Data Science for IoT

**TV-uitzetter**



Vak code: CMIDAT01K

Naam: Daniël van der Drift

Leerling nummer: 0986788

Klas: Ti1D

Inhoud

[Inleiding 3](#_Toc39003606)

[Project idee 4](#_Toc39003607)

[Hardware 5](#_Toc39003608)

[Sensor 5](#_Toc39003609)

[Actuator 5](#_Toc39003610)

[Schema 6](#_Toc39003611)

[Bestellijst 7](#_Toc39003612)

[Data-pipeline 8](#_Toc39003613)

[Sensoren 8](#_Toc39003614)

[Cloud opslag 8](#_Toc39003615)

[Data-analyse 8](#_Toc39003616)

[Actuator 9](#_Toc39003617)

[Software 10](#_Toc39003618)

[Raspbian 10](#_Toc39003619)

[Python 10](#_Toc39003620)

[Matlab analysis 10](#_Toc39003621)

[Reflectie 11](#_Toc39003622)

[Bronnen 12](#_Toc39003623)

# Inleiding

Ooit weleens thuisgekomen en erachter gekomen dat de TV de hele dag aan heeft gestaan? Bij mij thuis gebeurt dat heel vaak. Voor het keuzevak Data Science for IoT moest er een creatief project gemaakt worden met een IoT data-pipeline. Dit gaf me het idee om een IoT device te maken die de TV uitzet als deze te lang aan staat zonder mensen in de buurt.

In dit verslag ga je lezen hoe ik aan dit project heb gewerkt en wat er allemaal bij komt kijken.

# Project idee

Concept: TV-uitzetter

Het idee is dus dat ik en project maak dat de TV uitzet als iemand deze aan heeft laten staan en is weg gegaan. Dit moet uitgevoerd worden aan de hand van sensordata dat in de cloud geanalyseerd wordt.

Mijn doel voor dit project is om data in de cloud te kunnen verwerken en hier conclusies uit kunnen trekken.

Ook is dit een mooie gelegenheid om radar sensor uit te testen die ik al een tijdje had liggen.

# Hardware

Het centrale onderdeel van dit project is de Raspberry pi 3B. De Raspberry pi is een microprocessor waar een besturingssysteem op draait. Ook heeft de Raspberry pi gpio (general purpose input output) pinnen waar sensoren en actuatoren aan verbonden kunnen worden.

Om het doel van dit project te bereiken zijn er een aantal sensoren en actuatoren nodig.

## Sensor

Voor dit project is informatie nodig over twee dingen. Er moet bekend zijn of de TV op dit moment aan staat en of er nog mensen aanwezig zijn.

Om te herkennen of de TV aan staat zijn dit een aantal mogelijkheden:

* Power led uitlezen
* Afstandsbediening signalen opvangen
* Stroom verbruik meten

Ik heb ervoor gekozen om de afstandsbediening signalen op te vangen omdat dit me de meest interessante en leerzame optie leek. Dit gebeurt met een IR-ontvanger.

Om te herkennen of mensen aanwezig zijn:

* PIR-sensor
* Radar sensor

Ik heb hierbij voor de Radar sensor gekozen omdat het me heel interessant lijkt om deze sensor te gebruiken.

