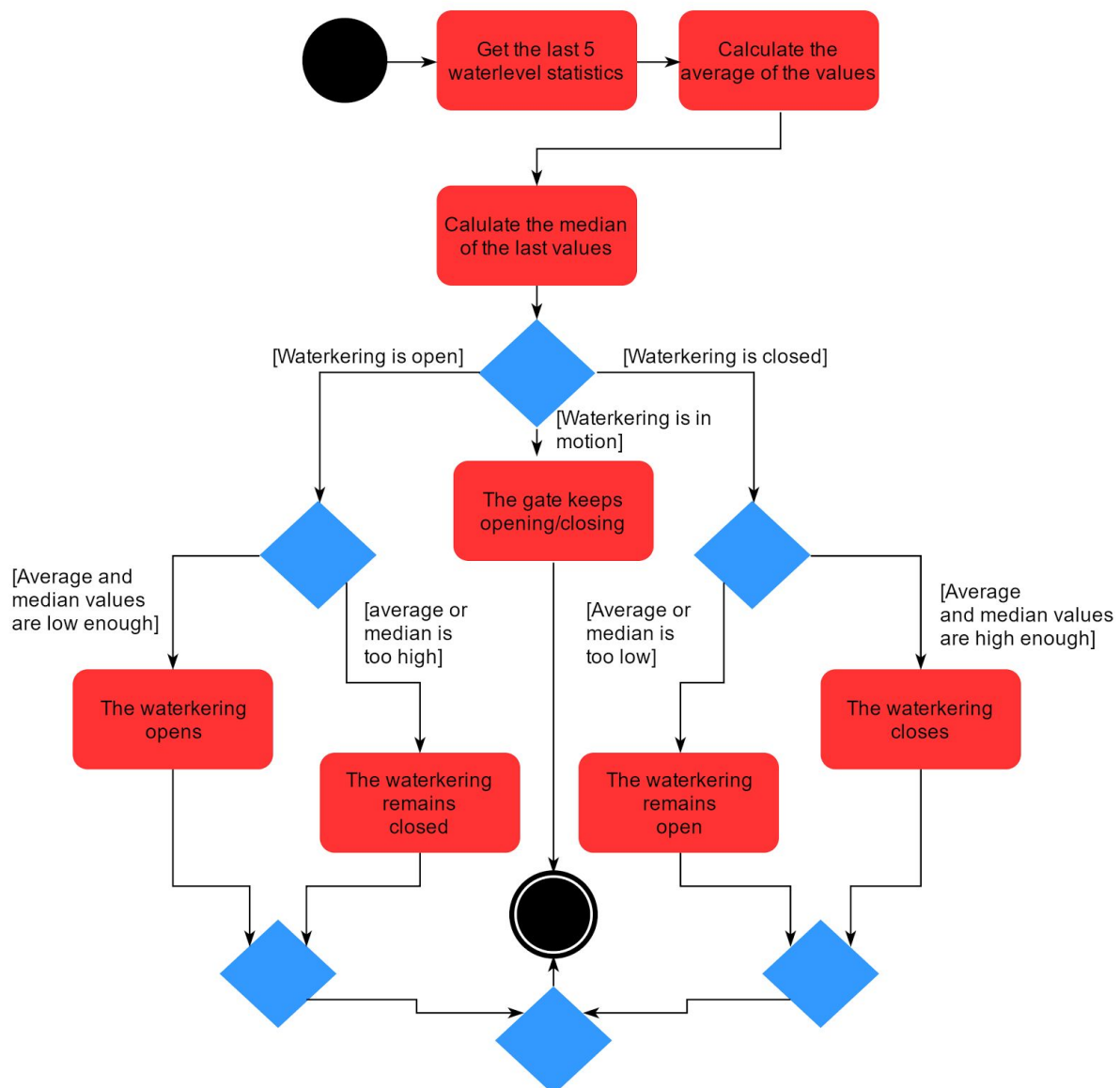
Verantwoording: betrouwbaarheid en synchrone database

Fritzing model

Voor het aansluiten van de 2 Raspberry Pi's op de ultrasonische sensor en de stappenmotoren is het onderstaande technisch model te volgen.



