

Th of

TASMOTA

André Reinink

Domotica programmeren? Thonny! Of toch Tasmota?

Je hebt een bak vol elektronica. Daarin wifi-schakelaars, internet-elektronica en je bent op zoek naar inspiratie voor een artikel in de SoftwareBus...

Je handen beginnen te jeuken, de soldeerbout lonkt naar jou, je wifi-netwerk is op zoek naar vriendjes om mee te babbelen. Wat doe je dan?

Even de draad weer oppakken.

Of beter gezegd: even de soldeerbout weer oppakken. Een aantal jaren geleden schreef ik een artikel met daarin een paar regels over het gebruik van domotica. Domotica? Domotica is het toepassen van elektronica en huisnetwerken ten behoeve van de automatisering van processen in en om een woning, een kantoor of een klein bedrijf. Het woord 'domotica' is een samentrekking van domus en het achtervoegsel -tica dat lijkt te duiden op 'toegepaste wetenschap'. Ik zou het zelf niet beter kunnen verwoorden.

Ik gaf toen aan te zijner tijd een uitgebreider artikel aan domotica te willen wijden. Het is er niet van gekomen. Er was altijd wel weer een ander idee voor een artikel dat bij mij boven kwam borrelen. Bovendien heb ik geen 'mancave' waar ik mijn elektronica kan of mag laten rondslingeren. Een goed excuus toch? Verwacht met mijn bijdrage nu geen high-tech artikel. Die bewaar ik voor later als ik verder ben met de materie. Dit wordt een praktisch artikel met simpele voorbeelden 'ter lering ende vermaeck'.

De trigger

Afgelopen jaar kocht ik een Bluetooth-hoofdtelefoon. Omdat mijn nieuwe tuner alles aan boord heeft behalve een Bluetooth-verbindingsmogelijkheid, kocht ik een losse Bluetooth-ontvanger. Deze losse ontvanger plaatste ik in de buurt van de tuner, de televisie en de eindversterker. Zo kon ik altijd draadloos luisteren zonder dat huisgenoten last hadden van mijn 'mediakeuzes'.

De laptop die ik eerder had aangeschaft is ook voorzien van Bluetooth. Handig? Nou, net niet helemaal. Als ik de hoofdtelefoon wil koppelen aan mijn laptop is de verbinding niet stabiel. Anders gezegd: De hoofdtelefoon weet niet of er een verbinding met de losse ontvanger of met de laptop in stand gehouden moet worden. En om nou de ontvanger iedere keer uit te schakelen als ik met mijn laptop de hoofdtelefoon wilde gebruiken vond ik niet ideaal.

Ik had natuurlijk grondiger onderzoek moeten doen voor die aanschaf. Tja.

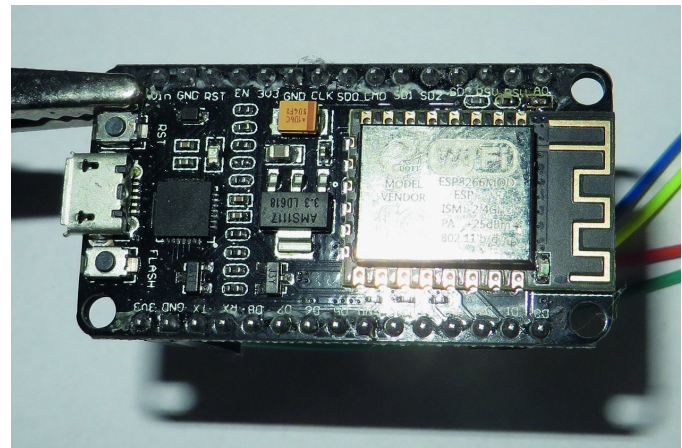
Het idee

Omdat ik al jaren wifi-schakelaars in de kast had liggen en omdat ik een aantal NodeMCU-modules (meer info verderop in dit artikel) aangeschaft had, besloot ik toch maar eens voorzichtig verder aan de slag te gaan met domotica. Ik wilde de wifi-schakelaar of de NodeMCU gebruiken om de Bluetooth-ontvanger in te schakelen als ik de televisie of tuner wilde gebruiken. Probleem: de wifi-schakelaar schakelt

alleen 230V wisselspanning door en de NodeMCU kan niet voldoende vermogen schakelen om de Bluetooth-ontvanger te activeren.



Een van de eerste wifi-schakelaars



Een van de eerste NodeMCU-printen

Dergelijke componenten komen (bijna) altijd uit China. Zo ook deze twee. De voordelen zijn voor de consument duidelijk: een lage prijs. Beide componenten waren toen (2017) ongeveer 3 euro per stuk. Nadelen zijn er ook. Het ontwerp is meestal 'economisch'. Goedkoop ontwerp en goedkope componenten.

Daarnaast geldt voor de wifi-schakelaar dat de fabrikant hoopt dat je de door hun geleverde software gebruikt. En voor de NodeMCU geldt dat de on-board-chip waarmee je via bijvoorbeeld je pc babbelt, niet de meest gangbare is. Gelukkig is er in vijf jaar veel verbeterd. De wifi-schakelaar heeft veel 'broertjes en zusjes' gekregen met sterk verbeterd ontwerp. De NodeMCU koop je tegenwoordig met meer geheugen, gangbare communicatiechip en Bluetooth op de koop toe.

