

Programmeerproject 2 verslag fase 2

Inhoud

1.	Inleiding	2
2.	Abstracte Data Types.....	2
2.1.	ADT majordomo	2
2.2.	ADT simulation	3
2.3.	ADT xbee-simulation	3
2.4.	ADT steward	4
2.5.	ADT device	5
2.6.	ADT database-connection	5
2.7.	ADT log file	7
2.8.	ADT gui	7
3.	Relaties tussen ADT's	8
4.	Communicatieprotocol.....	8
5.	Conclusie	9
6.	Bronnen	9

1. Inleiding

Dit verslag gaat over de tweede en laatste implementatiefase van een domotica-energie-monitoringsysteem. In de vorige fase is de majordomo (huisopzichter), de grafische interface en de databank geïmplementeerd; de input van de steward, de actoren en sensoren werden op een simplistische manier gesimuleerd. In deze fase wordt de steward en zijn actoren vervangen door een simulatie die overeenstemt met de werkelijkheid; het is mogelijk de simulatie te laten vallen en enkel echte devices te gebruiken. Tijdens de simulatie is er één Raspberry Pi die de stewards en devices runt.

2. Abstracte Data Types

Nu volgt er een kort overzicht van de gebruikte ADT's en de boodschappen die ze begrijpen, samen met hun argumenten. Verder wordt de samenwerking tussen de verschillende ADT's toegelicht.

2.1. ADT majordomo

Dit ADT houdt alle stewards bij en verzekert correcte communicatie met een zekere steward. De majordomo is ook verantwoordelijk voor het correct opstarten van het hele systeem. Verder zorgt de majordomo ervoor dat de communicatie naar de devices gecontroleerd wordt. Dit wil zeggen dat er maar één procedure tegelijk mag schrijven naar of wachten op antwoord van een device. Als laatste vraagt de majordomo elke 60 seconden het verbruik op van alle devices (die een verbruiksmeter hebben).

Message	Parameters meegegeven met oproep	Resultaat
Add-steward	Steward naam	Maakt een nieuwe steward aan met een correcte naam en voegt deze toe aan de lijst van stewards.
Add-device	Target steward Device naam Device protocol Product id	Stuurt een bericht naar de juiste steward om te vragen een zeker device aan te maken, met een zeker protocol.
Erase-steward	Steward naam	Verwijdert een zekere steward van het systeem. Bij de simulatie wordt er een bericht gestuurd naar de raspberry pi om de juiste steward te verwijderen. In de echte applicatie zal dit niet moeten gebeuren.
Erase-device	Target steward Device naam	Stuurt een bericht naar de juiste steward om te vragen een bepaald device te verwijderen.
Switch-device	Steward naam Device naam On (booleaan = #t/#f)	Van zodra de majordomo aangeeft dat het communicatiekanaal vrij is, stuurt deze procedure een bericht naar de juiste steward om een bepaald device aan of uit te zetten. Als de meegegeven booleaanse waarde true is wordt het device aangezet, anders wordt het device uitgezet.

Device-turned-on?	Target steward Device naam	Van zodra de majordomo aangeeft dat het communicatiekanaal vrij is, vraagt deze procedure aan de steward om te kijken of een bepaal device aan of uit staat. Dit wordt gebruikt bij het opstarten van de majordomo om te kijken of de lampen aan of uit staan.
Initiate	Geen	Start het systeem op: alle stewards en devices uit de database worden opgevraagd en aangemaakt. Dan wordt de gui opgestart, waaraan de log-values worden toegevoegd.

2.2.ADT simulation

Zoals al gezegd is zal er in de simulatie één of meerdere steward draaien op éénzelfde Raspberry Pi. Dit is het object dat alles zal aansturen in de simulatie, hij gaat permanent lopen en per steward kijken of er input klaar staat op de input poorten. Als dat het geval is, wordt het overeenkomstige slip object opgeroepen.

Message	Parameters meegeven met oproep	Resultaat
Add-steward	Steward naam	Gaat een nieuw steward object aanmaken en deze toevoegen aan de lijst van stewards.
Remove-steward	Steward naam	Gaat het juiste slip object verwijderen uit de lijst van stewards, in de volgende garbage collect wordt de eigenlijke steward verwijderd.