Algemene werking

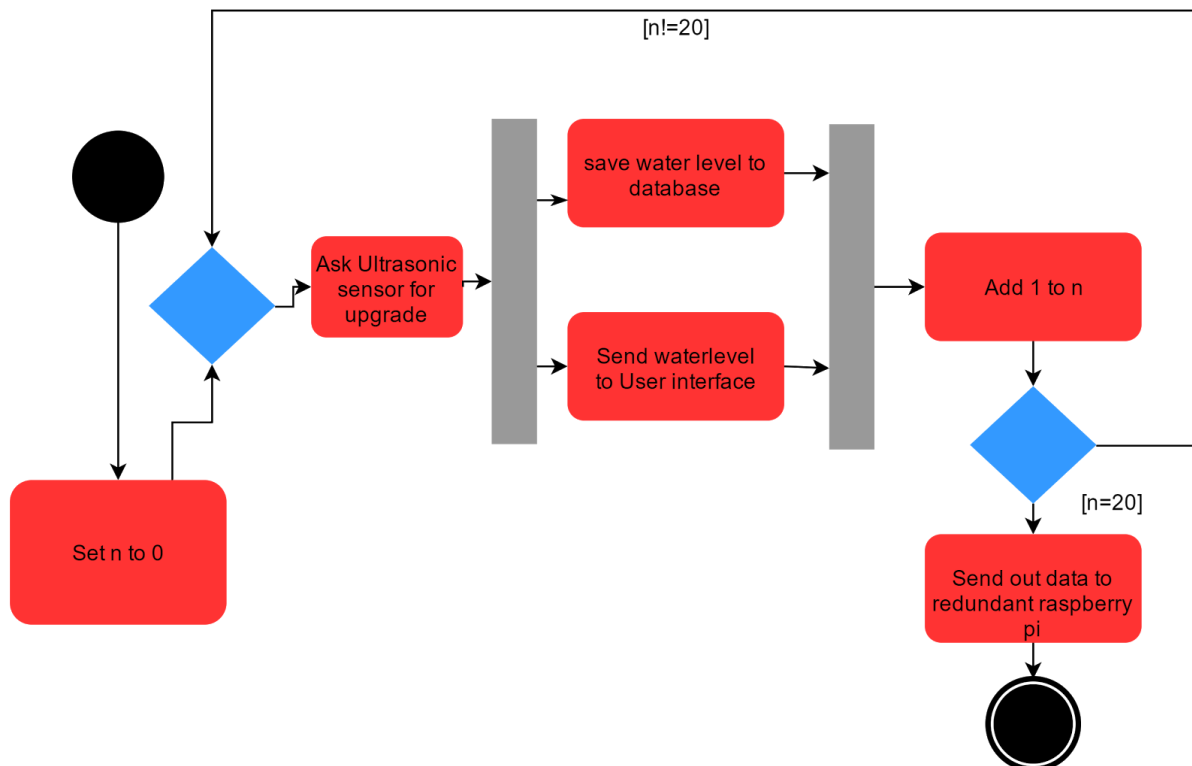


Motor thread

Algemene werking applicatie:

1. De controller haalt de laatste 5 waterstanden op uit de database.
2. Van deze 5 waarden wordt het gemiddelde en de mediaan (de mediaan is de middelste waarde uit een gesorteerde lijst (bijvoorbeeld in [1; **4**; 5] is 4 de mediaan)) berekend.
3. De waterkering controleert of de poort open, dicht of in beweging is:
 - a. als de poort open is wordt gecontroleerd of de poort dicht moet. Dit wordt gedaan door te checken of het gemiddelde en de mediaan hoger zijn dan de vastgestelde maximale waterhoogte.

- b. als de poort dicht is is wordt gecontroleerd of de poort open moet. Dit wordt gedaan door te checken of het gemiddelde en de mediaan lager zijn dan de vastgestelde maximale acceptabele waterhoogte.
- c. als de waterkering in beweging is veranderd er niks, de waterkering blijft open of dicht gaan.



Sensor thread

1. Tel 20 bij n op;
2. De ultrasone sensor krijgt de opdracht een mee om een nieuwe waarde op te meten;
3. voert de volgende 4 stappen 20 keer terug;
4. de ultrasone sensor geeft de waarden terug;
5. De thread slaat het opgehaalde waarde en de huidige tijd op in de django database;
6. De thread geeft de waterdata door aan de User Interface die dit weergeeft in zijn visuals;
7. tel een bij n op;
8. Update de database van de redundante raspberry pi.

