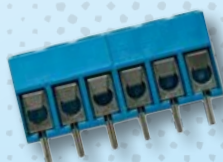
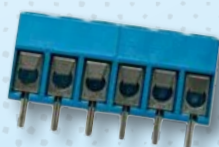
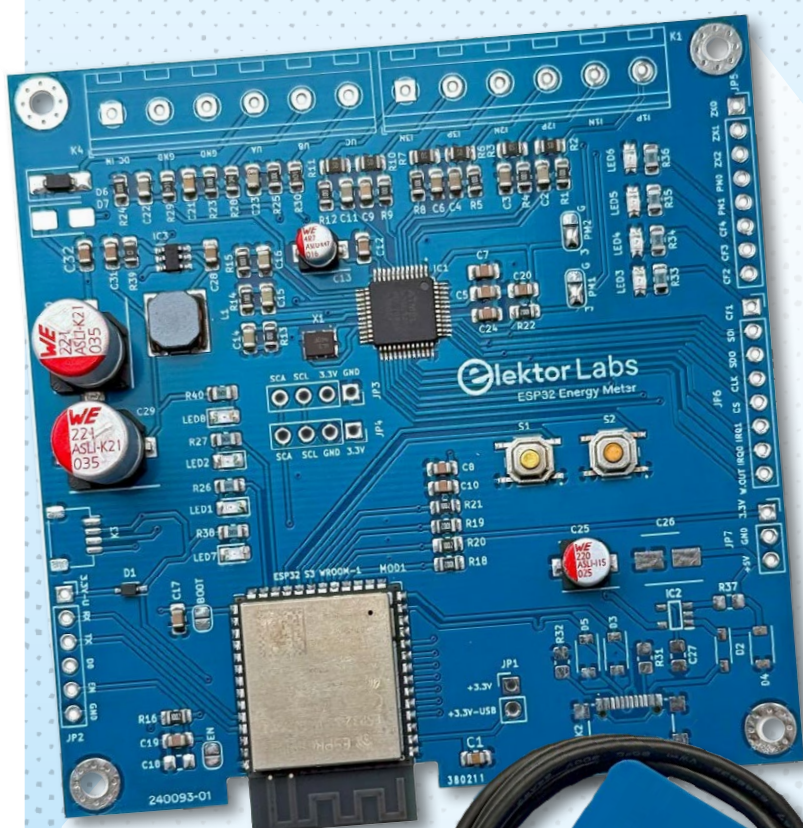




Elektor ESP32 Energymeter



IN DIT
NUMMER



Prototype van een ESP32-gebaseerde energiemeter



Figuur 1. Zo zou onze enkelfasige energiemeter eruit kunnen zien.

Saad Imtiaz (Elektor)

Dit artikel beschrijft de ontwikkeling van een energiemeter met behulp van een Espressif ESP32, waarbij de nadruk ligt op het observeren van het energieverbruik in real-time en op de veiligheid. Het belicht de eerste stappen, voorwaarden en overwegingen die plaatsvinden wanneer je aan een embedded project begint.

Naarmate het project vordert, zullen de resultaten worden gedeeld in de komende uitgaven van Elektor.

Op het gebied van techniek kan het combineren van de juiste technologieën tot aanzienlijke vooruitgang leiden. Dit project heeft als doel een energiemeter te ontwikkelen met behulp van de Espressif ESP32 microcontroller en Microchip's energiemeter-IC ATM90E32AS. In dit artikel wordt het beginstadium van dit project kort beschreven, van componentkeuze tot prototyping. Het doel is eenvoudig: het maken van een betrouwbaar systeem voor nauwkeurige energiemeting vanuit de groepenkast van je huis of werkplaats. Met deze meter kunnen gebruikers het stroomverbruik in realtime volgen en een inzicht krijgen dat kan leiden tot een efficiënter energiegebruik.

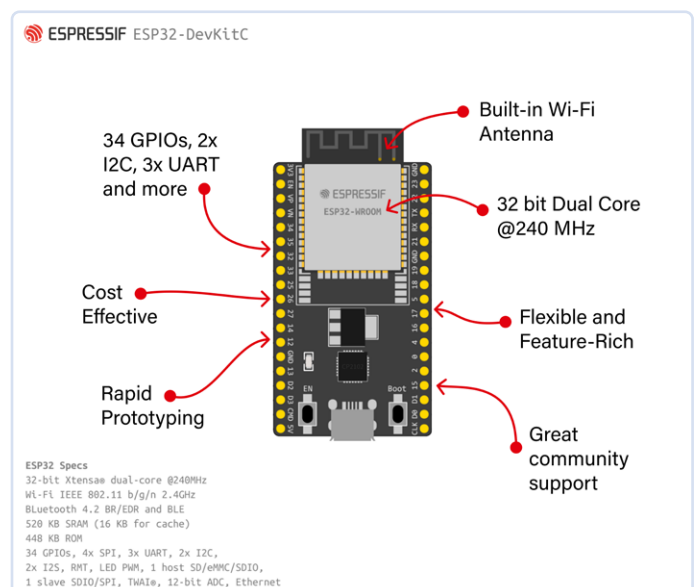
Ontwerp en voorwaarden

Het project heeft duidelijke doelen en ontwerpvoorwaarden: real-time monitoren van enkelfasig vermogen met behulp van drie stroomtransformatoren (CT's), het betaalbaar houden en het gebruiksvriendelijk maken. De keuze voor de ESP32 en ATM90E32AS IC-componenten werd bepaald door deze doelstellingen, die zowel een goede prijs-kwaliteitverhouding als betrouwbare prestaties bieden. Een ander doel was om de afmetingen kleiner dan 100×80×30 mm (L×B×H) te houden, zodat het in een groepenkast past. Om de gebruikerservaring te verbeteren, is er ook een mobiele interface voor monitoring op afstand en een OLED-scherm met knoppen voor directe bediening. Het ontwerp maakt ook toekomstige software-updates mogelijk, zodat de consument verzekerd is van langdurig gebruik. In **figuur 1** is een gerenderde weergave van het huidige prototype van de behuizing te zien.

