

Hack The Future 2016 - Mission to Mars

Vooreerst, kolonisten van het eerste uur, laat me jullie welkom heten op Mars. Als voorhoede van een grootschaligere volksverhuizing naar de rode planeet is het onze taak om deze nieuwe omgeving te gaan monitoren. Daarvoor krijgt elk team een kwadrant in onze sector toegewezen met bijhorende sensoren. Het is dus jullie taak om de gegevens van deze sensoren weer te geven op een digitaal dashboard en dan ook door te sturen naar het hoofdkwartier van onze sector.

De sensoren bestaan uit:

Druksensor

Om te beginnen implementeer je de Panic Button. Het is de bedoeling dat als je deze knop indrukt jullie eigen dashboard zal oplichten. Als je nogmaals op de button drukt, moet het alarm verdwijnen. Zorg er vervolgens ook voor dat het alarm het hoofdkwartier bereikt, zodat wij de cavalerie kunnen sturen.

Soundsensor

De geluidssensor moet het geluidsniveau van jullie omgeving in de gaten houden. Om de waarde van deze sensor om te zetten in decibels nemen jullie de \log_{10} van de uitgelezen waarde + 1 en vermenigvuldig je die met twintig ($20 * \log_{10}(\text{value} + 1)$). Geef de resultaten weer in jullie dashboard en stel een baseline vast. Wanneer er een te grote afwijking is moeten jullie wederom een alarm melding geven. Geef ook de actuele waarden om de minuut door aan het hoofdkwartier of onmiddellijk indien de alarmwaarden bereikt worden.

Ultrasound Distance sensor

In tegenstelling tot wat sommige extreem groene bewegingen ons willen doen geloven, zijn voor deze sensoren geen dolfijnen gebruikt. Dit geheel terzijde, is het de bedoeling dat jullie deze sensor plaatsen, gericht op een vast punt. Indien juist geïmplementeerd geeft deze jullie de afstand in centimeters terug. Stel de afstand in als jullie base value. Geef deze value eenmaal door aan het hoofdkwartier. Wanneer de sensor een afwijking van 10cm vertoont dienen jullie een alarm door te geven aan het hoofdkwartier. Op dit moment wordt het bestaan van Marsmannetjes noch ontkend noch bevestigd... Better safe than sorry...

Lightsensor

Naar analogie met de geluidssensor stellen jullie zelf een base waarde vast en sturen jullie deze om de minuut naar het hoofdkwartier. Bij sterke afwijkingen rapporteren jullie nu ook weer een alarm

Temperature and Humidity sensor

Deze sensor geeft verschillende waarden terug. Momenteel zijn we enkel geïnteresseerd in de temperatuur in graden Celsius. Bedenk zelf hoe je deze het best implementeert in je eigen dashboard en breng ook het hoofdkwartier op de hoogte.

In welke technologieën jullie wensen te werken zijn jullie eigenlijk vrij. Vanuit het hoofdkwartier kunnen wij technische expertise bieden voor Java, Python, NodeJS en Angular. Maar ben je meer vertrouwd met een ander framework... Wij houden je zeker niet tegen. Veel succes!

Waarop jureren wij

Het zal niet evident zijn om de gehele opdracht zoals hierboven beschreven te gaan implementeren. We kijken dan ook niet enkel naar het resultaat of hoever elk team geraakt is. We bekijken ook hoe jullie te werk zijn gegaan en werpen ook een blik op jullie code en 'architectuur'. Dus probeer ons van onze sokken te blazen met een kwalitatief resultaat eerder dan een kwantitatief zootje ;-).

Dashboard

De api vinden jullie in ons wifi netwerk op <http://dashboard.local:4000/api>

Hier stuur je een post request naartoe met volgende body:

```
{  
  team: { id: [teamID] }  
  sensor: {  
    id: [sensorID]  
    state: [boolean]  
    value: [sensor value]  
  }  
}
```

zorg er voor dat het een application/x-www-form-urlencoded (URL-Encoded Forms) is.

NodeJS

<https://github.com/DexterInd/GrovePi/tree/master/Software/NodeJS>

Python

Java