

LUA-Controlled Domoticsysteem

INTH-A, Groep 15

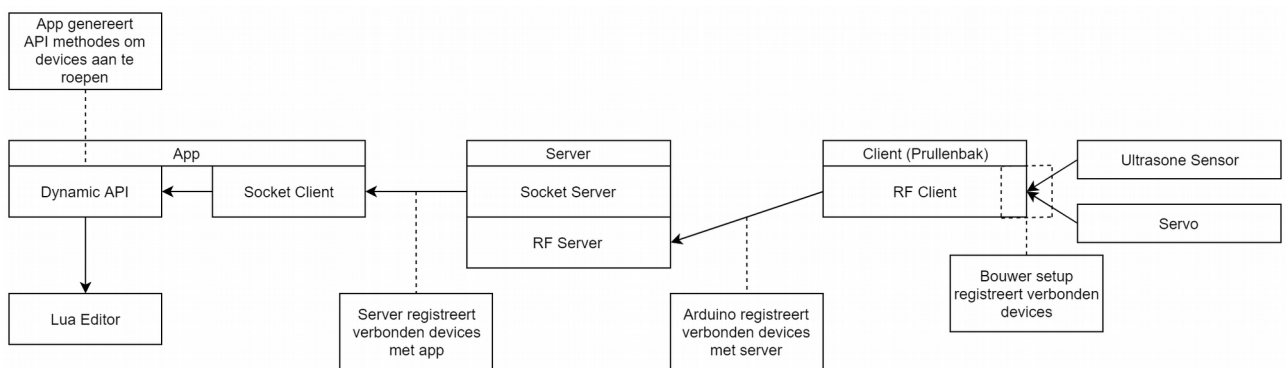
Rob Klein Iking,
Jan Vlasman,
Gerard van Meekeren,
Joshua Terra &
Robin Sinnema

Overzicht

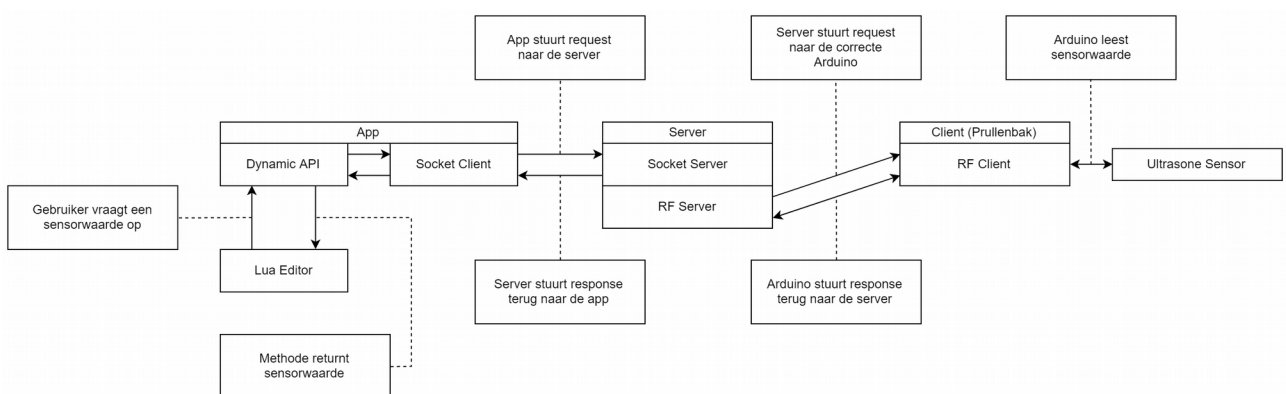
Ons idee is een app waarmee een of meer Arduino's kunnen worden aangestuurd, door middel van Lua scripts, die in de app geschreven en uitgevoerd worden.

Er is een hoofdarduino, de server, die in verbinding met het internet staat, waarmee de app communiceert. Deze Arduino verbindt zich via RF automatisch met andere nabijgelegen Arduinos. De Arduino stuurt een lijst van verbonden sensoren en actuatoren (hierna *attachables*) naar de app. De gebruiker kan deze lijst gebruiken om deze attachables direct vanuit de app (doormiddel van het schrijven en uitvoeren van een script) aan te sturen.