2.3.ADT xbee-simulation

Het doel van dit ADT is het simuleren van een echte xbee, het wordt ook enkel gebruikt in de simulatie en niet in het eigenlijke product. Het object houdt een lijst bij van tripples bestaand uit een product-id, een 64-bit adres en een racket object. Dit object maakt gebruik van 2 queue's, de eerste noemt de buffer, de tweede noemt de payloadlist.

Message	Parameters meegeven met oproep	Resultaat
Add-device	Device adres Device product id	Deze methode gaat een device toevoegen in de lijst van devices die gevonden kunnen worden in de omgeving van de xbee
Erase-device	Device adres	Verwijdert een zeker device uit de lijst van devices.
Xbee-tick	Geen	Zet de inhoud van de buffer naar de payloadlist en geeft terug hoeveel berichten er zijn toegevoegd aan de payloadlist.
Xbee-ready?	Geen	Geeft de waarde 'true' terug als er nog data in de payloadlist zit; als de queue leeg is wordt false terug gegeven.
Xbee-read	Geen	Geeft de eerste frame van de payloadlist terug indien deze aanwezig is, anders gaat er een error worden gegeven.

Xbee-write	Device adres Bericht	Gaat het racket object zoeken dat overeenkomt met het meegegeven device adres. Daarna gaat het dit object oproepen met het meegegeven bericht als argument. Er worden twee bytevector in de bufferqueue gestoken, één om te zeggen dat het bericht succesvol is aangekomen, één om de output van de oproep in op te slaan.
Xbee-discover	Geen	Geeft een lijst van tuples terug bestaand uit product-ids en hun overeenkomstige 64-bit-adressen.

2.4.ADT steward

Het doel van dit ADT is het regelen van communicatie tussen de majordomo en de devices. Deze communicatie wordt bereikt met behulp van een xbee. Het ADT onthoudt een lijst van devices die zijn toegevoegd, samen met hun protocol en hun 64 bit adres. Ten slotte is het verantwoordelijk voor het vormen van lijsten van datapakketten over al zijn toegevoegde devices.

Een van de belangrijkste functies van de steward is assure-arrival. Deze methode blijft een zeker bericht sturen tot de xbee zegt dat dit bericht is aangekomen. Daarna blijft deze methode frames uit de xbee lezen tot een 0X90 frame wordt terug gegeven die een acknowledge geeft van het opgestuurde bericht.

Message	Parameters meegeven met oproep	Resultaat
Add-device	Device naam Device protocol Device product id	Deze methode gaat kijken of een zeker product-id voorkomt in de lijst van devices, gevonden door de xbee. Als dit het geval is wordt er aan de lijst van devices een nieuw object toegevoegd met de naam, het protocol en het adres van het device. Dit adres wordt gevonden in de lijst van gevonden devices teruggegeven door de xbee. In de simulatie wordt er eerst aan de xbee simulatie gevraagd om een nieuw device toe te voegen.
Erase-device	Device naam	Verwijdert een zekere device uit de lijst van devices. In de simulatie wordt er aan de xbee gevraagd om hetzelfde device te verwijderen.
Turn-on	Device naam	Zoekt het adres van het meegegeven device op en zal zorgen dat het device wordt aangezet, gebruik makend van assure-arrival
Turn-off	Device-naam	Zoekt het adres van het meegegeven device op en zal zorgen dat het device wordt uitgezet, gebruik makend van assure-arrival

Turned-on?	Device-naam	Zoekt het adres van het meegeven device op en zal een 'get' bericht laten sturen naar dit adres, gebruik makend van assure-arrival. Daarna zal uit de teruggegeven bulk informatie ON of OFF gehaald worden en teruggegeven worden aan de majordomo.
Get-usage-all	Geen	Geeft een lijst terug met tuppels devicenaam en deviceverbruik. Enkel devices met een verbruiksmeter komen voor in deze lijst.

2.5.ADT device

Dit ADT is een simulatie van een Zigbee device in het domoticasysteem. Ik heb ervoor gekozen om maar één ADT te maken voor device in plaats van één voor elke actor. Het is de taak van het systeem om enkel de juiste berichten te sturen naar de juiste actoren. Dit betekent dat alle actoren alle berichten verstaan en antwoorden op ook diegene die ze logischerwijs niet verstaan. Een device zonder verbruiksmeter heeft wel de nodige code om juist te antwoorden maar het systeem moet slim genoeg zijn om zo iets nooit te vragen.

