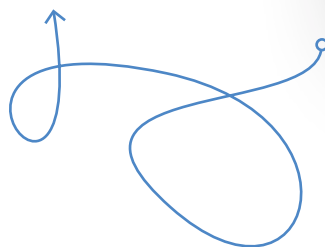




Connected Thermostaat met ESP32

Houd uw wijn op de juiste temperatuur!



Yves Bourdon (Frankrijk)

Om een goede rijping op lange termijn te realiseren, moet wijn in een donkere en koele ruimte worden bewaard. Hoewel specialisten het niet eens zijn over de beste temperatuur hiervoor, raden ze wel aan om de temperatuurwaarde zo constant mogelijk te houden. De hier gepresenteerde thermostaat probeert niet alleen dat te doen, maar stuurt ook waarschuwingen per e-mail wanneer de temperatuur te hoog wordt.

Het idee voor dit project ontstond tijdens het bouwen van de uitstekende WiFi-thermostaat van Elektor [1], die is ontworpen rond de ESP8266 van Espressif. Ik probeerde de software van dat project over te zetten naar een ESP32, maar gaf dat op omdat dat te

veel werk bleek. Ik besloot daarom vanaf nul opnieuw te beginnen en wat al gedaan was als inspiratie te gebruiken (webinterface, NTP-klok, temperatuurregeling met programmeerbare hysteresis, enz.) en de webinterface en zijn tekortkomingen te verbeteren (met name werd de gebruikersinterface niet in real-time ververst). Ik heb ook een klein OLED display toegevoegd, waardoor deze thermostaat zelfs nog bruikbaar is als de WiFi verbinding is onderbroken (degraded mode). Dit project is gemakkelijk aan te passen voor andere toepassingen. Als voorbeeld heb ik dezelfde hardware gebruikt met een vochtigheidssensor DHT11 (of BME280) om het mechanische ventilatiesysteem in mijn badkamer aan te sturen. De met een optocoupler geïsoleerde ingang detecteert of het licht brandt (12 V spots) zodat hij de lawaaiige luchtafzuiging tijdelijk kan uitschakelen wanneer je de badkamer bezoekt.

Het Circuit

Het schema van de ESP32-thermostaat is weergegeven in **figuur 1**. Om de kosten, betrouwbaarheid en afmetingen te optimaliseren, heb ik besloten om de Heltec WiFi Kit 32 module (**figuur 2**) te gebruiken in het hart van de thermostaat. Deze ESP32-gebaseerde module, die gemakkelijk op het Internet te vinden is, heeft een mooi blauw 0,96" 128 x 64-pixel OLED-display aan boord en



Specificaties

- Temperatuursensor DS18B20, $\pm 0,5^{\circ}\text{C}$ nauwkeurig, weergave tot op $0,1^{\circ}\text{C}$ nauwkeurig.
- Relais met gesynchroniseerde contacten voor het schakelen van AC-belastingen tot 2200 W.
- Voedingsspanning 100-240 VAC.
- I²C-uitbreidingsbus compatibel met Grove en Qwiic.
- Print ondersteunt Heltec WiFi kit 32 met geïntegreerd OLED-display of ESP32 DevKitC plus extern OLED-display.
- Zoemer voor akoestisch alarm.
- Externe schakelaar-ingang.
- Optisch geïsoleerde ingang.
- Drie programmeerbare drempels: minimum, maximum en alarm.
- Alarm kan e-mail versturen.
- Op NTP-gebaseerde tijdsaanduiding.
- Weekdag timer.
- Kan een koeler of een verwarming regelen.



heeft een door de gebruiker programmeerbare witte LED. Hij wordt aangesloten op K5 en K6. Merk op dat een ESP32 DevKitC module (connectoren K7 en K8) samen met een afzonderlijk 0.96" 128 x 64 I2C OLED-display gebruikt kan worden in plaats van de WiFi Kit 32 module, aangezien ze beide vrij gangbaar zijn (**Figuur 3**). Het OLED-display bestaat in twee pinout-versies (met VCC en GND verwisseld) en daarom zijn er twee connectoren voor voorzien (K9 & K10). De software is voor beide modules identiek.

Stroomvoorziening

De voeding is een kleine 3 W AC/DC-converter. Hij accepteert eeningangsspanning tussen 100 en 240 VAC en levert 5 V bij maximaal 600 mA. LED3 geeft aan of de voeding is ingeschakeld. Een zekering beschermt het apparaat en het lichtnet, terwijl varistor VAR1 overspanningsbeveiliging biedt (zeer gewaardeerd door de conformiteitsnormen). De netvoeding wordt aangesloten via aansluitblok K1.