Architectuurschema



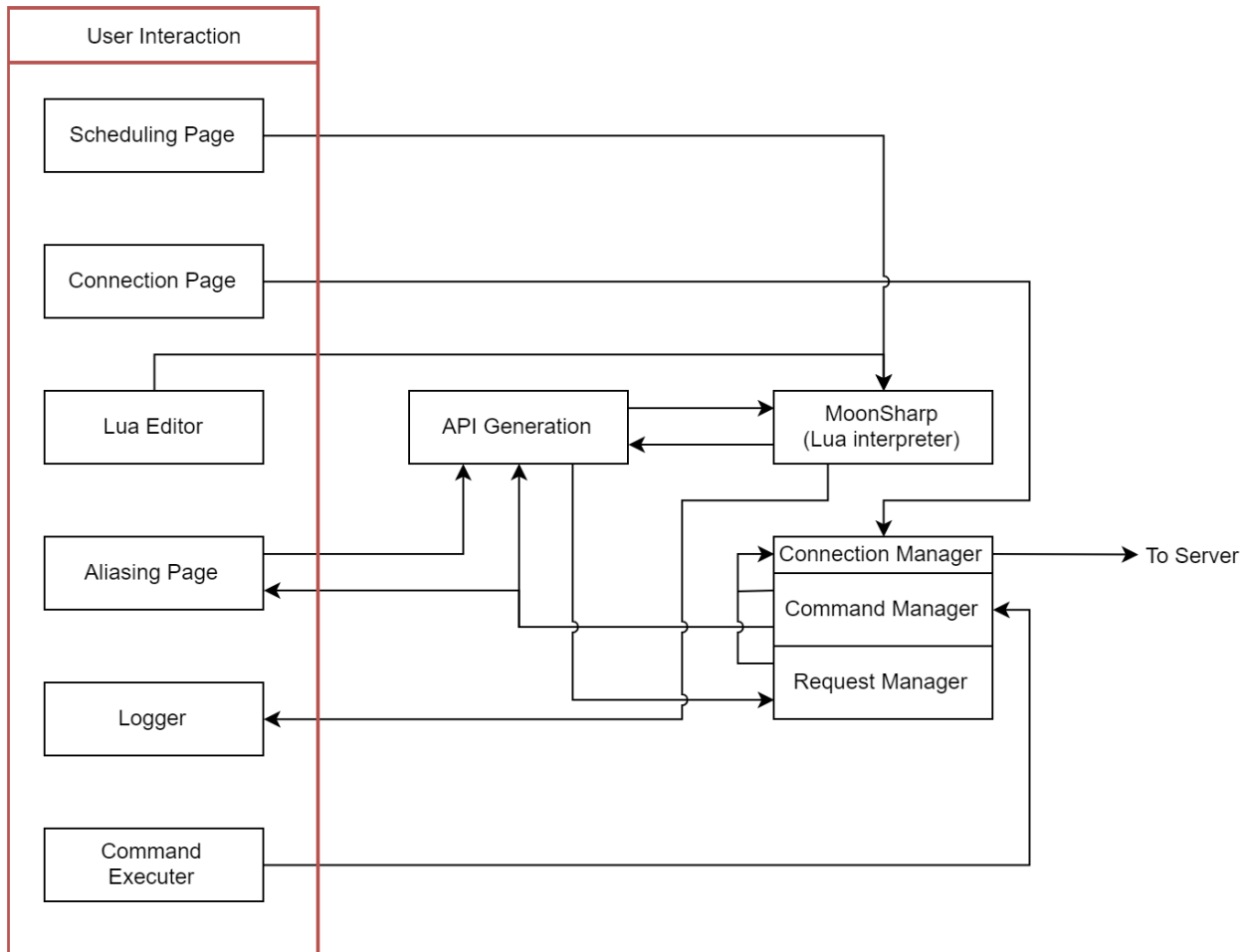
1. Schema initialisatie & verbinding



2. Schema request voor sensor vanuit app.

(Requests voor actuatoren werken vergelijkbaar, maar de response bevat geen data.)