Message	Parameters meegeven met oproep	Resultaat
GET	Geen	Geeft bulk informatie weer over het device. Dit houdt in power on/off, hoeveel keer het device wakker wordt, verbruik (als device aan staat),
SET POW=ON	Geen	Zet het apparaat aan en geeft een zigbee 0X90 frame terug met daarin ack: pow= on.
SET POW=OFF	Geen	Zet het apparaat uit en geeft een zigbee 0X90 frame terug met daarin ack: pow= off.

2.6.ADT database-connection

Dit ADT zorgt voor de persistentie van het programma. Het toevoegen van tuppels aan de devicetabel en stewardtabel vereist dat stewards een unieke naam hebben en de combinatie steward-device ook.

Message	Parameters meegeven met oproep	Resultaat
Put-device	Target steward Device naam Device adres Device protocol	Probeert een nieuwe rij aan te maken in de devicetabel. Als het protocol dat vraagt, wordt er ook een nieuwe tabel bijgehouden om gemeten verbruik in op te slaan. Als dit niet lukt wordt de booleaanse waarde 'false' weergegeven.
Devive-name-used?	Steward naam Device naam	Kijkt of er al een tuple is in de devices tabel met deze naam en deze steward.
Product-id-used?	Product id	Kijkt of er al een tuple is in de devices tabel met dit product-id.

Put steward	Steward naam Steward adres	Probeert een nieuwe rij aan te maken in de stewardtabel. Als dit niet lukt, wordt de booleaanse waarde 'false' weergegeven.
Steward-name-used?	Steward naam	Kijkt of er een tuple in de steward tabel zit met deze naam.
Steward-adres-used?	Steward adres	Kijkt of er een tuple in de steward tabel zit met dit adres.
Get-all-stewards	Geen	Geeft een lijst met de namen van alle stewards die momenteel in de database opgeslagen zijn.
Get-specific-devices	Steward naam	Geeft een lijst met namen van de devices die zijn toegevoegd aan de steward.
Get-steward-device-list	Geen	Geeft een lijst van tuples met als tweede element een device en als eerste element de steward waartoe het device behoort.
Get-steward-address	Steward naam	Geeft het adres van de steward.
Get-device-product-id	Target steward Device naam	Geeft het product-id van het device.
Get-protocol	Target steward Device naam	Geeft het protocol van het device.
Erase-device	Target steward Device naam	Verwijdert het meegegeven device uit de devicetabel. Als het device een verbruikstabel heeft, wordt die ook verwijderd.
Erase-steward	Target steward Device naam	Verwijdert de meegegeven steward en al zijn devices.
Add-data	Target steward Time Data-package	Voegt een rij toe aan de juiste tabel.
Get-data	Target steward Device naam	Vraagt alle ingevoerde verbruikswaarden op. De waarden worden chronologisch in een lijst teruggegeven. De oudste waarden komen eerst in de lijst.
Get-time	Target steward Device naam	Vraagt alle ingevoerde tijdswaarden op. Ze worden chronologisch in een lijst teruggegeven.
Get-steward-time	Steward naam	Vraagt een lijst van ingevoerde tijdswaarden op. Ze worden genomen van een device dat zijn verbruik meet.
Get-steward-data	Steward naam	Vraagt alle waarden van de devices op van de steward. Van deze waarden die op hetzelfde moment zijn waargenomen, wordt het gemiddelde genomen. De rest wordt weggelaten.
Erase-table-data	Tabel naam	Verwijdert alle tuples in de tabel.

2.7.ADT log file

Dit ADT is verantwoordelijk voor het beheren van de log. Bij het opstarten van het domoticasysteem worden alle oude logwaarden ingelezen. Elke keer als er een nieuwe logwaarde wordt toegevoegd, wordt de log weggeschreven naar de schijf. Aangezien het niet de bedoeling is dat er constant toegevoegd of verwijderd wordt, leek dit mij een betere oplossing dan om de zoveel minuten wegschrijven.