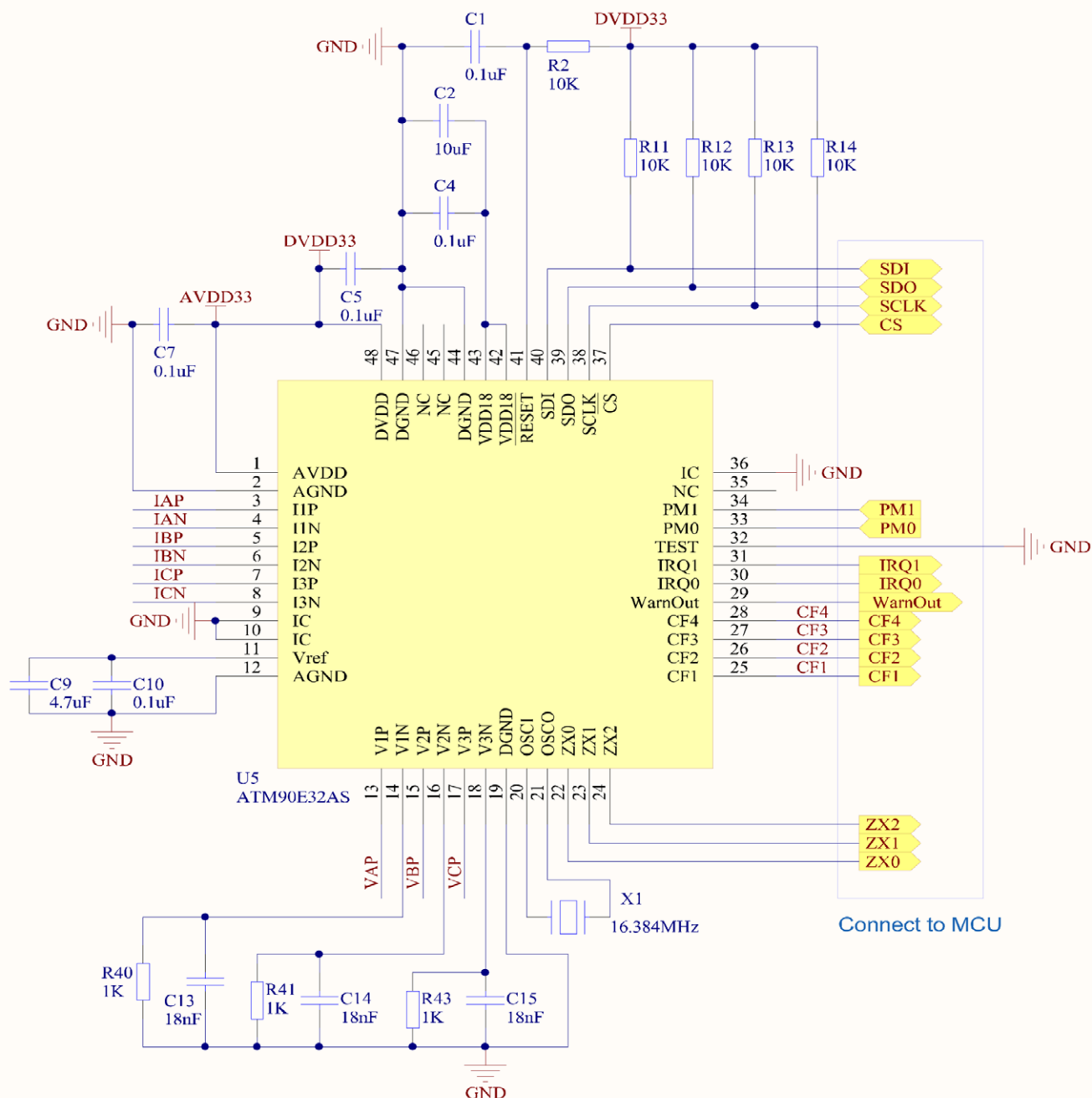
Keuze microcontroller

De keuze voor de ESP32-microcontroller was gebaseerd op een gedetailleerde evaluatie van zijn functies. De chip blinkt uit op verschillende gebieden die cruciaal zijn voor het succes van dit project. Ten

eerste het gemak waarmee hij in verschillende ontwerpen kan worden geïntegreerd; dat biedt flexibiliteit tijdens de ontwikkelingsfase. Ten tweede maakt de kosteneffectiviteit hem een aantrekkelijke keuze voor een prototype dat prestaties en kosten in evenwicht brengt. Ten derde biedt de compatibiliteit met een breed scala aan sensoren en IC's aanzienlijke voordelen. Tot slot vergroot de uitgebreide ondersteuning van de ESP32 chip door de community zijn geschiktheid voor dit project. **Figuur 2** laat de belangrijkste kenmerken en voordelen van de ESP32-D0WD-V3 zien, waardoor deze voor dit project is geselecteerd.



Figuur 2. Belangrijkste kenmerken en voordelen van de ESP32.



Figuur 3. De energiemeter is gebaseerd op een application note van Atmel [2]. Hier zie je de schakelingen rond het energiemeter-IC.

Integratie van het energiemeter-IC

Het ATM90E32AS-IC van Microchip werd geïntegreerd volgens een application note van de fabrikant; het document diende als leidraad om ervoor te zorgen dat het energiemeter-IC naadloos communiceerde met de ESP32-microcontroller. Deze fase was echter niet helemaal vrij van hindernissen. De aanschaf van de juiste componenten binnen de budgetbeperkingen vereiste een nauwgezette planning, omdat de beschikbaarheid beperkt was. In **figuur 3** wordt de application note van Atmel (nu Microchip) getoond.