Testen van het systeem

Doel

Het maken van een proof of concept waterkering die de waterstand meet en aan de hand van de binnengehaalde waarden bepaalt of de poort moet openen of sluiten. Verder moeten er een aantal tests uitgevoerd worden om te controleren of alle aspecten van het systeem goed werken.

Te testen onderdelen

Hieronder staan de testmethoden uitgewerkt met daarbij het resultaat van elke test.

Meten of de ultrasone sensor correct werkt

Benodigdheden

- Verstelbaar waterreservoir
- Ultrasone sensor
- Liniaal

Testmethode

De afstand tussen de ultrasone sensor en het wateroppervlak wordt gemeten met behulp van een liniaal. Vervolgens wordt de afstand afgelezen die de sensor geeft. Als de waardes overeenkomen dan is de ultrasone sensor correct ingesteld.

Testresultaat

De waardes van de ultrasone sensor komen overeen met wat de liniaal aangeeft.

Controleren of de poort opent/sluit op basis van de waterstand

Benodigdheden

- Geteste ultrasone sensor
- Gemotoriseerde poorten
- Verstelbaar waterreservoir

Testmethode

Het systeem wordt aangezet, waarna een waterreservoir wordt gevuld en vervolgens weer geleegd. Daarbij wordt gecontroleerd of de poorten correct sluiten wanneer het reservoir te vol zit, en weer openen wanneer het reservoir te leeg is. Deze test wordt 4 keer uitgevoerd om er zeker van te zijn dat alles werkt.

Testresultaat

De poort opent en sluit op het goede moment.

Testen of het failover systeem correct werkt

Benodigheden

- Een volledig redundant systeem met 2 Raspberry Pi's.

Testmethode

De Raspberry Pi die verantwoordelijk is voor het aansturen van de sluisdeuren wordt opzettelijk gestopt. Vervolgens wordt gekeken of de tweede Raspberry Pi het werk van de eerste overneemt. Deze test herhalen we 4 keer. Als het failover systeem in alle gevallen werkt wordt het als correct werkend beschouwd.

Testresultaat

Het failover systeem werkt correct.

Testen van de werking van de motoren

Benodigheden

- Alle andere tests succesvol uitgevoerd
- Het geheel opgestelde systeem.

Testmethode

De test knoppen 'Open gate' en 'Close gate' die zich in de webapplicatie bevinden worden kort na elkaar ingedrukt. Er wordt gekeken of de poort goed opent en weer sluit. Als dat zo is wordt de test als succesvol beschouwd.

Testresultaat

De poorten openen en sluiten correct na het drukken op de knoppen. De motoren werken dus zoals gewenst.

Bronvermelding

1. Rijkswaterstaat. Geraadpleegd van: <https://www.rijkswaterstaat.nl/water/waterbeheer/bescherming-tegen-het-water/waterkeringen/index.aspx>
2. Tutorials point. (z.j.). Python Multithreaded Programming. Geraadpleegd van: https://www.tutorialspoint.com/python/python_multithreading.htm
3. Python - Tutorial. (z.j.). Geraadpleegd van <https://www.tutorialspoint.com/python/>

4. Barnett, J. (2014, 11 april). Controlling DC Motors Using Python With a Raspberry Pi.
Geraadpleegd van <https://business.tutsplus.com/tutorials/controlling-dc-motors-using-python-with-a-raspberry-pi--cms-20051>
5. Raspberry, M. (2012, 13 augustus). Reading Analogue Sensors With One GPIO Pin.
Geraadpleegd van <http://www.raspberrypi-spy.co.uk/2012/08/reading-analogue-sensors-with-one-gpio-pin/>
6. Raspberry, M. (2012, 13 augustus). Ultrasonic distance measurement using python part 2. Geraadpleegd van <http://www.raspberrypi-spy.co.uk/2013/01/ultrasonic-distance-measurement-using-python-part-2/>
7. Wiles, F., Procida, D., Bennett, J., Conley, R., Love, K., & Alger, K. W. (z.j.). Django.
Geraadpleegd van <https://www.djangoproject.com/>
8. Stateham, B. (2014, 11 april). 28BYJ-48 Stepper Motor and ULN2003 Driver Intro.
Geraadpleegd van <https://www.youtube.com/watch?v=B86nqDRskVU>