Structuurschema

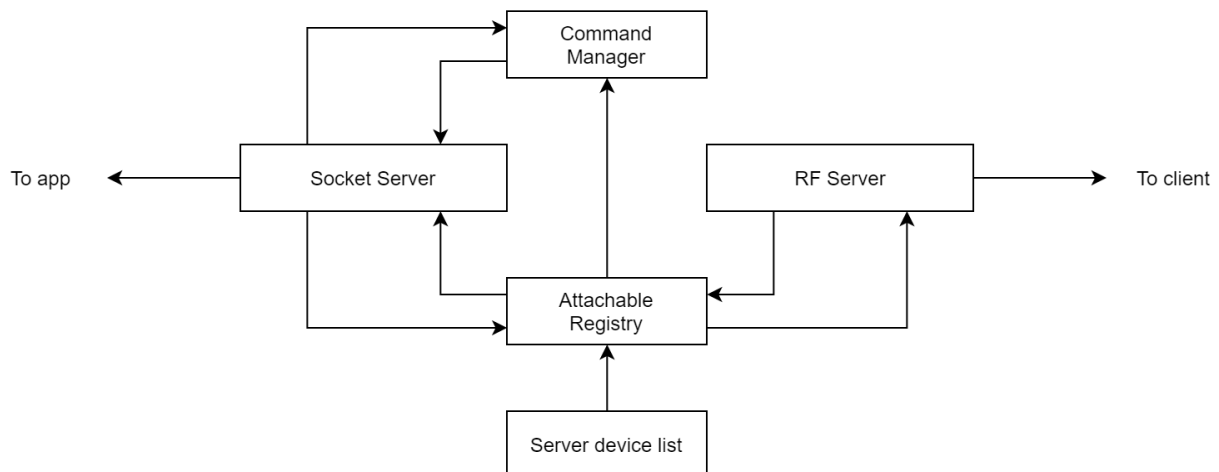


3. Schema dataflow binnen de app

Op de Aliasing Page kan de gebruiker aliasen voor de aan het netwerk verbonden attachables instellen, waarmee ze in de Lua scripts kunnen worden aangeroepen.

Op de Command Execution page kunnen commando's naar de server worden gestuurd, om de server zelf te beïnvloeden, in plaats van een verbonden attachable.

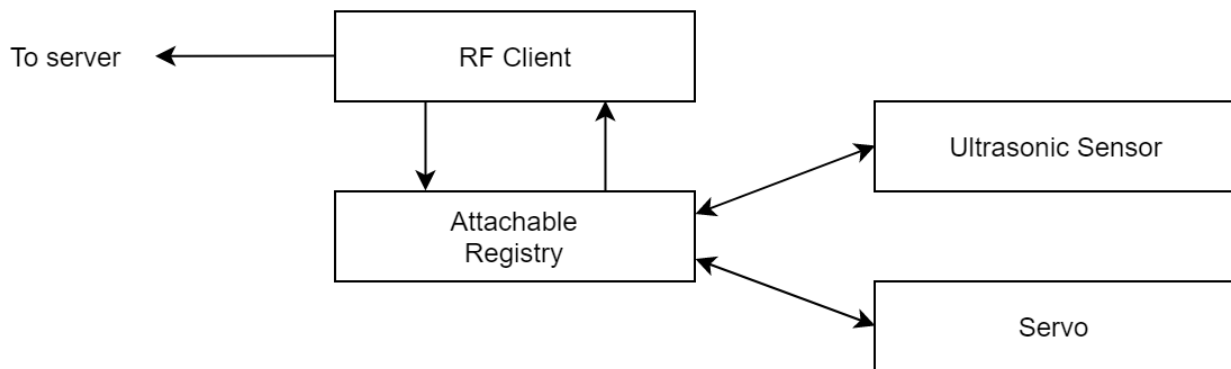
Via de Scheduling Page kunnen scripts worden ingesteld om op gekozen momenten te worden uitgevoerd.



