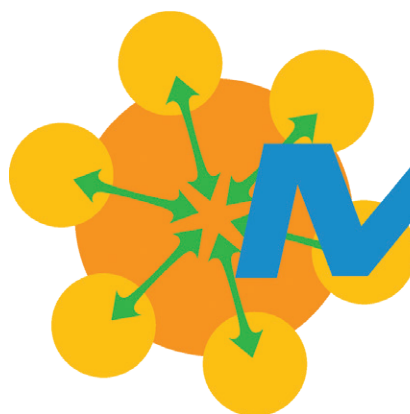




Modbus

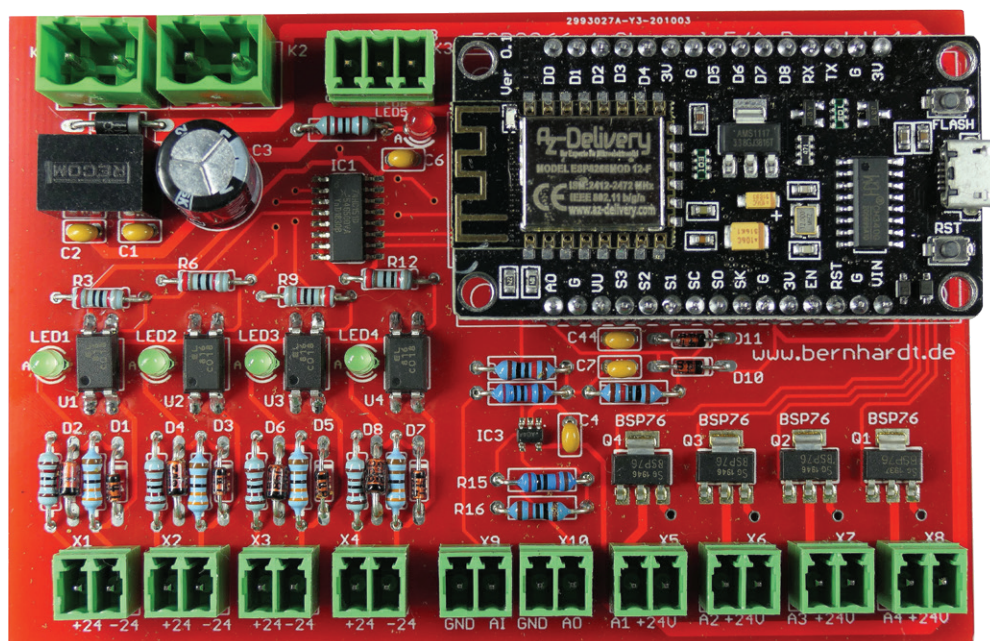
via WLAN (deel 1)

Hardware en programmering

Josef Bernhardt (Duitsland) en
Martin Mohr (Duitsland)

Het Modbus-protocol wordt in de industriële sector breed ingezet voor de communicatie tussen verschillende systemen. Meestal gaat dat via de oude, vertrouwde RS485-interface en conventioneel koperdraad. Hier presenteren we een module waarmee

u het Modbus-protocol kunt gebruiken in een draadloos netwerk (WLAN). De module is opgebouwd rondom een NodeMCU-kaart met een ESP8266-microcontroller van Espressif. Met een aanvullend Modbus-basisboard kunt u werken met 24V-signalen, een in de industriële omgeving gebruikelijk spanningsniveau. Om de werking te demonstreren, hebben de auteurs een eenvoudige besturing voor een liftdeur ontwikkeld.



Modbus-kaart met NodeMCU-module.

De meeste Elektor-lezers zijn waarschijnlijk al bekend met de NodeMCU-module van Espressif en met de Arduino-IDE. Als dat ook voor u geldt, kunt u deze inleiding overslaan en meteen doorgaan naar de beschrijving van de Modbus-TCP-kaart. Als dat niet zo is, kunt u hier in het kort lezen wat u moet weten.

Dit project is opgebouwd rondom een NodeMCU-module (verkrijgbaar in de ElektorShop). De module bevat een Espressif ESP8266-microcontroller, die half zo groot is als een postzegel en een WLAN-interface aan boord heeft. Ondanks zijn kleine afmetingen heeft hij een groot rekenvermogen. De

belangrijkste eigenschappen van de ESP8266 zijn samengevat in **Tabel 1**. De NodeMCU-kaart genereert de voedingsspanning voor de ESP8266 en verzorgt de programmeer-interface voor de microcontroller. In **figuur 1** ziet u een overzicht van de aansluitpennen van de NodeMCU in onze Modbus-schakeling.

