



Modbus

via WLAN (deel 2)

software voor de Modbus TCP WLAN-module

Josef Bernhardt en Martin Mohr (Duitsland)

De module die we in het eerste deel van deze serie hebben beschreven, is opgebouwd rond een Espressif NodeMCU-board. U kunt hiermee het Modbus-protocol gebruiken via een WLAN. In dit artikel gaan we in op de software en de configuratie van de Modbus-module.

In het eerste deel [1] van deze serie hebben we de hardware besproken. De module is opgebouwd rond een NodeMCU-board voorzien van een ESP8266-microcontroller met een extra basisboard voor industrie-compatibele poorten. Na een korte introductie van het Modbus-protocol kijken we hoe de module wordt bestuurd en welke software we daarbij gebruiken. We gebruiken bij dit voorbeeld weer de liftdeur-besturing uit het eerste deel.

Het Modbus-protocol

Het Modbus-protocol wordt veel toegepast in de automotive sector. Het werkt volgens het master/slave-principe; de bus-master kan maximaal 246 slave-apparaten aansturen. De busdeelnemers krijgen een adres van 1 tot 247; adres 0 is gereserveerd voor **broadcasts**: alle data die naar dat adres wordt verzonden, wordt ontvangen door alle nodes. De consistentie van de individuele datapakketten wordt gegarandeerd met CRC-checksums. De slaves hebben interne registers met verschillende functies. **Tabel 1** geeft een overzicht van de Modbus-functies. Het Modbus-protocol kent drie varianten:

- Modbus RTU: binaire overdracht via RS485 (EIA485);
- Modbus ASCII: tekstoverdracht via RS485 (EIA485). Dit is minder efficiënt dan binaire overdracht, maar leesbaar voor mensen. Commando's kunnen met een eenvoudig terminalprogramma worden verzonden;
- Modbus TCP: bij deze variant worden de Modbus-commando's verzonden met behulp van TCP/IP. Meestal gaat dat via Ethernet, maar deze module gebruikt WiFi.

Zie de Modbus-website [2] voor meer informatie en protocolbeschrijvingen.

Voorbereiding van de Modbus WLAN-module

Om de WLAN-module te kunnen gebruiken met het Modbus-protocol, moet u eerst de firmware laden. Maak het NodeMCU-board los van de Modbus-kaart en sluit het aan op een USB-poort van uw computer. Open dan de Arduino-IDE, die is geconfigureerd zoals beschreven in deel 1. U kunt de firmware die de module verandert in een Modbus-client downloaden van onze projectpagina [3].

Download het bestand *OpenPLC_ESP8266_1_0_MUX_V1_1.ino* en open dat in de Arduino IDE (*Bestand -> Open...*). De Arduino-IDE vraagt dan of u het project wilt verplaatsen naar het sketchbook. Antwoord met 'Ja'. Als de IDE een nieuwe map maakt, kopieer dan het header-bestand (*modbus.h*) naar de nieuwe map met het *ino*-bestand, zodat de compiler het kan vinden.

Om de module te kunnen bereiken via het WLAN, moeten de WLAN-gegevens aan het begin van de broncode worden aangepast (zie **listing 1**). Voer daar de gegevens van uw eigen netwerk in.

Na deze aanpassing kunt u het programma in de Modbus-module laden, zoals beschreven in deel 1. De module werkt met DHCP, dus hij krijgt automatisch een IP-adres van uw router. U kunt dat adres vinden

Tabel 1. Overzicht van de Modbus-functies.

Beschrijving	Mode	Bits
Individuele "Coil" input/output	Lezen/schrijven	1
Individuele "Discrete Inputs"	Lezen	1
"Input Registers" (analoog/digitaal)	Lezen	16
"Holding Registers" inputs/outputs (analoog/digitaal)	Lezen/schrijven	16

in de router. U kunt ook kijken naar de uitvoer in de seriële monitor van de Arduino-IDE (door op het vergrootglas rechts bovenaan te klikken). Kies in het dropdown-menu rechts onderaan voor een overdrachts-snelheid van 115.200 baud. In **listing 2** ziet u een voorbeeld van hoe het IP-adres wordt weergegeven in de seriële monitor. Om te voorkomen dat u dit elke keer moet doen als u de Modbus-module inschakelt, kunt u een statisch IP-adres toewijzen. Alle moderne routers bieden die mogelijkheid. In elk geval moet u het IP-adres kennen om de module te kunnen benaderen via uw netwerk. Als u geen uitvoer ziet, druk dan op de resetknop op de NodeMCU-kaart om de software opnieuw te starten, want het IP-adres wordt alleen weergegeven bij het opstarten.