4. Schema dataflow server

Via de Socket Server kunnen requests worden gemaakt naar de AttachableRegistry, waar de informatie om de attachables aan te sturen bewaard wordt. Via de RF Server kunnen requests die niet voor een attachable op de server zelf zijn worden doorgestuurd.

Via de Command Manager kunnen zaken rond de server zelf verricht worden, zoals het pingen van de server en het opvragen van de lijst van attachables.



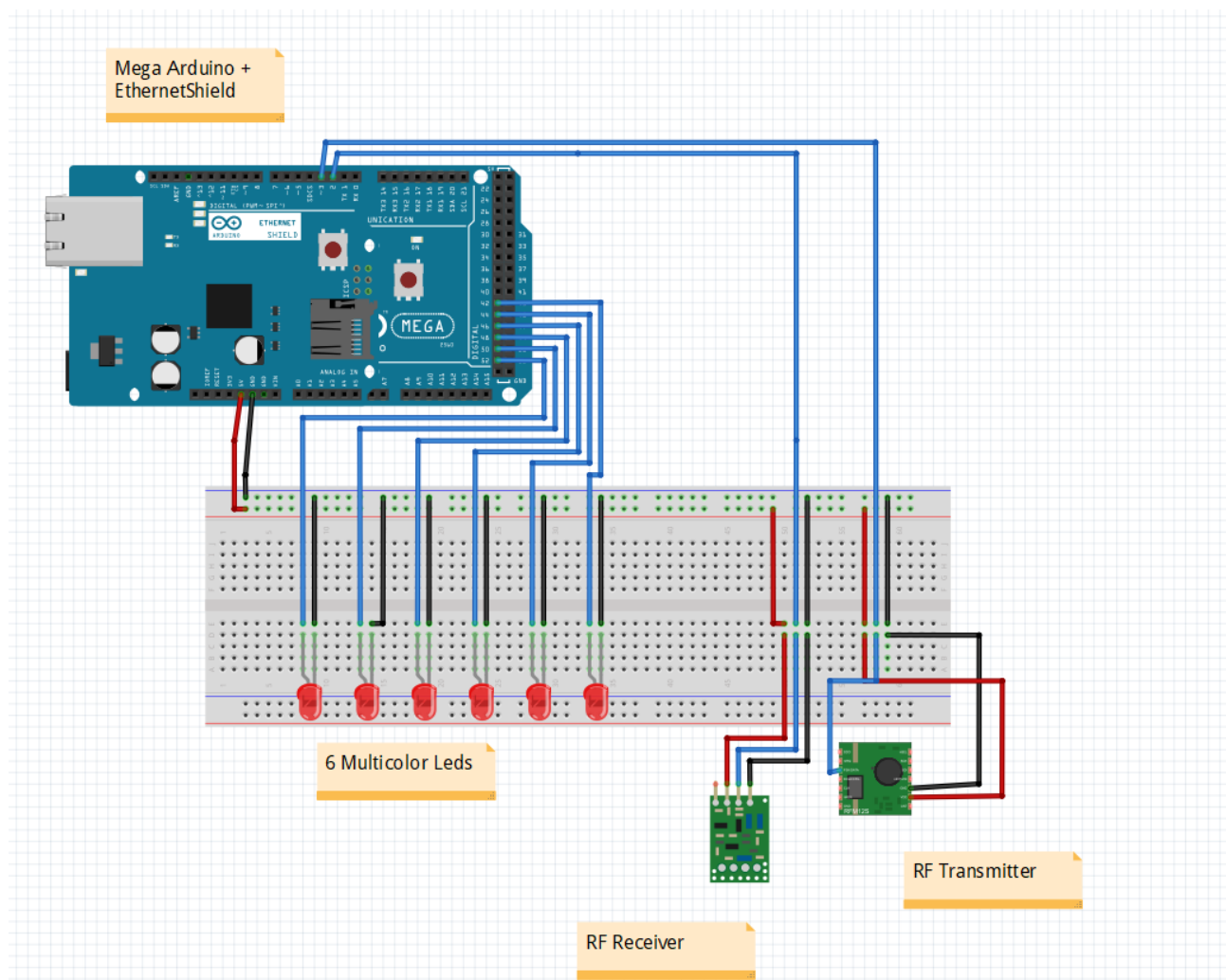
5. Schema dataflow client

Iedere client heeft zijn eigen AttachableRegistry, met daarin de attachables die aan de client verbonden zijn. Deze lijst wordt ingevuld door degene die de Arduino opzet.

