# TV-uitzetter

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 bedacht 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.



De data die verzameld moet worden is of de tv aan staat en of er mensen in de buurt zijn. Het resultaat moet zijn dat de tv uit gezet wordt als er een tijdje niemand in de buurt is. Dit kunnen we voor elkaar krijgen door een PID of vergelijkbare sensor te gebruiken om de aanwezigheid van de mensen te herkennen en de signalen van de afstandsbediening op te vangen.

Uit de gegenereerde data zal een conclusie getrokken moeten worden. Dit gebeurt in de cloud. De uitkomst beslist of de tv te lang aan staat en dus uitgezet moet worden.

Het uitzetten van de tv kan gedaan worden door het aan/uit signaal met een infrarode led uit te zenden naar de tv.

In het Word document staan meer details over de keuzes die bij dit project gemaakt zijn. Ook gaat deze dieper in op de technische werking van dit project.

# Data science en IoT

### Wat is data science

Data science is in het kort gezegd het waarde creëren uit data. Data zelf is namelijk niet per se nuttig. Met data science worden verbanden gevonden en conclusies getrokken uit data. Vaak worden hier geautomatiseerde systemen bij gemaakt die zelf conclusies kunnen trekken of kunnen voorspellen wat er gaat gebeuren.

### Wat is IoT

IoT staat voor Internet of Things. Dit staat voor het netwerk van apparaten die aan het internet verbonden zijn. Bij deze term wordt niet alleen maar gesproken over laptops en telefoons maar vooral over de andere apparaten in huis die aan het internet worden verbonden. Bijvoorbeeld je Roomba, weerstationnetje, Amazon Alexa of zelfs het koffiezetapparaat. Er wordt steeds meer informatie gedeeld tussen apparaten.

### Verband

Het verband tussen data science en IoT is dat er heel veel data wordt geproduceerd door de IoT producten. Deze data kan heel nuttig zijn. Met data science kan deze data verwerkt worden tot nuttige informatie. Deze informatie kan dan gebruikt worden om acties uit te voeren als reactie op de data.

Dit project laat je zien hoe je sensor data in de cloud kan opslaan en verwerken.

# Bouw instructies

## Benodigdheden

### Hardware

Project onderdelen

* Raspberry pi 3B
* 5 volt voeding
* Radar sensor
* IR-ontvanger
* IR led
* Breadboard
* Breadboard draden
* Transistor
* 220Ω weerstand
* 10kΩ weerstand

Accessoires voor de Raspberry pi:

* Toetsenbord
* Muis
* HDMI-beeldscherm met kabel

### Software

* Raspbian besturingssysteem (Ook te installeren via NOOBS)
* Python (Is standaard geinstalleerd op Raspbian full versie)
* Thingspeak website

