

# Eerste stappen met een ESP32-C3 en het IoT

## WiFi-knop en -relais



Mathias Claußen (Elektor)

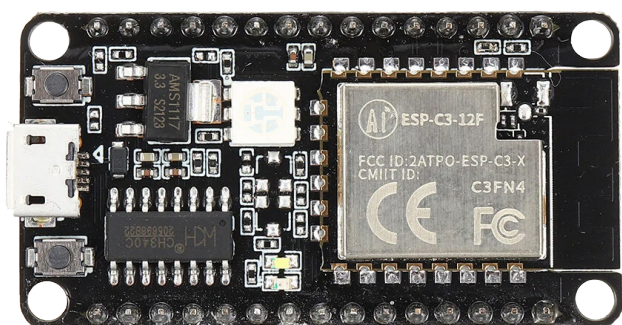
Het IoT is geen gesloten boek met geheime recepten. Krachtige controllers zoals de nieuwe ESP32-C3 en beginnersvriendelijke ontwikkelomgevingen zoals de Arduino IDE maken het ontwikkelen van kleine projecten een fluitje van een cent.

Als we het hebben over het Internet of Things (IoT), zien we dat steeds meer dingen in ons dagelijks leven met het internet zijn verbonden. Het begint met verlichting, verwarming en sensoren in en om huis en gaat verder met auto's, verkeerslichten, scheepscontainers en nog veel meer. In elk van de *connected* dingen worden kleine netwerkcompatibele componenten geïnstalleerd, die de uitwisseling van informatie mogelijk maken.

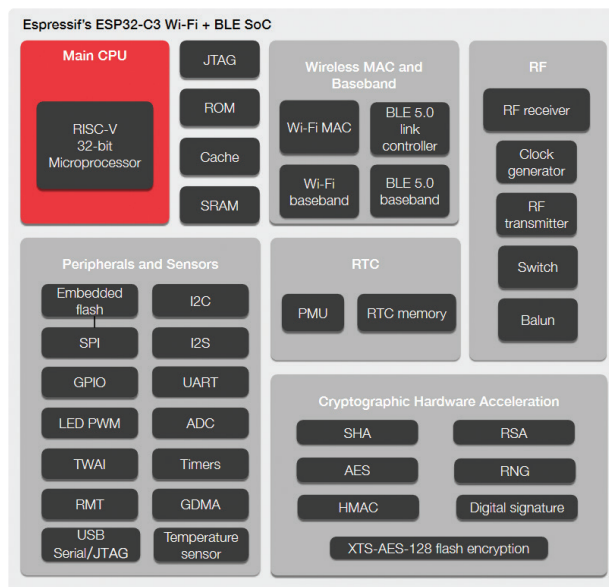
De beste manier om te leren hoe u uw eigen applicaties kunt verbinden met het IoT is door in de praktijk aan de slag te gaan. In dit artikel maken we een koppeling tussen een WiFi-drukknop en een WiFi-relais; het relais kan dan op afstand worden geactiveerd door de drukknoop en meldt zijn status terug aan de drukknoop.

### Keuze van de onderdelen

Zoals bij elk project moeten eerst de juiste componenten worden gekozen. De ESP-C3-12F-kit (**figuur 1**), verkrijgbaar in de Elektor Store, is hiervoor de ideale oplossing. Deze print is voorzien van een WiFi-enabled ESP32-C3 microcontroller van Espressif. De ESP32-C3 is een vervanger van de vertrouwde ESP8266. Naast een moderne CPU-kern heeft de chip een uitgebalanceerde mix aan periferie die zowel beginnersvriendelijk als krachtig is (zie onze bespreking van de ESP32-C3 [1]). Een blokschema van de geïntegreerde hardware is te zien in **figuur 2**. Naast de ESP32-C3 zijn ook een RGB-LED en een USB/serieel-converter op het board geïntegreerd. Voor ons project hebben we twee ESP-C3-12F kits nodig.



Figuur 1. De ESP32-C3-12F kit.



