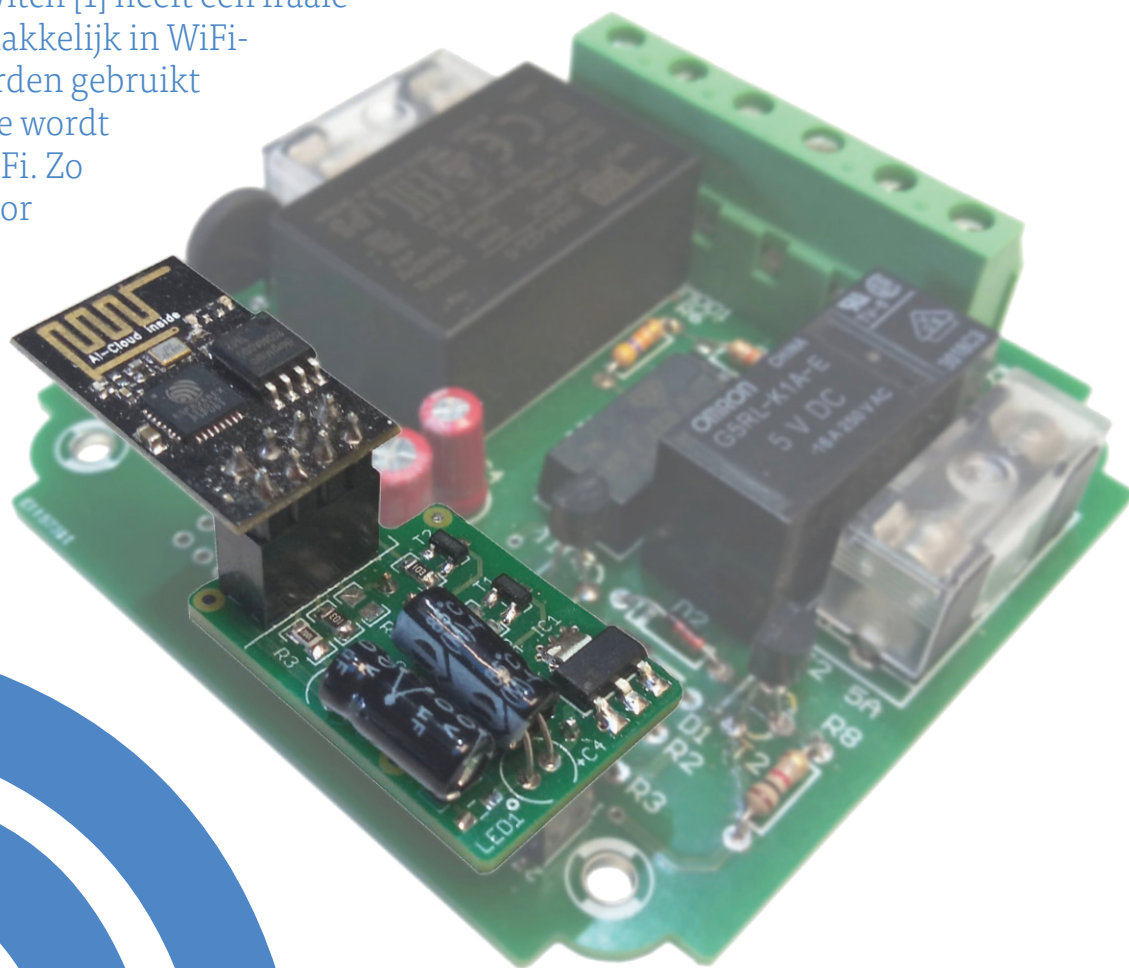


Wi-Fi voor de LoRa Switch

geïntegreerd in Home Assistant met ESPHome

Clemens Valens (Elektor)

De Elektor LoRa Switch [1] heeft een fraaie relaisprint die gemakkelijk in WiFi-netwerken kan worden gebruikt als de LoRa-module wordt vervangen door WiFi. Zo krijgen we de Elektor WiFi Switch.