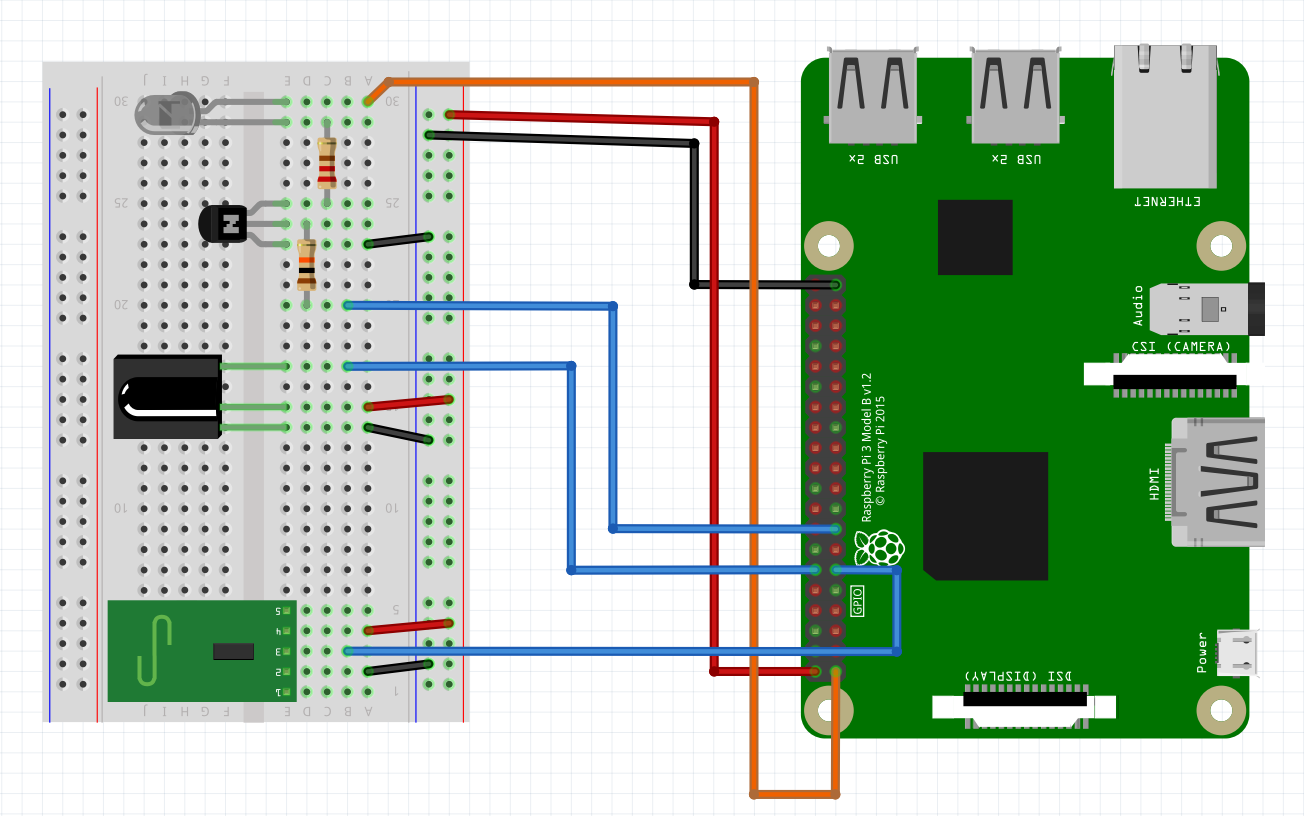
## Raspberry pi

Installeer een besturingssysteem. Dit doe je door een besturingssysteem op de SD kaart te zetten en deze in de Raspberry pi te steken.

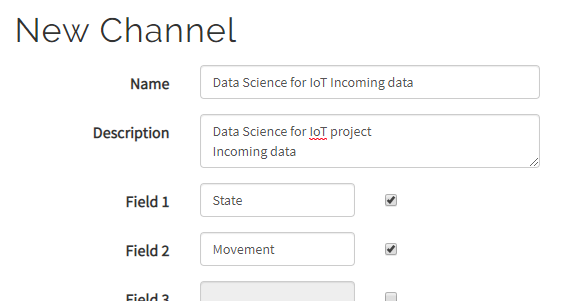
Voor beginners adviseer ik om noobs te installeren. Hiermee kan je makkelijk verschillende besturingssystemen proberen. Kies bij dit project voor Raspbian.

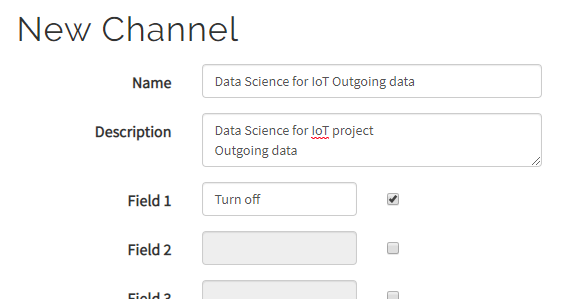
Dit zijn instructies om noobs te installeren: <https://thepi.io/how-to-install-noobs-on-the-raspberry-pi/>

Als Raspbian geïnstalleerd is moeten het elektrische schema opgebouwd worden. Verbind de onderdelen zoals op de afbeelding hieronder is voorgedaan. Let op, controleer de pinnen van de infrarood ontvanger, hier is namelijk verschil in.



## Thingspeak

Maak een account op [Thingspeak](https://thingspeak.com/) en klik op channels. Er zijn twee channels nodig. De eerste channel is voor de inkomende data met twee fields genaamd State en Movement. De tweede channel is voor de verwerkte data met een field die aan zal geven of de tv uit moet.