Figuur 2. Functieblokken van de ESP32-C3 (bron: ESP32-C3 datasheet).

Naast de ESP-C3-12F kits hebben we ook een sensor en een actuator nodig. Dat is waar de Elektor 37-in-1 sensor kit van pas komt, deze kit bevat 35 sensoren (de oorspronkelijke versie had er 37, maar twee daarvan bevatten kwik en zijn sindsdien uit veiligheidsoverwegingen weggelaten). Een overzicht van de sensoren van de kit (figuur 3) is te zien in figuur 4 en het informatieblad [2]. Eerst nemen we de Joystick-module en gebruiken de drukknopfunctie om het systeem een invoermogelijkheid te geven. De relaismodule kan nu worden aangesloten op het andere microcontrollerboard en fungeert als de actuator van het systeem. Voor het aansluiten van de modules zijn een paar (female/female) jumperkabels nodig. Deze maken deel uit van de Pimoroni "Mini Breadboards & Jumpers" kit (zie het kader **Gerelateerde producten**). We hebben ook een computer nodig, zoals een Raspberry Pi, die als lokale server voor de IoT-apparaten dient zodat deze hun data kunnen uitwisselen. Een originele Raspberry Pi versie 1 zou al volstaan, maar we raden ten minste een Raspberry Pi 2 aan voor deze toepassing. On-board WiFi was niet inbegrepen tot Raspberry Pi model 3B, dus om eerdere versies te kunnen gebruiken hebben we nog een eenvoudige WiFi-dongle of Ethernetkabel nodig zijn. Als u een kleine maar krachtige Raspberry Pi wilt kopen, kijk dan eens naar de Raspberry Pi

Zero 2 W bundel. Het hoeft echter niet per se een Raspberry Pi te zijn. Elke PC met een Linux-distributie zoals Ubuntu [3] zal ook voldoen. Voordat we aan het project beginnen, zullen we eerst bekijken hoe de besturing en uitwisseling van gegevens in deze opzet plaatsvindt.

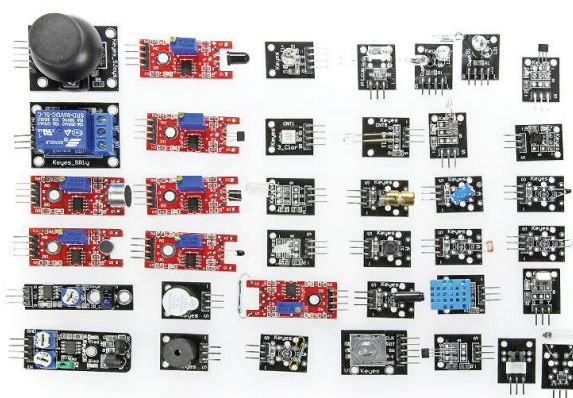
## MQTT

Elk IoT-apparaat, of het nu een sensor of actuator is, moet gegevens overdragen. Hiervoor kunnen we het ons moeilijk maken door ons eigen communicatieprotocol te ontwikkelen of we kunnen gebruik maken van gevestigde standaardprotocollen. Een inmiddels wereldwijd verbreid systeem is MQTT. Oorspronkelijk was dat een acroniem voor "Message Queuing Telemetry Transport", maar naarmate het systeem zich verder ontwikkelde beschreef de naam de functie steeds minder accuraat. In 2013 werd officieel besloten dat MQTT een label zou worden [4].