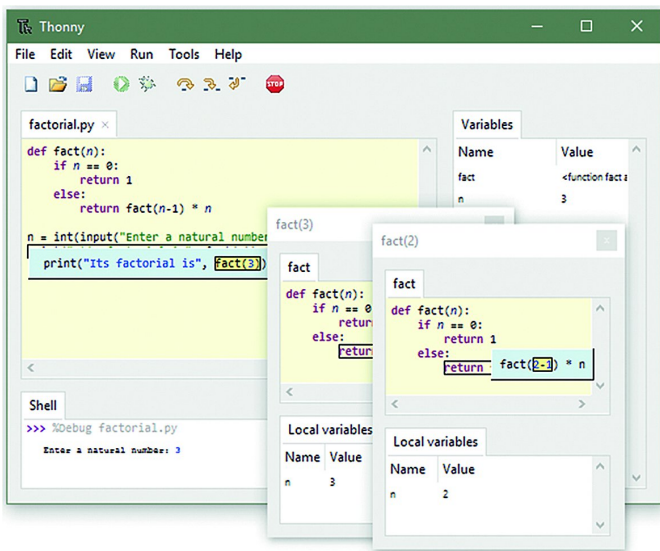
ESP(8266): het kloppend hart

Beide componenten hebben als kloppend hart een ESP8266 chip. Het was toentertijd de meest gebruikte chip voor consumententoepassing. Je vind ze echt overal in verstop. Inmiddels is er een sterk verbeterde opvolger: de ESP32. Een apparaat met een ESP8266 is bijvoorbeeld met Arduino

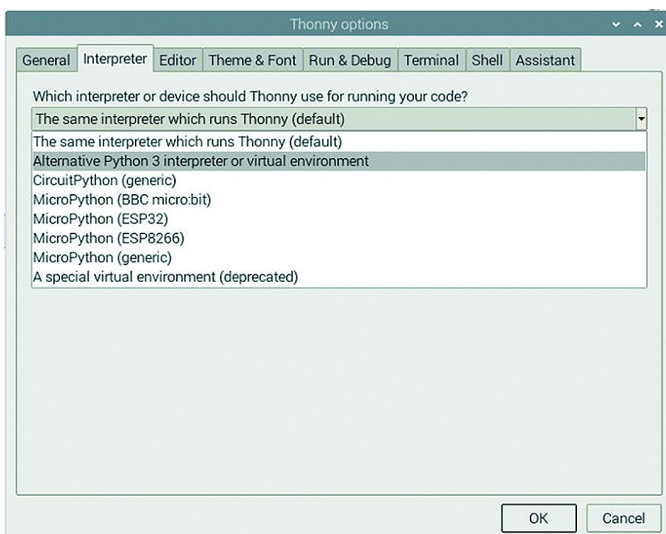
te programmeren. De software is gratis en het kost meestal weinig moeite om de software voor jouw toepassing te vinden. En als je niet de juiste software vindt, doe je meestal oneindig veel inspiratie op. Zo ook tijdens mijn zoektochten.

Thonny¹

Het eerste alternatief dat ik tegenkwam was Thonny. 'Thonny is an integrated development environment (IDE) for Python that is designed for beginners'. Python is op dit moment een van de meest populaire programmeertalen en is ontwikkeld door de Nederlander Guido van Rossum. Leuke bijkomstigheid. Thonny is beschikbaar voor Windows, Mac en Linux. Na installatie en starten van de software kom je in het openingsscherm terecht.



Met Thonny kun je een groot aantal apparaten programmeren. Daarvoor heeft Thonny een mogelijkheid ingebakken om een 'interpreter' te selecteren:



Je ziet dat er 'out of the box' al een aantal bekende namen tevoorschijn komen. Als laatste koppel je je device aan je pc of laptop aan en je probeert te communiceren. Ik schreef dat mogelijk de driver voor jouw device niet standaard ondersteund wordt door Windows. Met andere woorden: Windows herkent de aangeboden hardware niet en biedt geen mogelijkheid om te communiceren. In dat geval moet je dus zoeken naar een driver. Als je deze stappen doorlopen hebt, kun je beginnen aan het schrijven van een programma.

Het programma transfereer je dan naar het device en het bestand krijgt de naam main.py. Naast het reguliere programma is er ook een boot.py. Dit is bedoeld om bijvoorbeeld variabelen te declareren. Bij opstarten wordt eerst boot.py doorlopen en aansluitend main.py. Als er geen boot.py aangetroffen wordt volgt automatisch main.py. Voor de retrolezers onder ons: vergelijk het een beetje met config.sys c.q. autoexec.bat.

