

Van dingdong-deurbel naar IoT-deurbel

integreer uw deurbel in Home Assistant met ESPHome

Een bijdrage van

Idee, ontwerp, tekst en illustraties: **Clemens Valens**

Schema: **Kurt Diedrich**

Redactie: **C.J. Abate**

Vertaling: **Evelien Snel**

Layout: **Giel Dols**

In een eerder artikel [1] presenteerde ik de open-source domoticaprojecten ESPHome en Home Assistant. Ook legde ik uit hoe we die kunnen installeren en in gebruik nemen. Dit artikel laat zien hoe ik de deurbel van ons huis heb aangesloten op het systeem. Het belangrijkste doel was het detecteren van bel-events in Home Assistant, zodat deze kunnen worden doorgestuurd naar die plaatsen waar de deurbel zelf niet te horen is.



Figuur 1. Zo maken we van een eenvoudige deurbel een connected deurbelsysteem.

De drukknop van onze elektromechanische dingdong-deurbel is vele jaren geleden vervangen door een draadloos exemplaar met verschillende beltonen. Hij werd geleverd met twee ontvangers, waarmee het bereik van de deurbel sterk werd vergroot. Helaas wordt alles gevoed door batterijen, en als die leeg raken, wordt het systeem onbetrouwbaar. Bovendien raken bezoekers vaak in verwarring, omdat ze niets horen als ze op de knop drukken. Bij ons "belt de postbode altijd twee keer".

Omdat ik onlangs met domotica ben gaan experimenteren, was dit een mooie gelegenheid om de deurbel te updaten en in het systeem te integreren. Om geen kabels te

hoeven aanleggen, heb ik gekozen voor een WiFi-verbinding tussen de deurbel en de domoticacontroller.

De oude deurbel zelf en de aansluitdraden, inclusief de stroomvoorziening, waren op hun plaats gebleven. Dat bleek heel handig te zijn, omdat er binnenin de bel voldoende ruimte was om een zelfgebouwde draadloze deurbel-interface in te bouwen en ook te voeden.

Omdat ik de werkende draadloze deurbel niet wilde dumpen, besloot ik de drukknop ervan in te bouwen in mijn nieuwe WiFi-deurbel. Zo ontstond een deurbelsysteem met drie technologie-generaties in één systeem: ik heb nu de mechanische dingdong-bel plus het draadloze systeem dat verschillende

deuntjes speelt als er gebeld wordt, plus een interface naar de cloud (zie **figuur 1**). Ook de hoorbare feedback voor de bezoeker is daarmee hersteld.

Bouw een deurbelinterface

Ik heb voor de WiFi-deurbelinterface gekozen voor een op ESP8266 gebaseerd NodeMCU-board (**figuur 2**) omdat dit een analoge ingang heeft die ik wilde gebruiken (later meer hierover). Ook heeft het geen USB/serieel-converter nodig, dus naast een soldeerbout is er geen speciaal gereedschap voor dit project vereist.

Zoals gezegd is de knop van de draadloze deurbel met de oude, bestaande draden

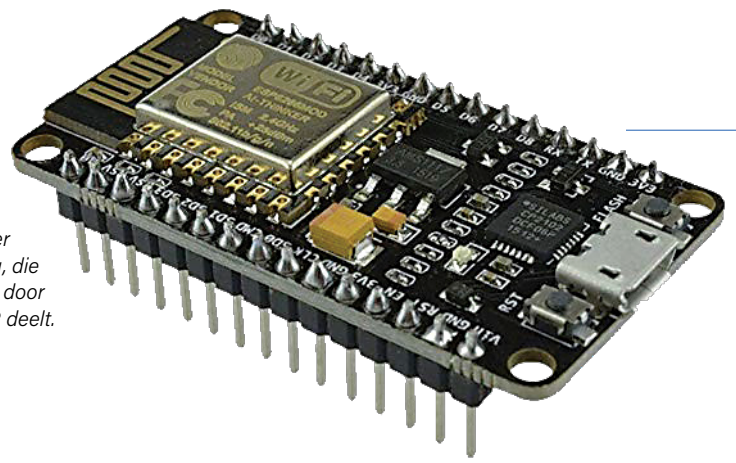
verbonden met de deurbelinterface. Omdat de drukknop buiten zit en blootgesteld is aan de elementen, heb ik de verbinding beschermd met serieweerstanden, een filtercondensator en klemdiodes.