De ESP32-module wordt gevoed met 5 V, aangezien hij over een eigen 3,3 V-regelaar beschikt. De module kan ook rechtstreeks worden gevoed via de micro-USB-bus.

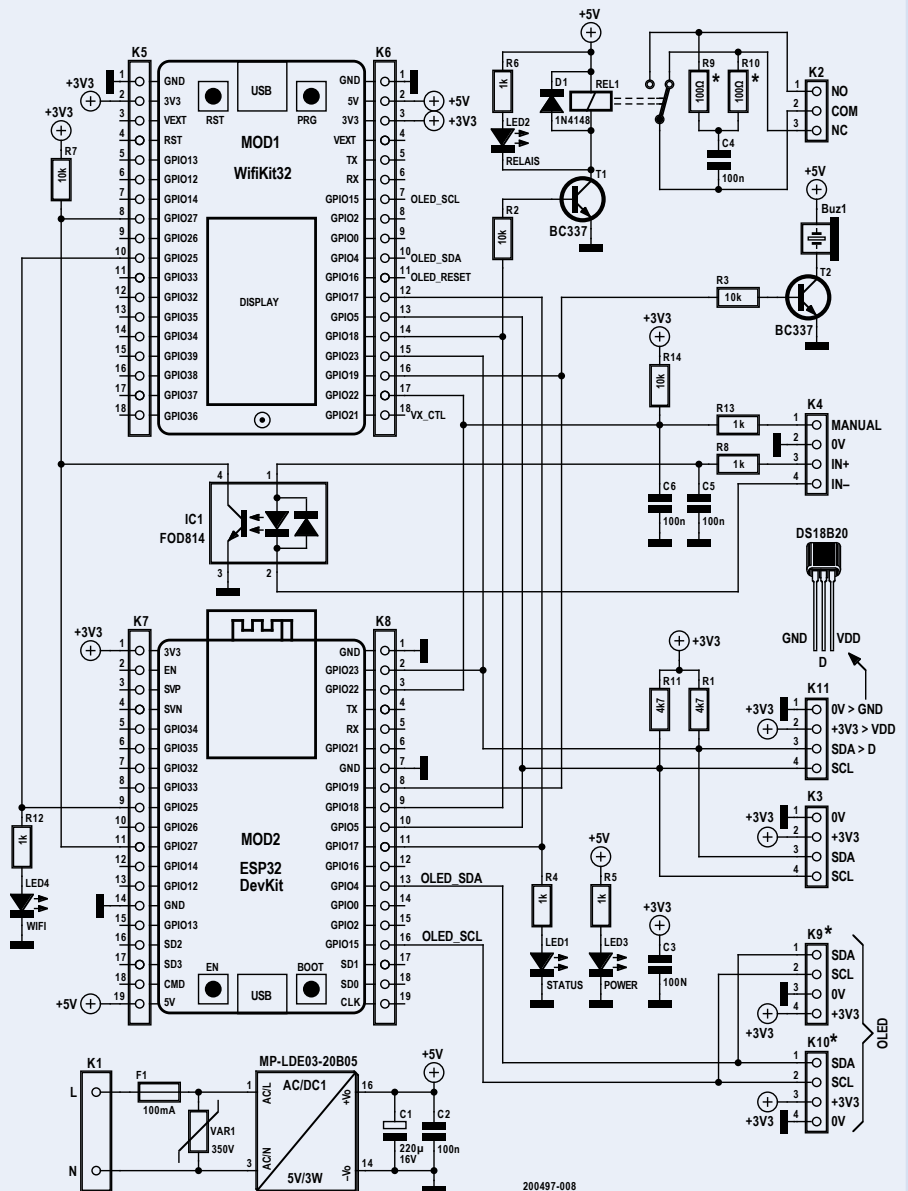
Relais

De belasting van de thermostaat wordt geschakeld door relais RE1, dat op zijn beurt wordt aangestuurd door een kleine NPN-transistor (T1) die is aangesloten op poortpin IO18. LED2 geeft de toestand van het relais aan. Diode D1 beschermt de elektronica tegen door het relais geïnduceerde schakelstromen, terwijl het snubbernetwerk C4/R9 of C4/R10 (monteer slechts één weerstand) de relaiscontacten beschermt door eventuele schakelvonken in geval van een zware belasting te dempen.