In het maart/april-nummer 2020 van *Elektor* hebben we een afstandsbediende schakelaar gepubliceerd [1]. Deze bood terugkoppeling van de stand van de schakelaar en communicatie via LoRa. Omdat hij was ondergebracht in een waterdichte IP66-behuizing, was hij geschikt voor gebruik buitenshuis. Het was

daarnaast een modulaire project met het relais en de voeding op de ene print en het LoRa-communicatiegedeelte op de andere. Dat interesseerde me, omdat ik op zoek was naar een buitenschakelaar die gemakkelijk in mijn domoticasysteem geïntegreerd kon worden. Dit systeem is gebaseerd op Home

Assistant en WiFi, maar niet op LoRa. Ik weet zeker dat het mogelijk is om LoRa eraan toe te voegen, maar dat wilde ik niet. In plaats daarvan dacht ik dat ik, door simpelweg de LoRa-module te vervangen door een WiFi-module, daar ESPHome op zou kunnen draaien, dat prima met Home Assistant werkt (**figuur 1**) [2].

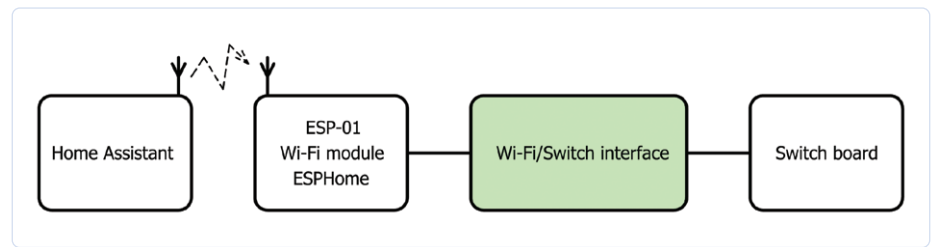
Definitie van de interface

Het eerste wat moet gebeuren als we iets willen aanpassen, is uitzoeken hoe het moet worden aangesloten. Het schema van de LoRa-schakelprint is niet erg ingewikkeld (figuur 2). Bovenaan hebben we de voeding op basis van een AC/DC-omvormer MOD1 die 5 V levert (inclusief wat spul ter bescherming en filtering).

In het midden zien we een bistabiel relais, RE1, dat de belasting in- en uitschakelt. Een bistabiel relais lijkt veel op een mechanische schakelaar doordat het zijn toestand behoudt, zelfs nadat de stroom is uitgeschakeld – net als een mechanische schakelaar dus. Er zijn twee stuursignalen om van toestand te veranderen: *set* en *reset*. Deze worden geleverd door de beide MOSFET's T1 en T2; hiervoor zijn actief-hoge stuursignalen vereist.

Merk op dat het relais is beveiligd met 5A-zekering F2 terwijl het relais maximaal 16 A kan schakelen. De reden hiervoor is dat de printsporen zoveel stroom niet verdragen; dezekering beschermt ze zodoende tegen doorsmelten. De schakelprint kan belastingen tot ongeveer 1 kW aan.

Linksonder zien we een optocoupler IC1 die parallel staat met de belasting. Als de belasting is ingeschakeld, is de optocoupler ook



Figuur 1. Blokschema. In dit artikel houden we ons vooral bezig met het lichtgroene blok.

ingeschakeld. De LED stuurt de transistor in geleiding en de output van de optocoupler wordt laag. Als de belasting is uitgeschakeld, is de output hoog. C4 filtert samen met R4 het AC-sig-naal (50 Hz of 60 Hz) weg, zodat een fraai stabiel niveau overblijft.

De connector rechtsonder maakt alle benodigde signalen toegankelijk. S1 is bedoeld voor aansluiting van een drukknop zodat de schakelaar ook lokaal bediend kan worden.