Message	Parameters meegeven met oproep	Resultaat
Add-log-value	New value	Voegt de nieuwe waarde toe aan de logvalue. Vóór deze value wordt eerst de tijd geplaatst. Verder wordt er gekeken dat er niet meer dan 300 waarden in de logfile zitten door recursief toe te voegen.
Return-log	Geen	Geeft alle logvalues in chronologische volgorde weer.
Get-time	Geen	Geeft de tijd weer in stringvorm met formaat uur:minuut dag/maand/jaar

2.8.ADT gui

Dit ADT is waar alles uiteindelijk om draait. Het staat de gebruiker toe zijn domoticasysteem te gebruiken. Het bestaat uit vier verschillende verticale tabs elk met een eigen doel.

- 1) Het steward tab: hier kan het verbruik per steward worden opgevraagd en kunnen nieuwe stewards worden toegevoegd of verwijderd.
- 2) Het device tab: hier worden alleen devices van één steward tegelijk afgebeeld. Hier kan het verbruik worden opgevraagd per device (indien het een verbruiksmeter heeft) en kunnen devices met switch aan-of uitgezet worden. Verder kan men ook devices toevoegen en verwijderen.
- 3) Het overzicht tab: hier kan het gemiddelde verbruik van verschillende devices bekeken worden. Verder kunnen de opgeslagen waarden van devices hier verwijderd worden.
- 4) Het log tab: hier kunnen de laatste 300 acties van het systeem worden nagekeken.

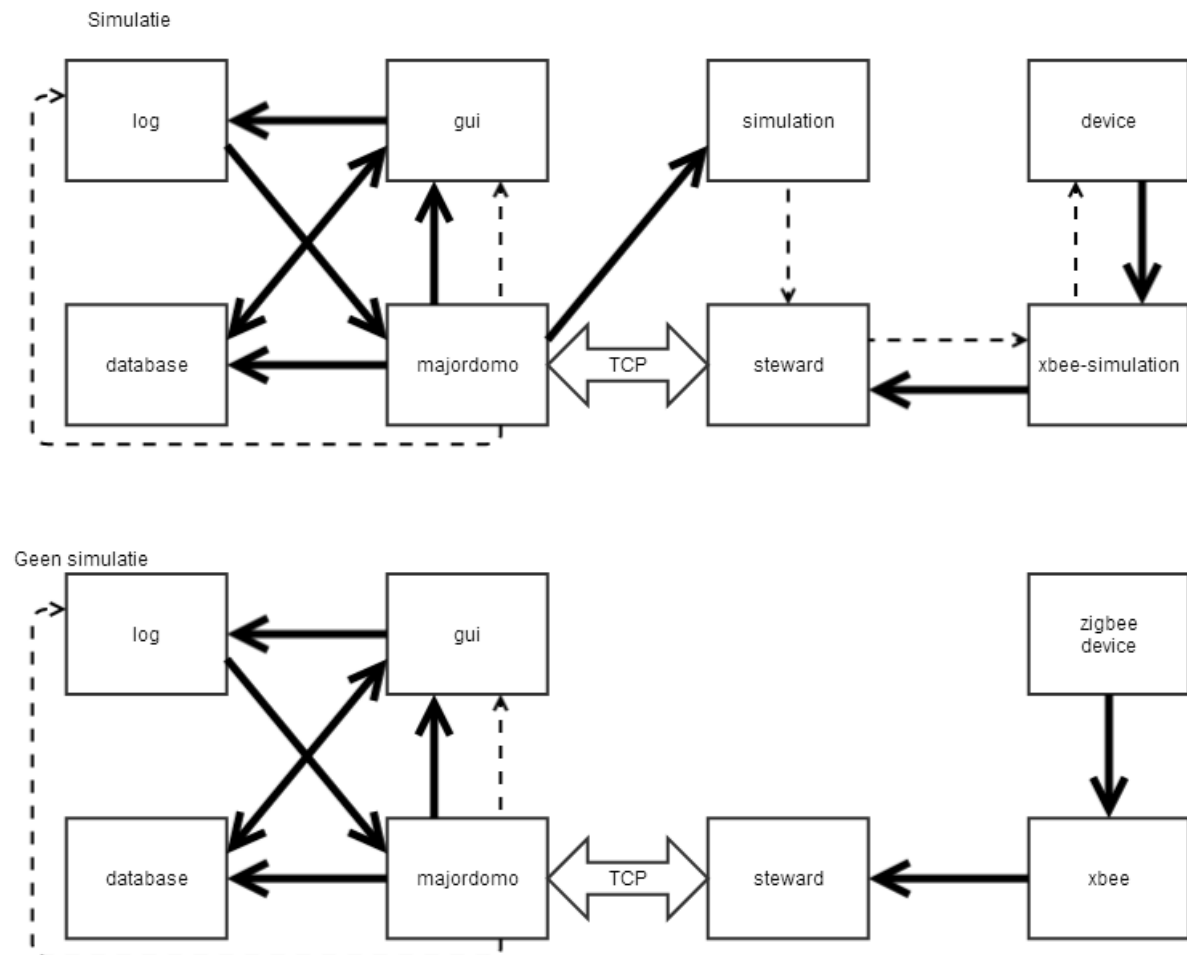
Zowel steward als device bestaan uit meerdere panels. Het eerste panel is een horizontaal gericht paneel en bestaat op zich weer uit een aantal verticale panelen. Elk van deze verticale panelen is bedoeld om er één soort object in te plaatsen. Zo ontstaat er een raster waar onder elkaar verschillende stewards of devices staan, terwijl er naast elkaar verschillende acties staan die ondernomen kunnen worden met betrekking tot die steward of dat device. We hebben ervoor gekozen om elke plot een nieuwe window te laten maken, zodat het makkelijk is deze naast en onder elkaar af te beelden.

