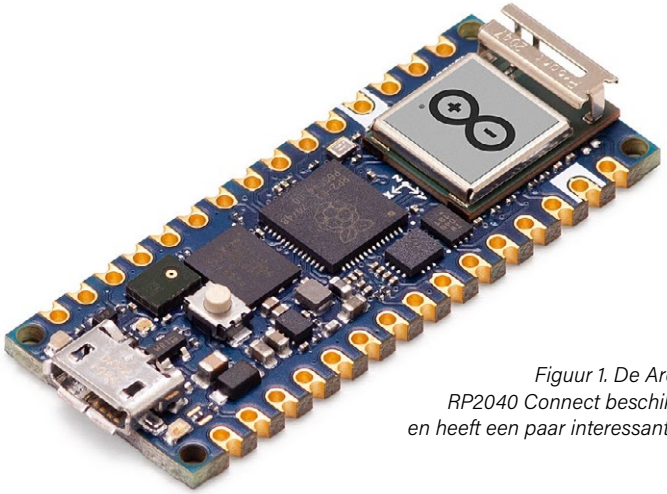
Met Home Automation de wereld redden?

MQTT op de Arduino Nano RP2040 Connect

Clemens Valens (Elektor Lab)

Met de juiste componenten en een beetje vindingrijkheid kun je je huis automatiseren, en dit op de Arduino Nano RP2040 Connect-gebaseerde project is daarvoor een uitstekend begin. Een extra bonus is dat je misschien helpt bij het redden van onze planeet.



Figuur 1. De Arduino Nano RP2040 Connect beschikt over WiFi en heeft een paar interessante sensoren.

Tegenwoordig lijkt voor veel mensen het milieu de grootste zorg te zijn. Onze planeet is groot, en er zijn veel vervuilers waar je geen controle over hebt. Maar er is één plek waar je een verschil kunt maken, en dat is thuis. Je huis automatiseren kan het energiezuiniger maken en zo een beetje helpen om de planeet te redden. Hier is een manier om daarmee te beginnen.

Home Automation (ook wel domotica) wordt over het algemeen toegepast op de volgende gebieden:

- klimaatregeling
- verlichting
- energiebeheer
- toegangscontrole
- watermanagement

De eerste drie hebben betrekking op energiebesparing, door de temperatuur en vochtigheid in het gebouw te regelen met kachels, koelers, ventilatoren en zonwering, en door verlichting, apparaten en machines uit te schakelen wanneer deze niet nodig zijn. Watermanagement is ook nuttig, tenzij het gaat om het besproeien van het gazon, wat altijd verspilling is.

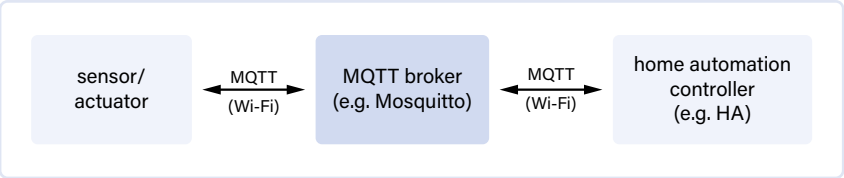
We hebben sensoren nodig

Het regelen van zaken zoals de temperatuur in een kamer of het stroomverbruik van een machine, vereist sensoren en actuatoren. In dit artikel presenteer ik een soort universeel apparaat dat sensoren uitleest en de meetgegevens via MQTT doorstuurt naar een domotica-controller. De actuatoren die dingen als motoren, pompen, relais en lampen aan en uit kunnen zetten worden hier niet besproken. Voor wie niet vertrouwd is met domotica: een controller is het hart van een domoticasysteem, de spin in het web tussen alle apparaten in het systeem, inclusief jezelf en andere gebruikers (als je aanvaardt dat we je hier als een levend apparaat beschouwen). MQTT is een protocol voor gegevensuitwisseling dat tamelijk populair is voor IoT en automatisering.