Een eerste test

Om te controleren of de Modbus WLAN-module goed werkt, kunt u EasyModbus gebruiken, wat u kunt downloaden van SourceForge [4]. EasyModbus heeft een server, een client en een library. Voor deze test is alleen de client nodig. Die kan rechtstreeks worden gedownload van [5]. Om hem te kunnen gebruiken, moet Java geïnstalleerd zijn op uw PC. Als u Java nog niet hebt geïnstalleerd, kunt u de gratis OpenJDK [6] gebruiken; deze is erg populair in de ontwikkelaars- en makers-community en is een standaardcomponent van alle Linux-distributies. In Ubuntu kan het bijvoorbeeld worden geïnstalleerd met het commando `sudo apt install openjdk-11-jdk`

Om de EasyModbus-client te starten, geeft u het commando `java -jar EasyModbusJavaClient.jar`. In **figuur 1** ziet u de uitvoer van de EasyModbus-client (links onder Linux; rechts onder Windows). Kies hier **Modbus TCP** en voer het IP-adres van uw module in. Laat het **Starting Address** op 1 staan en verander **Number of Values** in 4. Als u nu op **Read Discrete Inputs - FC2** klikt, worden de vier digitale inputs van de module ingelezen en weergegeven. Sluit nu een testsignaal



Listing 1. WLAN-configuratie Modbus-module.

```

/*****NETWORK CONFIGURATION*****/

const char *ssid = "<YOUR_SSID>";
const char *password = "<YOUR_PASSWORD>";

/*****/

```



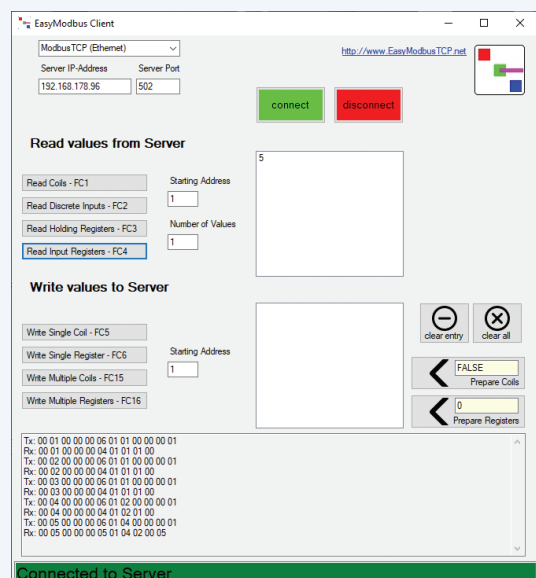
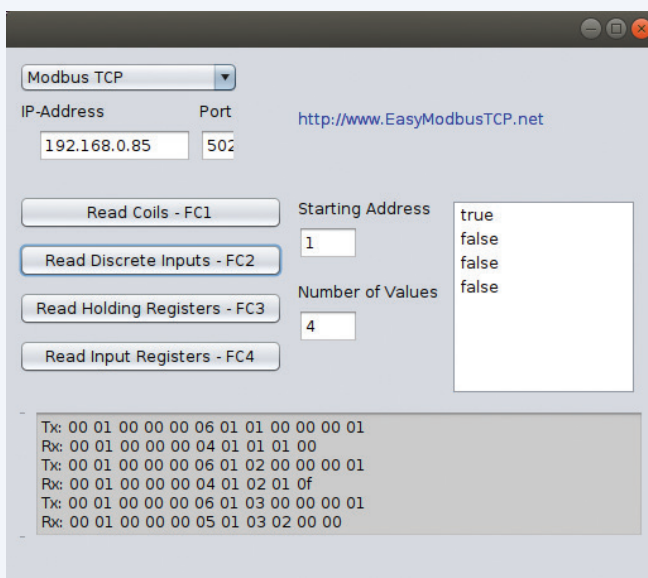
Listing 2. Weergave IP-adres Modbus-WLAN-module.

```

Connecting to Vodafone-3980
.....
WiFi connected
Server started
My IP: 192.168.0.85

```

aan op minstens één van de inputs om te zien of alles goed werkt. De EasyModbus-client kan alleen data lezen van Modbus; hij kan niet schrijven. Om het schrijven te testen, kunt u het voorbeeldprogramma gebruiken dat hieronder wordt beschreven. De data die wordt verzonden is te zien onderin het venster van de EasyModbus-client. Dat is heel handig bij het debuggen. Als de eerste test is geslaagd, kunt u de module gaan programmeren met Python.