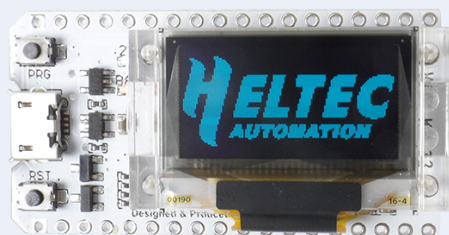
Het gebruikte relais kan 250 VAC belastingen aan tot 10 A. Zowel de normaal-open (NO) als normaal-gesloten (NC) contacten zijn beschikbaar op K2. Merk op dat K1 en K2 kunnen worden gecombineerd tot één 5-weg klemblok.

Zoemer

Een buzzer met ingebouwde oscillator (Buz1), aangestuurd door T2 via poortpin IO19 is beschikbaar voor hoorbare feedback. Omdat deze een ingebouwde oscillator heeft, hoeft



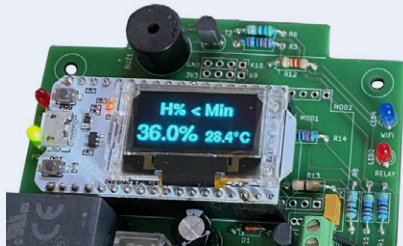
Figuur 1: Het schema van de ESP32-thermostaat toont veel aansluitingen omdat er verschillende configuratiemogelijkheden zijn.



Figuur 2: De WiFi Kit 32 van Heltec Automation.



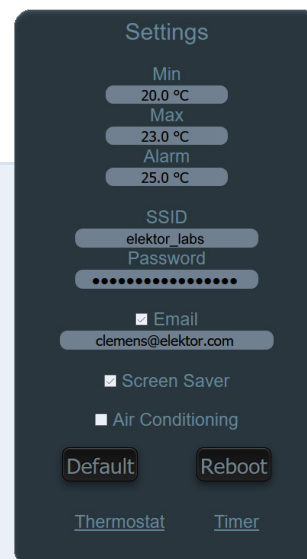
Figuur 3: De door Elektor Labs gebouwde thermostaat maakt gebruik van een ESP32 DevKitC-module met een afzonderlijke OLED-displaymodule.



Figuur 4: De thermostaat omgeprogrammeerd als vochtigheidsregelaar, nu uitgerust met een Heltec WiFi Kit 32 module met on-board OLED display.



Figuur 5: De hoofdpagina toont de actuele temperatuur samen met de schakel- en alarmdrempels.



Figuur 6: De minimum- en maximumtemperatuur en de alarmdrempel worden hier ingesteld. Dit is ook de pagina waar u de SSID en het wachtwoord van het WiFi-netwerk bepaalt dat de thermostaat moet gebruiken.

alleen met DC aangestuurd te worden om hem te laten piepen. Een zogenaamde AC-zoemer kan ook worden gebruikt, maar dan zal de ESP32 het geluid zelf moeten genereren.

Status-LED

LED1, verbonden met IO17, is voor statusindicatie. Snel knipperen duidt op een fout als gevolg van slechte of ongeldige EEPROM-gegevens of communicatieproblemen van de DS18B20-sensor. Deze fouten zijn blokkerend, en het programma stopt tot het probleem is opgelost. Korte flitsen tijdens de normale werkingsmodus geven aan dat de geïntegreerde webserver wordt benaderd.

Externe bediening

Klemblok K4 maakt de aansluiting mogelijk van een externe schakelaar (*Manual*) tussen poortpin IO22 en GND. Met de standaard software zal deze schakelaar de thermostaat overrulen en het relais activeren. Een RC peak/glitch/noise/debounce netwerk (1,5 kHz met de gegeven componentwaarden) kan op deze ingang worden gemonteerd. Wanneer een bi-stabiele schakelaar wordt gebruikt, kan C6 worden weggelaten en kan R13 worden vervangen door een draadbrug.

Een optisch geïsoleerde digitale ingang *IN* (optocoupler IC1) verbonden met poortpin IO2 is beschikbaar voor toekomstige uitbreiding en wordt door de huidige software niet gebruikt. Ik heb hem gebruikt in het eerder genoemde badkamervoorbeeld om

de ingeschakelde verlichting te detecteren (figuur 4).