De Arduino Nano RP2040 Connect

Het hier beschreven sensorapparaat is gebaseerd op een Arduino Nano RP2040 Connect-module (**figuur 1**) met een 3-assige gyroscoop, een 3-assige versnellingsmeter en een microfoon. Dat lijken misschien vreemde sensoren om te gebruiken in een domoticasysteem, maar ze kunnen erg nuttig zijn. (En het is weer eens wat anders dan de gebruikelijke temperatuur- vochtigheidsbenadering van een typisch IoT-sensorproject). Wanneer de gyro bijvoorbeeld op een deur of raam is gemonteerd, kan die zowel verwachte als onverwachte in- en uitgaande bewegingen detecteren, of een alarm geven bij een openstaand raam. Accelerometers hebben vele toepassingen in trillingsdetectie (draait de motor van de vriezer wel continu?) en een microfoon kan geluiden detecteren die er niet horen te zijn (lopende kraan) of een verandering in achtergrondgeluid of frequentie opmerken (draait de motor wel goed?). Maar het maakt niet uit of je dergelijke sensoren nodig hebt, het software-framework is gemakkelijk aan te passen aan andere sensoren.

De Arduino Nano RP2040 Connect heeft WiFi-connectiviteit dankzij de draadloze NINA-module (een vermomde ESP32) en die gaan we gebruiken om verbinding te maken met het WiFi-netwerk van het huis. De huisautomatiseringscontroller is Home Assistant (HA), die – in mijn geval – draait op een Raspberry Pi 3B+ board [1], maar elke andere controller die met het MQTT-protocol overweg kan (ongeveer 99%), is ook bruikbaar. De controller maakt ook verbinding



Figuur 2. Een blokschema van het op MQTT gebaseerde domoticasysteem dat in dit artikel wordt gepresenteerd.

met het WiFi-netwerk. **Figuur 2** geeft een blokschema van het systeem.

Voorbereidingen & vereisten

Het redden van de planeet is een dringende zaak, daarom zullen we ons hier niet verliezen in allerlei technische details over het Arduino-board, maar komen we meteen ter zake. Als je nog geen domoticacontroller hebt, begin dan met de installatie en configuratie daarvan. Dat is gemakkelijker gezegd dan gedaan, zie voor meer details bijvoorbeeld [1]. Vergeet niet dat de controller MQTT-compatibel moet zijn, waarvoor wellicht een add-on nodig is. Als je, zoals ik, HA als controller gebruikt, kun je de veelgebruikte Mosquitto (met twee 't's) 'integratie' installeren [2]. Dit is een zogenaamde MQTT-broker, een apparaat dat MQTT-berichten ontvangt en verzendt, bijvoorbeeld tussen ons Arduino-board en HA.

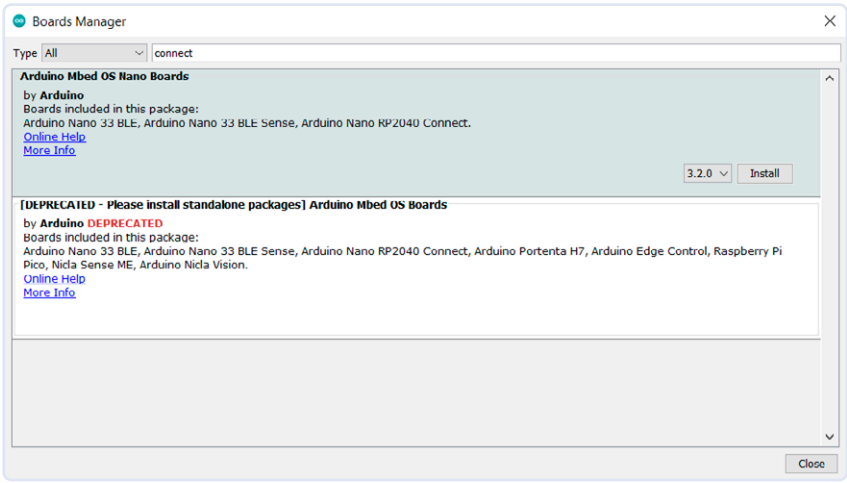
De Arduino IDE voorbereiden

Als je al een werkende MQTT-compatibele domoticacontroller hebt, kun je overgaan tot het opzetten van de Arduino-ontwikkelomgeving:

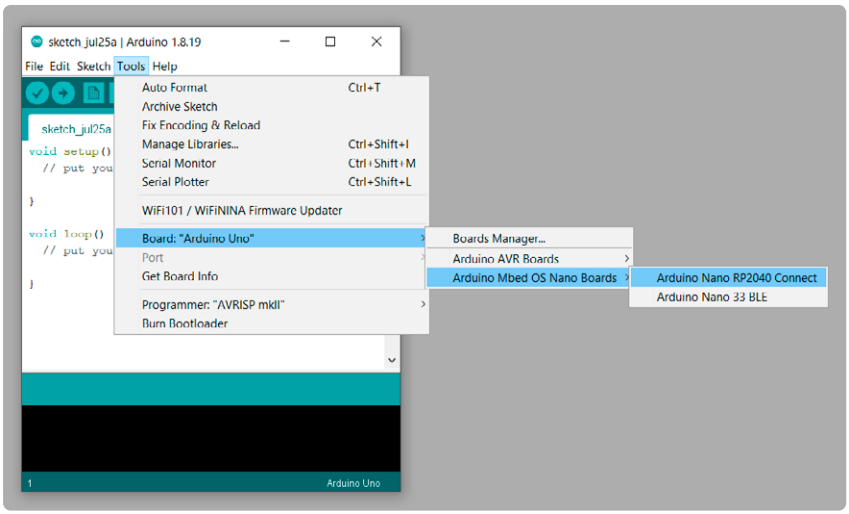
1. Installeer de Arduino IDE. Er zijn veel versies; ik heb 1.8.19 gebruikt.
2. Met behulp van de Boards Manager van de Arduino IDE (*Tools → Board → Boards Manager…*) installeer je het *'Arduino Mbed OS Nano Boards'* board package (ik heb v3.2 gebruikt; zie **figuur 3**).
3. Selecteer in de IDE het Arduino Nano RP2040 Connect board – zie **figuur 4** (*Tools → Board → Arduino Mbed OS Nano Boards → Arduino Nano RP2040 Connect*).
4. Gebruik de bibliotheekmanager van de IDE (*Tools → Manage Libraries…*) om de volgende bibliotheken te installeren (de versies die ik heb gebruikt staan tussen haakjes):

- *WiFiNINA* (v1.8.13)
- *ArduinoMqttClient* (v0.1.6)
- *Arduino_LSM6DSOX* (v1.1.0)

5. Download mijn sketch van [3] en pak hem uit in de map sketchbook van de IDE. Merk op dat de sketch bestaat uit twee bestanden, waarvan er één `arduino_secrets.h` heet. Voer in dit bestand je netwerkgegevens in.



Figuur 3. Gebruik de Boards Manager van de IDE om ondersteuning voor de Arduino Nano RP2040 Connect te installeren.



Figuur 4. Voordat je iets probeert te uploaden naar het board, moet je het juiste board (en de bijbehorende seriële poort!) selecteren.

Programmaconfiguratie

Voordat je mijn sketch kunt compileren, moet je wat informatie verzamelen van je MQTT-broker. In HA vereist de Mosquitto-broker dat je een gebruiker en een wachtwoord definieert; zonder dat komt er geen MQTT-verkeer door. Voer de MQTT-gegevens in het bestand `arduino_secrets.h` in als `SECRET_MQTT_USER` en `SECRET_MQTT_ PASSWORD`. Voer in dit bestand ook het SSID en wachtwoord van je WiFi-netwerk in.