De Arduino-IDE is heel geschikt voor het programmeren van de NodeMCU-kaart. U kunt de juiste versie van de Arduino-IDE voor uw besturingssysteem gratis downloaden van de Arduino-website [2] en hem installeren volgens de instructies op de site. Als u de IDE voor het eerst opstart, ziet u een venster als in **figuur 2**. In het programmeervenster staan twee voorgedefinieerde functies. Bovenaan staat de functie `setup()`. Die wordt één keer uitgevoerd bij de start van het programma en zorgt voor dingen als het initialiseren van de interfaces. Onderaan staat de functie `loop()`, die de rest van het programma vormt.

De functie `loop()` wordt uitgevoerd als de functie `setup()` is afgerond. Telkens als het programma het einde van de functie `loop()` bereikt, wordt die opnieuw uitgevoerd. De ESP8266 bestuurt de WLAN-interface tussen de aanroepen van de functie `loop()` door. U moet dus geen oneindige lussen opnemen in de functie `loop()`, want anders zal de ESP8266 gegarandeerd crashen. De code in de functie `loop()` moet zó worden ontworpen, dat hij herhaaldelijk kan worden uitgevoerd. Veel schijnbaar onbegrijpelijke crashes van de ESP8266 worden veroorzaakt door het feit dat de processor niet genoeg CPU-tijd krijgt om de WLAN-interface te besturen. Als er een kans is, dat uw programma veel tijd nodig heeft, bijvoorbeeld in grote lussen, dan kunt u gebruik maken van de functies `yield()` of `delay()` om de ESP8266 genoeg tijd te geven voor de besturing van het WLAN.

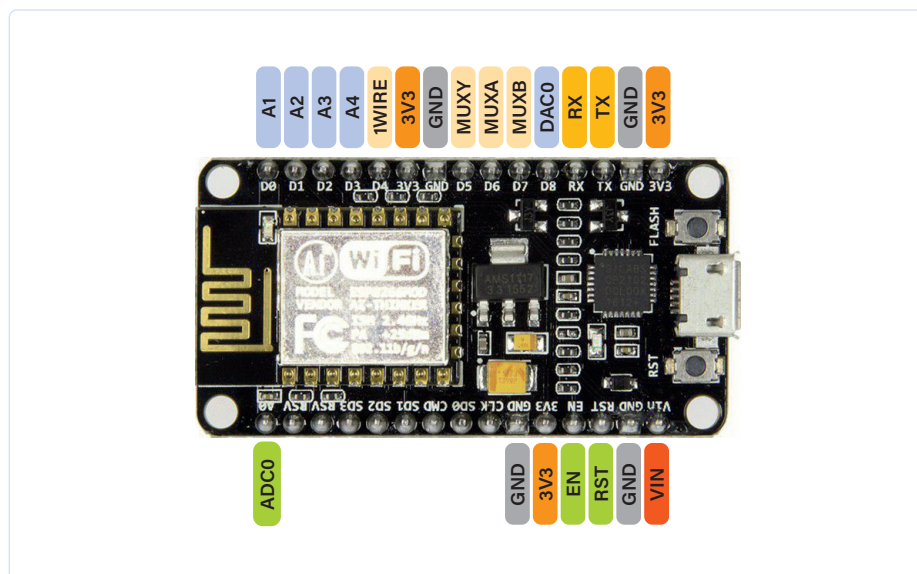
Voordat u de NodeMCU-module in de Modbus-kaart steekt, moet u hem aansluiten op een USB-poort van uw computer, maar er zijn nog meer voorbereidingen nodig. De ESP8266 wordt standaard niet ondersteund door de Arduino IDE, dus we moeten eerst de IDE updaten. Ga daarvoor naar *File* → *Preferences* en voer de volgende URL in bij *Meer Boards Managers URL's*:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Klik op *OK* en kies dan *Tools* → *Board* ... → om het venster *Board Beheer* te openen. Zoek daar naar *ESP8266* en installeer de *ESP8266 Community*-boards.

Als die zijn geïnstalleerd, ziet u onder *Tools* → *Board* → het board *NodeMCU 1.0 (ESP-E12 Module)*, en onder *Tools* → *Port* de poort waar het NodeMCU-board op is aangesloten (*/dev/ttyUSBx* bij Linux of *COMx* bij Windows).