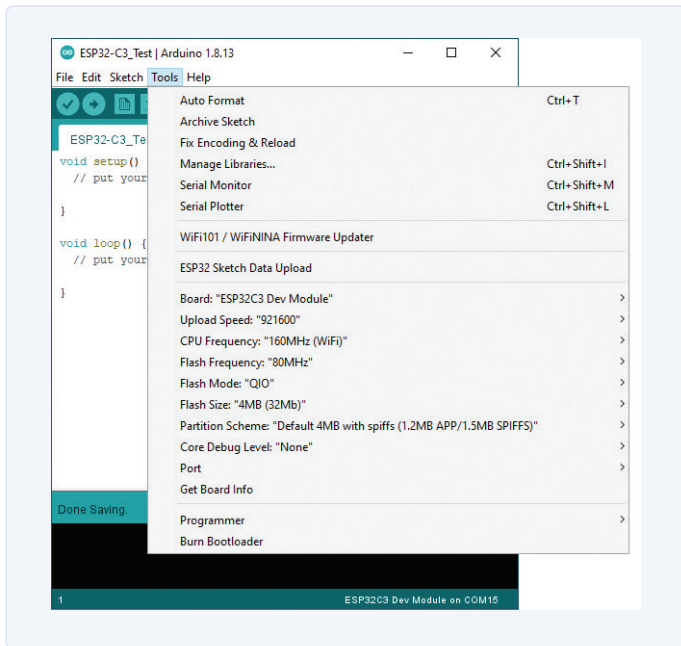
Het MQTT-protocol zorgt voor de uitwisseling van berichten met behulp van een broker (server) zonder te specificeren hoe de berichten eruit zien. Je kunt dit vergelijken met het versturen van een brief: de logistiek en het formaat van de envelop worden gespecificeerd door de postbode, maar het bericht in de envelop en de taal waarin het wordt



Figuur 3. De Elektor 37-in-1 sensor-kit.



Figuur 4. De kit bevat een heleboel sensoren en actuatoren.



Figuur 5. Configuratie van de ESP32-C3 in de Arduino IDE.

geschreven, is een zaak van de gebruiker. Welke 'taal' moeten we kiezen om onze boodschappen te versturen? Ook hier zijn er weer verschillende opties. Een populaire keuze (niet alleen voor MQTT) is Java Script Object Notation (JSON).

## JSON

JSON is een lichtgewicht formaat voor de overdracht van berichten dat gemakkelijk kan worden gegenereerd en geïnterpreteerd, zelfs door kleine microcontrollers. Bovendien is JSON-tekst niet alleen gemakkelijk te begrijpen voor mensen, maar ook gemakkelijk te schrijven. Een overzicht van de JSON-specificatie is te vinden op de JSON standaard-website [5]. JSON wordt niet alleen gebruikt in combinatie met MQTT, maar ook op vele andere gebieden. JavaScript, de programmeertaal waarvan JSON is afgeleid, is een van de kerntechnologieën waarop het World Wide

Web heden ten dage is gebaseerd. Een goede inleiding tot JSON met praktische voorbeelden staat op de site van Mozilla [6].