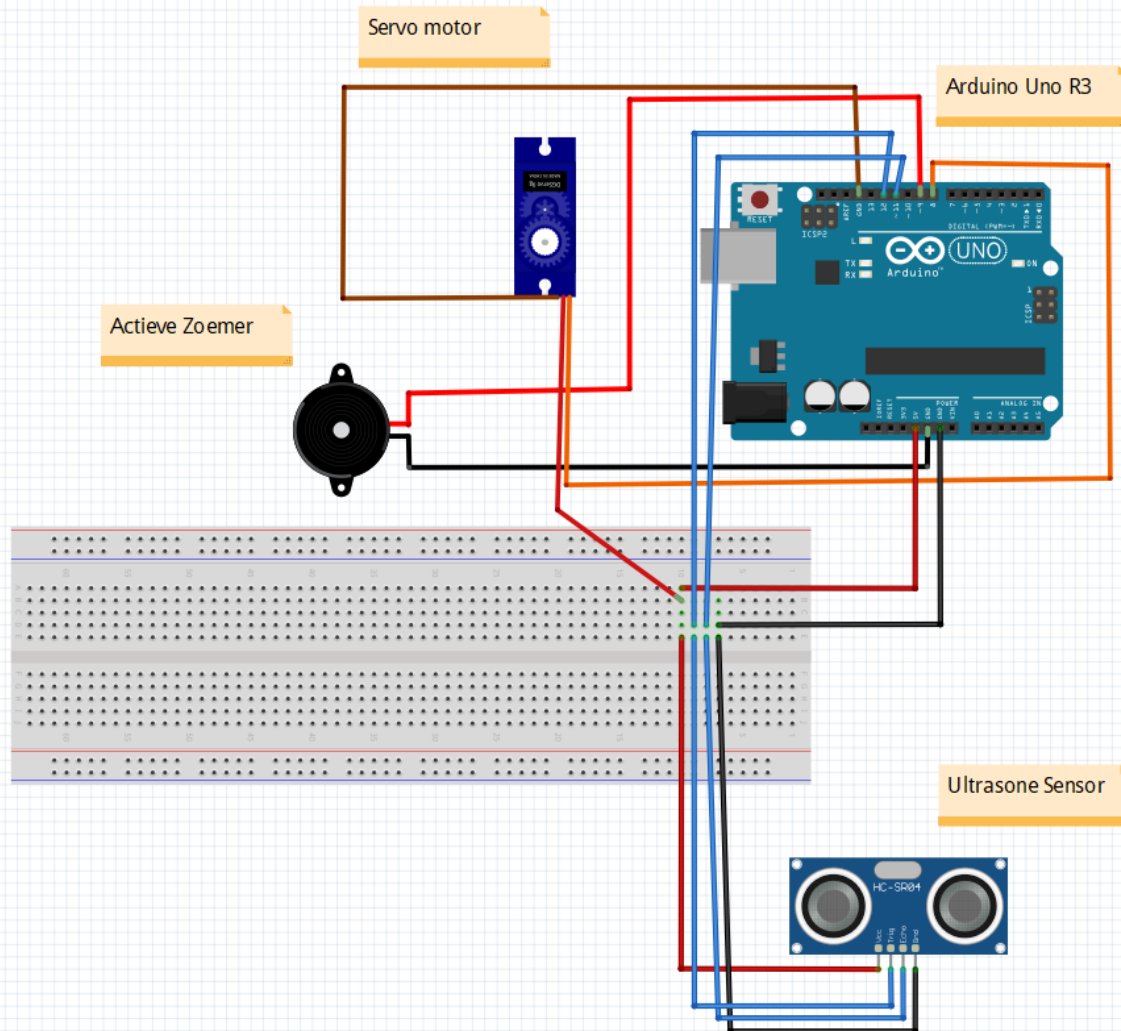
Overzicht hardware

Het idee achter ons systeem is een server waaraan nieuwe Arduinos kunnen worden gekoppeld, ongeacht hun attachable-configuratie. Om ons systeem te demonstreren hebben wij echter drie specifieke systemen opgesteld:

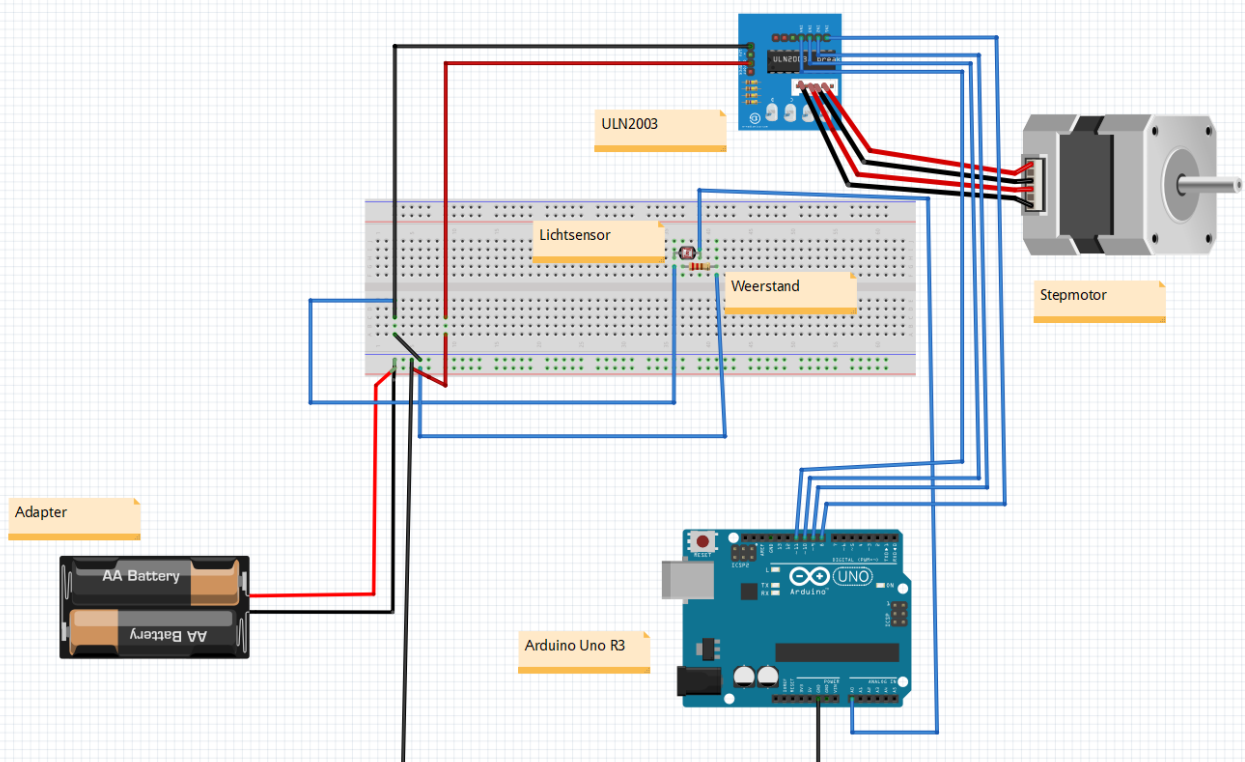
- Een automatisch openende en sluitende prullenbak
- Een automatisch openend en sluitend gordijn
- Een temperatuur-geactiveerd brandalarm