Message	Parameters meegeven met oproep	Resultaat
Make-steward-panel	Geen	Vraagt aan de database alle stewards op en maakt per steward een rij in het steward panel.
Initiate	Geen	Maakt communiceren met de majordomo mogelijk, voegt add knop toe aan steward panel en device panel.

3. Relaties tussen ADT's

Het volgende diagram stel de relatie voor tussen de verschillende ADT's.

Een pijl stelt hierbij het doorgeven van informatie voor. Een stippenlijn stelt het aanmaken van een object voor.



4. Communicatieprotocol

Het communicatie protocol tussen majordomo en steward staat enkel strings toe. Als een bericht wordt geschreven met meerdere argumenten worden alle strings aan elkaar ge-append met een punt tussen elk argument. Gebruik makend van number->string en omgekeerd kunnen nummers worden doorgegeven.

De communicatie tussen xbee en devices gebeurt in 3 stappen.

- 1) Het bericht (string) wordt omgezet in een bytevector en doorgestuurd via de xbee-write functie.
- 2) De xbee geeft een 0X8B frame terug, dit is een status report van het bericht. Op de 6^{de} bit staat er of het bericht is aangekomen of niet.
- 3) Eenmaal een bericht is aangekomen verzekert xbee dat er een 0X90 bericht (received package) zal worden ontvangen door de xbee. In deze bytevector staat hoe het device heeft gereageerd op het gestuurde bericht. Als het bericht een 'get' was, zal hier de gevraagde data instaan, als dit bericht een 'set pow=' was, zal hier staan 'ack: set pow='.

5. Conclusie

Het programma wordt aanzienlijk moeilijker vanaf het moment dat er communicatie moet zijn tussen verschillende talen (hier scheme, slip en sql). Externe devices maken een project lastig, Xbee is onbetrouwbaar en robuust in gebruik. Het is het niet makkelijk om devices die door verschillende bronnen aangesproken worden, in de juiste volgorde te laten antwoorden.

6. Bronnen

1. Racket documentatie GUI
<http://docs.racket-lang.org/gui/gui.pdf>
2. Racket documentatie PLOT
<http://docs.racket-lang.org/plot/>
3. Racket documentatie PLOT contract
[file:///C:/Program%20Files%20\(x86\)/Racket/doc/plot/contracts.html#%28def.%28%28lib.plot%2Futils..rkt%29.%28plot-colors%2Ffc%29%29](file:///C:/Program%20Files%20(x86)/Racket/doc/plot/contracts.html#%28def.%28%28lib.plot%2Futils..rkt%29.%28plot-colors%2Ffc%29%29)
4. Racket documentatie srfi-19
[file:///C:/Program%20Files%20\(x86\)/Racket/doc/srfi-std/srfi-19.html](file:///C:/Program%20Files%20(x86)/Racket/doc/srfi-std/srfi-19.html)
5. Racket development by Matthew Flatt, tab-panel platform dependency
<http://lists.racket-lang.org/dev/archive/2012-March/009100.html>
6. Xbee_documentation_API