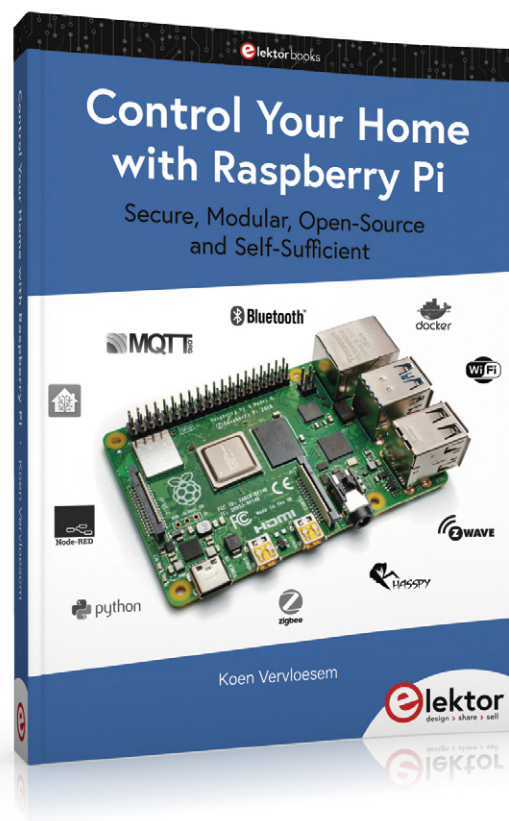


Domotica met de Raspberry Pi

RPi luistert op 433,92 MHz

Koen Vervloesem (België)

Deze aflevering van Elektor Boeken presenteert een hoofdstuk uit het boek *Control Your Home with Raspberry Pi* van Koen Vervloesem, dat onlangs door Elektor is uitgegeven. Koen legt uit hoe men van een RPi een krachtige ontvanger en decoder kan maken voor de meeste, zo niet alle, afstandsbedienings- en sensorsignalen die op 433,92 MHz, een van de bekendste en meestgebruikte ISM-frequenties, kunnen worden ontvangen.



In de doe-het-zelf gemeenschap (tenminste in Europa waar ik woon) zal '433,92 MHz' waarschijnlijk een belletje doen rinkelen. Veel goedkope draadloze apparaten zoals garagedeurenopeners, weersensoren en deurbellen gebruiken deze frequentie. Bovendien is de hardware om met deze apparaten te communiceren erg goedkoop. Het nadeel is dat de meeste van deze apparaten gebruik maken van eenvoudige, niet-versleutelde radiocommunicatie, en als ze al een of andere vorm van beveiliging gebruiken, is die vrij zwak en/of een eigen algoritme dat niet veel vertrouwen wekt. Maar er zijn zoveel beschikbare apparaten en ze zijn zo goedkoop dat ze niet genegeerd kunnen worden. Om veiligheidsredenen gebruik ik alleen temperatuursensoren op 433,92 MHz: in bijna alle kamers van mijn huis heb ik er wel een, en ook buiten. Voor werkelijk kritieke taken zou ik die 433,92MHz-apparaten echter niet willen vertrouwen.

In dit hoofdstuk laat ik zien hoe u metingen van deze draadloze temperatuursensoren leest en doorgeeft aan uw MQTT-broker voor verdere integratie in uw domoticasysteem.

Opmerking: in dit boek heb ik het over 433,92 MHz, maar afhankelijk van waar u woont, moet u dit vervangen door een andere frequentie. In Amerika is de betreffende frequentie bijvoorbeeld 915 MHz. Zorg ervoor dat u de juiste apparaten voor uw land koopt.

433,92MHz-protocollen

Apparaten die de 433,92MHz-frequentie gebruiken, werken in de vrije industriële, wetenschappelijke en medische (ISM) frequentieband. Maar de frequentie is één ding, het protocol dat ze gebruiken is een ander. Er is geen standaardprotocol voor deze frequentie. Dit is geen Z-Wave of Zigbee. Veel protocollen van deze apparaten zijn echter reverse-engineered en u kunt met ze communiceren zolang u een transceiver heeft voor de frequentieband rond 433,92 MHz en de juiste software om het protocol te decoderen en/of te coderen. Enkele interessante apparaten zijn:

- › temperatuur-, vochtigheids- en weersensoren van Alecto, Cresta, La Crosse en Oregon Scientific;
- › deur-/raamsensoren met Hall-sensor;
- › schakelaars en dimmers van Energenie, KlikAanKlikUit en LightwaveRF;
- › deurbelgongen van Byron en Chacon.