De spoel van de deurbel trekt ongeveer één ampère bij voeding met 12 V AC. Dat is veel te veel voor de kleine drukknop van de bel (zulke contacten kunnen doorgaans maximaal zo'n 50 mA aan), maar met een relais of een vermogenstransistor gaat het wel. Ik heb een klein 5V-relais gebruikt, dat ik via een transistor laat aansturen door de microcontroller. Het volledige schema is weergegeven in **figuur 3**.

Controleren van de batterijspanning

De draad naar de drukknop van de bel is rechtstreeks verbonden met de koude kant van de pullup-weerstand van de drukknop.

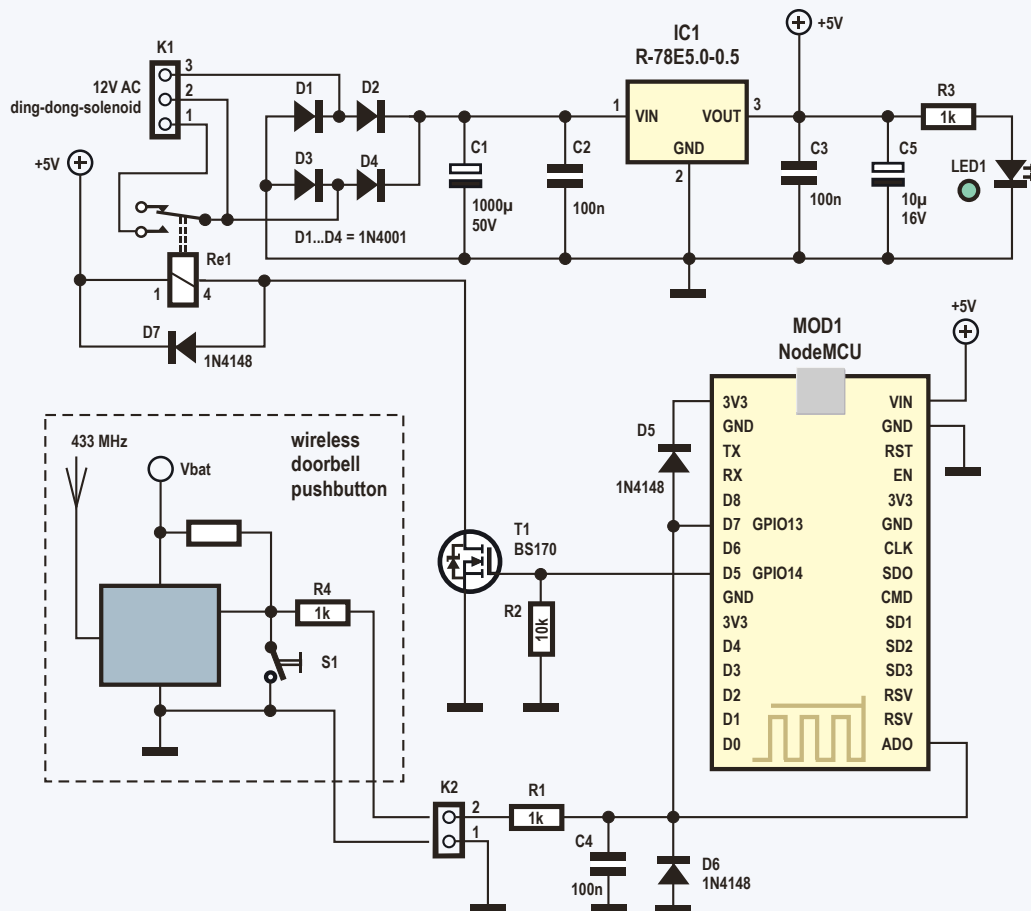
Figuur 2. De NodeMCU-module heeft een spanningsdeler op zijn analoge ingang, die deingangsspanning door 3,2 deelt.