## Het opzetten van de IoT-omgeving: de MQTT Broker

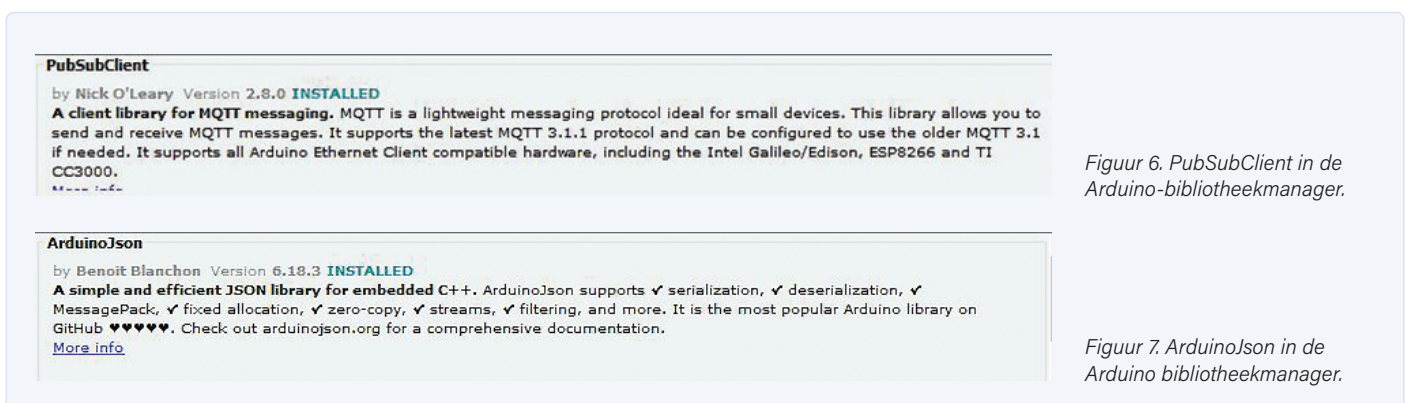
Zoals altijd helpt een goede voorbereiding onverwachte verrassingen achteraf te voorkomen. Om MQTT-berichten te verwerken, hebben we een broker nodig – ofwel eentje op het Internet ofwel een die we lokaal op ons netwerk kunnen installeren. Een lokale broker betekent dat we niet afhankelijk zijn van cloud-services voor IoT-functionaliteit die voor onze doeleinden alleen zal worden gebruikt om berichten over te dragen tussen lokaal aangesloten apparaten. De broker kan bijvoorbeeld worden gebouwd met behulp van een 'gepensioneerde' PC of Raspberry Pi. Node-RED biedt ons een complete gereedschapskist voor het ontwikkelen van netwerk-applicaties die zich niet beperken tot het verwerken van MQTT-berichten. Node-RED [7] is bij Elektor al vaak gebruikt voor het verwerken van MQTT-gegevens en kan dankzij de gedetailleerde instructies snel op een Raspberry Pi [8] of een PC [9] worden geïnstalleerd.

## De Arduino IDE

De Arduino IDE wordt hier gebruikt als ontwikkelomgeving. De editor van de IDE is niet de beste in zijn klasse, maar biedt momenteel wel de meest stabiele ondersteuning voor de ESP32-C3. De Arduino IDE [10] kan gratis van de Arduino-homepage worden gedownload en geïnstalleerd. Dan moet de ondersteuning voor de Arduino ESP32 (zoals beschreven in de Espressif-documentatie [11]) geïnstalleerd worden. De instellingen voor het board zijn aangegeven in **figuur 5**. Naast de Arduino ESP32-ondersteuning hebben we ook een paar bibliotheken nodig voor onze eerste stappen. Om in ons voorbeeld de ESP32-C3 in staat te stellen data via MQTT/JSON te versturen, moeten we *PubSubClient* van Nick O'Leary en de *ArduinoJson*-bibliotheken van Benoit Blanchon installeren. Dat kunnen we doen met behulp van de Arduino IDE Library Manager (**figuren 6 en 7**).

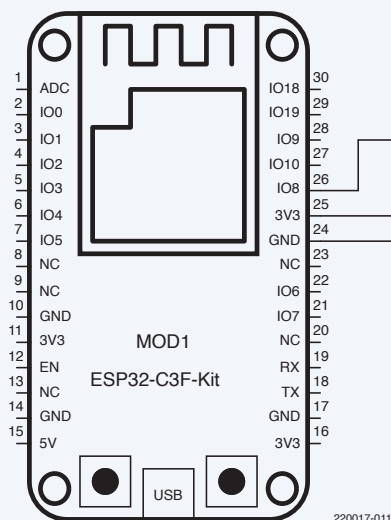
## Bouw van de hardware

De twee op de ESP32-C3 gebaseerde modules worden aangesloten conform de schema's in **figuur 8** en **figuur 9**. Voor de joystick- en relaismodules zijn slechts drie draden nodig die op de betreffende