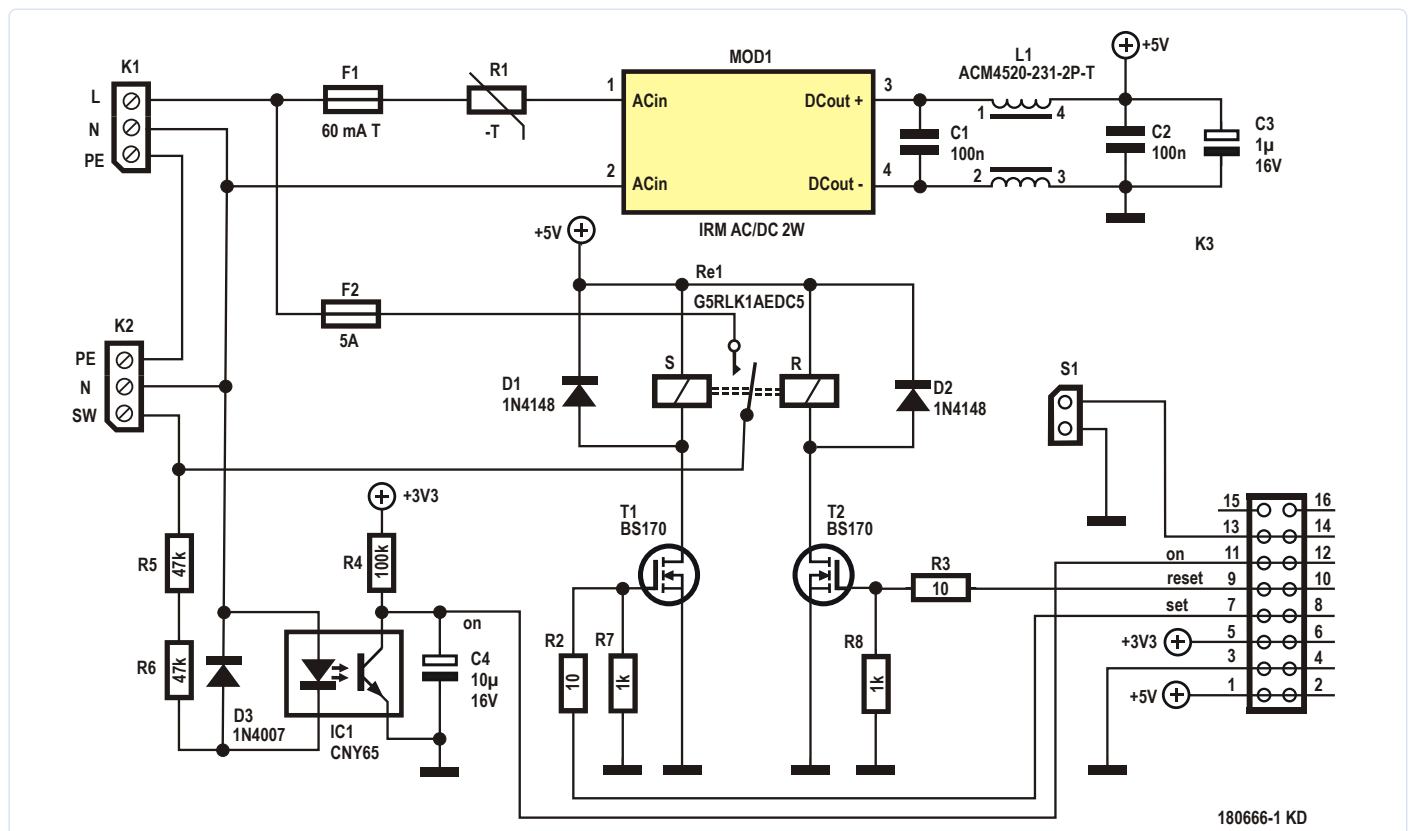
Een belangrijk detail is dat de optocoupler is aangesloten op een 3,3V-voeding, maar dat er geen 3,3V-voeding op het print voorhanden is. Deze spanning moet worden geleverd door de besturingsprint.