U kunt op AliExpress en Banggood ook veel nog goedkopere apparaten vinden die dezelfde protocollen ondersteunen. En er zijn zelfs kleine printjes zoals de STX882-zender die u kunt aansluiten op

Figuur 1. Voor ongeveer € 5 is een temperatuur- en vochtigheidssensor van DANIU verkrijgbaar die zijn meetwaarden op 433,92 MHz uitzendt en op een duidelijk display toont.



te vinden. Hun bereik is redelijk goed: ik kan sensoren overal in mijn huis uitlezen, ook in de koelkast en diepvriezer, en zelfs op mijn terras buiten.

Hardware-vereisten

Om ervoor te zorgen dat uw Raspberry Pi 433,92 MHz-sensormetingen kan ontvangen, hebt u een ontvanger en een antenne nodig.

Ontvanger

Een populair type ontvanger voor 433,92MHz-projecten is een op de Realtek RTL2832 gebaseerde DVB-dongle. Ja, u leest het goed: DVB zoals in Digital Video Broadcasting. Het blijkt dat de RTL2832-chip in veel van deze dongles veel meer kan dan alleen digitale videosignalen decoderen: met de juiste software kunt u er een echte software defined radio (SDR) mee maken.

Dus als je nog een oude DVB-dongle in de la hebt liggen, is de kans groot dat u daarmee signalen van uw (weer-)sensoren kunt ontvangen. En anders is de RTL-SDR (<https://www.rtl-sdr.com>) een goede keuze (figuur 2). Varianten van deze stick zijn voor 25 euro te koop en voor 40...45 euro in een kit met antenne en andere accessoires [1]. En voor nog minder geld kunt u al iets bruikbaars vinden: ik heb op AliExpress DVB-dongles voor 7 euro gezien die prima zouden moeten voldoen, maar ik heb daar geen ervaring mee.

Antenne

Vervolgens hebt u een goede antenne nodig. Er zijn hele boeken geschreven over antennetheorie, maar ik ga niet nader op dit omvangrijke onderwerp in omdat ik geen antennespecialist ben. Bij de keuze van uw antenne moet u de vereiste lengte kennen. Die hangt af van de golflengte van het signaal. De golflengte is gelijk aan de snelheid (in m/s) gedeeld door de frequentie (in Hz), en wordt gemeten in meters (m). Laten we de berekening doen voor 433,92MHz-communicatie. In lucht is de voortplantingssnelheid vrijwel de snelheid van het licht. Dus voor de golflengte geldt:

$$299.792.458 \text{ m/s} / 433.920.000 \text{ Hz} = 0,69 \text{ m}$$

Dus een hele golflengte is 69 cm, de halve golflengte is 34,5 cm en de kwart golflengte is 17,25 cm. Dit zijn de theoretisch optimale lengtes voor een antenne om 433,92MHz-transmissies te ontvangen. In de praktijk worden de eigenschappen van de antenne door meerdere factoren beïnvloed, waaronder de positionering, een vuistregel zegt nog 5% van deze theoretische lengte af te trekken. Nogmaals, het gaat hier niet over antennetheorie, dus ik niet uitwijken over de verschillende soorten antennes. Bovendien maakt het voor het uitlezen van sensorwaarden in huis niet eens zoveel uit wat voor antenne u gebruikt. U kunt naar hartelust te experimenteren, maar de kans is groot dat alles meteen werkt als u voor een paar euro een '433MHz-antenne' koopt op AliExpress of Banggood. Als u geen risico wilt nemen, gebruik dan de antenne die bij de RTL-SDR wordt meegeleverd. De officiële antennekit (figuur 3) heeft dipool-telecoopantennes die u van 5 cm tot 1 m kunt uitschuiven – optimaal voor de 433,92MHz-frequentie [2].

Sensorwaarden ontvangen met rtl_433

Aan de softwarekant is 'rtl_433' populair om 433,92MHz-signalen te lezen (https://github.com/merbanan/rtl_433); ondanks de naam is het



Figuur 2. De RTL-SDR decodeert veel draadloze signalen, waaronder weersensoren die op 433,92 MHz zenden.