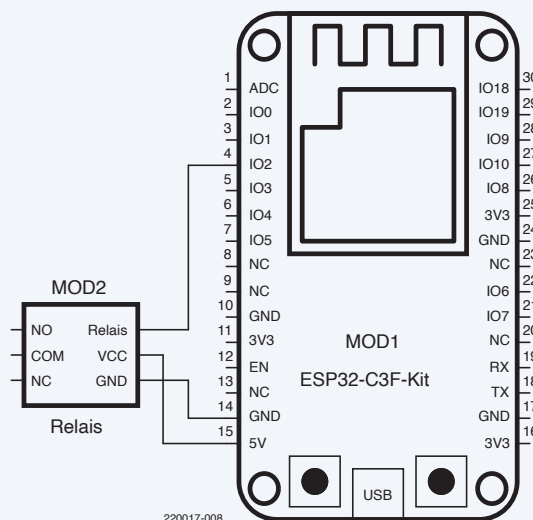


Figuur 6. PubSubClient in de Arduino-bibliotheekmanager.

Figuur 7. ArduinoJson in de Arduino bibliotheekmanager.



Figuur 8. Schema van de ESP32-C3 plus joystick.



Figuur 9. Schema van de ESP32-C3 plus relais.

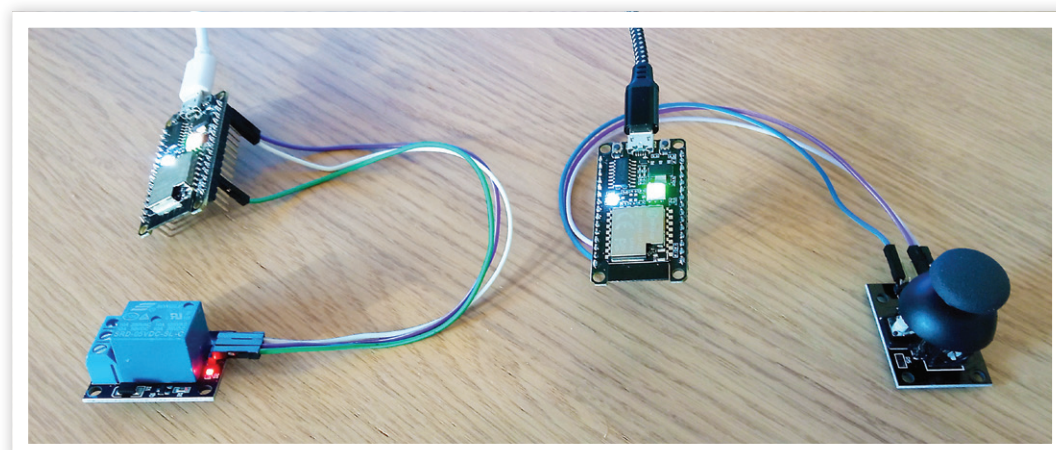
ESP32-C3 print worden aangesloten. Merk op dat de relaisprint wordt gevoed via de 5V-voedingspen. In **figuur 10** zijn alle modules en controllers met elkaar verbonden.

### Software-setup

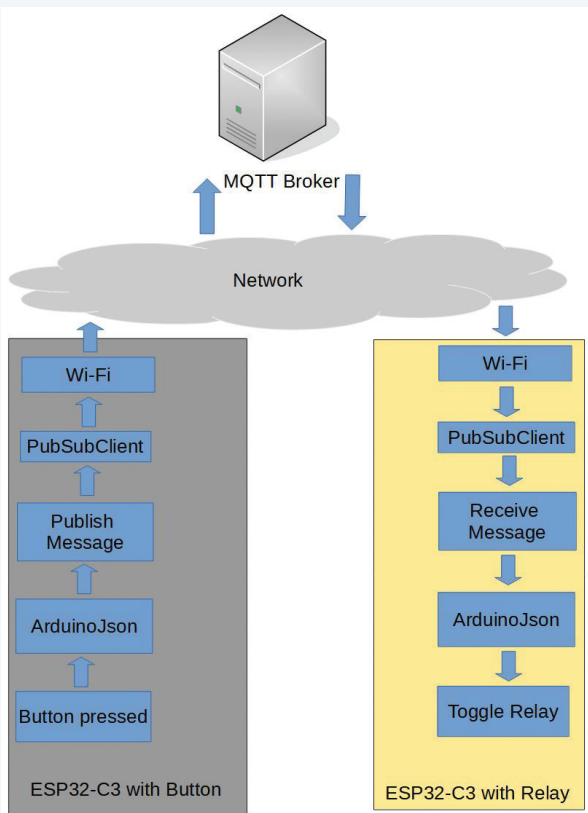
De broncode voor dit project is beschikbaar op GitHub [12]. De sketches voor de beide ESP32-C3 controllers kunnen van daar gedownload worden. U moet nog enige informatie over uw lokale netwerk in deze bestanden invoeren voor u ze kunt uploaden naar de ESP32's. Alles moet juist ingesteld worden zodat de twee controllers gegevens kunnen uitwisselen via de lokale MQTT-broker. Daartoe staan aan het begin van de twee Arduino-sketches enkele `#define`-statements:

```
#define WIFI_SSID "changeme"
#define WIFI_PASS "changeme"
#define MQTT_SERVER "test.mosquitto.org"
```