Ik had een basis gevonden voor een programma waar een webserver gebruikt werd om de hardware te bedienen. M.a.w.: ik kon dan met behulp van een smartphone of tablet via een webpagina een functie op de hardware aan- en uitzetten. Maar, zoals ik al even benoemd had was de NodeMCU niet krachtig genoeg om de Bluetooth-ontvanger in te schakelen. De uitgangen leverden maximaal 30mA en de Bluetooth-ontvanger vroeg meer. Bovendien waren de uitgangen 3.3V en de ontvanger was gebaseerd op USB, dus op 5V. Daarmee kwam ik op een dood spoor.

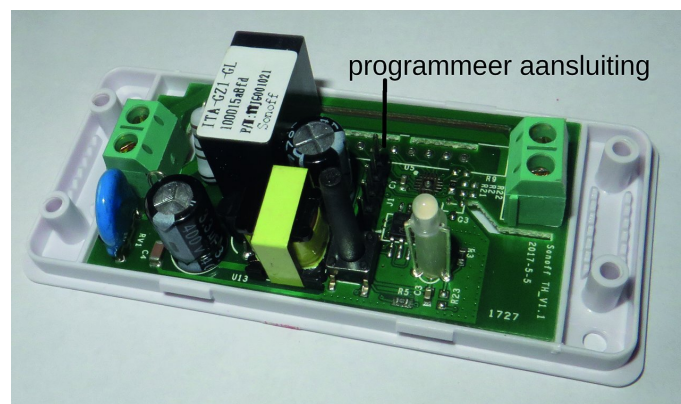
Ik besloot om over te stappen naar de wifi-schakelaar.

Tasmota²

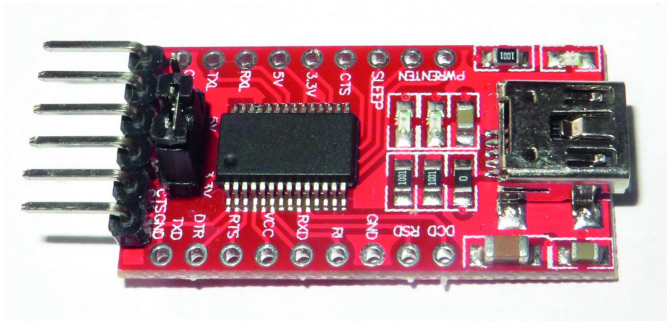
Tijdens mijn zoektocht naar een oplossing op basis van een Sonoff wifi-schakelaar stuitte ik op 'Tasmota'. Tasmota is ontwikkeld door Theo Arends. Tasmota is open source firmware voor hardware gebaseerd op de Espressif ESP8266, ESP32, ESP32-S of ESP32-C3 chipset. Het begon als Sonoff-MQTT-OTA op 25 januari 2016. Het eerste doel was op ESP8266 gebaseerde Sonoff hardware met MQTT en 'Over the Air' of OTA-firmware. Het begon allemaal als een hack om cloud-gebonden devices te ontkoppelen en beschikbaar te maken voor niet-cloud-, maar lokaal aangestuurde installaties. Immers: als je de software installeert om de wifi-schakelaar via de cloud te bedienen, heb je natuurlijk geen idee waarmee je 'ergens in de wereld' verbinding maakt. En zo ontstond een compleet ecosysteem voor elke willekeurige hardware met een ESP8266 chip. Tasmota is een afkorting van Theo Arends Sonoff Mqtt Ota, Het werd me al snel duidelijk dat Tasmota een oplossing bood voor mijn probleem en voor nog veel meer.

Flashen

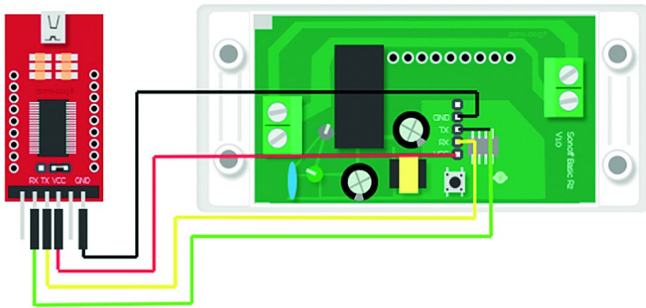
In tegenstelling tot een NodeMCU waar je een programma in 'schiet', moet je jouw apparaat met Tasmota software 'flashen'. Daarna is het enkel nog configureren. Bij de NodeMCU is het zaak om de juiste driver te installeren. Bij de Sonoff is het zaak om over het juiste toeltje te beschikken. Bij de Sonoff is de programmeeraansluiting op de printplaat beschikbaar gemaakt d.m.v. een aantal aansluitingen. NB! Tijdens het programmeren mag de schakelaar niet aangesloten zijn op het lichtnet. Het aansluiten op het lichtnet is pas nodig als de wifi-schakelaar geflasht is en de configuratie van de wifi-schakelaar gereed is. Anders kun je je laptop, of elektronica in het algemeen, onherstelbaar beschadigen.



Om deze programmeeraansluiting te kunnen gebruiken heb je een stukje elektronica nodig. Het tooltje wordt vaak aangeduid met 'FTDI232'. Pas voor elektronica met de ESP8266 de 3.3V variant toe. De pinnen op de wifi-schakelaar verbind je met de pinnen op de FTDI232.