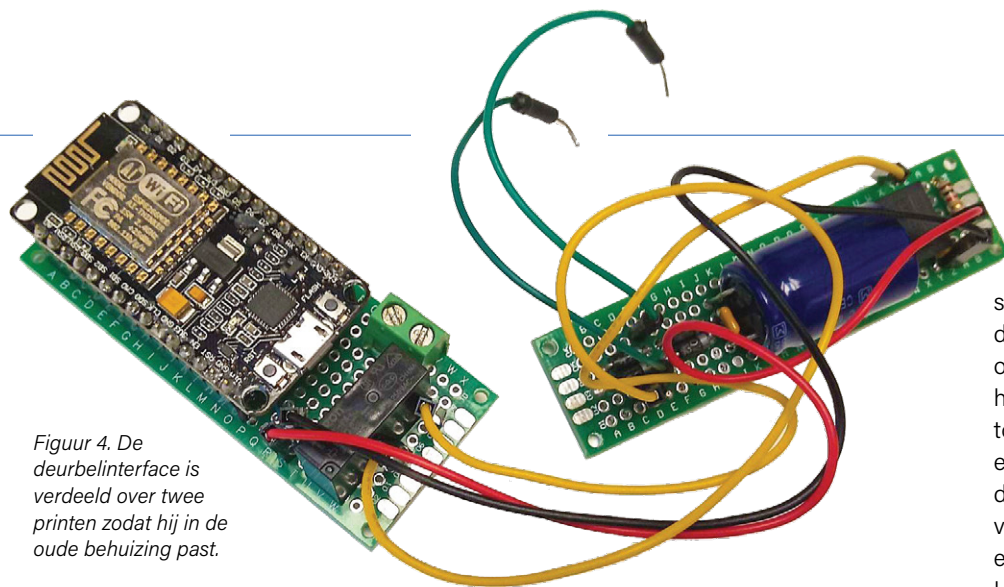
Dat betekent dat zolang de drukknop niet wordt ingedrukt, op deze draad de spanning van de batterij in de behuizing van de drukknop staat. Daarom heb ik de draad niet alleen aangesloten op een GPIO-pen, maar ook op een analoge ingang. Daardoor kan Home Assistant het batterijniveau van de draadloze deurbel in de gaten houden plus de deurbel-events doorgeven. Dat is nuttig, want het draadloze bereik van de bel is evenredig met de batterijspanning. Wanneer de

spanning te laag wordt, bereikt het signaal de ontvangers niet en wordt het systeem onbetrouwbaar.

Natuurlijk is het ook mogelijk om de analoge ingang zowel te gebruiken voor het detecteren van het indrukken van de knop als het monitoren van het batterijniveau, maar dat maakt de firmware ingewikkelder. Door een GPIO-pen te gebruiken die anders toch overbleef, houd ik het lekker simpel.



Figuur 3. De deurbelinterface heeft een 'schone' voedingsspanning nodig. De pennen die met de drukknop zijn verbonden, moeten worden beschermd omdat deze buiten is gemonteerd, op een plaats waar hij wordt blootgesteld aan de elementen.



Figuur 4. De deurbelinterface is verdeeld over twee printen zodat hij in de oude behuizing past.

WEET U NOG?

ESPHome is open-source firmware voor een op een ESP8266 of ESP32 gebaseerd remote apparaat, zoals een sensor of een schakelaar (en nog veel meer).

<https://esphome.io/>

Home Assistant is een open-source domoticaproject dat remote apparaten, zoals sensoren en schakelaars (en nog veel meer), verbindt met actuatoren en diensten en dergelijke, zodat ze door complexe regels kunnen worden aangestuurd.

<https://www.home-assistant.io/>

Zie [1] voor meer informatie.

Voeding

De voeding was lastiger, omdat de oude bel alleen een 12V-transformator had, terwijl ik 5 V DC nodig had voor het relais en het NodeMCU-board.

De spoel van de gong is een vrij zware belasting voor de transformator en zorgt ervoor dat de uitgangsspanning van de transformator inzakt als hij wordt geactiveerd. Een grote buffercondensator filtert deze spanningsdips eruit. Zonder die condensator wordt de NodeMCU-module bij elke druk op de knop gereset.

De 5V-regelaar is een kleine 7805-compatible schakelende regelaar. Dat voorkomt problemen met de koeling.

Ik heb ook een indicatie voor de voedingsspanning toegevoegd omdat de NodeMCU die niet heeft en het toch wel praktisch bleek om er een te hebben.

Door de schakeling van de deurbelinterface te verdelen over twee prototypeprintjes, één voor de voeding en de andere voor de rest,

heb ik alles goed in de behuizing van de oude deurbel kunnen inpassen (figuur 4).