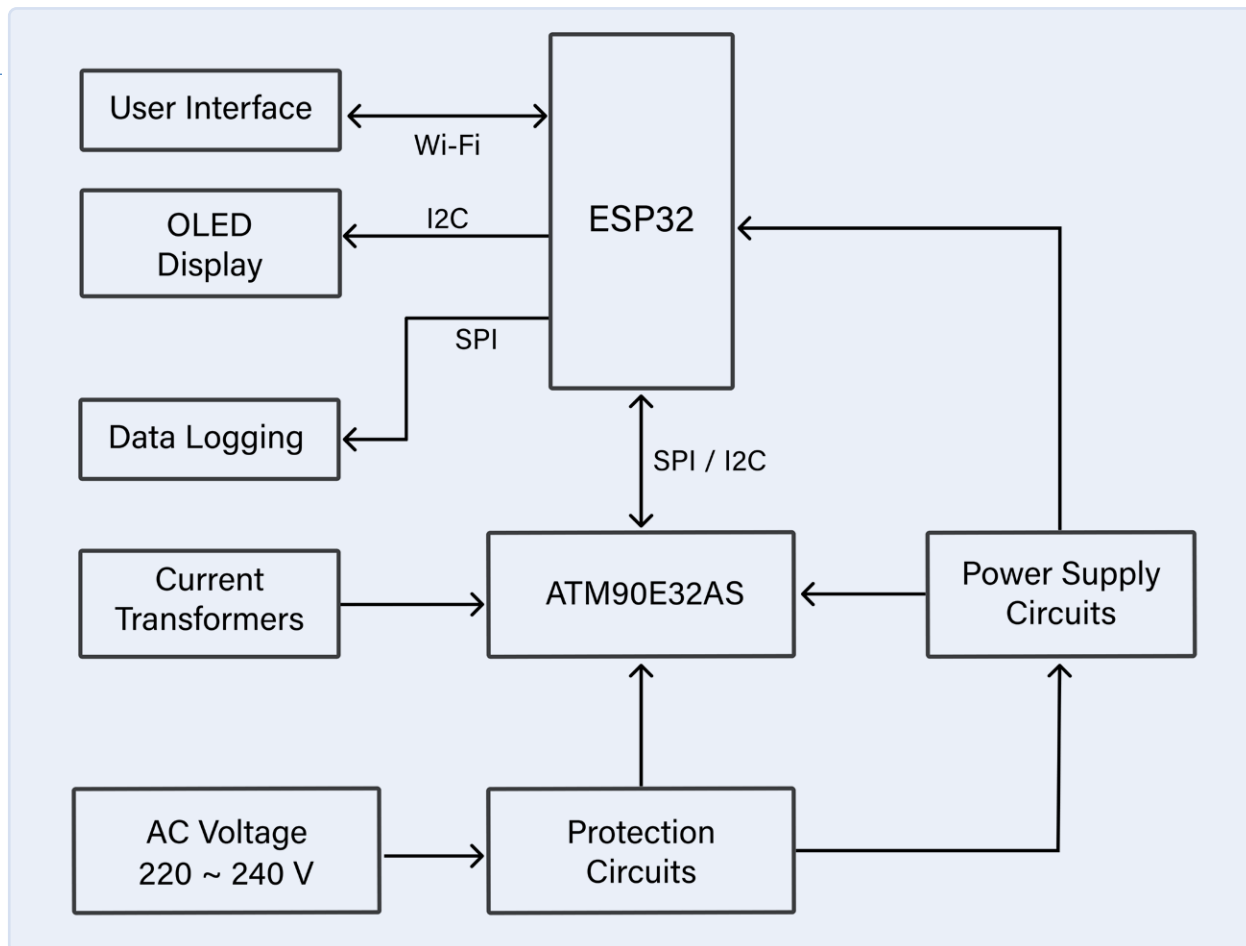
Ontwerpfase en elektrische veiligheidsnormen

De ontwerpfase is inderdaad een cruciaal onderdeel van het ontwerp-proces, vooral wanneer veiligheid een belangrijk aandachtspunt is. Bij een apparaat dat ontworpen is om te werken met de lichtnet-wissel-

spanning moet er nauwgezet op worden toegezien dat het voldoet aan de vastgestelde veiligheidsnormen. In **figuur 4** wordt het blokschema van het project getoond.

Om de veiligheid te garanderen werden verschillende speciale elektrische componenten in het ontwerp opgenomen. Metaaloxide-varistoren (MOV's) werden gebruikt om het circuit te beschermen tegen kortstondige spanningspieken. Verder werden er zekeringen ingebouwd als essentiële beveiliging tegen te hoge stromen.

Behalve op de keuze van componenten richtte het ontwerp zich ook op printlayout-overwegingen die zouden voldoen aan de veiligheidsnormen. Er werden voldoende kruip- en onderlinge afstanden aangehouden tussen de geleidende elementen op de printplaat om overslag te voorkomen. De spoorbreedte voor AC-spanningslijnen werd zorgvuldig berekend om de verwachte stromen aan te kunnen, volgens de



Figuur 4. Blokschema van ons energiemeterproject.

IPC-2221 normen [1]. Dit was cruciaal om de thermische prestaties van de print onder maximale belasting te kunnen waarborgen. Om de integriteit van de aarding te garanderen, werd een degelijk aardvlak gebruikt. Er werd speciale aandacht besteed aan het ontwerp van differentiële paren voor een goede signaalintegriteit, waarbij ervoor werd gezorgd dat de routing een nauwkeurige geometrie volgde om elektromagnetische interferentie tot een minimum te beperken.

Fabricage: we kozen voor JLC PCB

Na het bestuderen van verschillende bedrijven voor PCB-assemblage werd JLC PCB geselecteerd. De belangrijkste reden voor deze keuze was de balans tussen kosteneffectiviteit en betrouwbaarheid die zij bieden. Deze beslissing was belangrijk om het project binnen het budget te houden zonder afbreuk te doen aan de kwaliteit van de geassembleerde print. Op dit moment wordt de laatste hand gelegd aan het prototype-schema en de printlayout, en deze zullen binnenkort worden aangeboden voor productie.

Terugblik en vooruitzicht

Achteraf gezien toont dit project wat er bereikt kan worden als zorgvuldige planning en goede techniek hand in hand gaan. De hindernissen die we tegenkwamen hebben ons geholpen ons ontwerp te verbeteren. Nu we overgaan van het maken van een prototype naar de mogelijke massaproductie ervan, verwachten we dat het echt een verschil zal maken in de manier waarop mensen omgaan met energie. Dit project zal in de komende edities van dit tijdschrift uitgebreid aan bod komen – we zijn nog steeds bezig met het prototype, dit te testen en de software te ontwikkelen waarmee het zal werken. Er komt nog meer, dus blijf op de hoogte van updates over dit project. We zullen de voltooiing en

laatste updates delen in het januari/februari 2024 nummer van Elektor, met de focus op voeding en energie.

vertaling: Hans Adams — 230646-03

Vragen of opmerkingen?