Nu kunt u aan de slag met het eerste eenvoudige testprogramma op de NodeMCU-module. Open daarvoor het



Figuur 1: Aansluitpennen van de NodeMCU-module voor het Modbus-project.

Tabel 1: Technische gegevens van de ESP8266.

32-Bits-RISC-CPU	Tensilica Xtensa LX106 met een klokfrequentie van 80 MHz
Programmageheugen	64 kB
Datageheugen	96 kB
Extern flashgeheugen (optioneel)	512 kB tot 4 MB
GPIO	10 pennen; ondersteunde interfaces: SPI, I ² C, I ² S, UART. Ook DMA en IRQ worden ondersteund.
WiFi	IEEE 802.11b/g/n WEP- of WPA/WPA2-authentisering WiFi-Direct (P2P), Soft AP Geïntegreerde TCP/IP-protocolstack
ADC	1x 10bits-converter
Antenne	Geïntegreerd
Uitgangsvermogen	+19.5 dBm in 802.11b-modus
Afmetingen	17,2 x 12,3 mm
Standby-stroom	< 1,0 mA

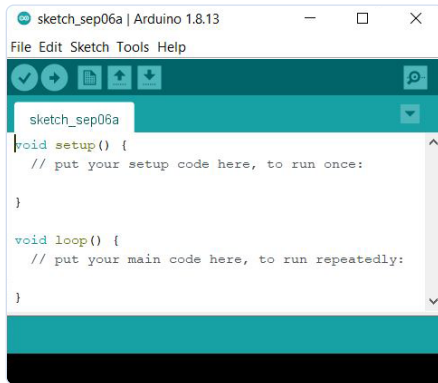
voorbeeldprogramma 'Blink' onder *File* → *Examples* → *ESP8266* → *Blink*. Het programma in **listing 1** laat de LED op de ESP8266-kaart knipperen. De code die de LED in- en uitschakelt zit in de functie `loop()`. In de functie `setup()` wordt de GPIO-pen van de LED geconfigureerd als een output.

Om het programma in de NodeMCU-kaart te laden, klikt u op het *Upload*-pictogram (een pijl naar rechts). Het duurt een paar seconden om het programma over te dragen, en dan begint de LED te knipperen. De installatie