De FTDI232 verbind je een pc of laptop via USB. Dat ziet er dan schematisch zo uit:



Het daadwerkelijke flashen kan via een Chrome-compatibele browser, maar er zijn ook lokale installatiemogelijkheden. Voor dit artikel gebruik ik de webinstaller³: [o/install/](#)

Install Tasmota

1. Connect the ESP device to your computer using USB or serial-to-USB adapter
2. Select the firmware variant suitable for your device
3. Hit "Install" and select the correct port or find help if no device found

Tasmota (english) ▾

All ▾

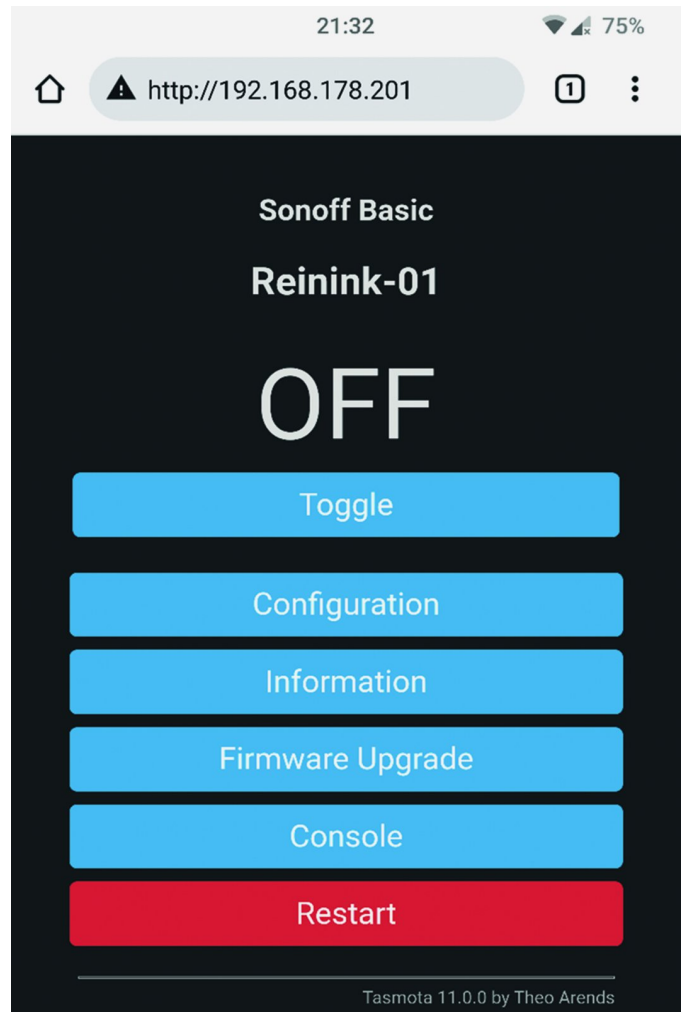
*Your browser does not support Web Serial.
Open this page in Google Chrome or
Microsoft Edge instead (but not on your mobile device).*

Tasmota Installer powered by ESP Web Tools

Kies voor een eerste test 'Tasmota (english)' en bij 'All' kies je voor ESP8266. Als alles goed gaat wordt de ESP8266 geflasht. Tasmota zal een vrij IP-adres toekennen aan jouw device. Wil je dat niet, dan kun je in Tasmota of in jouw router een vast IP-adres toekennen.

Je kunt in het keuzemenu ook kiezen voor andere firmware. Dit omdat de firmware enigszins groot is en je gehouden bent aan de grootte van het geheugen op je device. Deze procedure is per device maar een keer noodzakelijk,

tenzij je andere firmware wilt gebruiken. Alle vervolgacties lopen via een browser. En dat hoeft niet noodzakelijk een Chrome-browser te zijn. Ik heb bij de installatie een vast IP-adres toegekend en heb een aantal zaken naar smaak aangepast.