De ESP-01 WiFi-module

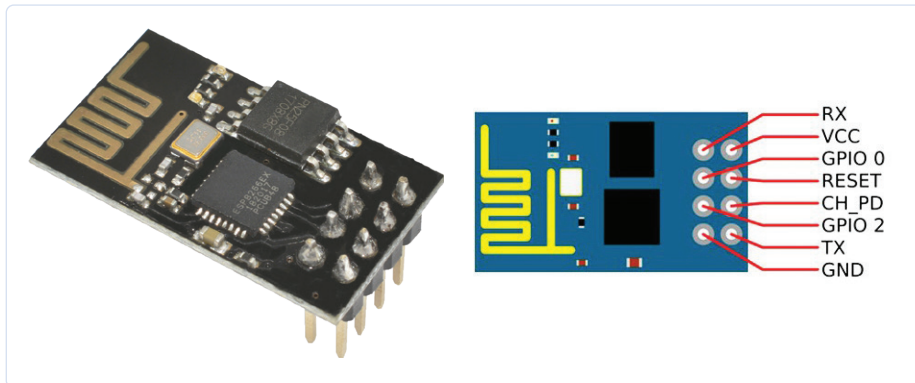
We kunnen nu de specificaties opschrijven voor de WiFi-besturingsprint die de relaisprint moet aansturen:

- er zijn twee digitale ingangen nodig, één om de output van de optocoupler te lezen en één voor S1;
- er zijn twee digitale uitgangen nodig om T1 en T2 aan te sturen;
- hij moet 3,3 V leveren voor de optocoupler.

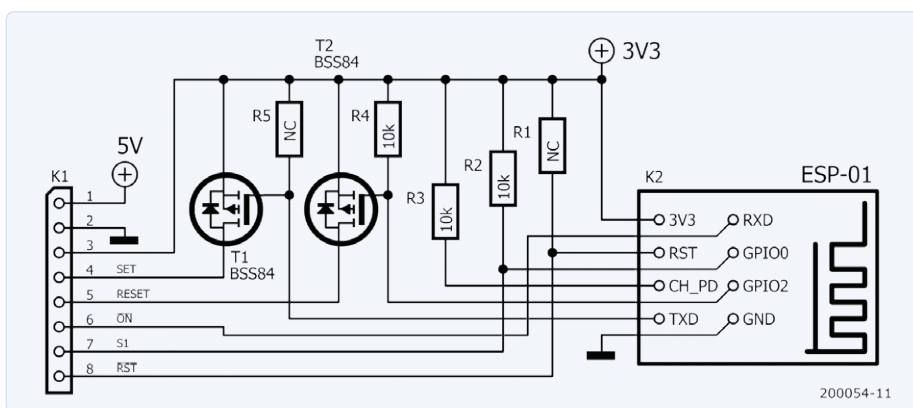
Wie zegt "vier digitale I/O's en WiFi en 3,3 V", denkt meteen aan de ESP-01-module, want deze heeft precies vier GPIO-poorten en WiFi



Figuur 2. De relaisprint van de Elektor Lora Node (maart/april 2020, [1]) heeft een bistabiel relais als hart.



Figuur 3. De kleine en goedkope ESP-01-module is uitgerust met een ESP8266EX-microcontroller met ingebouwde WiFi-ondersteuning. Hij heeft vier I/O-poorten (GPIO 0, GPIO 2, RX & TX) en heeft een 3,3V-voeding (VCC) nodig.



Figuur 4. Zo kan een ESP-01-module worden aangesloten op de LoRa-relaisprint. De MOSFET's ontkoppelen de ingangen van de uitgangen.



ONDERDELENLIJST

Weerstanden (alle 5%, 50V, 0,1, 0805):

R1,R2,R3,R4,R5 = 10k
R6 = 470Ω

Condensatoren:

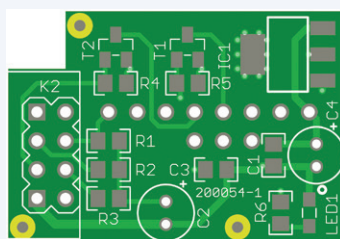
C1,C3 = 100n, 0805
C1,C4 = 10μ, 16V, steek 2 mm

Halfgeleiders:

IC1 = LD1117AS33
LED1 = LED groen
T1,T2 = BSS84

Diversen:

K1 = 8-polige pinheader, steek 2,54 mm
K2 = 2x4-polige busstrip, steek 2,54 mm
K3 = 3-polige pinheader, steek 2,54 mm
print 200054-1



afgebeeld op 150% van de ware grootte



en hij draait op 3,3 V. Deze populaire module op basis van de ESP8266EX is goedkoop en goed verkrijgbaar (figuur 3).