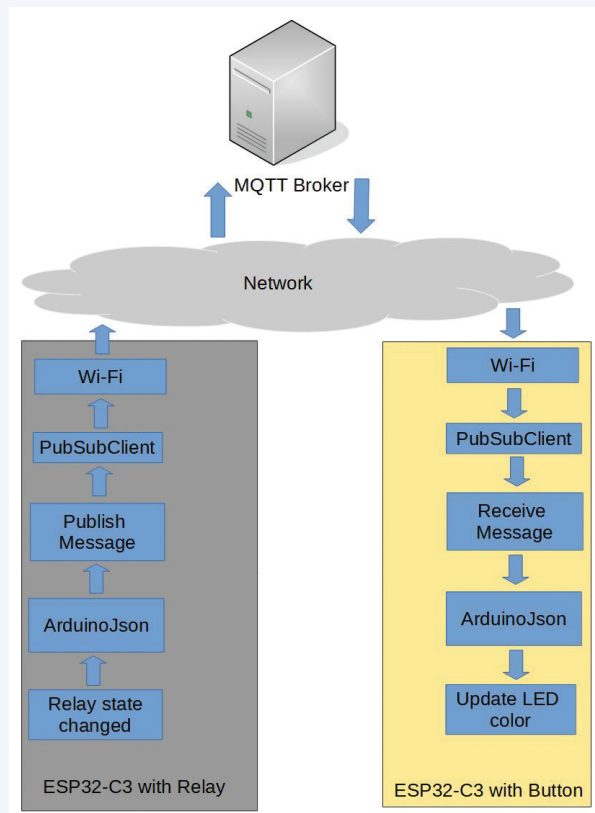
Deze drie `#defines` aan het begin van de Arduino sketch moeten worden aangepast aan uw eigen netwerk. De SSID en PASSWORD van uw netwerk moeten tussen de aanhalingstekens worden ingevuld. Het IP-adres van de Node-RED computer in het eigen netwerk wordt opgegeven voor de MQTT-server. Nadat beide sketches (voor het relais en voor de knop) zijn bewerkt kunnen ze worden in de betreffende ESP32-C3 worden geladen. Beide ESP32's kunnen dan worden ingeschakeld; de grote LED op elk board moet wit gaan knipperen. Dit geeft aan dat de ESP32-C3 verbinding probeert te maken met het WiFi-netwerk. De LED moet continu oplichten wanneer een print verbinding heeft gemaakt. De kleur van de LED hangt af van de functie van het board: het board waarop het relais is aangesloten zal wit oplichten. De print waarop de drukknop is aangesloten, zal rood (relais uit) of groen (relais aan) oplichten – afhankelijk van de status van het relais. We weten nu dat beide ESP32-C3's correct werken. Door op de knop te drukken verandert de status van het relais en verandert de kleur



Figuur 10. Alle hardware is aangesloten.



Figuur 11. Gegevensoverdracht naar de broker en terug.