Configuratie van de ESPHome-firmware

De software voor de deurbelinterface maakt gebruik van ESPHome. Het voordeel hiervan is duidelijk: ongeveer 30 minuten programmeren (of moet ik zeggen typen en bewerken?) is genoeg om een volledig functioneel apparaat met OTA-programmering (Over the Air), een eigen webinterface en een perfecte integratie in de Home Assistant te bedenken. Bovendien kan het naar believen worden aangepast met uw smartphone (of tablet of computer) wanneer u daar zin in hebt.

Het programmeren gebeurt niet in C, C++ of Python, maar in YAML. YAML is geen programmeertaal, maar een opmaaktaal die alleen regels geeft voor het opmaken van tekst. ESPHome accepteert een YAML-script of -configuratiebestand als invoer en zet dit

om in C/C++-code die vervolgens wordt toegevoegd aan en gecompileerd met de rest van de bibliotheek. Het script specificeert de onderdelen van ESPHome die u wilt opnemen (WiFi-instellingen, wel of geen webinterface – dat soort dingen), hoe en welke soorten sensoren en actuatoren zijn verbonden met de I/O-pennen, en wat de controller met die pennen moet doen. De meestvoorkomende dingen zijn voorgedefinieerd, zodat het programmeren eenvoudig en snel gaat.

Het kan even duren om aan YAML te wennen, maar het is niet ingewikkeld zolang u het inspringen goed in de gaten houdt.

Koppeling van relais en drukknop

Het relais dat de spoel van de dingdong-bel aanstuurt, is gedefinieerd als een interne 'switch' die is aangesloten op GPIO14. Door hem te specificeren als 'internal' wordt hij onzichtbaar voor Home Assistant, zodat uw systeem overzichtelijk blijft.

switch:

```
- platform: gpio
  pin: GPIO14
  id: ding_dong_relay
  internal: true
```

De schakelaar krijgt een ID, zodat we hem kunnen gebruiken in automatiseringen (zie hieronder).

De drukknop van de bel is een 'binary_sensor' op pin GPIO13. Omdat hij actief laag is, vertellen we dat aan ESPHome met het keyword 'inverted'.

binary_sensor:

```
- platform: gpio
  name: "my doorbell pushbutton"
  pin:
    number: GPIO13
    inverted: true
  on_press:
    then:
      - switch.turn_on:
ding_dong_relay
      - delay: 200ms
      - switch.turn_off:
ding_dong_relay
```

Als een drukknop wordt ingedrukt, genereert ESPHome een 'on_press'-event dat het naar de Home Assistant stuurt. Deze events kunnen ook binnen het YAML-bestand worden gebruikt als signalen om een of meer acties te triggeren. Dat wordt 'automatise-

Vragen of opmerkingen?

Hebt u vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur of naar de redactie van Elektor, via redactie@elektor.com.

ring' genoemd en is een krachtige functie van ESPHome.

Hier gebruiken we het 'on_press'-event om een puls van 200 ms te produceren op de relaisuitgang. Om dit te bereiken specificeren we het ID van het relais. Nu zal de dingdong-bel geactiveerd worden als iemand op de knop drukt, zelfs als het netwerk niet actief is. Kortom, dit herstelt de oorspronkelijke functie van de deurbel, zoals die meer dan 40 jaar geleden werd geïnstalleerd, maar in plaats van een eenvoudige directe draadverbinding met de drukknop hebben we enkele duizenden transistors gebruikt.

Houd de batterijspanning in de gaten

De batterij van de beldrukknop is gedefinieerd als een 'sensor' op A0. Het type ('platform') is 'adc' omdat het een spanning is die moet worden omgezet in een digitale waarde. De NodeMCU-module deelt de ingangsspanning op A0 door 3,2. Met behulp van een 'filter' kunnen we in ESPHome de uitgangswaarde van een sensor wijzigen, en een vermenigvuldigingsfactor wordt als een filter beschouwd. Als we dus de waarde van de sensor met 3,2 vermenigvuldigen, krijgen we de oorspronkelijke waarde terug.

sensor:

```
- platform: adc
  name: "doorbell battery level"
  pin: A0
  filters:
    - multiply: 3.2
```

Specificeren van 'lights'

De NodeMCU-module heeft een Flash-knop op GPIO0 en een LED op GPIO16 die we ook kunnen gebruiken. De Flash-knop kan voor testdoeleinden worden gebruikt; de LED kan een visuele indicatie geven dat iemand heeft aangebeld en dat Home Assistant het event heeft gezien. Daarom heb ik GPIO0 gedefinieerd als een binaire sensor (toegevoegd aan de binaire-sensorsectie).