Er is echter een maar. De vier GPIO-poorten van de ESP-01 moeten omzichtig worden behandeld, omdat ze ook bepalen hoe de module na het inschakelen zal werken. Voor normaal gebruik moeten GPIO0, GPIO1 en GPIO2 hoog zijn tijdens het inschakelen. Dit staat ook in de opmerkingen onder de tabel 'Pin Definitions' in de datasheet van de ESP8266EX. Veel ESP8266-gebruikers zijn op de hoogte van de vereisten voor GPIO0 en GPIO2, maar niet iedereen kent die voor GPIO1, beter bekend als TXD – de seriële uitgang.

De GPIO-poorten en de stuursignalen

Op de relaisprint zijn twee van de vier stuursignalen altijd laag bij het opstarten; dat zijn 'Set' en 'Reset' vanwege de weerstanden R7 en R8. Als de belasting is ingeschakeld terwijl de relaisprint opstart, is het 'On'-signaal ook laag. Bedenk dat dit heel wel mogelijk is omdat het relais bistabiel is en ook zonder bekrachtiging in de laatste stand blijft staan. Ingang S1 zweeft (is open). De toestand bij het opstarten wordt bepaald door de WiFi-besturingsprint, tenzij iemand toevallig op de drukknop drukt wanneer het systeem opstart...

Dus al met al hebben we te maken met vier signalen die allemaal *laag* kunnen zijn bij het opstarten. Deze gaan naar vier poorten, waarvan er drie *hoog* moeten zijn bij het opstarten. Het gebruik van een ESP-01 module is in deze situatie dan ook een uitdaging.

De oplossing: voeg transistors toe

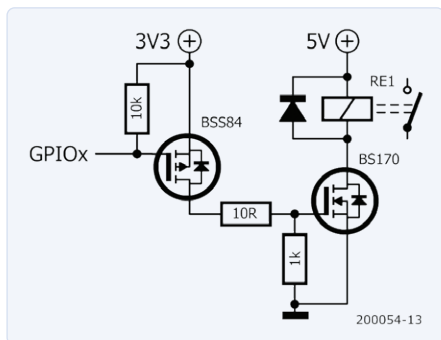
Figuur 4 toont de oplossing die ik bedacht heb:

- > RXD (of GPIO3) is de enige pin die vrijelijk kan worden aangesloten. Ik heb hem daarom aangesloten op de uitgang van de optocoupler omdat diens niveau bij het opstarten onvoorspelbaar is. RXD wordt een ingang.
- > Als GPIO0 laag is bij het opstarten, start de module op in de flash-programmeermodus. Dit kan handig zijn voor het actualiseren van de firmware, dus heb ik deze op S1 aangesloten. GPIO0 is dus ook een ingang.
- > GPIO2 en TXD (of GPIO1) zijn nu over en moeten daarom de uitgangen worden. Om ze te ontkoppelen van de lage impedanties ten gevolge van R7 en R8

op de relaisprint, heb ik P-MOSFET's gemonteerd, met een pull-up weerstand aan hun gate. Dit zorgt ervoor dat bij het opstarten GPIO2 en TXD een hoog niveau zullen zien. Wanneer de ESP-01 eenmaal is opgestart, kunnen ze worden geconfigureerd als actief-lage uitgangen.

De uitgangsdriver

De uitgangsdriver is getekend in **figuur 5**. Wanneer het GPIO-sigitaal laag wordt, zal de P-MOSFET BSS84 geleiden. Hierdoor wordt de gate van de N-MOSFET BS170 hoog getrokken, zodat deze ook gaat geleiden en een stroom door de spoel van het relais loopt. Als het GPIO-sigitaal hoog is, spert de P-MOSFET BSS84. De N-MOSFET BS170 spert nu ook vanwege de 1 kΩ pull-down weerstand aan de gate, en de relaisspoel krijgt geen stroom meer. De weerstand van 10 kΩ aan de gate van de BSS84 zorgt ervoor dat de GPIO-pin hoog wordt getrokken bij het opstarten.