Figuur 3. Met het statief en de dipool- en telescoopantennes uit de RTL-SDR-set hebt u alles wat nodig is om metingen te ontvangen van al uw 433,92MHz-sensoren.

een microcontroller of een Arduino-board om uw eigen draadloze sensorboards te maken.

In dit hoofdstuk concentreer ik me op de eerste soorten apparaten: temperatuur- en vochtigheidssensoren. Deze zijn voor nog geen 10 euro (figuur 1) of nog minder op AliExpress of Banggood

een generieke data-ontvanger, voornamelijk voor de 433,92 MHz, 868 MHz (SRD), 315 MHz, 345 MHz en 915 MHz ISM-banden.

Elke op de Realtek RTL2832 gebaseerde DVB-dongle zou met rtl_433 moeten werken – ook de officiële RTL-SDR-dongle. Ik gebruik die dongle met een dipoolantenne uit de RTL-SDR-antennekit. Sluit de antenne gewoon aan op de RTL-SDR en prik de RTL-SDR in een USB-poort van uw Raspberry Pi.

Waarschuwing: de RTL-SDR produceert tijdens bedrijf heel wat warmte. Let dus op waar u die plaatst.

rtl_433toMQTT installeren

Het rtl_433-programma wordt actief ontwikkeld en onderhouden en omvat meer dan 150 protocoldecoders voor verschillende apparaten die op 433,92 MHz zenden. Bovendien kan het de ontvangen waarden naar een MQTT-broker sturen. Gelukkig heeft iemand juist voor dit doel een Docker-container met rtl_433 gemaakt (https://github.com/bademux/rtl_433toMQTT) [3].

Maak eerst een map voor de container aan:

```
mkdir -p /home/pi/containers/rtl433tomqtt
```

Voeg vervolgens de containerdefinitie toe aan uw *docker-compose*.yml-bestand:

```
version: '3.7'
service:
  mosquitto:
    # mosquitto config
    rtl433tomqtt:
      image: bademux/rtl_433tomqtt:latest
      container_name: rtl433tomqtt
      restart: always
      volumes:
        - ./containers/rtl433tomqtt:/home/user/.config/
          rtl_433:ro
        - /etc/localtime:/etc/localtime:ro
      devices:
        - /dev/bus/usb:/dev/bus/usb
```

Voer in plaats van `#mosquitto config` de configuratie van de Docker-container voor Mosquitto in. De container heeft toegang tot de USB-bus nodig om het RTL-SDR-apparaat te kunnen uitlezen. Merk ook op dat de directory die u hebt aangemaakt als een volume is gemount. U hoeft nog geen configuratiebestand aan te maken.

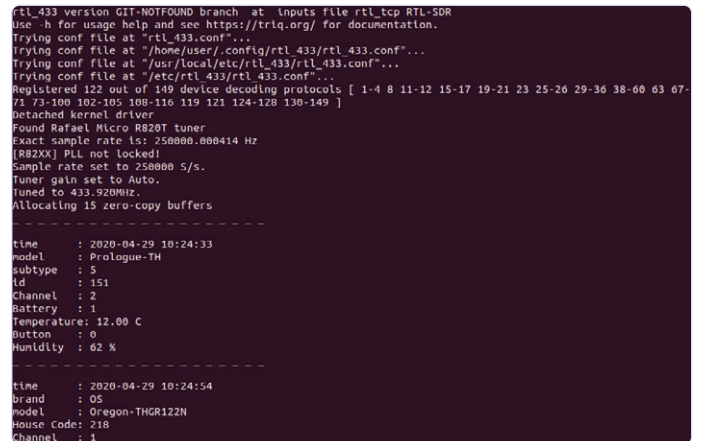
Maak eerst een *udev*-regel om de juiste machtigingen aan het USB-apparaat te geven:

```
sudo nano /etc/udev/rules.d/20.rtl-sdr.rules
```

Voer de volgende regel in:

```
SUBSYSTEM=="usb", ATTRS=="0bda",
  ATTRS=="2838",OWNER="pi", MODE="0660"
```