Sensoraansluiting en I2C-poort

De temperatuursensor, een OneWire DS18B20, moet worden aangesloten op K3. Pen 1 gaat naar GND, zijn DQ-pen (pen 2) wordt verbonden met pen 3 van K3, en de VDD-pen met 3V3. Dit betekent dat de pootjes 2 en 3 elkaar kruisen. Voor typische thermostaattoepassingen mag de temperatuursensor echter niet in de behuizing worden gemonteerd, aangezien de lucht binnenin wordt verwarmd door de voeding en de ESP32-module, waardoor temperatuurregeling onmogelijk wordt.

R1 is nodig voor de goede werking van de DS18B20. Als men in plaats van deze temperatuursensor kiest voor een vochtigheidssensor (zoals een BME280, SHT11 of DHT22) is R1 wellicht niet nodig.

Eigenlijk is K3 zo bedraad dat het compati-

bel is met de facto I2C-bus maker standaarden zoals Grove (Seeed Studio) en Qwiic (SparkFun). In feite is K11 een Grove connector (2 mm pitch). Er zijn veel uitbreidingsprinten beschikbaar die compatibel zijn met dit connectorformaat, waardoor de thermostaat-schakeling een uitstekende basis vormt voor andere toepassingen.

Print

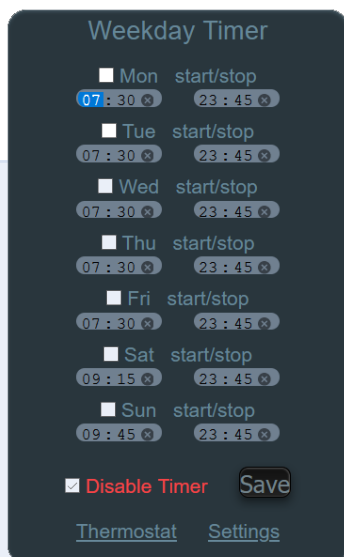
Elektor heeft een print ontwikkeld die past in een goedkope behuizing met transparant deksel. De sporen van het relais die met de belasting in verbinding staan, kunnen en moeten met soldeer worden versterkt.

Software en programmering

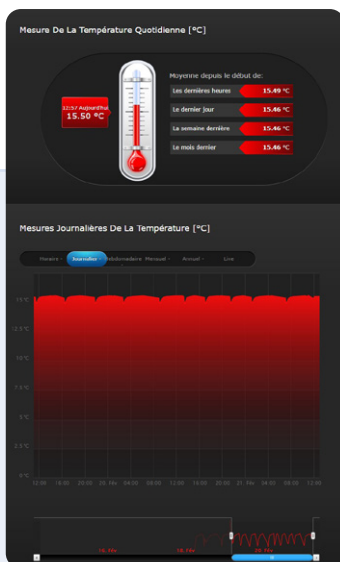
Het programma voor de thermostaat is geschreven in C/C++ in de Arduino IDE (ik heb 1.8.12 gebruikt). Het is vrij groot -meer dan 1.500 regels code- maar het is uitgebreid commentarieerd (zij het in een mengeling van Engels en Frans). Het is vrij eenvoudig aan te passen aan je eigen behoeften en ik heb alleen vrij beschikbare en bewezen open-source bibliotheken gebruikt.

Voordat je de ESP32 met de Arduino IDE kunt gebruiken, moet je eerst het bijbehorende Boards Package installeren. Er zijn vele webpagina's geschreven om deze procedure uit te leggen, zie [2] voor details (met video). Het officiële ESP32 Boards Package bevat ondersteuning voor zowel de Heltec WiFi Kit 32 als de DevKitC ('ESP32 Dev Module').





Figuur 7: Op de pagina Timer kun je voor elke dag van de week een tijdsperiode definiëren gedurende welke de thermostaat de temperatuur moet regelen.



Figuur 8: De regeling is vrij goed, zoals blijkt uit deze schermafbeelding van mijn domotica-systeem. Een variatie van 1°C in de kelder komt in feite overeen met een variatie van slechts 0,1°C van de temperatuur van de wijn in de fles.