Figuur 5. De uitgangsdriver: een P-MOSFET stuurt een N-MOSFET aan.

Laatste loodjes

Voor de 3,3V-voeding heb ik eenvoudig een low dropout-regelaar opgenomen in de 5V-voedingslijn (**figuur 6**). In de praktijk zijn R1, R6 en LED1 niet nodig omdat ze al op de ESP-01 module zitten, daarom hebben ze 'NC' als waarde (dus niet gemonteerd). R5 is ook niet nodig, maar ik heb hem voor alle zekerheid gemonteerd. Voor R1 en R5 is een waarde van 10 kΩ geschikt, en voor R6 iets als 470 Ω. Voor de interface tussen de WiFi-module en de relaisprint heb ik een kleine print ontworpen (**figuur 7**). De bestanden zijn te vinden op [3].

Het YAML-bestand voor ESPHome

Nu de in- en uitgangen zijn gedefinieerd, kunnen we het YAML-configuratiebestand voor ESPHome schrijven (zie **kader**; het bestand is beschikbaar op [3]). Dit is iets

YAML-configuratie voor de WiFi-switch

```
esphome:
  name: wifiswitch
  platform: ESP8266
  board: esp01_1m

wifi:
  ssid: "my_ssid"
  password: "my_passphrase"

# Logging impossible as UART0 pins are
# used for switches & sensors.
logger:
  baud_rate: 0 # Hardware UART off.

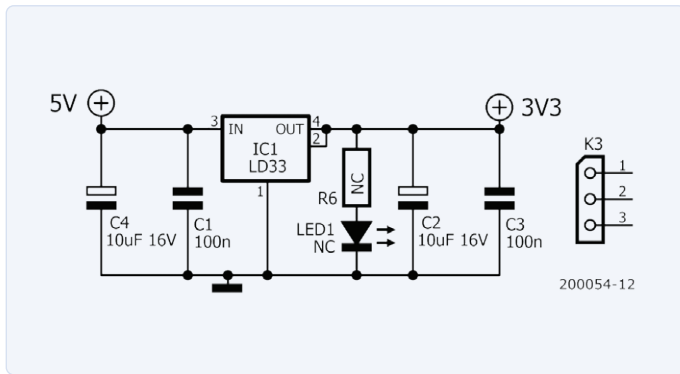
# Enable Home Assistant API
api:

# Enable Over-the-Air programming
ota:

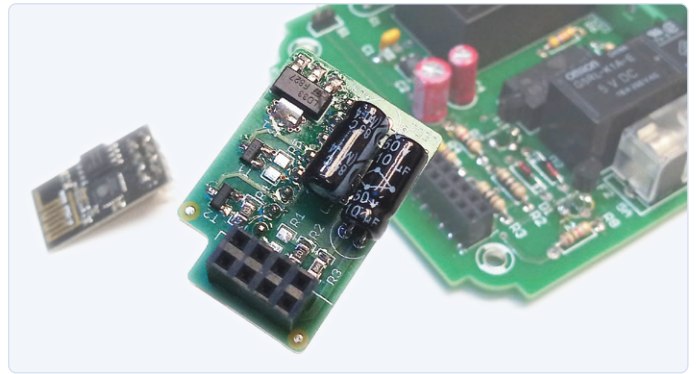
output:
  - platform: gpio
    id: relay_set
    pin: GPIO1 # TXD, may not be pulled low at startup.
    inverted: true
  - platform: gpio
    id: relay_reset
    pin: GPIO2 # Do not pull low at startup.
    inverted: true

switch:
  - platform: template
    name: "Wi-Fi switch"
    id: wifiswitch
    turn_on_action:
      # Pulse the Set pin.
      - output.turn_on: relay_set
      - delay: 0.1s
      - output.turn_off: relay_set
    turn_off_action:
      # Pulse the Reset pin.
      - output.turn_on: relay_reset
      - delay: 0.1s
      - output.turn_off: relay_reset
      # Use optocoupler state as switch state.
      lambda: return id(optocoupler).state;

# Pushbutton & optocoupler.
binary_sensor:
  - platform: gpio
    name: "Wi-Fi switch pushbutton"
    id: pushbutton
    pin:
      number: GPIO0 # Do not pull low at startup.
      inverted: true
    on_press:
      then:
        - if:
            condition:
              binary_sensor.is_on: optocoupler
            then:
              switch.turn_off: wifiswitch
            else:
              switch.turn_on: wifiswitch
  - platform: gpio
    name: "Wi-Fi switch optocoupler"
    id: optocoupler
    pin:
      number: GPIO3 # RXD, may be pulled low at startup.
      inverted: true
```