## Python

Nu gaan we verder op de Raspberry pi. Python hoeft niet meer geïnstalleerd te worden omdat dit standaard op Raspbian geïnstalleerd is. Clone de hele Github repository door een nieuwe terminal te openen en dit commando uit te voeren: git clone https://github.com/Danielvdd1/Data-Science-for-IoT.git

Om te weten of de tv aan of uit gezet wordt, moet het juiste infrarood signaal van de aan/uit knop van de afstandsbediening bekend zijn. Voer de python code uit in TestCode > IrReceiver > CLI.py om de IR-signaal van de gewenste knop uit te lezen. Dit geeft een signaalcode die in de hoofdcode ingevuld moet worden. Open de hoofdcode in de Code folder en vul de signaalcode in.

Ook moeten een aantal waarde van Thingspeak ingevuld worden. Open het python bestand in de Code folder in een tekst editor zoals nano en vul de 4 variabelen in genaamd writeAPIkey, writeChannelID, readAPIkey en readChannelID. Deze informatie kun je vinden in Teamspeak in de bijbehorende channels. De Channel id staat bij ‘Channel settings’ en de read of write API keys staan bij het kopje ‘API Keys’.

De write API key en channel moeten de waarde hebben van de channel van de Incomming data. En de read API key en channel moeten de waarde hebben van de channel van de Outgoing data.

Als de sensoren goed zijn aangesloten en de Teamspeak channels goed zijn ingesteld kan de python code aangezet worden en zal de eerste data naar Teamspeak verzonden kunnen worden.

## Matlab analysis

Matlab Analysis is een app op Teamspeak waarmee de data in de channels verwerkt kan worden. Open Matlab Analysis op Teamspeak > Apps > MATLAB Analysis. Maak een nieuwe lege code aan en copieer de code hieronder in Matlab Analysis.

```

% Turn the tv off when the tv is on without people nearby

% Replace the [] with channel ID to read data from:

readChannelID = [];

% TV state Field ID

stateFieldID = 1;

% Movement Field ID

movementFieldID = 2;

% Channel Read API Key:

% If your channel is private, then enter the read API Key between the '' below:

readAPIKey = '';

% Replace the [] with channel ID to write data to:

writeChannelID = [];

% Enter the Write API Key between the '' below:

writeAPIKey = '';

% Delay in minutes befor turning off

waitDuration = 30;

[state, time] = thingSpeakRead(readChID, 'Fields', stateFieldID);

movement = thingSpeakRead(readChannelID, 'Fields', movementFieldID, 'NumMinutes', waitDuration, 'ReadKey', readAPIKey);

% Calculate the maximum and minimum temperatures

[maxMovement,maxMovementIndex] = max(movement); % 1 or 0

display(maxMovement,'Movement for the past 30 minutes is');

turnTvOff = 0

if (maxMovement > 0) && (state == 1)

turnTvOff = 1;

else

turnTvOff = 0;

end

thingSpeakWrite(writeChannelID,turnTvOff,'timestamp',time,'WriteKey',writeAPIKey);

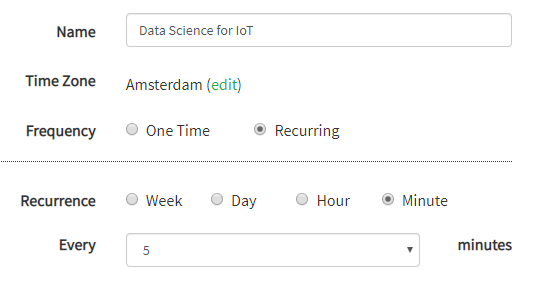
```

Vul de 4 variabelen in genaamd readAPIKey, readChannelID, writeAPIKey en writeChannelID.

De readAPIKey en readChannelID moeten de waarde hebben van de channel van de Incomming data. En de writeAPIKey en writeChannelID moeten de waarde hebben van de channel van de Outgoing data.

Klik op ‘Sava and Run’ om de code uit te voeren. Deze code wordt dan helaas maar 1 keer uitgevoerd. Om deze code constant uit te voeren en dus up-to-date te houden moeten we TimeControle gebruiken. Dit is een app op Teamspeak die acties uit kan voeren op ingestelde tijdstippen. Ook kan deze app acties herhalen.

De makkelijkste manier om dit in te stellen is om bij de Matlab Analysis code omlaag te scrollen naar ‘Schedule Actions’ en op ‘TimeControl’ te klikken. Dit brengt je meteen naar een pagina om TimeControl in te stellen op de MatlabAnalysis code.

Stel deze TimeControl in zoals op de afbeelding hieronder.

Als alle stappen goed zijn opgevolgd stuurt de Raspberry pi data naar Thingspeak en weer terug om te besluiten of de tv uitgezet kan worden of niet.