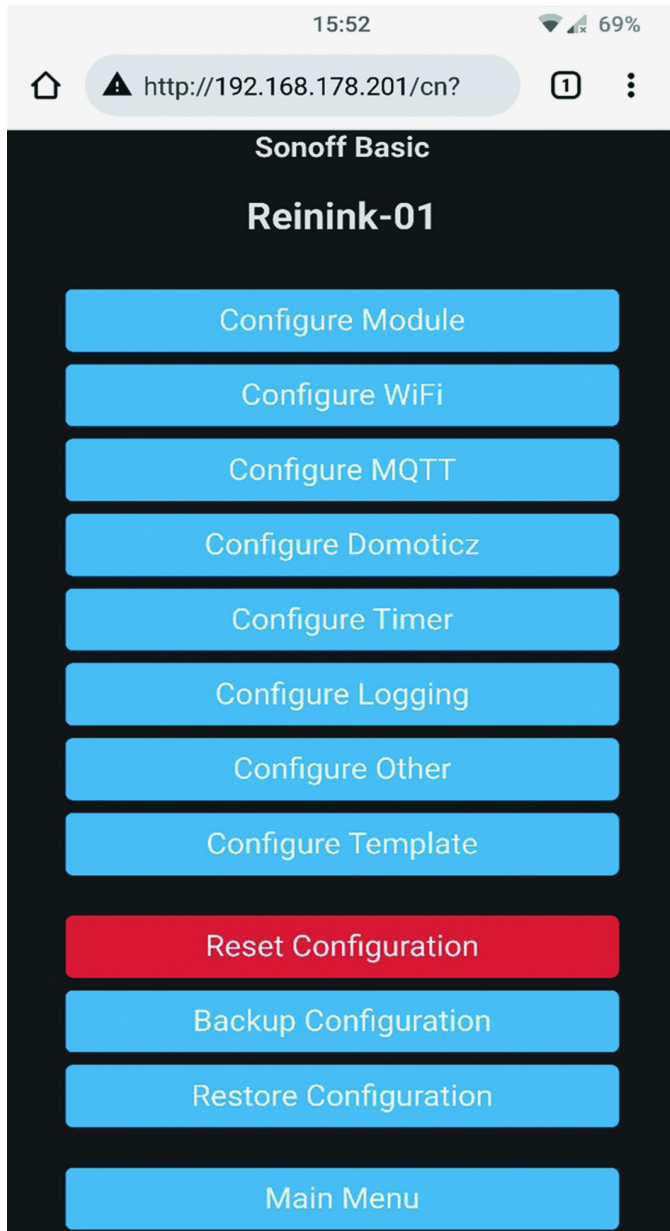
Op mijn telefoon heb ik het vaste IP-adres van de wifi-schakelaar in de URL-balk ingegeven. De schermafbeelding toont een weergave van de webinterface van de wifi-schakelaar.

Je ziet dat ik de wifi-schakelaar een naam heb gegeven en je ziet ook de status van de schakelaar. Het voert te ver voor dit artikel om alle verdere details te behandelen. Maar de optie 'Configuration' in het menu wil ik nog even onder de aandacht brengen. Als je dit menu binnengaat kom je in een submenu met nog veel meer instelmogelijkheden.

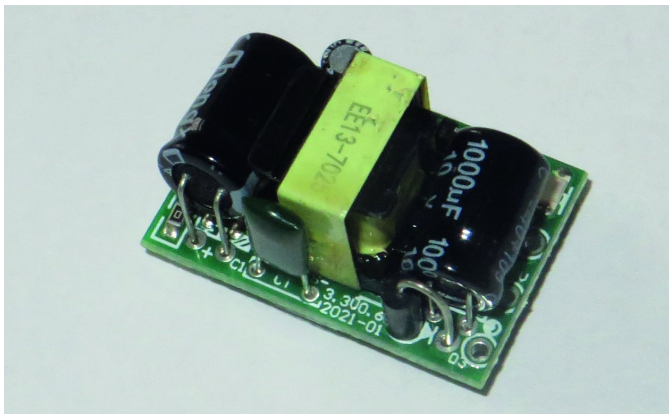
Onder module kun je je eigen module selecteren en onder 'Configure Template' kun je je module configureren. Ik heb voor het gemak de Sonoff-basic genomen. Met 'Toggle' in het hoofdscherm kan ik het relais op de print in- en uitschakelen.

Het relais schakelt de ingangsspanning (230V) door. Ik moet dus nog iets verzinnen om deze spanning om te zetten naar 5V voor de Bluetooth ontvanger. Daarvoor vond ik, wederom in China, een convertor van 230V wisselspanning naar 5V gelijkspanning. Deze omzetter monteerde ik achter de uitgang van de wifi-schakelaar. Als ik nu de wifi-schakelaar toggle dan schakelt het relais van de wifi-schakelaar in en zet 230V op de ingang van de convertor. De convertor geeft aan haar uitgang een gelijkspanning van 5V. Ik monteer er voor het gemak een USB-connector aan en kan dan met een standaard USB-kabel de Bluetooth ontvanger aansluiten. Missie volbracht.

Het configuratiemenu met een scala aan mogelijkheden:



Deze omzetter van 230V wisselspanning naar 5V gelijkspanning heb ik gebruikt:

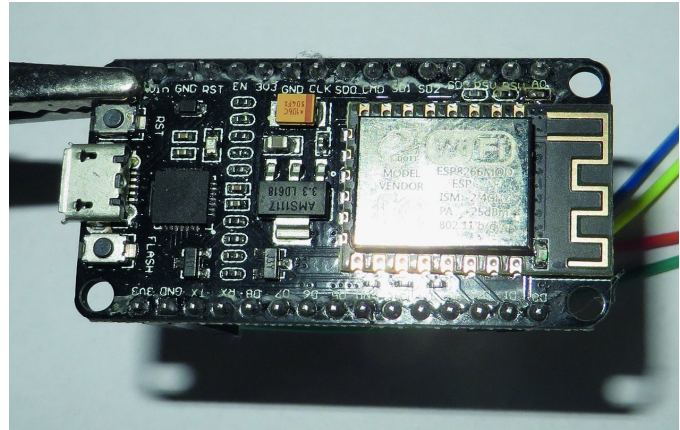


De omzetter is ongeveer 15x25mm (!) groot. Voor dit soort artikelen is China een ware schatkamer, waarin je je uren kunt vermaken. Vaak is alles wat je nodig denkt te hebben er wel te vinden.