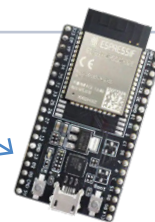
Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de redactie van Elektor via redactie@elektor.com.



Gerelateerde producten

> **ESP32-DevKitC-32E** www.elektor.nl/20518

> **ESP32-C3-DevKitM-1** www.elektor.nl/20324

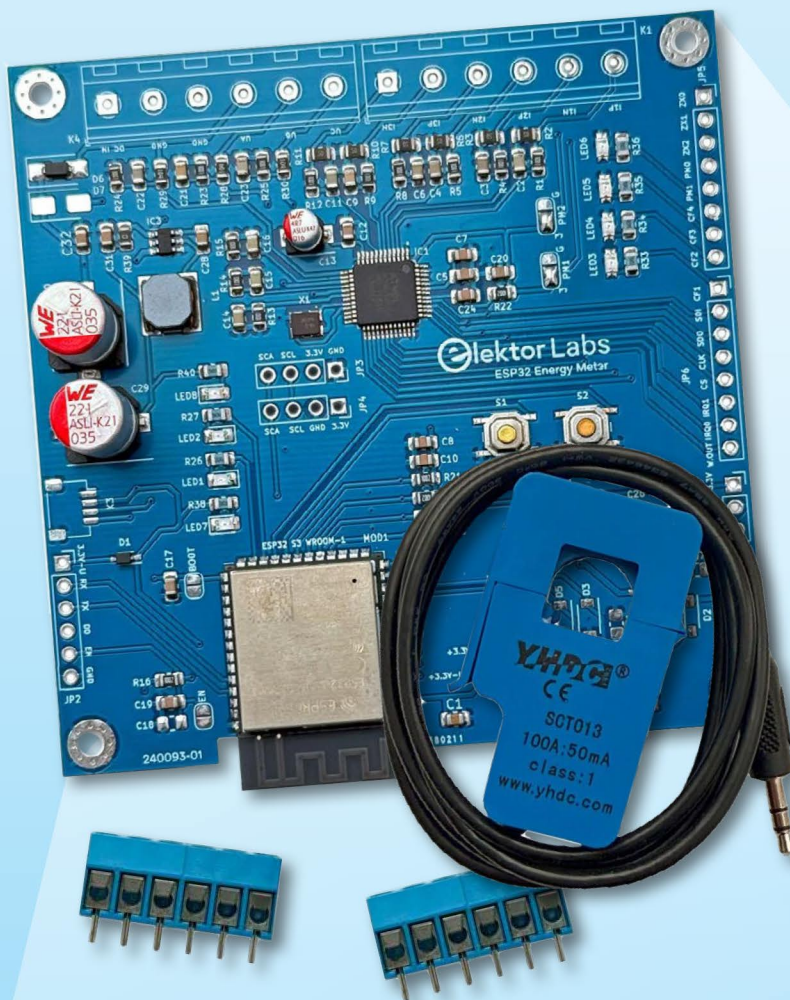


WEBLINKS

- [1] IPC-2221A standaarden:
[https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A\(L\).pdf](https://www-eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf)
- [2] ATM90E32AS application note:
<https://ww1.microchip.com/downloads/en/Appnotes/Atmel-46103-SE-M90E32AS-ApplicationNote.pdf>

NIEUW

Elektor ESP32 Energymeter



- > SMD-voorgemonteerde printplaat
- > Stroomtransformator inbegrepen
- > Compatibel met ESPHome, Home Assistant en MQTT
- > Gerelateerde Elektor-artikelen zijn gratis te downloaden

Bestel nu:

www.elektor.nl/21044



Figuur 1. De Elektor-energiemeter kan er zo uit zien.

Project-update: ESP32- gebaseerde energiemeter

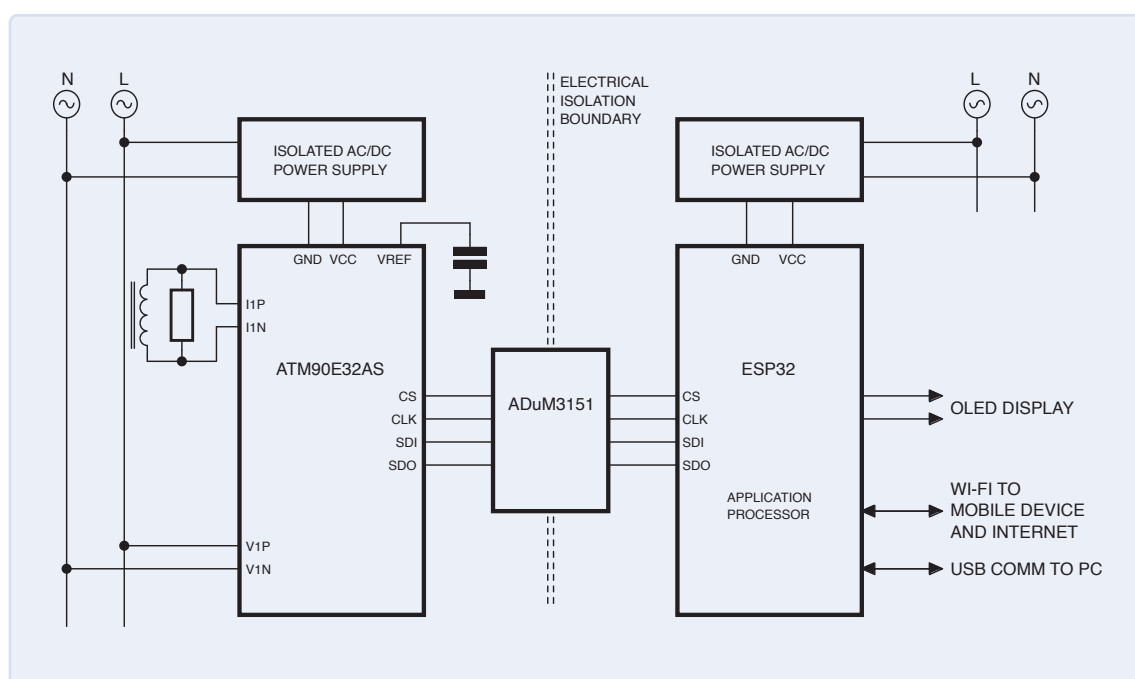
we gaan verder met het prototype

Saad Intiaz (Elektor)

In de eerste aflevering van deze serie hebben we het basisonwerp van de Elektor-energiemeter onderzocht. In deze aflevering kijken we naar de volgende fase in het energiemeterproject met de ESP32, waarbij we ons richten op gedetailleerde schema's, isolatiestrategieën en belangrijke verbeteringen.