light:

```
- platform: binary
  name: "LED1"
  output: led1
```

output:

```
- platform: gpio
  id: led1
  pin:
    number: GPIO16
    inverted: true
```

ESPHOME OP EEN MAAGDELIJKE CONTROLLER ZETTEN

Omdat een nieuwe (maagdelijke) ESP8266 of ESP32 nog niet ESPHome-compatibel is, moet hij de eerste keer via de seriële poort worden geprogrammeerd. Afhankelijk van het apparaat moet u mogelijk een USB/serieel-driver installeren op de computer waarop u uw ESPHome-ontwikkeling uitvoert (dus de computer waarop de Home Assistant draait) om dit te laten werken. De NodeMCU-boards met een Silabs (Silicon Laboratories) CP2102 USB-chip die ik gebruikte werkten rechtstreeks uit de verpakking op een Raspberry Pi.

Nadat u het apparaat hebt aangesloten op de computer met de Home Assistant, opent u het ESPHome-dashboard via de zijbalk (als u die optie hebt ingeschakeld) of door te klikken op 'Open Web UI' van de add-on kaart in de Supervisor Dashboard-view. Controleer of de seriële poort beschikbaar is in de keuzelijst rechtsboven die standaard op 'OTA (Over-The-Air)' staat. Als dat niet het geval is, start dan de ESPHome add-on opnieuw op (door terug te gaan naar het Supervisor Dashboard). Klik dan op de roze '+'-knop op het ESPHome Dashboard. Dit opent een wizard die u door het eerste gedeelte leidt. Zelf heb ik nooit een toegangswachtwoord opgegeven voor een apparaat, maar dat zou wel eens dom van me kunnen zijn. Selecteer als uploadpoort de seriële poort waarop het apparaat is aangesloten.

Om het principe te illustreren heb ik een 'light' gedefinieerd dat verbonden is met een uitgang die ik 'led1' heb genoemd. Hierdoor ontstaat een 'light-entity' in de Home Assistant, die andere mogelijkheden biedt dan een schakelaar. Ook heeft hij een ander pictogram. De LED zelf (op GPIO16) is gedefinieerd als een uitgang met ID 'led1'. Merk op dat ook deze actief laag is ('inverted' is 'true').

Wanneer de Home Assistant nu een bericht stuurt om 'led1' in of uit te schakelen, bestuurt dat de uitgang met de bijbehorende ID, 'led1'. Het volledige YAML-configuratiebestand voor ESPHome kan worden gedownload van [2]. Natuurlijk moet u de SSID en het wachtwoord van uw eigen WiFi-netwerk daarin invullen. U kunt ook de naam van het apparaat wijzigen.

Advertisement



Bedienen zonder aanraken

- Extra hygiëne door bedienen zonder oppervlak aan te raken
- Geschikt voor PCAP touchscreens en capacitieve toetsen
- Een vinger in de buurt houden is voldoende voor een goed schakelcontact
- Bepaal zelf of de "touchless" optie in- of uitgeschakeld is

info.nl@schurter.com | 0523-281200
schurter.nl

SCHURTER
ELECTRONIC COMPONENTS



GERELATEERDE PRODUCTEN

- > **NodeMCU ESP8266-microcontrollerboard**
www.elektor.nl/17952
- > **Engelstalig boek: IoT Home Hacks with ESP8266**
www.elektor.nl/19158

Gebruik de ESPHome add-on voor Home Assistant

Omdat ik de ESPHome add-on heb geïnstalleerd in Home Assistant, heb ik het YAML-script in de add-on geschreven. Op die manier zit het YAML-bestand van het apparaat al in het systeem, waardoor OTA-updates later gemakkelijk zijn. Maar de eerste keer, in ieder geval met een maagdelijk ESP8266-systeem, moet u de seriële poort gebruiken om de MCU te programmeren, omdat de OTA-programmering nog niet is geactiveerd (zie **kader**).