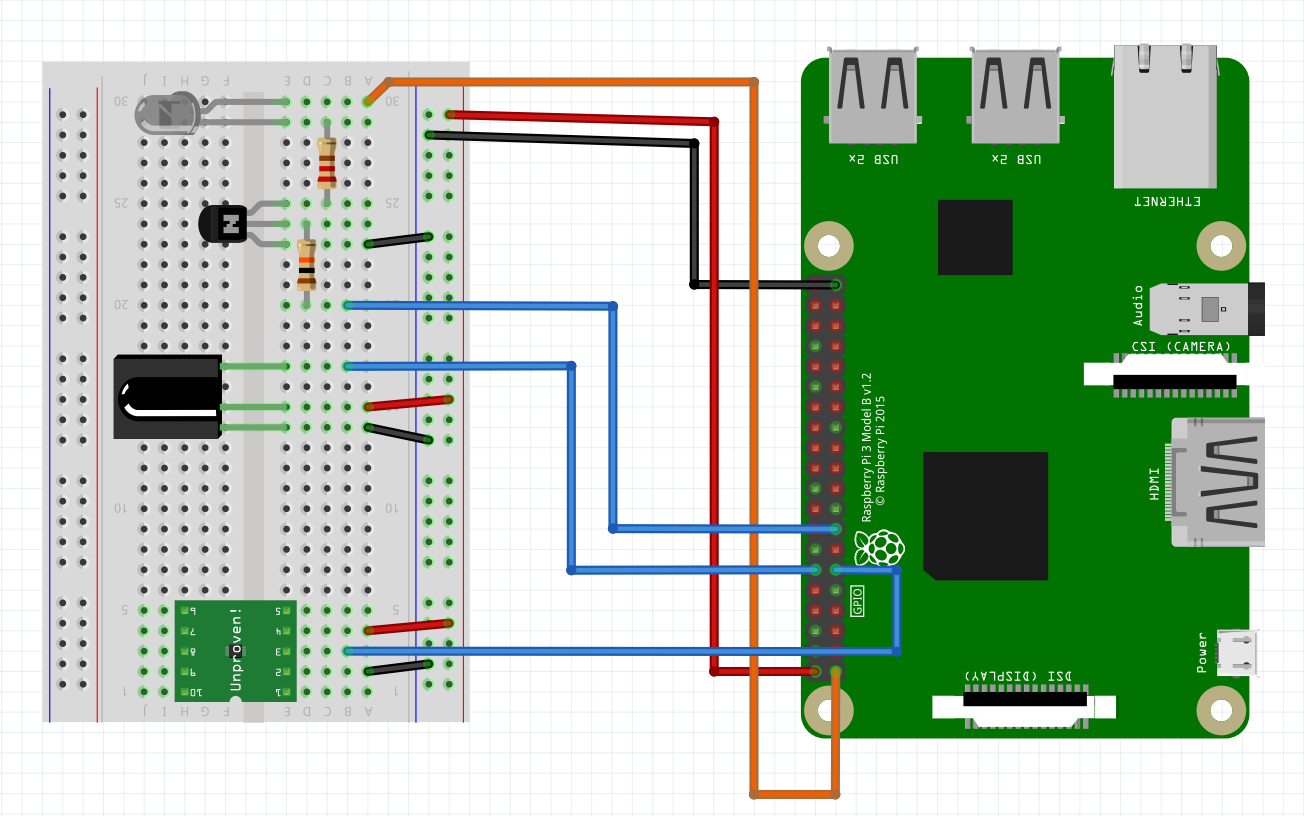
## Actuator

De actie die uitgevoerd moet worden is dat de TV uitgezet moet worden. Dit kan gedaan worden door het aan/uit signaal te sturen met een Infrarode led. Omdat de gpio van de Raspberry pi niet veel stroom kan leveren is er een transistor gebruikt die de hoofdspanning van 5 volt aan de led geeft.

## Schema

Dit is het schema dat is gebruikt in dit project.

Let op! De pinnen van de infraroodsensor kunnen op een andere volgorde zitten. Dit verschilt vaak.



## Bestellijst

Bestellijst met prijzen van [Tinytronics](https://www.tinytronics.nl/).