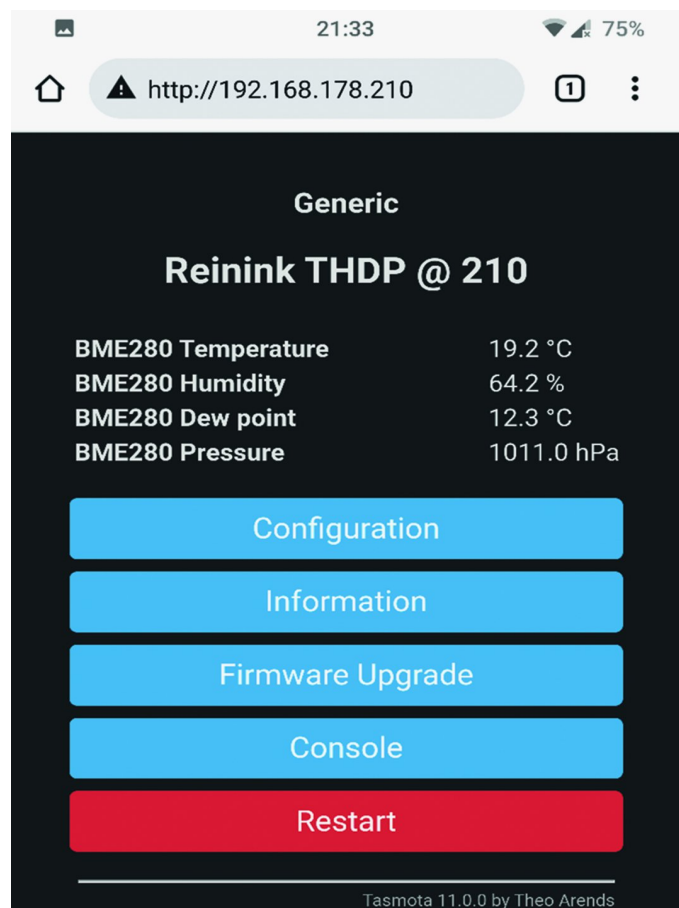
Dat smaakt naar meer!

Tijdens deze eerste kennismaking met Tasmota zag ik dat behalve een Sonoff wifi-schakelaar er nog veel meer 'geprogrammeerd' kan worden. Als je op de site van Tasmota kijkt zie je dat behalve aan-uit functies ook sensoren aangesloten kunnen worden. En ik had nog een aantal sensoren in de rommelbak liggen.

Zo ook een BME280. Een sensor waarmee je temperatuur, luchtvochtigheidsgraad, dauwpunt en luchtdruk kunt meten. Ik besloot om de NodeMCU (26x48mm) daarvoor te gebruiken:



Net als bij de wifi-schakelaar paste ik het een en ander aan. Het uitzoeken van de juiste pinning op de NodeMCU en configureren van deze aansluitingen in Tasmota was het meeste werk. Na enige aanpassingen ziet het er dan zo uit:



Even voor de duidelijkheid: ik had de sensor ook kunnen aansluiten op de wifi-schakelaar. De wifi-schakelaar is veel meer dan 'aan-uit' omdat de ESP8266 bepaalt wat je er mee kunt. Bij de wifi-schakelaar zijn er slechts een beperkt aantal in- en uitgangen naar buiten gebracht via pinnen. Sommige zijn niet op pinnen aangesloten, maar zijn op de print via soldeeraansluitingen benaderbaar. En dat is natuurlijk voor een knutselaar een ideale uitdaging.

Zo kun je op een Sonoff meerdere sensoren of tot wel 8 relais aansluiten. Maar de beste en de meest interessante mogelijkheden moeten nog komen.

Het loggen van gegevens

Alle data van de sensor is terug te vinden in het hoofdmenu onder 'Console'. Naast de data kun je in de console ook commando's geven.

Alle mogelijke commando's met uitleg van de functie zijn te vinden op website van Tasmota.