Listing 1:
Blink-programma voor de ESP8266

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  digitalWrite(LED_BUILTIN, HIGH);
  delay(2000);
}
```



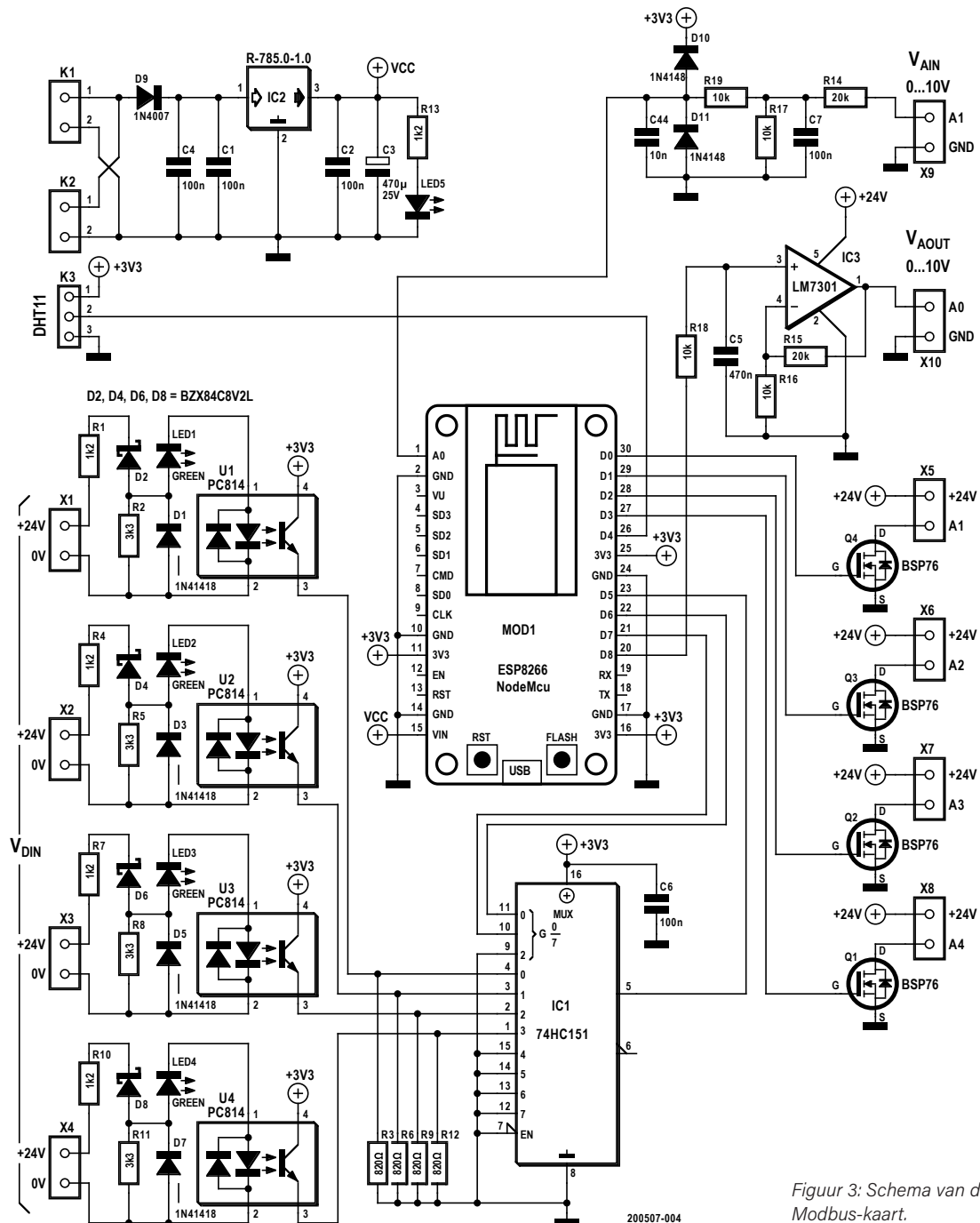
en de inrichting van de IDE zijn daarmee afgerond.

Het Modbus-TCP-Board

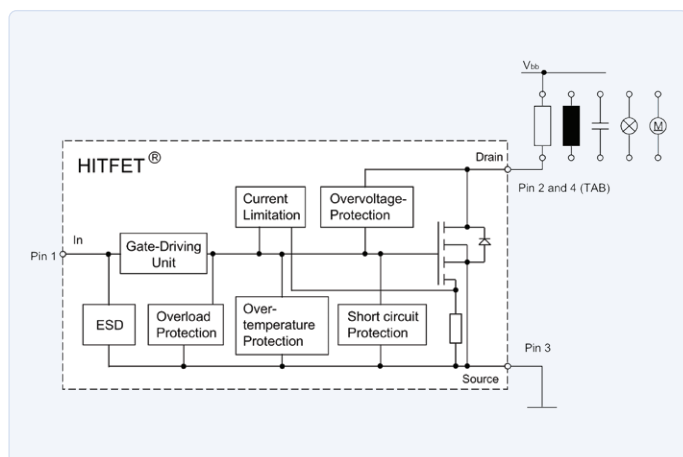
Zoals u in de titelfoto kunt zien, wordt de NodeMCU-module geplaatst op de Modbus-TCP-kaart, die er behoorlijk "industrieel" uitziet. Alle inputs en outputs zijn voorzien van

connectors voor schroefterminals. Dat vereenvoudigt de opbouw en maakt het mogelijk om de schroefterminals te vervangen als ze beschadigd raken. De belangrijkste taak van de Modbus-TCP-kaart is het omzetten van de signaalniveaus van 3,3 V van de microcontroller naar de 24 V van de industriële omgeving. In het schema in **figuur 3** ziet u dat er vier galvanisch van de microcontroller geïsoleerde inputs zijn met optocouplers van het type PC814. Deze inputs zijn niet rechtstreeks verbonden met de microcontrollerkaart, maar

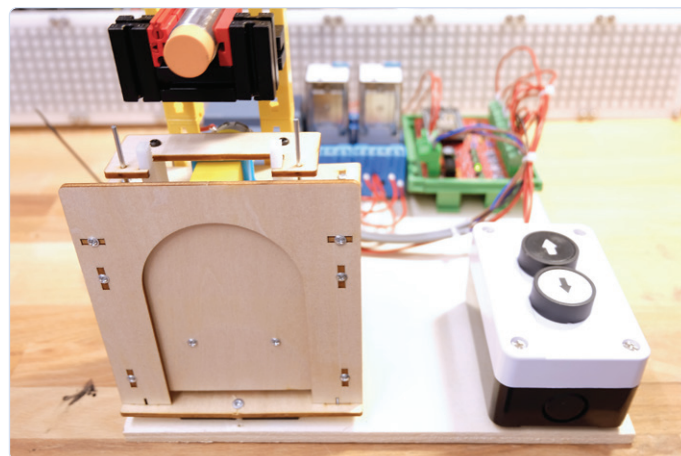
Figuur 2: Het beginscherm van de Arduino-IDE.



Figuur 3: Schema van de Modbus-kaart.



Figuur 4: Interne structuur van de BSP76 (bron: Infineon datasheet).



Figuur 5: Houten model van een liftdeur.

via een 74HC151-multiplexer. Dat spaart één GPIO-pen uit, waardoor ruimte ontstond voor een DHT11-sensor voor het meten van de temperatuur en luchtvochtigheid. Dat maakt de Modbus-kaart ook geschikt voor eenvoudige HVAC-besturingstaken.

De uitgangsdrievoren zijn voorzien van HITFET BSP76's (figuur 4). Dat zijn geen simpele vermogens-MOSFET's, maar schakel-IC's met ingebouwde beschermingsfuncties, zoals overspannings-, stroom- en overtemperatuur-bescherming. U kunt daarmee probleemloos ohmse, inductieve of capacatieve belastingen schakelen.

Naast de digitale inputs en outputs heeft de Modbus-TCP-kaart een analoge input en een analoge output. Beide analoge poorten werken in standaard spanningsbereik in de industrie van 0 tot 10 V. Bij de analoge input reduceert de spanningsdeler R14/R17 de ingangsspanning tot een maximum van 3,3 V. Deze input is niet galvanisch gescheiden, maar C7, D10 en D11 bieden enige bescherming tegen pieken en overspanning. De seriële weerstanden R14 en R19 beperken de ingangsstroom om te voorkomen dat de diodes in rook opgaan als er iets fout gaat aan de input.