Sla het bestand op met Ctrl+ en sluit nano af met Ctrl+x. Ontkoppel vervolgens de RTL-SDR en sluit hem dan opnieuw aan. Bekijk nu de lijst met aangesloten USB-apparaten:



```
rtl_433 version GIT-NOTFOUND branch at inputs file rtl_tcp RTL-SDR
Use -h for usage help and see https://trig.org/ for documentation.
Trying conf file at "/home/user/.config/rtl_433/rtl_433.conf"...
Trying conf file at "/usr/local/etc/rtl_433/rtl_433.conf"...
Trying conf file at "/etc/rtl_433/rtl_433.conf"...
Registered 122 out of 149 device decoding protocols [ 1-4 8 11-12 15-17 19-21 23 25-26 29-36 38-60 63 67-
71 73-100 102-105 108-116 119 121 124-128 130-149 ]
Detached kernel driver
Found Rafael Micro R820T tuner
Exact sample rate is: 250000.000414 Hz
[82XX] PLL not locked!
Sample rate set to 250000 S/s.
Tuner gain set to Auto.
Tuned to 433.920MHz.
Allocating 15 zero-copy buffers
-----
time      : 2020-04-29 10:24:33
model     : Prologue-TH
subtype   : 5
id        : 151
channel   : 2
Battery   : 1
Temperature: 12.00 C
Button    : 0
Humidity  : 62 %
-----
time      : 2020-04-29 10:24:54
brand     : Oregon-THGR122N
model     : Oregon-THGR122N
House Code: 218
channel   : 1
```

Figuur 4: De opdracht rtl_433 vindt de RTL-SDR-ontvanger automatisch en geeft de ontvangen sensormetingen weer.

```
lsusb
```

Zoek in deze lijst naar een regel als deze:

```
Bus 001 Device 008: ID 0bda:2838 Realtek
Semiconductor Corp. RTL2838 DVB-T
```

Noteer het bus- en apparaatnummer: 001 en 008. Bekijk nu de apparaatbestanden voor bus 001:

```
ls -l /dev/bus/usb/001
```

U zou een regel als deze moeten zien:

```
crw-rw---- 1 pi root 189, 10 May 7 20:22 008
```

Hier ziet u dat het apparaat eigendom is van de gebruiker pi die lees- en schrijfrechten heeft.

Maak hierna de container aan met:

```
docker-compose up -d
```

Bekijk de logs nadat de container is gemaakt:

```
docker logs -f rtl433tomqtt
```

U zou nu enkele berichten moeten zien (zoals in **figuur 4**) dat het programma rtl_433 op een paar plaatsen een configuratiebestand probeert te vinden, dat het meer dan 120 decoderingsprotocollen heeft geregistreerd en dat het een ontvanger heeft gevonden. Dan zou u de melding 'Tuned to 433.920MHz' moeten zien en als alles goed gaat zie u nu sensorwaarden binnenkomen. Heb een beetje geduld, want veel van deze sensoren zenden maar één keer per minuut.