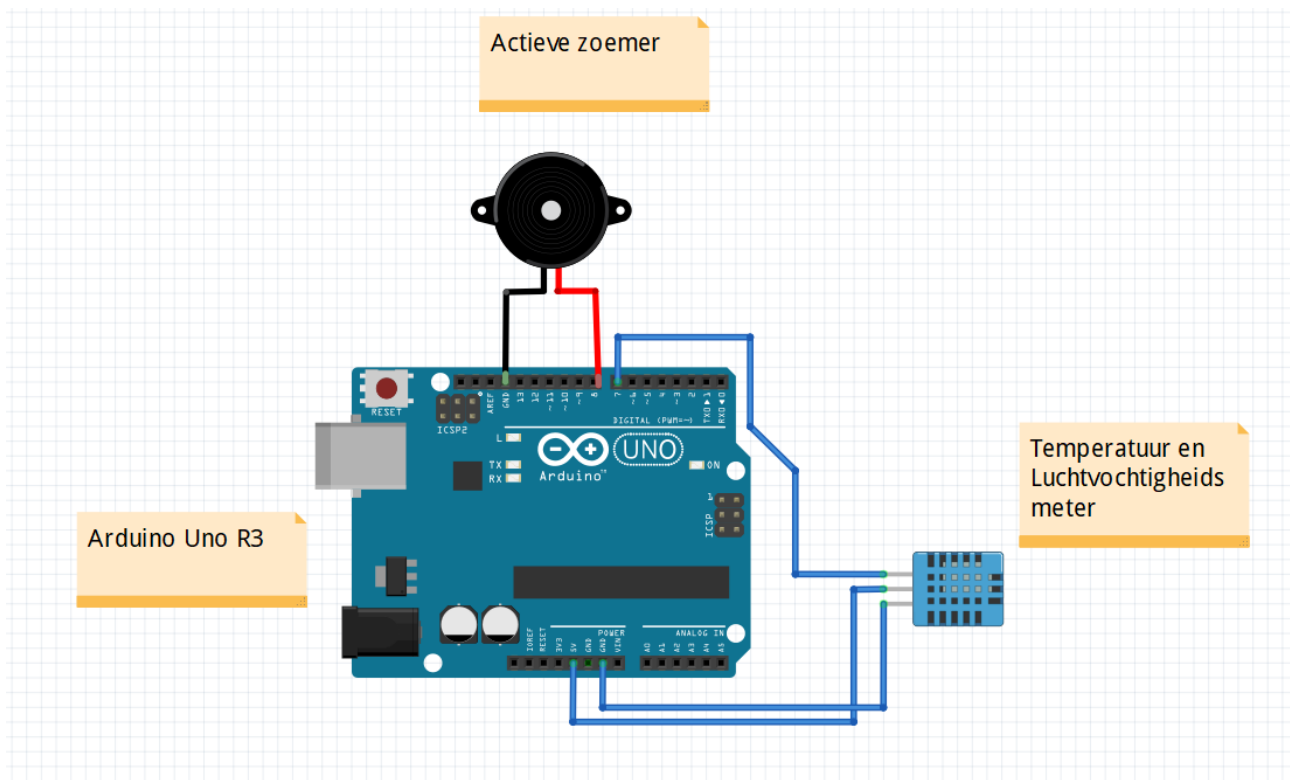
6. Fritsingschema Arduino Server



7. Fritsingschema Automatische Prullenbak



8. Fritsingschema Automatisch Gordijn



9. Fritsingschema brandalarm

Plan van aanpak

Voor dit project hebben wij geprobeerd volgens de principes van Scrum te werken. Wij hebben de tijd voor het project opgedeeld in twee sprints: in de eerste sprint hebben wij de basisfuncties van de app opgezet: de verbinding tussen de app en de server en het uitvoeren van de Lua scripts op de telefoon.

Wij hebben gezamenlijk besproken welke features wij haalbaar achtten en hebben daarmee een MoSCoW lijst en een GitHub backlog opgesteld.

Must Have:

[M1] Als gebruiker wil ik dat ik de Arduino vanaf mijn telefoon kan programmeren, zodat ik deze niet fysiek met mijn computer hoeft te verbinden wanneer ik de code wil veranderen.

[M2] Als gebruiker wil ik dat ik de Arduino in een higher-level programmeertaal dan C++ kan programmeren, zodat ik met minder tijd en kennis de Arduino opzetten kan.