We zijn begonnen met de ontwikkeling van een betrouwbare, gebruiksvriendelijke energiemeter met behulp van de ESP32-microcontroller. In onze vorige aflevering, "Prototype van een ESP32-gebaseerde energiemeter" [1], bespraken we de uitgangspunten voor het ontwerp, het blokschema en het plan om dit project te starten. Voordat we een update geven, even een samenvatting. Het conceptuele ontwerp van de energiemeter wordt getoond in de rendering in **figuur 1**.

Onze focus lag op real-time energiemonitoring, met de nadruk op veiligheid en betaalbaarheid. Om de energiemeting nauwkeurig te maken, kozen we voor de Atmel ATM90E32AS, een meerfasig energiemonitoring-IC [2]. Dit IC krijgt de enkelfasige spanning van het lichtnet en gebruikt CT-transformatoren met deelbare kern om de stroom veilig te meten. Als belangrijkste microcontroller werd de ESP32 gekozen, omdat deze ingebouwde WiFi heeft en zeer voordelig is in vergelijking met



Figuur 2. Blokschema van de Elektor-energiemeter.



Figuur 4. De YHDC SCT013 (bron: YHDC).

andere MCU's. In **figuur 2** wordt het bijgewerkte blok van het project getoond. De geplande grootte van de uiteindelijke energiemeter is 100×80×30 mm (L×B×H), maar voor het prototype meet onze print 100×100 mm. Het doel van dit prototype is een proof-of-concept en als we daarin succesvol zijn, schalen we de afmetingen terug naar 100×80 mm of zelfs minder voor de uiteindelijke versie. Het belangrijkste doel van het maken van deze energiemeter was de ontwikkeling van een betaalbaar apparaat met IoT-ondersteuning dat nauwkeurige energiemetingen kan doen en de gebruiker via een mobiel apparaat realtime-energiegegevens kan leveren, zodat de gebruiker zijn energieverbruik in realtime kan volgen en energiezuiniger kan worden. In dit artikel gaan we dieper in op de ontwikkeling van het project, met aandacht voor het schema-ontwerp, de implementatie van de isolatie van de schakeling en de belangrijkste verbeteringen die we hebben aangebracht sinds ons eerste concept.

Ontwerp van het schema

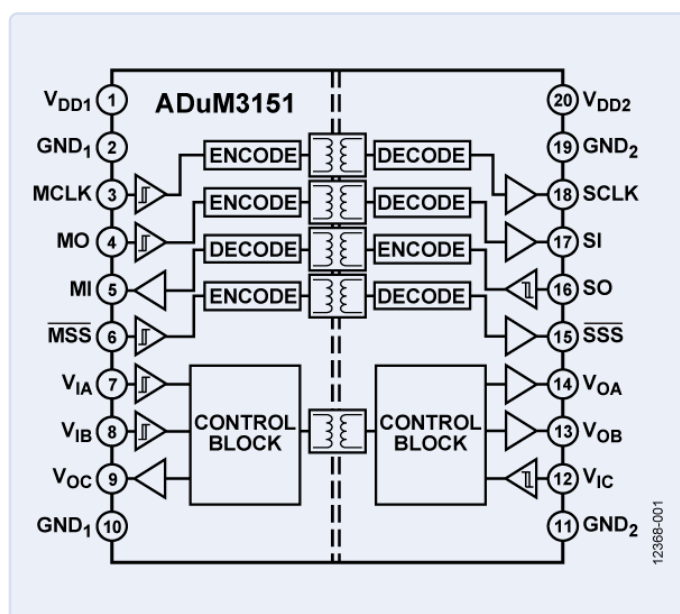
Het hart van ons project is het ontwerp van het schema. De ESP32-microcontroller blijft centraal staan in onze architectuur en sluit naadloos aan op de ATM90E32AS voor de energiemeting. Ons bijgewerkte schema weerspiegelt een meer gestroomlijnde aanpak, waarbij ruis wordt verminderd en signaalintegriteit, isolatie en meer worden verbeterd. In **figuur 3** zie je het volledige schema van het project. IC1 is de ATM90E32AS – het brein van dit hele project. Deze koppelt de netspanning via zeven 240kΩ-weerstanden (R1...R7) naar de pinnen VIP, V2P en V3P. Om het eenvoudig te houden, krijgen al deze pinnen een enkelfasige spanning van het lichtnet. Je kunt je afvragen waarom we geen transformator gebruiken in plaats van deze reeks weerstanden? Omdat we door de gekozen aanpak beperkingen hebben wat betreft afmetingen en kosten. Afgezien van het kleine formaat van de gebruikte weerstanden is er nog een voordeel, namelijk minder fasevertraging. Transformatoren kunnen een fasevertraging introduceren tussen de primaire en secundaire wikkelingen, wat invloed kan hebben op de timing en nauwkeurigheid van spanningsmetingen bij het meten van energie. Bij gebruik van weerstanden wordt deze fasevertraging aanzienlijk verminderd, wat mogelijk leidt tot nauwkeurigere realtime-spanningsmetingen. Maar het gebruik van deze serie-weerstanden heeft een groot nadeel, namelijk het ontbreken van galvanische scheiding. We zullen het hier later in het artikel over hebben. Nu gaan we over naar de stroommeting. Daarvoor gebruiken we

stroomtransformatoren (CT's). Connectoren K1...K3 zijn audio jack-connectoren waarop de SCT013 van YHDC wordt aangesloten, een CT met een deelbare kern, zie **figuur 4**. De reden om voor een CT te kiezen is dat deze voordelig en gebruiksvriendelijk is en geen verbindingen hoeven te worden onderbroken.