|  |  |  |
| --- | --- | --- |
| Onderdeel | Prijs | Link |
| Raspberry pi 3B | €37,50 | <https://www.tinytronics.nl/shop/nl/raspberry-pi/main-boards/raspberry-pi-3-model-b-1gb> |
| Radar sensor | €2,- | <https://www.tinytronics.nl/shop/nl/communicatie/rf/microgolf-bewegingssensor-rcwl-0516> |
| IR-ontvanger | €0,75 | <https://www.tinytronics.nl/shop/nl/communicatie/infrarood/hx1838-ir-ontvanger-38khz> |
| IR led | €0,20 | <https://www.tinytronics.nl/shop/nl/verlichting/led-los/ir-led-940nm-5mm-helder> |
| Breadboard | €3,- | <https://www.tinytronics.nl/shop/nl/prototyping/breadboards/breadboard-400-points> |
| Breadboard draden | €3,- | <https://www.tinytronics.nl/shop/nl/kabels/prototype-draden/breadboard-draden-65-stuks-verschillende-maten> |
| Transistor | €0,15 | <https://www.tinytronics.nl/shop/nl/componenten/transistor-fet/npn-transistor-bc547> |
| 220Ω weerstand | €0,05 | <https://www.tinytronics.nl/shop/nl/componenten/weerstanden/220%CF%89-weerstand-(led-voorschakelweerstand)> |
| 10kΩ weerstand | €0,05 | <https://www.tinytronics.nl/shop/nl/componenten/weerstanden/10k%CF%89-weerstand-(standaard-pull-up-of-pull-down-weerstand)> |
| Totaal: | €46,70 |  |
| Verzendkosten | €2,50 |  |
| Totaal: | €49,20 |  |

# Data-pipeline

In dit hoofdstuk wordt beschreven hoe de data ontstaat, verplaatst en gebruikt wordt.

## Sensoren

De Data waar gewerkt mee zal worden ontstaat omdat sensoren deze dingen waarnemen.

De radar sensor geeft bijvoorbeeld een hoog signaal voor 5 seconden als deze bewegingen herkent van voorwerpen van metaal of water (ook mensen). Dit signaal zal minimaal elke 5 seconde afgelezen moeten worden om elke menselijke beweging op te kunnen vangen.

De IR-ontvanger vangt het signaal op dat door de afstandsbediening verstuurd is. Dit signaal bestaat uit lichtpulsen en moet vertaald worden naar enen en nullen. Deze enen en nullen worden dan opgeslagen als bits en vormen een unieke code voor elk knopje op de afstandsbediening. Verschillende merken hebben over het algemeen verschillende codes. In de Python code zal de signaalcode van de aan/uitknop genoteerd moeten worden zodat deze herkent kan worden. Als de signaalcode van de aan/uitknop onderschept is zal dit verstuurd worden naar de cloud.

Opvallend is dat de waardes van deze sensoren binair zijn en dus voor makkelijke berekeningen zorgen.

De Raspberry pi noemen we een Publisher omdat deze data publiceert door het naar de cloud te sturen.

## Cloud opslag

Thingspeak is een IoT analytics platform dat het mogelijk maakt om live data streams te verwerken en visualiseren in de cloud.

De data die verzonden wordt vanaf de Raspberry pi wordt opgeslagen in een channel met 2 fields. In deze twee fields wordt de geschiedenis van de stand van de TV en de aanwezigheid van mensen bijgehouden.

Er zijn ook andere vergelijkbare cloud services maar er is voor Thingspeak gekozen omdat deze service als voorbeeld in de Cursus is gebruikt.

## Data-analyse

Bij het verwerken van de data zal het resultaat opgeslagen worden in een nieuwe channel.

In Thingspeak zijn er verschillende apps die je samen kan laten werken met je data. Ook de app MATHLAB Analysis is daar een van. Hier kan Matlab code in geschreven worden om de data aan te passen of verwerken en het te plaatsen in een nieuw channel.

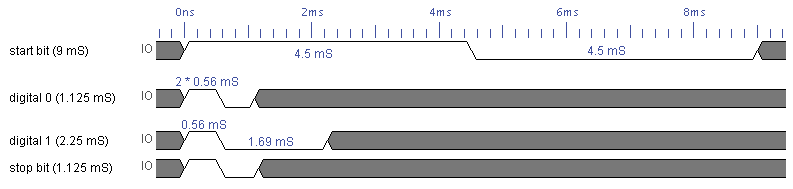
MATHLAB Analysis code werkt eenmalig. Daarom is er nog een ander app genaamd TimeControl die acties kan uitvoeren op bepaalde tijden. Deze app kan je gebruiken om de MATHLAB Analysis code te herhalen.