Bij de analoge output versterkt IC3 (een LM7301 general-purpose opamp met rail-to-rail inputs en output) de uitgangsspanning van de DAC van de NodeMCU. De versterking van de opamp is met de weerstanden R15 en R16 ingesteld op 3. Deze opamp wordt gevoed uit de 24V-voedingsslijn.

De DC/DC-converter van Recom verdient een aparte vermelding. Hij zet de voedingsspanning van 24 V om naar de 5 V die nodig is

voor het voeden van de NodeMCU-module. Hij heeft een rendement van meer dan 90%, en is heel veelzijdig in te zetten. Hij kan bijvoorbeeld ook een negatieve voedingsspanning voor opamps genereren. Het is zeker de moeite waard om het datasheet van deze DC/DC-converter te bekijken [4]. Aan de 24V-input zit een 1N4007-diode om de schakeling te beschermen tegen verkeerd-om aansluiten van de voeding.

De Gerber-files voor het maken van de print zijn te downloaden van de projectpagina [8]. De kaart is bij de auteur verkrijgbaar [7] als kale print of volledig bestukt en klaar voor gebruik.

Demonstratie met een houten model

Om te laten zien, hoe de Modbus-module kan worden ingezet in een typische industriële taak, hebben de auteurs een houten

model van een liftdeur uit een bekende Chinese webshop gebruikt [5] (figuur 5). De deur wordt geopend en gesloten met twee drukknoppen. De eindposities van de deur worden gedetecteerd met twee capacatieve nabijheidssensoren. De motor die de deur opent en sluit wordt aangestuurd door een H-brugschakeling bestaande uit twee vermogensrelais.

Het besturingsprogramma is te zien in Listing 2. Eerst worden mnemonische namen voor de inputs en outputs gedefinieerd. De individuele I/O-pennen worden geïnitialiseerd als inputs en outputs in de functie `setup()`. Daarna worden `input()`- en `output()`-functies gedefinieerd, die het wat gemakkelijker maken om de I/O-pennen te gebruiken. Het hoofdprogramma zit in de functie `loop()`. Daar worden de outputs voor het sturen van de motor aangestuurd, afhankelijk van de drukknoppen en de eindschakelaars. Het

WEBLINKS

- [1] **NodeMCU-kaart:** www.elektor.nl/nodemcu-microcontroller-board-with-esp8266-and-lua
- [2] **Download de Arduino-IDE:** www.arduino.cc/en/software
- [3] **BSP76-datasheet:** <https://bit.ly/3r3GP99>
- [4] **Datasheet van de DC/DC-converter:** <https://recom-power.com/pdf/Innoline/R-78-0.5.pdf>
- [5] **Houten model van een liftdeur:** www.aliexpress.com/item/33008243573.html
- [6] **De liftdeur op YouTube:** <https://youtu.be/VHIBQswdA0E>
- [7] **Josef Bernhardt:** www.bernhardt.de
- [8] **Projectpagina:** www.elektormagazine.nl/200507-03

Listing 2: Liftdeurbesturing.