Configureren van rtl_433

In het begin van de logs zag u dat rtl_433 probeerde enkele configuratiebestanden te vinden. Dat is niet gelukt, dus het programma

```
rtl_433 version Git-NOTFOUND branch at inputs file rtl_tcp rtl_50R
use -h for usage help and see https://triq.org/ for documentation.
Trying conf file at "/home/user/.config/rtl_433/rtl_433.conf"...
Trying conf file at "/home/user/.config/rtl_433/rtl_433.conf"...
Reading conf from "/home/user/.config/rtl_433/rtl_433.conf"...
Registered 120 out of 149 device decoding protocols [ 2-3 8 11-12 15-17 19-21 23 25-26 29-36 38-60 63 67-71 73-100 102-105 108-116 119 121 124-128 130-149 ]
Detached kernel driver
Found Rafael Micro R820T tuner
Exact sample rate is: 250000.000414 Hz
(R820T) PLL not locked!
Sample rate set to 250000 S/s.
Tuner gain set to Auto.
Tuned to 433.920MHz.
Allocating 15 zero-copy buffers
{"time": "2020-04-29 11:01:58.043744", "protocol": 12, "brand": "OS", "model": "Oregon-THGR122N", "id": 218, "channel": 1, "battery_ok": 1, "temperature_C": -21.000, "humidity": 45, "mod": "ASK", "freq": 433.927, "rssi": -0.160, "snr": 5.945, "noise": -6.105}
{"time": "2020-04-29 11:02:36.885857", "protocol": 12, "brand": "OS", "model": "Oregon-THGR122N", "id": 218, "channel": 1, "battery_ok": 1, "temperature_C": -21.000, "humidity": 45, "mod": "ASK", "freq": 433.925, "rssi": -0.082, "snr": 5.968, "noise": -6.049}
{"time": "2020-04-29 11:02:37.045495", "protocol": 12, "brand": "OS", "model": "Oregon-THGR122N", "id": 218, "channel": 1, "battery_ok": 1, "temperature_C": -21.000, "humidity": 45, "mod": "ASK", "freq": 433.926, "rssi": -0.085, "snr": 6.055, "noise": -6.140}
{"time": "2020-04-29 11:03:03.976743", "protocol": 31, "model": "TFA-TwinPlus", "id": 42, "channel": 2, "battery_ok": 1, "temperature_C": -50.000, "humidity": 39, "nrc": "CHECKSUM", "mod": "ASK", "freq": 433.980, "rssi": -0.115, "snr": 5.970, "noise": -6.085}
{"time": "2020-04-29 11:03:15.807792", "protocol": 12, "brand": "OS", "model": "Oregon-THGR122N", "id": 218, "channel": 1, "battery_ok": 1, "temperature_C": -21.000, "humidity": 45, "mod": "ASK", "freq": 433.922, "rssi": -0.088, "snr": 5.887, "noise": -5.895}
{"time": "2020-04-29 11:03:16.047633", "protocol": 12, "brand": "OS", "model": "Oregon-THGR122N", "id": 218, "channel": 1, "battery_ok": 1, "temperature_C": -21.000, "humidity": 45, "mod": "ASK", "freq": 433.924, "rssi": -0.096, "snr": 6.080, "noise": -6.176}
```

Figuur 5. De opdracht `rtl_433` kan de sensorwaarden in veel formaten weergeven, inclusief JSON.

```
rtl_433/cdoedeb2dc2/events {"time": "2020-04-29 11:18:12.855076", "protocol": 12, "brand": "OS", "model": "Oregon-THGR122N", "id": 218, "channel": 1, "battery_ok": 1, "temperature_C": -21, "humidity": 45, "mod": "ASK", "freq": 433.92355, "rssi": -0.151859, "snr": 5.69722, "noise": -5.84988}
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/time 2020-04-29 11:18:12.855076
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/protocol 12
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/id 218
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/channel 1
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/battery ok 1
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/temperature_C -21
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/humidity 45
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/mod ASK
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/freq 433.92355
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/rssi -0.151859
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/snr 5.69722
rtl_433/cdoedeb2dc2/devices/Oregon-THGR122N/1/218/noise -5.84988
```

Figuur 6. De opdracht `rtl_433` kan de ontvangen sensorwaarden naar uw MQTT-broker sturen.

gebruikte een standaardconfiguratie, wat voor een eerste test prima is maar waarbij geen MQTT is gebruikt. Laten we nu een voorbeeld-configuratiebestand kopiëren naar een pad waar `rtl_433` zoekt:

```
docker cp rtl433tomqtt:/usr/local/etc/rtl_433/
rtl_433.example.conf /home/pi/containers/
rtl433tomqtt/rtl_433.conf
```

Omdat ik de directory `containers/rtl433tomqtt` in de homedirectory van uw pi-gebruiker heb toegewezen aan `/home/user/.config/rtl_433` in de container, kunt u dit nu configuratiebestand bewerken en de container opnieuw starten om deze configuratie te gebruiken. U kunt bijvoorbeeld protocollen uitschakelen die u niet nodig hebt, of protocollen inschakelen die standaard zijn uitgeschakeld. Het configuratiebestand heeft heel veel commentaar, wat u zou moeten helpen uit te vinden wat u moet wijzigen. Bovendien kunt u in de online-documentatie van `rtl_433` (https://triq.org/rtl_433/) veel informatie vinden, inclusief stappen om ondersteuning toe te voegen voor een sensor die (nog) niet wordt ondersteund. Start de container opnieuw na het wijzigen van uw configuratiebestand:

```
docker restart rtl433tomqtt
```

Als alles goed gaat, zouden de logbestanden moeten tonen dat `rtl_433` stopt naar een configuratiebestand te zoeken na het tweede bestand dat het vindt:

Reading conf from `"/home/user/.config/rtl_433/rtl_433.conf"`

Met het voorbeeld-configuratiebestand toont `rtl_433` nu de uitvoer van elke sensor als een JSON-dictionary (figuur 5).