Figuur 12. Terugkoppeling van het relais.

van de LED van rood naar groen of omgekeerd. De knop van de ene ESP32-C3 kan dus met succes het relais van de andere ESP32-C3 aansturen, en hij krijgt ook feedback over de status van het relais. Tijd om te vieren. Uw eerste IoT-toepassing draait! Maar hoe werkt de gegevensuitwisseling precies?

### Naar het relais en weer terug

Laten we eerst eens kijken naar de weg die een druk op een knop naar het relais aflegt. **Figuur 11** laat zien hoe het bericht laag voor laag wordt ingepakt en dan via WiFi naar de broker wordt gestuurd. In de broncode wordt dit gedaan met `client.publish(MQTT_TOPIC_OUT, (const uint8_t*)buffer, n, true);`.

Waarom wordt deze functie `publish` genoemd en niet gewoon `send`? Dit heeft te maken met de manier waarop de gegevens later in MQTT worden gedistribueerd. Op een MQTT-broker worden berichten gedistribueerd op basis van een onderwerp; in dit geval, het onderwerp (`MQTT_TOPIC_OUT`) "BUTTON". Bij de verbinding met de MQTT-broker kan de client (dus de ESP32-C3 met het relais) aangeven welk onderwerp van belang is (om zich dus op dit nieuwskanaal te 'abonneren'). Elke deelnemer die aan de MQTT-broker heeft meegedeeld dat hij geïnteresseerd is in een bepaald onderwerp, ontvangt de berichten die onder dit onderwerp zijn verzonden. De afzender hoeft zich daarentegen

geen zorgen te maken over de distributie. Hij verzendt ('publiceert') alleen zijn berichten naar de broker.

De ESP32-C3 met het relais abonneert zich op het BUTTON-onderwerp in zijn code met `client.subscribe(MQTT_TOPIC_IN);`, waarbij `MQTT_TOPIC_IN` hier "BUTTON" is. Telkens wanneer een bericht wordt getriggerd door de knop, gaat het naar de MQTT-broker. Het wordt dan afgeleverd bij de ESP32-C3 met het relais, waardoor het relais wordt omgeschakeld.

Wanneer het relais van toestand verandert, stuurt zijn controller een bericht naar de MQTT-broker onder het topic "RELAIS"; waarin de huidige toestand (aan of uit) is opgenomen. De ESP32-C3 met knop heeft zich op zijn beurt geabonneerd op het "RELAIS"-topic bij de MQTT-broker en ontvangt daardoor het bericht met de nieuwe status, zoals te zien is in **figuur 12**. Hierdoor verandert de kleur van de LED in rood of groen.

Het mooie van deze structuur is dat ook een tweede knop en controller in het netwerk kunnen worden geïntegreerd om – net als de eerste knop – berichten onder het thema "BUTTON" te verzenden. De ESP32-C3 met het relais zal dan reageren op de stuurgegevens van een van beide knoppen en de juiste schakelhandeling uitvoeren. Als u geïnteresseerd bent in verdere experimenten met MQTT, kijk dan eens naar andere projecten van Elektor die MQTT gebruiken om gegevens te

versturen. Voorbeelden zijn het weerstation [13] en de monster-LED-klok met een externe temperatuursensor [14]. Voor meer info over MQTT en hoe u het kunt gebruiken om gegevens naar bijvoorbeeld cloud-platforms te sturen, kunt u de artikelreeks "Mijn pad naar het IoT" [15] lezen.

## Een ideaal platform voor het IoT

IoT hoeft niet ingewikkeld te zijn. Zoals we hier hebben laten zien, is de technologie behoorlijk volwassen, en kunt u IoT-apparaten snel ontwikkelen met eenvoudige tools en de juiste componenten. Toepassingen kunnen natuurlijk complexer zijn dan een simpele drukknop plus relais. Van het regelen van de verwarming in huis tot de voordeurbel – er zijn tal van praktische toepassingen die baat kunnen hebben bij IoT-connectiviteit. De ESP32-C3 is een krachtig en goedkoop board dat een ideaal platform is om uw eigen ideeën te ontwikkelen! ◀