Figuur 1. De EasyModbus-client leest data uit de module uit. Links de Linux-versie en rechts de Windows-versie.



Listing 3. Voorbeeldprogramma `tor.py` voor de liftdeurbesturing.

```
from pyModbusTCP.client import ModbusClient
client = ModbusClient(host="192.168.0.85",
    port=502, auto_open=True, debug=False)
while(True):
    inputs=client.read_discrete_inputs(0,4)
    end_switch_top = inputs[0]
    end_switch_bottom = inputs[1]
    push_button_down = inputs[2]
    push_button_up = inputs[3]
    motor_up = 0
    motor_down = 1
    # request end_switch
    if(end_switch_top):
        client.write_single_coil(motor_up,False)
        print("gate open")
    if(end_switch_bottom):
        client.write_single_coil(motor_down,False)
        print("gate closed")
    # request push button
    if (push_button_up and not end_switch_top):
        client.write_single_coil(motor_up,True)
    if (push_button_down and not
        end-switch_bottom):
        client.write_single_coil(motor_down,True)
```

Installeren van de Modbus+ bibliotheek

Om de test wat realistischer te maken, gebruiken we weer de liftdeurbesturing uit het eerste deel [7]. We hebben een PC-programma geschreven dat de toestand van de drukknoppen en de eindcontacten ontvangt via Modbus om de motor (ook weer via Modbus) aan te sturen.

Ons Python-programma heeft een bibliotheek [8] nodig om met de Modbus te communiceren. Dat is niet ongebruikelijk; er zijn Modbus-bibliotheken voor praktisch alle programmeertalen. We zouden desnoods zelf de bitreeksen voor de commando's kunnen genereren en ze over het netwerk sturen. Maar daar beginnen we hier niet aan. Zorg eerst, dat u Python's package installer *pip* aan boord hebt, dat hoort bij moderne Python-installaties altijd zo te zijn. U kunt dat testen met `$ python -m pip --version` onder Linux of met `C:\> py -m pip --version` onder Windows.

Als u *pip* nog niet hebt, kunt u het installeren met `get-pip.py` [9] (gebruik daar niet de Linux package manager voor):

```
wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
get-pip.py
```

Installeer daarna de Modbus-library:

```
$ python -m pip install -U pymodbusTCP
C:\> py -m pip install -U pymodbusTCP
```

Uitgebreide documentatie en voorbeeldprogramma's voor deze bibliotheek zijn te vinden onder [10].

Voorbeeldprogramma

In **listing 3** ziet u een voorbeeldprogramma. De eerste regel importeert de Modbus-bibliotheek. De tweede regel instantieert een nieuw verbindingsobject om te communiceren met een specifiek Modbus-apparaat. Daarbij kunnen verschillende parameters worden meegegeven. Wij geven het IP-adres van het target-device mee als `host`. Poort 502 is de default poort voor TCP-communicatie met Modbus. De parameter `auto_open` geeft aan of de verbinding automatisch of met de hand moet worden gemaakt. In ons voorbeeld kiezen we voor het gemak voor automatisch, maar als u alles zelf in de hand wilt houden, kiest u voor `False`. Als u de verbinding met de hand maakt, hebt u meer mogelijkheden voor besturing en foutafhandeling. Als u de parameter `debug` op `True` zet, geeft de software alle uitgezonden data weer op het console. Dat is heel handig bij het foutzoeken.

We gebruiken een `while`-lus om het programma uit het eerste deel meerdere malen uit te voeren. Het programma werkt dus cyclisch, net als alle industriële controllers. De code in de lus is het belangrijkste deel van het programma, hij leest de invoer en zet de gelezen waarden om naar bruikbare variabelen. We definiëren ook variabelen voor de twee digitale uitgangen. Dat maakt het programma gemakkelijker te lezen. U ziet dat de motor wordt uitgeschakeld als de eindpunten worden bereikt. Er verschijnt een tekstbericht op het console, zodat u de huidige positie van de deur kunt zien op de computer. Verder laat de software de motor in de juiste richting draaien als er op Up of Down is gedrukt. De logische AND met de eindschakelaars voorkomt dat het relais gaat stuiten als de deur al in de eindstand staat.