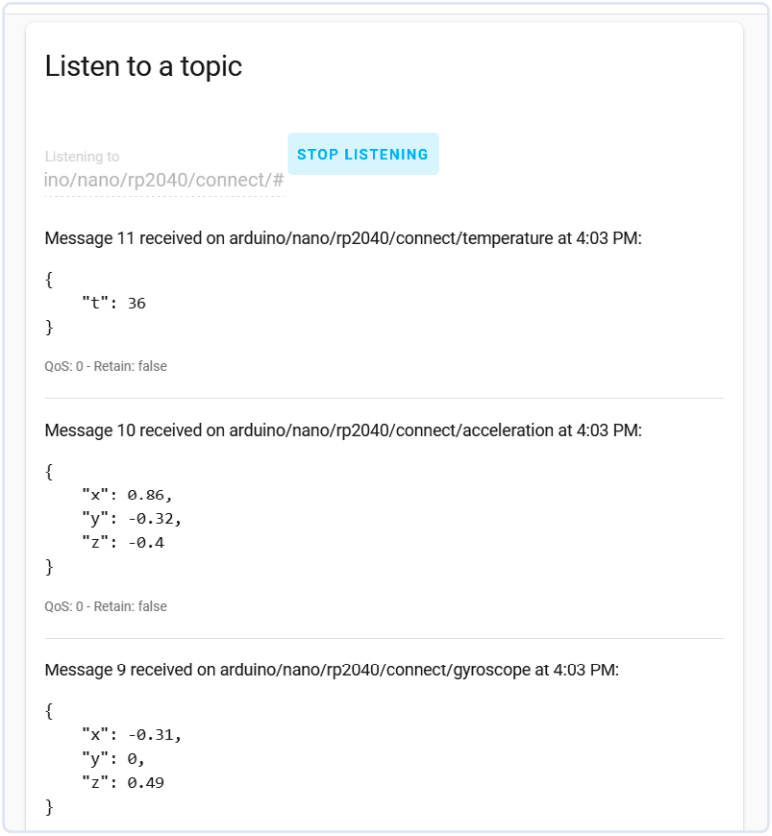
Voer in het hoofdbestand van de sketch, op regel 37 op het moment van schrijven, het IP-adres van de MQTT-broker in. In HA is dit gewoon het IP-adres van HA, te vinden bij Settings → System → Network (daar stond het in HA Core v2022.8.6 met HA OS v8.4):

```
const char broker[] = 'xxx.xxx.xxx.
   xxx';
```

Als je om een of andere reden de standaard MQTT-poort hebt gewijzigd, moet je de volgende regel ook wijzigen:

```
int port = 1883;
```

Als de sketch correct is geconfigureerd, kun



Figuur 5. MQTT-berichten ontvangen met Mosquitto in Home Assistant.

je hem compileren en de executable uploaden naar het Arduino Nano RP2040 Connect-board. Er zouden geen waarschuwingen of fouten mogen verschijnen, zelfs niet met de compiler ingesteld op 'verbose output' en alle waarschuwingen geactiveerd (Arduino IDE *File → Preferences*).

De proef op de som

Ervan uitgaande dat alles is goed gegaan, en je geen fouten hebt gemaakt bij het invoeren van de wachtwoorden enzovoort, zou het board nu verbinding moeten maken met het WiFi-netwerk en MQTT-berichten gaan versturen met een snelheid van 0,1 Hz (dus elke 10 seconden). De RGB-LED van het board knippert telkens wanneer een bericht wordt verzonden. Als de LED niet knippert of blijft branden, is er iets mis. Rood betekent geen netwerkverbinding en blauw een probleem met de sensor.

De domotica-controller zou de MQTT-berichten moeten ontvangen. Klik in HA met de Mosquitto add-on op 'Configure' op de Mosquitto broker-integratiekaart, en scroll naar beneden naar 'Listen to a topic.' Voer `arduino/nano/rp2040/connect/#` in en klik op 'Start listening.' Nu moeten berichten verschijnen met een snelheid van één per tien seconden (**figuur 5**).

Mijn voorbeeldsketch stuurt elke tien seconden gyro- en acceleratiegegevens, en ook een temperatuurwaarde (Ha! Dat had je niet verwacht, hè? De LSM6DSOX IMU-chip heeft een ingebouwde temperatuursensor, vandaar). De temperatuurwaarde zal meestal hoger zijn dan de omgevingstemperatuur, omdat de sensor wordt verwarmd door de draadloze module van het board. Als de microfoon een hard geluid hoort, stuurt hij een bericht. Klap in je handen om het uit te proberen. (Vergeet niet dat er tot tien seconden vertraging kan optreden).

Als je in 'Publish a packet' `arduino/nano/rp2040/connect/incoming/xxx` met in plaats van `xxx` (het 'topic') een woord of getal, en dan op Publish klikt, zou je het moeten zien verschijnen in de Serial Monitor van de Arduino IDE. Je kunt optioneel een payload toevoegen (kan van alles zijn).