De energiemeter wordt gevoed door twee Hi-Link HLK-5M05 modules ACDC1/2, voor een galvanische scheiding tussen de MCU en de schakeling van de energiemeter, als bescherming tegen het risico van hoogspanning. De AMS1117-3.3 spanningsregelaars leveren voor de voeding een stabiele 3,3 V, essentieel voor de ESP32 en andere laagspanningsonderdelen. De veiligheid wordt verder verhoogd door de zekering F1 voor overstroombeveiliging en een metaaloxide-varistor (MOV) (R23) tegen spanningspieken. Voor diagnosedoeleinden geven LED1 en LED2 de voedings- en bedrijfsstatus aan. Connector K6 is verbonden met alle uitgangen van de MCU voor het debuggen.

Isolatie van de schakeling

In het schema heb je misschien twee DC-massa's gezien, GND en GNDA. De massaklem (GND) is verbonden met IC1 en is ook verbonden met de nulleider van het lichtnet. GNDA is een geïsoleerde aardklem die is verbonden met de ESP32-WROOM-32D (MOD1). Voor de veiligheid is het noodzakelijk om de ESP32 te isoleren van de nulleider van het lichtnet. Omdat IC1 geen galvanische scheiding heeft, is het noodzakelijk om deze componenten van elkaar te isoleren. Nu rijst de vraag hoe de SPI tussen deze twee chips wordt gecommuniceerd. Hier komt IC2, een ADuM3151 van Analog Devices om de hoek kijken. De ADuM3151 is van cruciaal belang voor een veilige communicatie tussen IC1 en de ESP32-WROOM-32D en zorgt voor galvanische isolatie van de SPI-lijnen. In **figuur 5** zie je het functionele blokschema van IC2 [3]. Het maakt gebruik van inductieve koppelingen om digitale signalen door een isolatiebarrière te sturen, waardoor de op de computer



Figuur 5. Functioneel blokschema van de ADuM3151 SPI isolator (bron: Analog Devices [3]).

aangesloten ESP32 effectief wordt afgeschermd van de piekspanningen van het lichtnet. Deze opzet is cruciaal voor het voorkomen van schade tijdens het coderen en debuggen, terwijl de ondersteuning van meerdere geïsoleerde kanalen zorgt voor betrouwbare en veilige SPI-communicatie, waarbij de gegevensintegriteit behouden blijft en afgestemd wordt op de veiligheids- en prestatiedoelen van het project.

Gebruikersinterface en interactie

De gebruikersinterface van het ESP32-energiemeterproject is ontworpen om informatief en gebruiksvriendelijk te zijn. Een OLED-display, aangesloten op connector K5, aangesloten op de I²C-pinnen van de ESP32, doet dienst als primair display. Dit display toont alle relevante gegevens in real-time aan de gebruiker, inclusief energieverbruik en systeemstatus. De keuze voor OLED-technologie zorgt voor een duidelijke weergave en een snel reagerende interface.

Naast het hardware-display bevat het project een webserver die op de ESP32 wordt gehost. Deze webinterface bevat de gegevens die op het OLED-scherm worden weergegeven en biedt gebruikers een alternatieve manier om hun energieverbruik in de gaten te houden. De firmware is erop gericht om een web-UI en -UX te maken die zowel gebruiksvriendelijk als gedetailleerd is, zodat de gegevens toegankelijk en uitgebreid worden gepresenteerd. Dankzij deze dubbele interfaces kunnen gebruikers zowel ter plekke als op afstand communiceren met de energiemeter, wat de algehele bruikbaarheid van het systeem verbetert.

Volgende stappen en toekomstplannen

Naarmate het project vordert, is het eerste PCB-ontwerp opgestuurd voor productie. Na levering wordt de aandacht verlegd naar de firmware-rekant van het project. De ontwikkeling van de firmware omvat het programmeren van de ESP32 om nauwkeurig gegevens over energieverbruik te verwerken en weer te geven, de webserver te beheren en te zorgen voor een soepele communicatie tussen alle componenten. Voor de toekomst zijn er plannen om extra functies te integreren om de functionaliteit van de energiemeter te vergroten. Deze kunnen het volgende omvatten:

- monitoring op afstand: gebruikers kunnen via de webinterface vanaf elke locatie hun energieverbruik controleren;
- waarschuwingen en meldingen: implementatie van een systeem om gebruikers te waarschuwen over ongewone energieverbruikspatronen of potentiële systeemproblemen;
- tools voor gegevensanalyse: analysetools opnemen in de webinterface om gebruikers te helpen inzicht te krijgen in trends van hun energieverbruik en te bepalen waar verbeteringen in efficiëntie mogelijk zijn.

We streven ernaar om voortdurend te verbeteren en te innoveren, waarbij we ons richten op feedback van gebruikers om toekomstige verbeteringen door te voeren. Het doel is om niet alleen een betrouwbare tool voor energiemonitoring te bieden, maar ook om gebruikers inzicht te geven in hun energieverbruik, waardoor bewustwording en efficiëntie worden bevorderd. ◀

vertaling: Hans Adams — 230709-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via saad.imtiaz@elektor.com of naar de redactie van Elektor via redactie@elektor.com.



Over de auteur

Saad Imtiaz (Senior Engineer, Elektor) is een mechatronica-technicus met vijf jaar ervaring in embedded systemen, mechatronische systemen en productontwikkeling. Hij heeft met talloze bedrijven, variërend van startups tot wereldwijde ondernemingen, samengewerkt aan product-prototyping en -ontwikkeling. Saad heeft ook gewerkt in de luchtvaartindustrie en heeft een technologisch startup-bedrijf geleid. Sinds kort werkt hij bij Elektor en is hij verantwoordelijk voor projectontwikkeling op het gebied van software en hardware.