GERELATEERDE PRODUCTEN

- **ESP32 DevKitC (SKU 18701)**
www.elektor.nl/18701
- **0.96" 128x64 I²C OLED display (SKU 18747)**
www.elektor.nl/18747
- **Elektor ESP32 Smart Kit Bundle (SKU 19033)**
www.elektor.nl/19033

gangspunt (AP) met SSID 'Thermostat ESP32'; er is geen wachtwoord. Open een browser en ga naar het IP adres 192.168.4.1. De webinterface zou nu moeten verschijnen. Klik op de link **Settings** en vervolgens op de knop **Default**. Dit zal alle parameters -inclusief de timer- en temperatuurinstellingen- initialiseren met hun standaardwaarden en de SSID en passphrase activeren zodat de thermostaat verbinding kan maken met je WiFi-netwerk. Herstart het apparaat om de parameters actief te maken. Met de knop **Reboot** start je een zachte reset van de thermostaat.

Merk op dat wanneer de SSID en/of het wachtwoord wordt gewijzigd, het systeem automatisch opnieuw opstart.

Wat de timer- en temperatuurinstellingen betreft, is het niet nodig een veld te valideren nadat de waarde ervan is gewijzigd. De muis uit het veld bewegen is voldoende. Er is ook een sanity-check op de minimum- en maximumtemperatuurdrempels. Als de maximumwaarde lager is dan de minimumwaarde, worden de twee omgewisseld.

Kleurrijke gebruikersinterface

Er is veel aandacht besteed aan het ontwerp van de gebruikersinterface. Het doel was om een nette webinterface te maken die correct wordt weergegeven op de meest gangbare en recente browsers en op iPhone en Android smartphones. De interface is zeer responsief zonder merkbare vertraging en alle waarden worden elke seconde ververs.

De web-based gebruikersinterface heeft drie pagina's: Main (**Figuur 5**), Settings (**Figuur 6**) en Timer (**Figuur 7**).

Op de hoofdpagina worden kleurcodes gebruikt om de parameterstatus aan te geven. De WiFi-siginaalsterkte (RSSI) wordt bijvoor-

Welke module je ook gebruikt, je kunt alle andere instellingen (CPU-snelheid, grootte van het flash-geheugen, enz.) op hun standaardwaarden laten staan. Zorg er echter wel voor dat je de juiste COM-poort selecteert.

Naast het installeren van het ESP32 Board-package, moet ook de SPIFFS *ESP32 Sketch Data Upload*-tool worden geïnstalleerd. Raadpleeg ook hier [2] voor details over hoe je dit moet doen.

Download de thermostaat-sketch van [3] en pak deze uit in de sketch-map van de Arduino IDE. Het bestand *Thermostat.ino* en de bijbehorende bestanden en gegevensmap moeten zich bevinden in een map met de naam *Thermostat*. De submap *data* bevat de HTML- en CSS-bestanden en de JavaScript-programma's die nodig zijn voor de webserver. Het bestand *D7MR.woff2* bevat het 7-segment lettertype dat ik heb gebruikt. Een grote "Dank u!" aan de auteur daarvan. Voor het gemak heb ik alle bibliotheken die nodig zijn voor het project opgenomen in de download [3]. Ze moeten correct worden geïnstalleerd, dat wil zeggen door ze te kopiëren naar de submap *libraries* van de sketch-map van je IDE. Waar die op je computer staat, hangt af van hoe je de Arduino IDE hebt ingesteld. Zie [4] voor details over de Arduino IDE.

Zodra alle software is gedownload en geïnstalleerd, moet je enkele standaardparameters van het programma aanpassen aan je thuis- of toepassingsomgeving. Deze bevinden zich

bovenaan het programma. Aangezien de thermostaat een vast IP-adres gebruikt, moet je er een definiëren samen met de gateway en andere typische netwerkzaken.

E-mail configuratie

Omdat de thermostaat e-mails kan versturen, is het nodig om de e-mail-servergegevens in te voeren. Ik raad aan om hiervoor een Gmail-account aan te maken. Als je dat doet, is de e-mailserver *smtp.gmail.com* en de te gebruiken poort 465. Om veiligheidsredenen moet het Gmail-account enigszins worden geconfigureerd. Als je account één-factor authenticatie gebruikt (d.w.z. dat alleen een wachtwoord nodig is), moet je het account configureren met link [5]. In het geval van twee-factor authenticatie (d.w.z. wachtwoord plus code per SMS), raadpleeg dan [6].

Nu is het tijd om de sketch te compileren en te uploaden zoals elke andere Arduino-sketch. Dit zou geen problemen mogen opleveren. Indien wel, controleer dan je installatie.