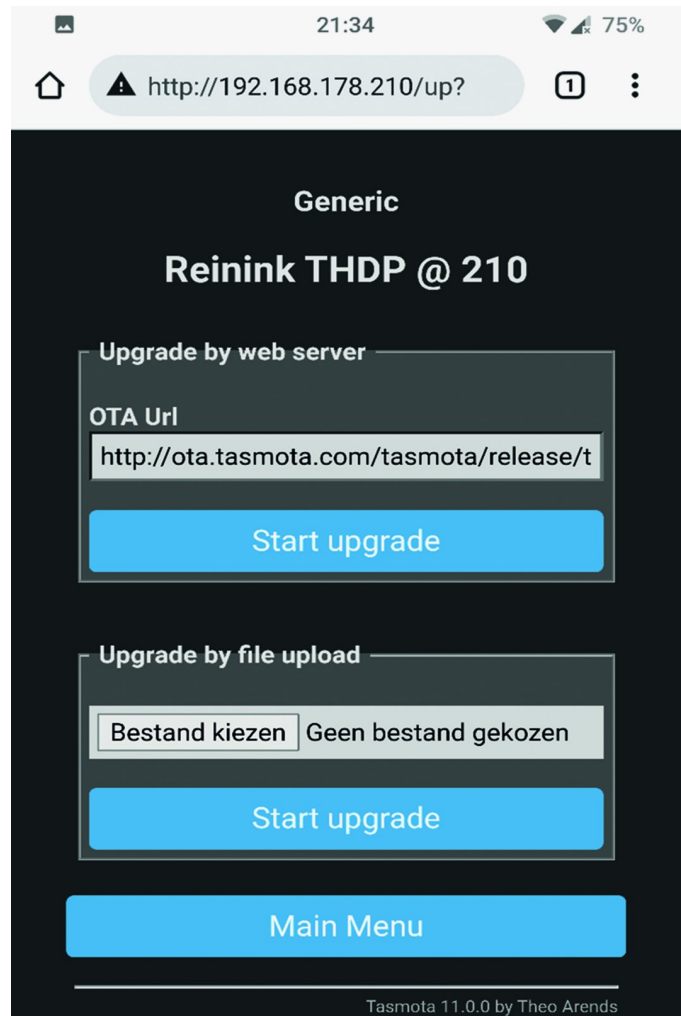
De uitvoering van commando's is afhankelijk van de gekozen hardware.



Upgrade van de firmware

Upgraden gaat heel simpel. Je hebt de mogelijkheid om via dezelfde procedure als de eerste flashprocedure een update uit te voeren. Maar de meeste gebruikers kiezen er voor om de upgrade via een draadloze verbinding te doen: OTA = Over The Air. Eigenlijk net zoals je dat tegenwoordig doet met een smartphone. Theo Arends waarschuwt op zijn website voor te enthousiaste 'upgraders'. Als iets goed werkt, upgrade dan niet. The first rule of upgrading: If it ain't broke, don't fix it!

Vergeet niet om voorafgaand aan de upgrade een back-up te maken. Op de website vind je uitgebreide en aanvullende informatie.



De status tot nu toe

We zien dat je met Tasmota betrekkelijk eenvoudig apparaten met een Espressif-chip kunt programmeren en kunt integreren in je eigen (wifi)netwerkomgeving. De vele mogelijkheden zijn met één artikel onmogelijk volledig te beschrijven. Hoe meer je je met Tasmota bezighoudt, hoe meer je ontdekt. De software die Theo Arends heeft ontwikkeld is zeer uitgebreid en zeer doordacht. Chapeau! De software is het zeker waard om te waarderen met een donatie. Overigens is er ook concurrentie voor Tasmota. Maar daarover later meer.

En hoe nu verder?

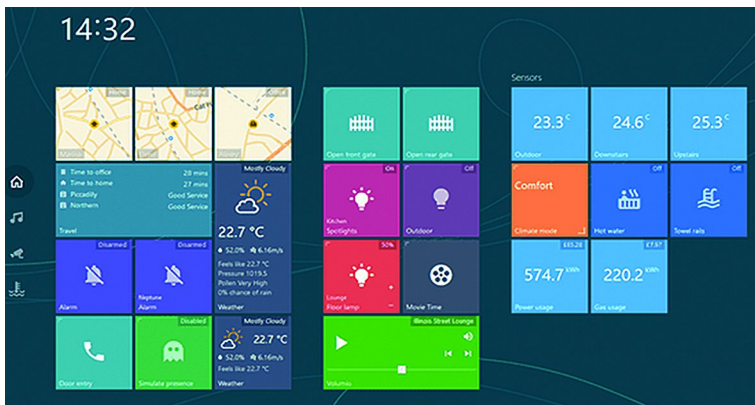
In de screenshots heb je ook de term MQTT zien staan. MQTT staat voor MQ Telemetry Transport. Ik noem het voor het gemak maar even berichten. O.a. met behulp van MQTT heb je de mogelijkheid om deze berichten te gebruiken. M.a.w.: je kunt deze 'data' aanschouwelijk maken in een overzicht. Denk daarbij aan de stand van een schakelaar, de temperatuur in een ruimte, de aanwezigheid van een persoon, een deurbel, jouw slimme energiemeter, of algemener: de actuele tijd. Meestal noemen we zo'n overzicht een dashboard. In principe geldt 'The sky is the limit'. Immers, je kunt je bv. voorstellen dat je de afzuigkap inschakelt als er iemand in de keuken is en het kooktoestel aan staat. En als de kookplaat 15 minuten uit is schakel je de afzuigkap uit en stuur je een berichtje aan de tafelgasten. Een mogelijk voorbeeld van domotica. De meeste lezers kennen Apple Homekit, Amazon Alexa en Google Home.

Visualiseren

Alle data kun je visualiseren met gratis software. Daar heb je geen Apple, Amazon of Google voor nodig. Liever nog helemaal geen van deze. Immers jouw data moeten jouw data blijven, toch? Bovendien: wat gebeurt er als een fabrikant de stekker uit een project trekt?