Dit is al een grote stap voorwaarts om uw 433,92 MHz-sensoren te integreren in uw domotica-installatie, maar als laatste stap moeten we deze waarden nog naar uw MQTT-broker publiceren.

433,92MHz-sensorwaarden publiceren naar MQTT

Het `rtl_433`-programma ondersteunt het verzenden van de sensorwaarden die het ontvangt naar een MQTT-broker. Open het configuratiebestand (`/home/pi/containers/rtl433tomqtt/rtl_433.conf`) en zoek deze regel:

`output json`

U kunt deze regel hier laten staan omdat u meerdere outputs kunt specificeren, of de regel veranderen in `output kv` als u de voorkeur geeft aan een meer mensvriendelijke standaard-uitvoer. Voeg de volgende regel toe om de MQTT-uitvoer te definiëren:

`output mqtt://mosquitto:1883,user=home,pass=PASSWORD`

Zorg ervoor dat u de juiste gebruikersnaam en wachtwoord invoert voor uw MQTT-broker.

Waarschuwing: het `rtl_433`-programma ondersteunt geen MQTT via TLS. Als u de `rtl_433toMQTT`-container op dezelfde Raspberry Pi draait als uw `mosquitto`-container, dan is het geen probleem dat ze onversleuteld communiceren: ze bevinden zich toch op dezelfde machine. Als u uw RTL-SDR-ontvanger op een andere machine hebt geïnstalleerd dan uw MQTT-broker (bijvoorbeeld omdat u daar een betere ontvangst hebt), dan raad ik u aan om naast de `rtl_433toMQTT`-container een `mosquitto`-container te gebruiken en deze als een brug naar uw hoofd-MQTT-broker via een versleutelde verbinding te configureren. Zie de bijlage aan het einde van dit boek voor details.

Na een herstart van uw container, zou u voor elke sensorwaarde een MQTT-bericht moeten zien gepubliceerd onder het kopje `rtl_433`. U kunt dit bekijken met:

```
mosquitto_sub -t 'rtl_433 / #' -v
```

Telkens wanneer de temperatuur- en vochtigheidssensor in mijn diepvriezer een bericht verstuurt, zie ik zo iets als in **figuur 6**.

Nu kunt u nog een aantal dingen aanpassen aan de MQTT-configuratie. Raadpleeg de opmerkingen in het voorbeeld-configuratiebestand voor de details. Deze configuratie maakt bijvoorbeeld de MQTT-topics iets korter:

`output`

```
mqtt://mosquitto:1883,user=home,pass=PASSWORD,device
s=rtl433/[model]/[channel]/[id]
```

En als u niet geïnteresseerd bent in de low-level metadata over de radioverbinding, zoals modulatie, frequentie, RSSI (received signal

strength indicator), SNR (signaal/ruis-verhouding) en ruis, verwijderd u de regel met `report_meta` level in het configuratiebestand. En dan zijn we klaar: alle metingen van uw 433,92MHz-sensoren worden naar uw MQTT-broker gestuurd en u kunt zich abonneren op hun topics in uw Python-scripts of andere domotica-software.

Samenvatting en verdere experimenten

Draadloze apparaten voor de 433,92MHz-frequentieband zijn erg populair in de doe-het-zelf gemeenschap; ze zijn goedkoop en gemakkelijk verkrijgbaar. Vooral interessant zijn temperatuur- en vochtigheidssensoren die u kunt gebruiken om uw koelkast, diepvriezer of plantenkast te bewaken.