220017-03

### Een bijdrage van

Idee en tekst: Mathias Claußen

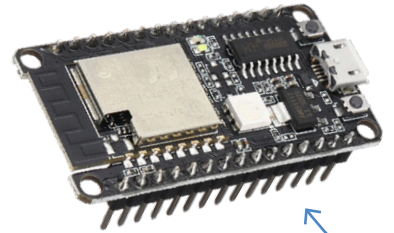
Redactie: Jens Nickel

Vertaling: Eric Bogers

Layout: Giel Dols

### Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via [mathias.claussen@elektor.com](mailto:mathias.claussen@elektor.com) of naar de redactie van Elektor via [redactie@elektor.com](mailto:redactie@elektor.com).



## GERELATEERDE PRODUCTEN

- ▶ **ESP-C3-12F-Kit Development Board with 4 MB Flash (SKU 19855)**  
[www.elektor.nl/19855](http://www.elektor.nl/19855)
- ▶ **Elektor 37-in-1 Sensor Kit (SKU 16843)**  
[www.elektor.nl/16843](http://www.elektor.nl/16843)
- ▶ **Raspberry Pi Zero 2 W Bundle (SKU 19952)**  
[www.elektor.nl/19952](http://www.elektor.nl/19952)
- ▶ **Pimoroni Maker Essentials - Mini Breadboards & Jumper Jerky (SKU 18430)**  
[www.elektor.nl/18430](http://www.elektor.nl/18430)



## WEBLINKS

- [1] M. Claußen, "Aan de slag met de ESP32-C3 RISC-V MCU", Elektor januari/februari 2022: [www.elektormagazine.nl/210466-03](http://www.elektormagazine.nl/210466-03)
- [2] Documentatie Elektor 37-in-1 Sensor Kit: [www.elektor.com/amfile/file/download/file/1170/product/6171/](http://www.elektor.com/amfile/file/download/file/1170/product/6171/)
- [3] Ubuntu Linux-distributie: <https://ubuntu.com/>
- [4] OASIS MQTT TC Minutes van 25.04.2013:  
[www.oasis-open.org/committees/download.php/49028/OASIS\\_MQTT\\_TC\\_minutes\\_25042013.pdf](http://www.oasis-open.org/committees/download.php/49028/OASIS_MQTT_TC_minutes_25042013.pdf)
- [5] JSON.org: [www.json.org/json-de.html](http://www.json.org/json-de.html)
- [6] Mozilla Web Docs: Working with JSON: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- [7] Node-RED: <https://nodered.org/>
- [8] Node-RED installatie op de Raspberry Pi: <https://nodered.org/docs/getting-started/raspberrypi>
- [9] Node-RED installatie op een PC: <https://nodered.org/docs/getting-started/local>
- [10] Arduino IDE-download: [www.arduino.cc/en/software](http://www.arduino.cc/en/software)
- [11] Espressif Arduino-ESP32 installatie-instructies: <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>
- [12] Elektor GitHub-repository: [https://github.com/ElektorLabs/220017-ESP32-C3-and-IoT\\_First-steps](https://github.com/ElektorLabs/220017-ESP32-C3-and-IoT_First-steps)
- [13] R. Aarts, "ESP32-Weerstation", Elektor januari/februari 2019: [www.elektormagazine.nl/magazine/elektor-72/42284](http://www.elektormagazine.nl/magazine/elektor-72/42284)
- [14] M. Claußen, "Gigantische LED-klok met WLAN en temperatuurweergave", Elektor mei/juni 2019:  
[www.elektormagazine.nl/magazine/elektor-98/42624](http://www.elektormagazine.nl/magazine/elektor-98/42624)
- [15] J. Nickel, "Mijn pad naar het IoT", Elektormagazine.nl: [www.elektormagazine.nl/news/mijn-pad-naar-het-iot-1-de-cloud](http://www.elektormagazine.nl/news/mijn-pad-naar-het-iot-1-de-cloud)