



▲ Afbeelding 3. Gezichtsdetectie.

hierboven voor de groene led. Zorg wel dat je nu pin **GPI019** selecteert en als naam **Rode led** invoert. Verbind tenslotte de twee nodes met elkaar en druk op **Deploy**.

Door nu met je hand langs de bewegingssensor te bewegen, springt het rode ledje aan. Na enkele seconden dooft hij weer. Hiervan willen we een melding hebben op het dashboard. Voeg uit de categorie **function** een **function** node toe en uit de categorie **dashboard** een **text** node. Verbind **Bewegingssensor** met de **function** node en die op zijn beurt met de **text** node. Dubbelklik op de functie en geef als naam **Bepaal melding**. Bij **Function** tik je de volgende code in:

```
> if (msg.payload == 1)
> {
>   msg.payload = "Iemand
aanwezig";
> }
> else
> {
>   msg.payload = "Alles
rustig";
> }
> return msg;
```

In Node-RED is **msg** het bericht dat

nodes onderling uitwisselen. Dat heet een object. Zo'n object heeft kenmerken en properties. **payload** is het standaard kenmerk. De inhoud daarvan vraag je op met de combinatie **msg.payload**. Bij beweging vult **Bewegingssensor** die met de waarde 1, anders blijft die 0.

Dubbelklik op de **text** node. Kies bij **Group** voor **[Mijn schakeling] Mijn ledjes**. Dat zijn het tabblad en de groep, die je hierboven bij het groene ledje hebt gedefinieerd. Geef als label **Bewegingsmelding**. Vergeet niet op **Deploy** te klikken en ga dan naar het dashboard. Beweeg de hand over de sensor. Het ledje en de bewegingsmelding reageren hierop synchroon.

MAAK EEN FOTO

Laten we nu vastleggen wat zich bij de sensor afspeelt. Voeg allereerst de module **node-red-contrib-camera-api** toe via het menu **Manage palette**. Plaats nu de extra **camerapi-takephoto** node uit de categorie **Raspberry Pi** in de workspace en verbind die met **Bewegingssensor**.

Om te zien wat de camera opneemt, voeg je de module **node-red-contrib-image-tools** toe aan de palette. Plaats uit de nieuwe categorie **image tools** de **viewer** node in

de workspace en koppel hem aan de uitvoer van de **camerapi-takephoto** node. Dubbelklik op deze laatste en zet **File Mode** op **Buffered Mode**. Hiermee zorg je dat de gemaakte foto als object beschikbaar blijft voor de viewer en niet als bestand naar de Raspberry Pi wordt weggeschreven. Druk op **Deploy**. Door nu in de buurt van de sensor te komen zie je in de preview een opname van jezelf.

LAAT JE GEZICHT ZIEN

Als kers op de taart maken we het af met gezichtsdetectie. Daarvoor heb je de module **node-red-contrib-opencv** nodig. De informatie daarvan zegt dat het pakket **libopencv-dev** nodig is:

```
> sudo apt-get install libopencv-dev
```

Installeer daarna de module via het menu **Manage palette**.

De **opencv** node vind je in de categorie **analysis**. Voeg die toe aan de flow en koppel daaraan een tweede **viewer** node. Dubbelklik op de **opencv** node en zet **Detect** op **Face**. Standaard staat bij **Output** de optie **JSON**. De node vult dan **msg.payload** met de coördinaten van de rechthoek rondom het gezicht. In

overige gevallen is **msg.payload** leeg. Verander dit in **Image buffer**, zodat je een foto krijgt samen met een rode cirkel rondom het gezicht.

Voor de gezichtsdetectie heeft **opencv** wat tijd nodig. Om een wachtrij te voorkomen, begrenzen we het aantal foto's. Voeg daartoe uit de categorie **function** een **delay** node toe. Zet die tussen de **camerapi-takephoto** en **opencv** node. Dubbelklik erop, kies als actie **Rate Limit**, zet **Rate** op **1 msg per 60 Seconds** en vink de optie **drop intermediate messages** aan. Klik op **Deploy** en test of je gezicht gevonden wordt door voor de camera te verschijnen.

EN VERDER

Ga nu zelf aan de slag. Gebruik het bovenstaande om bijvoorbeeld een automatisch kattenluik te maken of laat een LCD-scherm aanspringen als iemand aanbelt. Verder zijn er meer dan genoeg modules om zonder al teveel moeite een geheel nieuw project te beginnen. Met Node-RED is programmeren gewoon weer leuk. ◀

LINKS

Node-RED: nodered.org
 Raspbian: raspberrypi.org
 Etcher: github.com/balena-io/etcher/releases