Gerelateerde producten

- **LILYGO T-Display-S3 ESP32-S3 Development Board**
www.elektor.nl/20299
- **ESP-C3-12F-Kit Development Board (4 MB Flash)**
www.elektor.nl/19855
- **Joy-IT NodeMCU ESP32**
www.elektor.nl/19973



WEBLINKS

- [1] Saad Imtiaz, "Prototype van een ESP32-gebaseerde energiemeter," Elektor Guest Edition 2023: <https://www.elektormagazine.nl/230646-03>
- [2] ATM90E32AS Meerfasig energiemeter-IC: <https://www.microchip.com/en-us/product/atm90e32as>
- [3] Analog Devices Inc. ADuM3151 SPI Isolator™ Digital Isolators: <https://eu.mouser.com/new/analog-devices/adi-adum3151-isolators>



Project-update #2: ESP32-gebaseerde energiemeter

enkele verbeteringen



Figuur 1. De nieuw ontworpen behuizing van de ESP32-energiemeter.

Saad Intiaz (Elektor)

In de vorige aflevering van deze serie bespraken we de schema's en de isolatiestrategie van de ESP32 Energy Meter. Hier komen verdere verdere verbeteringen en het printontwerp aan de orde – en meer.

In 2023 begonnen we met de ontwikkeling van een betrouwbare, gebruiksvriendelijke energiemeter op basis van een ESP32-microcontroller. In het vorige artikel, "Project-update: ESP32-gebaseerde energiemeter" [1], hebben we het blokschema, de uitgewerkte schema's, de isolatiestrategie, de functies en de projectopzet besproken. We beginnen voor alle zekerheid met een korte samenvatting voordat we de jongste updates behandelen.

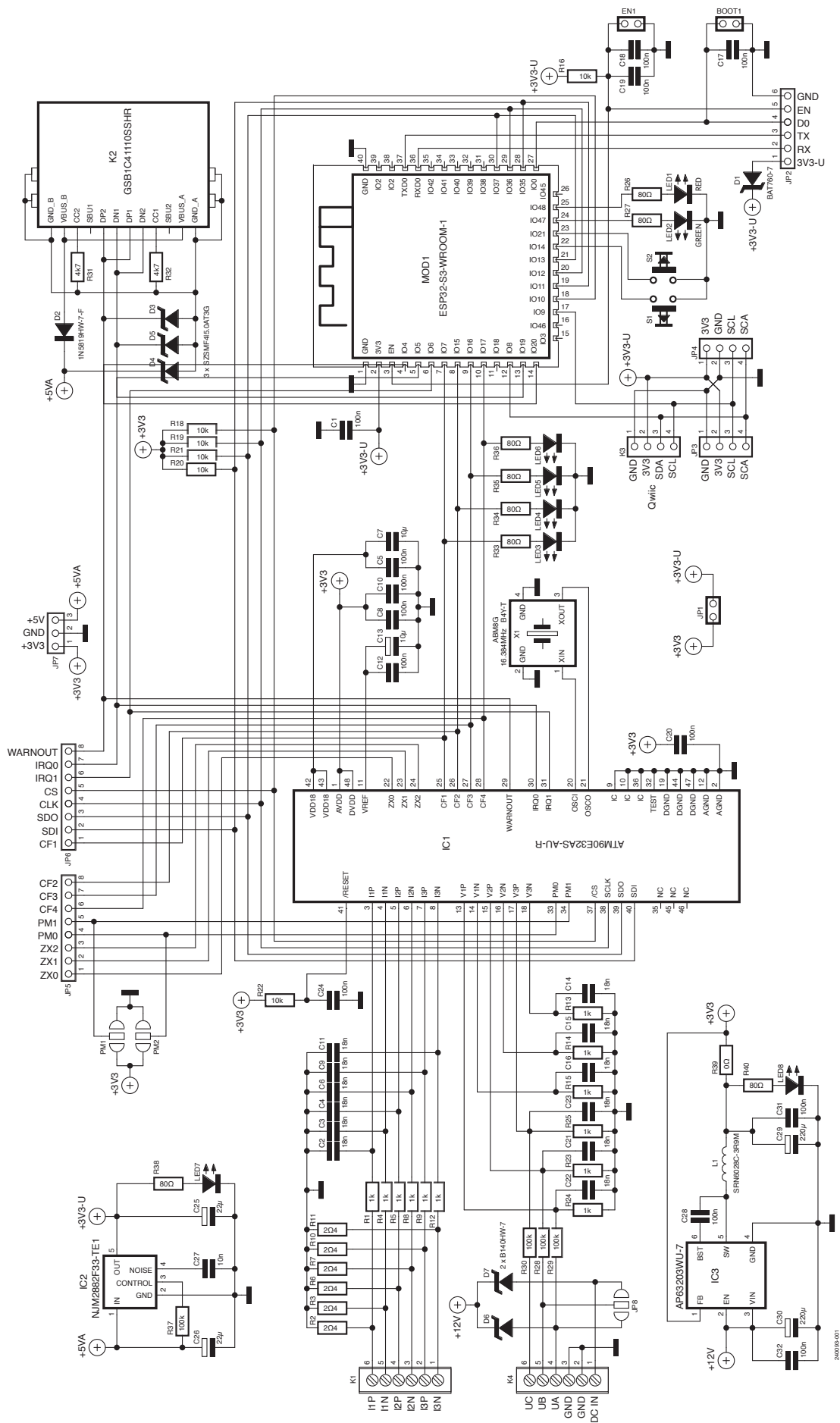
De achterliggende gedachte was het ontwikkelen van een nauwkeurige en efficiënte energiemeter die gebruik maakt van de mogelijkheden van de Espressif ESP32-microcontroller en het ATM90E32AS-IC voor energiemeting. Het project was gericht op het verbeteren van de gebruikerservaring en de betrouwbaarheid door middel van een nauwgezet schema-ontwerp en isolatie met behulp van de ADuM3151 voor een veilige communicatie tussen de ESP32 en de ATM90E32AS van Atmel (nu Microchip). De nadruk lag op veiligheid en efficiëntie door de integratie van ruisonderdrukkingstechnieken, verbeteringen van de signaalintegriteit en beschermingsmechanismen zoals zekeringen en MOV's. Met een focus op toekomstgerichte functies omvatte het plan ook de integratie van tools voor bewaking op afstand en gegevensanalyse voor een beter energiebeheer en meer inzicht in efficiëntie. In dit artikel blijven onze hoofddoelen hetzelfde en zijn er veel wijzigingen aangebracht om het project veiliger in gebruik te maken, de productiekosten te verlagen en de afmetingen te verkleinen. Zoals

al eerder opgemerkt, was de afmetingen van de prototype-print 100×100 mm. Na het testen zijn enkele componenten verwijderd en is de layout van de print geoptimaliseerd, waardoor de afmetingen voor deze versie zijn teruggebracht tot 79,5×79,5 mm – ongeveer 20% minder dan voorheen. In **figuur 1** is de behuizing voor de nieuwe versie van de print te zien. Om de ESP32 Energy Meter veiliger in het gebruik te maken, wordt de print niet langer direct gevoed uit het lichtnet, maar gebruiken we nu een 220 V/12 V stepdown-transformator voor de spanningsbemonstering en de voeding van de schakeling. Het toevoegen van een transformator heeft wel wat nadelen in termen van fasevertraging, maar veiligheid voor alles! Aangezien we geen spanningspieken of snelle spanningsdips willen meten, maar energie, zou dit onze meting niet mogen schaden.