Figuur 6. Een simpele 5V-naar-3,3V adapter. K3 zorgt voor wat extra mechanische stabiliteit wanneer de WiFi-module op de connector van de relaisprint is geprikt.



Figuur 7. De interface tussen de EPS-01 WiFi-module en de relaisprint is op een klein printje gemonteerd.

gecompliceerder dan normaal vanwege het bistabiele relais dat twee pinnen gebruikt in plaats van één. ESPHome en Home Assistant kennen wel een zogenaamde 'cover'-component die dergelijke periferie kan aansturen, maar die is bedoeld voor garagedeuren en jaloezieën en dergelijke met een knop voor openen en sluiten. Deze hebben ook een speciaal pictogram. Onze schakelaar is gewoon een schakelaar en daarom willen we dat deze zich ook als een schakelaar gedraagt en er ook zo uitziet. Hiervoor gebruiken we een sjabloonschakelaar waarmee het gedrag bij in- en uitschakelen afzonderlijk kan worden gespecificeerd.

Eerst declareren we GPIO1 (beter bekend als TXD) als een geïnverteerde uitgang met

de naam *relay_set*. GPIO2 wordt gedeclareerd als een geïnverteerde uitgang met de naam *relay_reset*. We definiëren GPIO3 ook als een actief-lage binaire sensor voor de optocoupler-uitgang.

Vervolgens specificeren we een korte puls van 100 ms op de *relay_set*-uitgang om de belasting in te schakelen, en doen we hetzelfde op de *relay_reset*-uitgang om de belasting weer uit te schakelen. Als schakelaarstand retourneren we de toestand van de optocoupler. Om drukknop S1 als verwacht te laten werken (dus door indrukken wordt de belasting in- en uitgeschakeld) definiëren we GPIO0 als een actief-laag binaire sensor. Wanneer S1 wordt ingedrukt, wordt eerst de toestand van de optocoupler gecontroleerd. Als de belasting

is ingeschakeld, wordt het relais afgeschakeld; als de belasting is uitgeschakeld, wordt het relais geactiveerd. Dit krijgen we met een *if-then-else* constructie voor elkaar.

Klaar

Na het compileren van ESPHome met dit YAML-bestand en het flashen van de ESP-01-module moet de WiFi-schakelaar in Home Assistant verschijnen en klaar zijn voor automatiseringsgebruik. We kunnen hem nu aansluiten voor het betere schakelwerk. 

200054-04

Vragen of opmerkingen?

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

Een bijdrage van

Idee, ontwerp, tekst en illustraties:

Clemens Valens

Vertaling: **Eric Bogers**

Layout: **Giel Dols**



GERELATEERDE PRODUCTEN

> ESP8266 ESP-01 WiFi-module

www.elektor.nl/17326

> IoT Home Hacks with ESP8266

www.elektor.nl/19158

> LoRa-controlled Switch with State Feedback – Slave Unit kale print

www.elektor.nl/180666-1

WEBLINKS

- [1] L. Lemmens en M. Claussen, "De Elektor LoRa Node", Elektor, maart/april 2020: www.elektormagazine.nl/180666-04
- [2] C. Valens, "Domotica helemaal niet moeilijk", Elektor, september/oktober 2020: www.elektormagazine.nl/200019-04
- [3] Projectbestanden op Elektor Labs: www.elektormagazine.com/labs/wifi-switch