Automatisering in Home Assistant

De LED op GPIO16 is alleen gedeclareerd in het YAML-bestand van ESPHome, we hebben er geen automatiseringsregel voor opgegeven, en dus zal hij niets doen zonder tussenkomst van Home Assistant. Het idee was om hem te laten oplichten als er iemand aanbelt. Dat kan met een eenvoudige automatisering in Home Assistant. Het enige wat hij hoeft te doen is wachten op een druk op de knop en het inschakelen van de LED als dat gebeurt. Er zijn verschillende manieren om de automatiserings-editor te openen in Home Assistant. Een manier is door te klikken op 'Configuration' in het menu aan de linkerkant, en daarna op 'Automations' aan de rechterkant. Maak een nieuwe automatisering aan door op de ronde oranje '+'-knop te klikken in de rechter benedenhoek.

U kunt nu een regel invoeren en Home Assistant laten proberen om die om te zetten in de gewenste automatisering. Maar het is niet eenvoudig om een geschikte zin te vinden en dan kost het een hoop tijd om de door de Home Assistant aangemaakte kreupel automatiseringsregel te herstellen, dus hier klikken we op 'Skip'.

Voer een naam in voor de automatisering en eventueel een beschrijving. Laat de modus op 'Single (default)' staan.

Triggers en acties

Er zijn verschillende manieren om een trigger-event te specificeren dat de automatisering kan starten. Voor mij is de meest intuïtieve manier in dit geval om een 'Device' te kiezen als 'Trigger type'. Zoek in de 'Device'-lijst naar de naam die u hebt opgegeven in het YAML-bestand. Zoek op dezelfde manier in de 'Trigger'-lijst naar de naam die u de drukknop heeft gegeven.

Er zijn geen voorwaarden, dus dat deel kunt u overslaan.

Acties zijn vergelijkbaar met triggers, en net als triggers is er meer dan één manier om het gewenste resultaat te bereiken. Een manier is om 'Device' te kiezen als 'Action type' en uw apparaat te selecteren uit de lijst 'Device'. Selecteer in de lijst 'Action' 'Turn on led1' (als u de LED anders heeft genoemd, gebruik dan die naam).

Sla uw werk op door op het ronde oranje diskette-icoontje rechtsonder te klikken. Controleer of de automatisering werkt door aan te bellen. Als u dat doet en het werkt, zult u merken dat u nog geen manier hebt om de LED weer uit te schakelen.

Voeg een lampje toe aan de gebruikersinterface

Om dat op te lossen, voegen we een zogenaamde 'card' toe aan de overzichtspagina van de Home Assistant. Hierdoor is het niet alleen mogelijk om de LED uit te schakelen (door op de kaart te klikken), maar ook om op afstand de status van de LED te bekijken, zodat u niet helemaal naar de deurbel hoeft te lopen om te zien of hij wel of niet brandt.

Klik op 'Overview' in het menu aan de linkerkant. Klik op de drie stippen in de rechterbovenhoek ('Open Lovelace UI menu') en selecteer 'Configure UI'. Klik dan op de ronde oranje '+'-knop in de rechter benedenhoek (in Home Assistant muist u veel veel heen-en-weer). Zoek de 'light'-kaart en klik erop. Zoek in de lijst 'Entity' naar de LED en selecteer deze. Voer ook een naam in, want dat maakt het gemakkelijker om het lampje te herkennen in het overzicht. Voorlopig is het misschien uw enige lampje, maar als uw domoticasysteem evolueert, komen er waarschijnlijk meer bij en dan is een goede naam heel handig.

Sla de kaart op en verlaat de UI-configuratiemodus door op het kruisje in de linkerbovenhoek ('Close') te klikken.

Door op de kaart te klikken kunt u nu de status van de LED veranderen.

Conclusie

Als u de stappen in dit artikel hebt gevolgd (en mijn eventuele fouten hebt gecorrigeerd, en alles hebt aangepast aan de nieuwste versies van de zich steeds verder ontwikkelende projecten ESPHome en Home Assistant), dan hebt u nu een deurbel met WiFi-aansluiting die u kunt automatiseren in Home Assistant. U kunt er veel meer (nuttige) dingen mee doen dan het laten branden van een LED, maar dat laat ik aan u over. Dit artikel gaf u de basisprincipes om aan de slag te gaan, en nu is het aan u om dingen te verbeteren en aan te passen aan uw eigen wensen en behoeften. 

200089-04

WEBLINKS

- [1] **Domotica – helemaal niet moeilijk:**
www.elektormagazine.nl/magazine/elektor-157/59027
- [2] **How-To: Integrate Your Doorbell in Home Assistant Using ESPHome:**
www.elektormagazine.com/labs/esphome-doorbell