[M3] Als gebruiker wil ik dat ik de resources van de telefoon kan gebruiken, zodat ik grotere scripts kan schrijven dan in het 2KB geheugen van de Arduino passen.

Should Have:

[S1] Als gebruiker wil ik de attachables verbonden met de Arduino met een door mij gekozen naam aanroepen, zodat ik duidelijkere scripts kan schrijven.

[S2] Als gebruiker wil ik dat ik mijn code kan scheduleren, zodat deze kan worden uitgevoerd wanneer ik wil.

[S3] Als gebruiker wil ik dat mijn code wordt opgeslagen op de telefoon, zodat ik hem niet iedere keer opnieuw hoeft te schrijven.

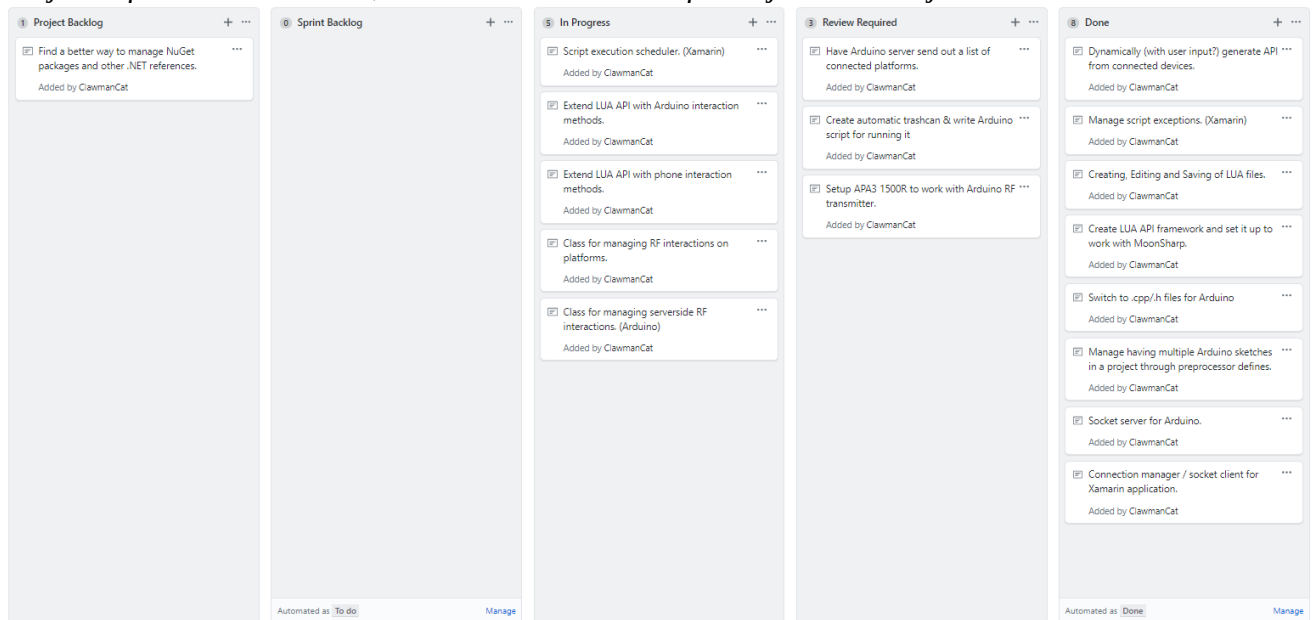
Could have:

[C1] Als gebruiker wil ik kunnen zien welke datatypes een attachable nodig heeft en teruggeeft, zodat ik ze in mijn code correct kan aanroepen.

[C2] Als gebruiker wil ik de sensoren van mijn telefoon, zoals de GPS, net als de sensoren van de Arduino kunnen aanroepen, zodat ik deze kan gebruiken in mijn code.

Won't Have:

[W1] Als gebruiker wil ik dat mijn code ook wordt uitgevoerd als er geen verbinding meer is tussen mijn telefoon en de Arduino, zodat ik niet online hoeft te blijven voor mijn Arduino om te werken.



10. Project backlog op 27/06/19