Omdat ik op mijn BananaPi via Armbian (voor de gebruikers van een Raspberry Pi: Raspbian) de hardware kan configureren en software kan installeren, stuitte ik laatst op 'Home Assistant'⁴. Voor een volgend artikel ben ik van plan om daar mee aan de slag te gaan.

Zo'n dashboard zou er dan zo uit kunnen zien:



Voor de lezers die gebruik maken van zonnepanelen en/of een warmtepomp is een dashboard niets nieuws. Veelal krijg je thuis een soort panel op een centrale plaats opgehangen waarin je alle gegevens kunt inzien. Of op je smartphone natuurlijk. In dit voorbeeld wordt alles voorgeconfigureerd aangeleverd.

De juiste keuze

Maar wat is de juiste keuze qua componenten en software? Ik had al aangegeven dat er een concurrent is voor Tasmota: ESPHome⁵. Ook met ESPHome kun je je sensor programmeren en configureren. Dat gaat net iets anders dan in Tasmota. In Tasmota flash je de gewenste firmware in de elektronica. Dat kost relatief veel geheugenruimte. Vandaar dat Tasmota verschillende firmware aanbiedt. Bijvoorbeeld firmware voor alleen sensoren. Maar daarna kun je met de webinterface dan alle sensoren simpel gebruiken. ESPHome pakt het anders aan. Bij ESPHome maak je eerst een configuratiebestand van de te gebruiken sensor. Daarna compileer je dit bestand en transfereer je het naar de doel-elektronica.

Wat dat betreft kun je ESPHome enigszins vergelijken met

Arduino. Als je de diverse informatiebronnen leest lijkt het een wedstrijd zonder overtuigende winnaar. Eerlijkheidshalve moet ik wel schrijven dat enkele vrienden hun installatie omgezet hebben naar een ESPHome basis⁶. De verwachting is dat door de komst van nieuwe componenten met meer geheugen Tasmota in de race blijft.

Protocollen

Misschien wel het lastigste onderdeel. Welke componenten schaf ik aan? Wat is voor mijn doel het beste geschikt? Wil ik een 'drukknoploos' huis of wil ik achter elke drukknop een intelligente component. En waarom zijn er zoveel protocollen? Hue, Z-Wave, KNX, Zigbee. En dan vergeet ik er vast nog wel een paar. Niet alle protocollen kunnen met elkaar babbelen. Het is dus belangrijk dat je vooraf een plan de campagne maakt.

Componenten

Bij het aanschaffen van componenten is het verstandig om vooraf te onderzoeken of de component te 'hacken' is. Ik denk dat de Sonoff voor een kleine revolutie gezorgd heeft. Er zijn op basis van deze schakelaar vele klonen op de markt gebracht. Maar een merk dat bij mij erg in de smaak valt is Shelly. Het assortiment van Shelly wordt steeds uitgebreider en interessanter. Misschien moet ik voor een volgend artikel maar eens een aantal Shelly⁷ componenten aanschaffen. En dan ook maar eens ervaring opdoen met programmeren c.q. compileren met ESPHome.

Veiligheid

Tot slot nog een paar woorden over veiligheid.

Het is verstandig om na te denken over hoe je het een en ander wilt gaan inrichten thuis. Voor een eerste test thuis kun je natuurlijk de verschillende componenten opnemen in je eigen netwerk.

Maar als je de smaak te pakken hebt en meer wilt gaan doen met domotica is het verstandig om het anders aan te pakken. Het is verstandig om je domotica zoveel mogelijk ontogankelijk te maken voor de buitenwereld. Een simpele thermometer die van buiten benaderd kan worden door derden is niet rampzalig. Wel als daarmee een gebruikersnaam en/of wachtwoord gevonden kan worden. Het is verstandig om een scheiding aan te brengen in de verschillende netwerken. Een netwerk dat verbonden is met de buitenwereld, een netwerk dat gedeeltelijk verbonden is met de buitenwereld en een netwerk dat niet verbonden is met de buitenwereld.

Makkelijker gezegd dan gedaan.

Ook daar hoop ik een volgende keer over te schrijven.

1 Thonny

<https://thonny.org/>
<https://en.wikipedia.org/wiki/Thonny>
<https://forum.pycom.io/topic/5999/boot-py-vs-main-py>
<https://tinyurl.com/mtu82jxn>

2 Tasmota

<https://tasmota.github.io/docs/>
<https://tasmota.github.io/docs/About/>
<https://www.ct.nl/reviews/tasmota-firmware-sonoff/>
<https://tinyurl.com/mr3hsn67>

3 Tasmota webinstaller

<https://tasmota.github.io/install/>

4 Home-assistant

<https://www.home-assistant.io/>
<https://www.home-assistant.io/integrations/tasmota/>
<https://tinyurl.com/un95dp6t>

5 ESPHome

<https://esphome.io/>

6 Tasmota vs. ESPHome

<https://www.youtube.com/watch?v=nHaFM0tK0vY>
<https://tinyurl.com/yzwhrwey>

7 Shelly

<https://www.shelly.cloud/en-nl>