Geef het commando `python tor.py` om het programma te starten. U kunt het programma weer stoppen door op `Ctrl-C` te drukken.

AdvancedHMI

Er zijn verschillende PC-frameworks beschikbaar waarmee u gemakkelijk een afstandsbesturing met visuele feedback kunt realiseren. Het programma AdvancedHMI is open source-software voor het maken van HMI-toepassingen (mens-machine-interface) die communiceren met uw PLC of andere I/O-apparaten. Deze software verschilt van andere standaardpakketten omdat u er uitvoerbare bestanden mee kunt maken in plaats van configuraties die door een *runtime engine* moeten worden geïnterpreteerd. Dat levert zeer snelle, efficiënte toepassingen op.

AdvancedHMI [11] is gebaseerd op het Microsoft .NET-framework. De toepassingen worden gegenereerd in *Microsoft Visual Studio Community 2019*, die gratis verkrijgbaar is. Hiermee kunt u eenvoudige HMI's maken door symbolen te slepen zonder dat u code hoeft te schrijven. Het .NET-framework wordt gebruikt door veel ontwikkelaars en er is veel ondersteuning beschikbaar. Er is veel meer ondersteuning voor AdvancedHMI dan voor alle andere standaard HMI-pakketten bij elkaar.

De auteur heeft dit platform gebruikt voor het genereren van een gebruikersinterface om de basiskaart te besturen. Deze geeft de status van de ingangen weer en heeft buttons voor het schakelen van de uitgangen (**figuur 2**).

.NET-programma's zoals deze gebruikersinterface kunnen ook draaien onder Linux met Mono. Het is succesvol getest op een Raspberry Pi.

Aan de slag

Met de Modbus WLAN-module kunt u eenvoudig Modbus-apparaten besturen vanaf een PC of een smartphone. Deze module is ook een goed project om te leren werken met het Modbus-protocol. Het hardware-ontwerp is zo robuust en fout-tolerant dat u het ook kunt gebruiken in uw eigen ESP8266-projecten voor het besturen en uitlezen van industriële apparaten zoals elektromagnetische kleppen, motoren en andere sensoren of actuatoren. ◀

200507-B-03

Een bijdrage van

Idee, ontwerp en tekst: **Josef Bernhardt** en **Martin Mohr**

Redactie: **Rolf Gerstendorf**

Vertaling: **Evelien Snel**

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 josef@bernhardt.de of naar de redactie van Elektor via redactie@elektor.com.

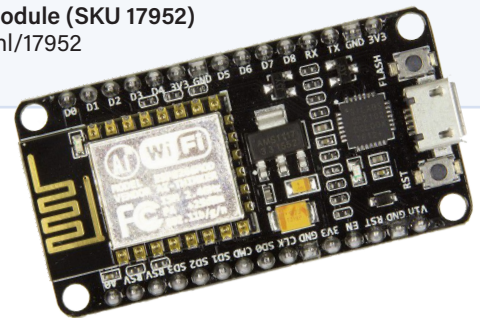


Figuur 2. De AdvancedHMI-software genereert HMI's.



GERELATEERD PRODUCT

> **NodeMCU Module (SKU 17952)**
www.elektor.nl/17952



WEB LINKS

- [1] Modbus via WLAN (deel 1), ElektorMagazine september/oktober 2021: <http://www.elektormagazine.nl/200507-03>
- [2] Modbus-website: <https://modbus.org/>
- [3] Projectpagina bij dit artikel: <https://www.elektormagazine.nl/200507-B-01>
- [4] EasyModbus-website: <https://bit.ly/2QDIDJz>
- [5] Download EasyModbus-client: <https://bit.ly/3d2zzFp>
- [6] Download OpenJDK: <https://bit.ly/2OVK4m6>
- [7] De liftdeur op YouTube: <https://youtu.be/VHIBQswdA0E>
- [8] Modbus-library voor Python: <https://pypi.org/project/pyModbusTCP/#description>
- [9] get-pip.py: <https://bootstrap.pypa.io/get-pip.py>
- [10] Voorbeelden Modbus-bibliotheek: <https://pymodbusTCP.readthedocs.io/en/latest/>
- [11] AdvancedHMI: https://www.advancedhmi.com/index.php?main_page=page&id=14&chapter=0
- [12] Meer informatie van de auteur: <https://bit.ly/3ch1PFI>