## Actuator

De Raspberry pi noemen we een Subscriber omdat deze de nieuwe data van de cloud wilt ontvangen als dit er is. Meerdere apparaten kunnen gesubscribed zijn op een channel.

Als de data een waarde heeft hoger dan nul weet de Raspberry pi dat de TV uitgezet moet worden. Het juiste IR-signaal wordt uitgezonden door de IR LED waardoor de TV uit gaat.

Omdat de signaalcode van de aan/uitknop bekend omdat we dit hebben opvangen, is het mogelijk om ditzelfde signaal uit te zenden met een infrarode led. Om te laten weten dat er een signaal aan komt wordt de led eerst 4.5 milliseconden aangezet en volgt er een pauze van 4.5 milliseconden. De test van het signaal bestaat uit een aantal enen en nullen die overgebracht worden als korte of lange pauzes tussen lichtpulsen.



# Software

## Raspbian

Raspbian is het besturingssysteem van de Raspberry pi. Dit is een UNIX-besturingssysteem dat speciaal is gemaakt voor de Raspberry pi.

## Python

De code op de Raspberry pi is geschreven met python. Ik heb Node-red niet gebruikt omdat dit geen infrarood afstandsbediening functionaliteit heeft.

Het was lastig om de infrarood ontvanger aan de praat te krijgen. Op internet werd veel gesproken over de LIRC. Dit is een opensource programma dat het mogelijk maakt om infrarood signalen te ontvangen. Dit kreeg ik helaas niet werkend op de Raspberry pi. Meer mensen schijnen hier last van te hebben op de nieuwere Raspbian versie. Toch heb ik een python script kunnen vinden die zonder LIRC werkt.

## Matlab analysis

De code om de data te verwerken is geschreven in Matlab code en is actief op Thingspeak MATLAB Analysis en word continu uitgevoerd door de TimeControl app.

De code bestaat uit de volgende stappen. Stap 1, de TV-stand en de bewegingswaardes van het afgelopen half uur worden in het programma geladen. Stap 2, de maximale waarde van de beweging field wordt berekend. Als dit getal nul is betekent het dat er een half uur niemand is of zich niet heeft bewogen. Stap 3, als de TV op dit moment aan staat wordt er een word er een 1 naar de andere channel geschreven, anders een 0. De 1 geeft aan dat de TV aan staat zonder mensen in de buurt en moet dus uitgezet worden.

# Reflectie

Een van mijn doelen was om te leren hoe data in de cloud verwerkt kan worden en hier conclusies uit te kunnen trekken. Dit heb ik zeker geleerd alleen denk ik dat er betere manieren zijn dan dat ik heb gebruikt. De implementatie van Matlab was niet ideaal vind ik. Het maakt me wel nieuwsgierig over hoe andere IoT platformen dit doen zoals AWS of Azure.

Een ander doel was om de Radar sensor gebruiken. Dit bleek niet lastig te zijn maar ik heb nu wel kennis over een nieuwe interessante sensor.

Ik begrijp nu hoe data in de cloud geanalyseerd kan worden met Thingspeak. Om dit beter te kunnen gebruiken moet ik meer kennis krijgen van het programmeren van Matlab code. Het was alleen jammer dat deze code niet direct is en van mee

IR was lastiger dan verwacht en nam veel tijd in. Het probleem had ik waarschijnlijk op kunnen lossen door een andere versie van Raspbian te installeren.

Corona heeft dit keuzevak wel een stuk lastiger gemaakt met namen dat het werk makkelijker uitgesteld wordt als je thuis werkt.

# Bronnen

Infrarood:

Video: <https://www.youtube.com/watch?v=KhiqINyHx08>

Github: <https://github.com/Lime-Parallelogram/IR-Code-Decoder-->

Protocollen: <https://www.techdesign.be/projects/011/011_waves.htm>

Matlab Analysis voorbeelden: <https://nl.mathworks.com/help/thingspeak/examples.html?category=matlab-analysis-and-visualization>