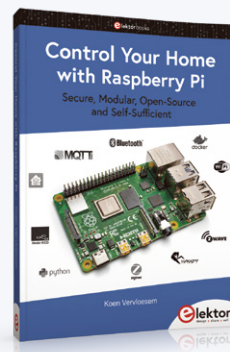
Om deze sensoren uit te lezen hebt u slechts een goedkope DVB-dongle en een antenne nodig. Het `rtl_433`-programma ondersteunt meer dan 150 protocollen van 433,92MHz-apparaten en stelt u in staat de ontvangen signalen als berichten door te sturen naar uw MQTT-broker, zodat uw andere domoticasystemen hierop kunnen reageren.

Er zijn nog veel interessante onderwerpen waar ik nauwelijks aandacht heb besteed, zoals de keuze van de optimale antenne en de beste plaatsing voor een betere ontvangst. Het is ook een aardige oefening ondersteuning voor een niet-ondersteund apparaat toe te voegen aan `rtl_433`. Het project bevat gedetailleerde documentatie over hoe u de 'ruwe' signalen opvangt en hoe u kunt proberen het protocol te ontcijferen.

U kunt ook andere ontvangers proberen, bijvoorbeeld de RFXtrx-familie, die ook een transceiver bevat waarmee u Somfy RTS-roluiken kunt bedienen. Er is zelfs een Python-bibliotheek, `pyRFXtrx` (<https://github.com/Danielhiversen/pyRFXtrx>) om te communiceren met uw 433,92MHz-apparaten met behulp van een RFXtrx-transceiver. Home Assistant (zie hoofdstuk 10) gebruikt deze bibliotheek voor de ondersteuning van 433,92MHz-apparaten. De RFXtrx-transceiver kost echter veel meer dan een RTL-SDR.

Wanneer u aan de andere kant van het prijspectrum wilt proberen wat mogelijk is met de goedkoopste mogelijke apparatuur, vormen de STX882-zender en SRX887-ontvanger een goed uitgangspunt. 

200534-04



CONTROL YOUR HOME WITH RASPBERRY PI

In dit boek leest u hoe u uw huis kunt automatiseren met een Raspberry Pi. U leert hoe u verschillende draadloze protocollen voor domotica kunt gebruiken, zoals Bluetooth, 433,92MHz-radio, Z-Wave en Zigbee. Al snel automatiseert u uw huis met Python, Node-RED

en Home Assistant, en kunt u zelfs met uw domoticasysteem spreken. En dit alles verloopt veilig, met een modulair systeem, volledig open-source, zonder afhankelijk te zijn van services van derden. U hebt zelf de controle over uw huis, en niemand anders.

Tegen het einde van het boek kunt u uw Raspberry Pi installeren en configureren als een zeer flexibele gateway voor huisautomatisering met protocollen naar keuze, en verschillende services koppelen aan MQTT om zo uw eigen systeem te bouwen.

Verander uw Raspberry Pi in een betrouwbare gateway voor verschillende domotica-protocollen. Maak uw domotica-installatie reproduceerbaar met Docker Compose. Beveilig al uw netwerkcommunicatie met TLS. Creëer een videobewakingssysteem voor uw huis. Automatiseer uw huis met Python, Node-RED, Home Assistant en AppDaemon. Krijg veilig toegang tot uw domotica-dashboard van overal ter wereld. Gebruik volledig offline spraakopdrachten in uw eigen taal.

De software en errata voor het boek zijn beschikbaar op GitHub:
<https://github.com/koenvervloesem/raspberry-pi-home-automation>

> Boek:

www.elektor.nl/control-your-home-with-raspberry-pi

> E-boek:

www.elektor.nl/control-your-home-with-raspberry-pi-e-book

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.

Een bijdrage van

Tekst en illustraties:

Koen Vervloesem

Redactie: **Jan Buiting**

Vertaling: **Eric Bogers**

Layout: **Giel Dols**

LINKS & LITERATUUR

- [1] **Elektor RTL-SDR-kit met antennes, statief en verlengkabel:** www.elektor.nl/rtl-sdr-software-defined-radio-with-dipole-antenna-kit
- [2] **Gebruik van een dipoolantenne-kit met de RTL-SDR:** www.rtl-sdr.com/using-our-new-dipole-antenna-kit/
- [3] **Vlak voordat dit boek af was, kondigde de ontwikkelaar aan dat hij dit Docker-image (dat meer dan 100.000 pulls heeft) niet meer actief zou ontwikkelen, dus er zullen geen nieuwe functies komen. Het valt nog te bezien of een ander image populair wordt.**