Hoe nu verder?

We hebben nu het punt bereikt vanwaar je alleen verder moet. Er zijn verschillende opties:

a. Gegevens- en signaalverwerkingsroutines toevoegen aan de sensorsoftware zodat deze alleen berichten verstuurt als bepaalde gebeurtenissen worden gedetecteerd.
b. Voeg meer of andere sensoren toe.
c. Voeg automatiseringsregels toe aan de home automation controller om deze op een nuttige manier te laten reageren op ontvangen berichten.
d. Al het bovenstaande.
e. Doe niet aan domotica (zie **kader**).

Om je op weg te helpen met de opties 'a' en 'b' zal ik nu kort de werking van de sketch uitleggen. Voor optie 'c' verwijs ik naar de documentatie van de domotica-controller.

Het programma

Het programma begint met het definiëren van een aantal zaken zoals netwerkdetails, maar ook de MQTT-topics die zullen worden gebruikt. Alle berichten die worden verzonden door ons apparaat, beginnen met

`arduino/nano/rp2040/connect/`

Het apparaat luistert dus alleen naar berichten die beginnen met deze prefix. Het is een lange prefix, maar zeer geschikt voor debugging. De prefix wordt gevolgd door een zogenaamd topic dat van alles kan zijn. Het enige wat hier belangrijk is, is dat de MQTT-broker of bestemming naar hetzelfde topic moet luisteren, anders worden de berichten gewoon genegeerd. Het is een goede praktijk om zinvolle topics te gebruiken. Er kunnen zoveel onderwerpen zijn als je wilt, en ze kunnen een (bijna) onbeperkte lengte hebben.

Geluidsverwerking

De sensoren worden gelezen met helperfuncties die betekenisvolle namen hebben, behalve de microfoon, die PDM heet (het is een digitale microfoon). Ook anders dan de andere sensoren is dat de microfoon in een soort thread op de achtergrond draait die continu audiosamples in een buffer pompt. De andere sensoren worden om de tien seconden door de hoofdlus afgevraagd. Een laatste verschil tussen audiogegevens en de rest is dat de audiomonsters worden gefilterd door een laagdoorlaatfilter met een kantelfrequentie van 10 Hz, om alleen te reageren op signalen met een lage frequentie (dus geluiden van dingen die kapot gaan). Na detectie van een geschikt geluid wordt een bericht verzonden met het topic `microphone/alarm` (plus prefix natuurlijk). De gegevens van de andere sensoren worden zonder filtering doorgestuurd.

MQTT-topics

De paar helperfuncties om uitgaande MQTT-berichten samen te stellen spreken voor zich. Goed om te weten is dat de

Tel uit je winst

Natuurlijk kan technologie je helpen huis en op kantoor energie te besparen, maar er hangt een prijskaartje aan dat zwaarder kan wegen dan de potentiële winst. Het in dit artikel beschreven project gebruikt een klein Arduino RP2040 Nano Connect-board dat via WiFi communiceert met een Raspberry Pi met Home Assistant. Al deze apparaten, inclusief de WiFi-router, verbruiken voortdurend energie omdat ze altijd aan staan. Je kunt de kosten schatten door naar je energierekening te kijken. Deze kosten zijn zichtbaar, en je betaalt ze zelf. Maar er zijn ook verborgen kosten die mensen graag vergeten: de grondstoffen die verbruikt worden om alles wat je gebruikt te produceren.

Ik heb geen idee wat de ecologische footprint is van de productie van een Arduino of Raspberry Pi board, maar die is duidelijk niet nul (vergeet niet de koolstof-footprint van de onderdelen mee te rekenen). Ook de software die door ons systeem wordt gebruikt, inclusief onze eigen kleine sketch, is ontwikkeld op vele computers wereldwijd en opgeslagen op servers in de cloud. Net als ons eigen kleine systeem, verbruikt dit enorme ecosysteem (veel) energie om online te blijven. En vergeet de middelen niet die werden gebruikt om het allemaal in elkaar te zetten en te fabriceren. Zeker, deze kosten worden gedeeld door vele gebruikers, maar er moet rekening mee worden gehouden.