```
#define A1 16
#define A2 5
#define A3 19
#define A4 0
#define MUXA 12
#define MUXB 13
#define MUXY 14
#define E1 0
#define E2 1
#define E3 2
#define E4 3

void setup() {
    pinMode(A1, OUTPUT);

    pinMode(A2, OUTPUT);
    pinMode(A3, OUTPUT);
    pinMode(A4, OUTPUT);
    pinMode(MUXA, OUTPUT);
    pinMode(MUXB, OUTPUT);
    pinMode(MUXY, INPUT);
}

bool input(int i){
    digitalWrite(MUXA, (i&1));
    digitalWrite(MUXB, (i&2)>>1);
    delay(1);
    return digitalRead(MUXY);
}

void output(int i, int v){
    digitalWrite(i,v);
}

void loop() {
    if(input(E4)& !input(E1)){output(A1,HIGH);}
    if(input(E3)& !input(E2)){output(A2,HIGH);}
    if(input(E1)){output(A1,LOW);}
    if(input(E2)){output(A2,LOW);}
}
```

programma in de functie `loop()` kan herhaaldelijk worden uitgevoerd. Als u de liftdeur in actie wilt zien, kunt u ook gewoon naar de YouTube-video [6] kijken.

Vooruitblik

De werkende liftdeurbesturing is natuurlijk een leuke demo, maar hij doet geen recht aan de mogelijkheden van de Modbus-kaart en de WLAN-module. Daar gaan we op in in het tweede deel van dit artikel, waar we het Modbus-protocol gaan behandelen en bespreken welke software u kunt gebruiken om over de bus te communiceren. We gaan dan ook in op het configureren van de Modbus-module. 

200507-03

Met bijdragen van:

Ontwerp en artikel: **Josef Bernhardt** en **Martin Mohr**

Redactie: **Rolf Gerstendorf**

Lay-out: **Giel Dols**

Vertaling: **Evelien Snel**

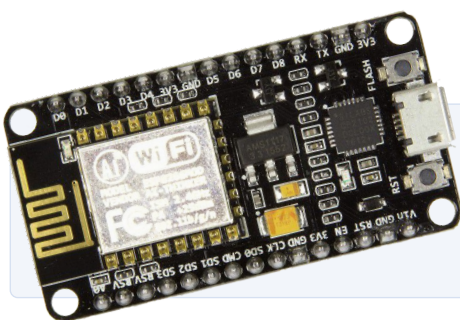
Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen? Neem dan contact op met de auteur via josef@bernhardt.de of met de redactie via editor@elektor.com.

Over de auteurs

Josef Bernhardt was al op jonge leeftijd geïnteresseerd in elektronica. Hij bouwde zijn eerste kristalontvanger toen hij twaalf was en ging door met andere schakelingen. Hij leerde programmeren in de jaren tachtig met de Commodore VIC20 en hij weet ook de weg in het programmeren in assemblertaal voor de 8088. Hij kan terugkijken op meer dan 30 jaar elektronica-evolutie bij de University of Regensburg, waar hij werkte aan de ontwikkeling van elektronica en software. Met zijn eigen SMD-productiefaciliteit implementeert hij ook elektronische projecten voor klanten.

Martin Mohr zag het levenslicht in het tijdperk van magnetische ringkernegeheugens en hefdraaikiezers en kon dus persoonlijk de hele ontwikkeling van de moderne computertechnologie meemaken. Zijn fascinatie voor alles wat knippert, gaat terug naar zijn vroege jeugd en werd verder versterkt door zijn elektronicaopleiding. Na het afronden van een studie in de informatica werkte hij voornamelijk aan de ontwikkeling van Java-toepassingen, maar de Raspberry Pi heeft zijn passie voor elektronica opnieuw doen opleven.



GERELATEERDE PRODUCTEN

> NodeMCU Module

www.elektor.nl/nodemcu-microcontroller-board-with-esp8266-and-lua