Nadat je de sketch hebt geprogrammeerd, upload je de webpagina's en andere gegevensbestanden met het *ESP32 Sketch Data Upload*-commando, dat beschikbaar is via het Tools-menu van de IDE. Hierdoor wordt alles in de datamap overgebracht naar het flashgeheugen (SPIFFS) van de ESP32.

Eerste boot

Als je het apparaat voor de eerste keer opstart, maak je verbinding met het eigen WiFi-toe-

beeld weergegeven in groen, oranje of rood, afhankelijk van de signaalkwaliteit. De tijd, verkregen van een NTP-server ergens op het internet, wordt in het rood weergegeven als de timer het relais aanstuurt; anders is de tijd groen. Evenzo wordt de temperatuur lichtblauw weergegeven als de temperatuur onder de minimumwaarde ligt; hij is groen als de temperatuur tussen minimum en maximum ligt. Als de temperatuur boven de maximumwaarde ligt, is hij rood. Als het oranje begint te knippen, heeft het systeem een probleem ondervonden.

Een klein LED-symbool wordt op de hoofdpagina getoond om in real time de status van het relais te tonen (groen voor Uit, rood voor Aan). De CSS-stylesheet vergde veel werk om deze kleurrijke weergave mogelijk te maken. Het OLED-display geeft ook de status van het systeem aan. Aangezien is gebleken dat deze displays de neiging hebben vrij snel te verouderen wanneer ze permanent aan staan, is een screensaver-optie toegevoegd. Door op een drukknop te drukken die is verbonden tussen GPIO05 (K11, pin 4) en GND (K11, pin 1) wordt het display gedurende enkele minuten ingeschakeld. Deze functie kan worden geactiveerd of gedeactiveerd op de Settings-pagina.

Timer pagina

De Aan- en Uitschakelmomenten voor elke dag, evenals het bijbehorende activeringsvinkje, moeten worden opgeslagen met de knop **Save**. Door hierop te klikken wordt de hele pagina opgeslagen. Het selectievakje *Timer disabled* hoeft niet te worden gevalideerd met de knop **Save**. Als de timer is uitgeschakeld, regelt de thermostaat de temperatuur continu.

Debugmodus

Door de seriële poort via de ingebouwde USB-interface (115.200 baud) te monitoren, kan men het initialisatieproces van de thermostaat na het inschakelen observeren. Dit is zeer nuttig om zaken als WiFi-verbindingsproblemen op te sporen.

Een betrouwbaar systeem

De in dit artikel beschreven thermostaat regelt nu al meer dan een jaar probleemloos de temperatuur in mijn kelder. Het systeem is betrouwbaar, en de regeling is zeer goed



Figuur 9: De DS18B20 sensor werd ingebouwd in een apart klein plastic doosje dat op een strategische plaats in de kelder werd bevestigd. Een tweede, identieke sensor rapporteert de temperatuur aan mijn domoticasysteem via een speciale module. Dit is een extra veiligheidsmaatregel voor het geval de thermostaat het zou laten afweten. Als dit gebeurt, zal mijn domotica-systeem mij waarschuwen.

(Figuur 8). Ik heb de webinterface kunnen testen op de meeste internetbrowsers (Chrome, Edge, Safari, enz.) en ook op iPhone en enkele Android-telefoons, en ik heb geen compatibiliteitsproblemen vastgesteld.

Ik wil Stephane Calderoni (PhD, Computerwetenschappen) bedanken die op Réunion woont (een eiland in Oost-Afrika) en die me veel heeft geholpen met de ontwikkeling van de HTML-interface en de bijbehorende JavaScript. Ik heb gebruik gemaakt van de lockdown-periode van maart 2020 in Frankrijk om mijn kennis over dit onderdeel, dat ik nog helemaal niet onder de knie had, te verbeteren. ◀

Een bijdrage van:

Idee, ontwerp & tekst: **Yves Bourdon**

Print-ontwerp en -bewerking:

Clemens Valens

Lay-out: **Harmen Heida**

Vertaling: **Jelle Aarnoudse**

Vragen of opmerkingen?

Heb je technische vragen of opmerkingen over dit artikel? Stuur een e-mail naar de redacteur via clemens.valens@elektor.com.