Waar heb je de hardware voor je systeem gekocht? Grote kans dat tenminste een deel ervan online is besteld. Het maakt niet uit waar je je spullen kocht, het werd vervoerd van ergens op de planeet naar jou – opnieuw kostbare hulpbronnen.

Tenslotte blijft de technologie vooruitgaan, en dus zal ons domotica-automatiseringssysteem binnenkort verouderd zijn en moeten worden geactualiseerd of weggegooid, wat weer veel verborgen kosten met zich meebrengt. Misschien bespaar je dus een beetje energie in uw huis, maar heb je waarschijnlijk veel meer uitgegeven dan je ooit terug zult verdienen. Daarom is het veel beter voor de planeet om niet te proberen energie te besparen met slimme domotica. Wennen aan het aan- en uitdoen van de lichten is goedkoper en laat je een beetje rondlopen, waardoor je in conditie blijft. Ook gezond verstand is een krachtige planeetspaarder.

`MqttClient print`-interface alleen van toepassing is op het payload-gedeelte van een bericht – het topic moet op een andere manier worden samengesteld.
Alle inkomende MQTT-berichten die beginnen met de prefix

```
arduino/nano/rp2040/connect/incoming/
```

worden geaccepteerd. Dit wordt afgehandeld door de *MqttClient*-library die de functie `mqtt_message_receive` aanroept wanneer aan alle ontvangstvoorwaarden is voldaan. Standaard abonneert de sketch zich op het 'joker'-topic, '`#`', wat betekent dat elk topic geldig is. Je kunt dit veranderen door het te vervangen door meer specifieke topics.
Je kunt je natuurlijk op meer dan één topic tegelijk abonneren. Abonnementen worden beheerd door de MQTT-broker, niet door de sketch, dus het is de broker die beperkingen kan opleggen. De hoeveelheid beschikbaar geheugen voor de sketch heeft geen echte invloed.

Een beetje JSON

De sensorgegevens worden als payload toegevoegd aan de topics. Ik heb hiervoor JSON-achtige opmaak gebruikt, maar handmatig, zonder hulp van een speciale JSON-bibliotheek. Het voordeel van JSON is dat het door veel andere programma's wordt begrepen, wat het parsen ervan veel gemakkelijker maakt.

Conclusie

Goed, dat is het voor nu. Ook al heb ik mijn best gedaan om het simpel en duidelijk te houden, de kans is groot dat je op een gegeven moment tegen problemen aanloopt. Het onderwerp is ingewikkelder dan uit dit artikel blijkt. Aarzel niet om het internet te raadplegen voor meer informatie – er zijn talloze sites over MQTT en Home Assistant (en zelfs over beide), die vol staan met mensen die je om raad kunt vragen, hints, trucs en tips. En als je problemen krijgt en dingen niet meer werken, keer dan terug naar de laatst bekende werkende configuratie.

220420-03 (vertaling: Jelle Aarnoudse)

Over de auteur

Clemens Valens is de creatieve technoloog van Elektor. Hij heeft een BSc in elektronica en een MSc in elektronica en informatietechnologie. Clemens begon in 2008 bij Elektor als hoofdredacteur van Elektor Frankrijk. Momenteel produceert hij technische tutorials en productrecensies bij Elektor TV. Clemens is ook verantwoordelijk voor de community-website van Elektor Labs, waar elektronica-enthousiastelingen hun werk kunnen publiceren en kunnen communiceren met gelijkgezinden van over de hele wereld.

WEBLINKS

[1] C. Valens, "Domotica helemaal niet moeilijk", Elektor september/oktober 2020: https://www.elektormagazine.nl/magazine/elektor-157/59027
[2] MQTT in Home Assistant: https://www.home-assistant.io/docs/mqtt/broker/
[3] Downloads bij dit artikel: https://www.elektormagazine.nl/220420-03

Gerelateerde Producten

- **Arduino Nano RP2040 Connect with Headers**
  www.elektormagazine.nl/arduino-nano-rp2040-connect
- **Clemens Valens, *Mastering Microcontrollers Helped by Arduino* (SKU 17967)**
  www.elektor.nl/17967
- **Elektor Ultimate Sensor Kit (SKU 19104)**
  www.elektor.nl/19104