200497-03



ONDERDELENLIJST

Weerstanden (5%, 0,25W)

R9, R10 = 100 Ω , 1 W (zie tekst) R4, R5, R6, R8, R12, R13 = 1 k
R1, R11 = 4k7
R2, R3, R7, R14 = 10 k

Condensatoren

C2, C3, C5, C6 = 100 n, steek 2,5 of 5 mm
C4 = 100 n, X2, steek 15 mm
C1 = 220 μ , 16 V, steek 3,5 mm

Halfgeleiders

D1 = 1N4148
LED1, LED2 = LED, rood, 3 mm
LED3, LED4 = LED, groen, 3 mm
IC1 = FOD814, DIP4
T1, T2 = BC337

Diversen

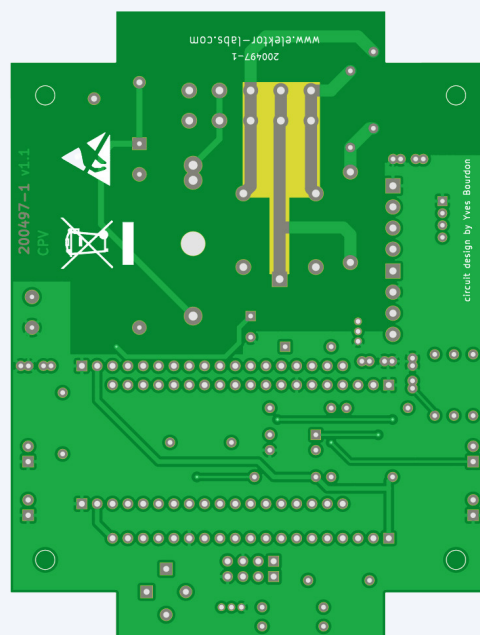
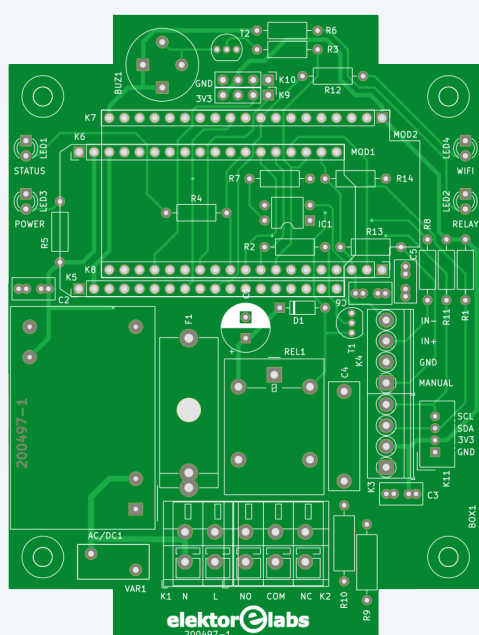
BUZ1 = Zoemer met oscillator, steek 6,5 of 7,6 mm
F1 = Zekeringhouder, 5x20 mm + zekering 250 V, 100 mA, traag
K1 = 2-weg klemmenblok, steek 5 mm
K2 = 3-weg klemmenblok, steek 5 mm
K3, K4 = 4-weg klemmenblok, steek 3,5 mm
K5, K6 = 18-weg pennenblok, steek 0,1" (alleen MOD1)
K7, K8 = 19-voudige pennenbus, steek 0,1" (alleen MOD2)
K9, K10 = 4-voudige pennenbus, steek 0,1", 20 mm hoog (alleen MOD2)
K11 = 4-pins header, steek 2 mm (Grove)
REL1 = Relais, 10 A, SPDT, 5VDC
VAR1 = Varistor 350 V

AC/DC1 = 5 V, 3 W, AC-DC-omvormer (b.v. Multicomp MP-LDE03-20B05)

BOX1 = Behuizing, IP65, transparant deksel (Multicomp MC001106)

MOD1 = Heltec WiFi Kit 32
of

MOD2 = ESP32 DevKitC + OLED display, I2C, 128x64 pixels



WEBLINKS

- [1] R. Aarts & C. Valens, "WiFi desktop-thermostaat," Elektor, Jan/Feb 2018: www.elektormagazine.nl/magazine/elektor-201801/41242
- [2] ESP32 – Getting Started: www.elektormagazine.com/labs/1874
- [3] ESP32 Thermostat at Elektor Labs: www.elektormagazine.com/labs/4216
- [4] Arduino FAQ: www.elektormagazine.com/labs/1876
- [5] Gmail one-factor authentication: <https://myaccount.google.com/lesssecureapps>
- [6] Gmail two-factor authentication: <https://security.google.com/settings/security/apppasswords>