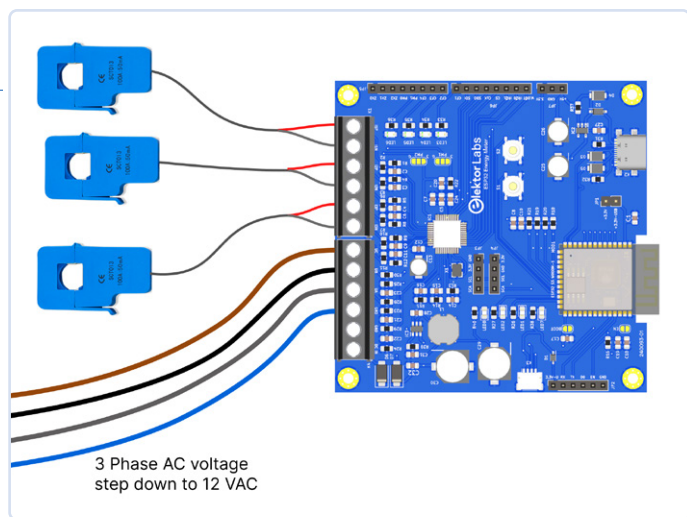
Het bijgewerkte schema

We hebben een aantal verbeteringen doorgevoerd en in plaats van de ESP32 is nu de ESP32-S3 aan boord. Dit geeft de energiemeter meer mogelijkheden. De ESP32-S3 biedt aanzienlijke verbeteringen ten opzichte van de ESP32, waaronder verbeterde verwerkingskracht, AI en signaalverwerkingsmogelijkheden, meer geheugen en betere beveiligingsfuncties. Het bijgewerkte schema verbetert de mogelijkheden van de energiemeter nog verder en biedt meer functionaliteit. De ontwerp-referentieguides van Espressif [2] en andere nuttige internetbronnen [3...6] werden gebruikt om de ESP32-S3 in het project te integreren. **Figuur 2** toont het schema van het project.

De layout van de print is geoptimaliseerd om de veiligheid, bruikbaarheid en efficiëntie van de ESP32-S3 te verbeteren. We hebben aanzienlijke aanpassingen gedaan door de print te verkleinen voor een compactere footprint, over te stappen op transformatorgebaseerde voeding voor meer veiligheid en de veelzijdigheid te vergroten (compatibiliteit met één en drie fasen). Het gebruik van een efficiëntere AP63203WU-7 buck converter in plaats van Hi-Link modules, samen met de toevoeging van gebruiksvriendelijke zaken zoals een USB-C- en een Qwiic-connector voor uitbreidingsmogelijkheden, hebben allemaal



Figuur 2. Het schema van het project.



Figuur 3. Bedrading van een driedfasig spanningsstelsel plus transformatoren met de ESP32-energiemeter.

bijgedragen aan de ontwikkeling van het project. Deze verbeteringen bouwen voort op de mogelijkheden van de ESP32-S3 en zijn gericht op het bieden van een praktische, aanpasbare en veiligere oplossing voor energiemonitoring.

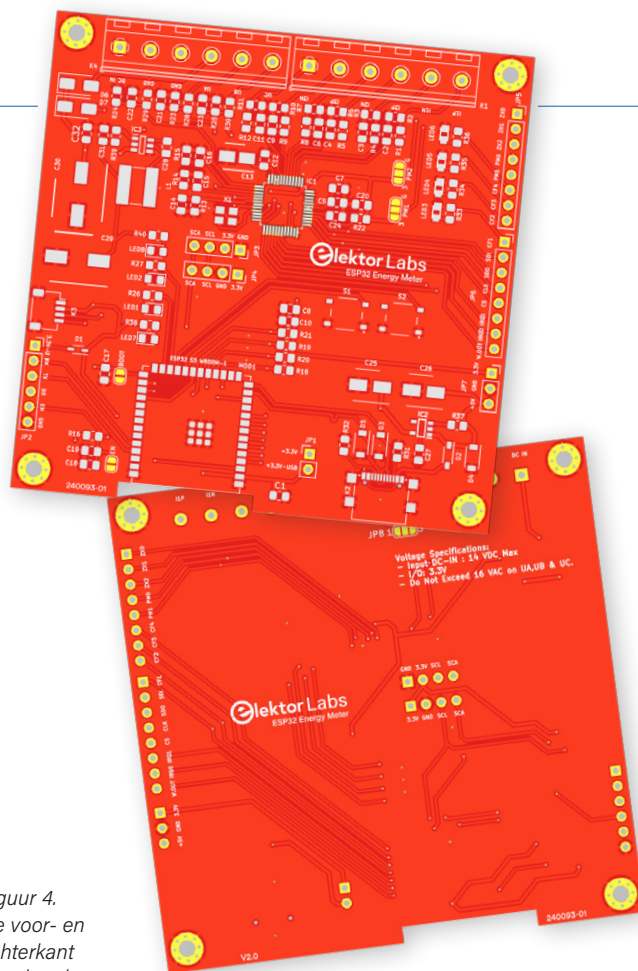
Nauwkeuriger bemonstering van spanning en stroom

IC1 blijft dezelfde ATM90E32AS, maar de verandering is dat er nu een 220 VAC/12 VAC step-down transformator tussen komt. Dit is gedaan om het project veiliger te maken om te testen en te gebruiken, omdat de transformatoren zorgen voor galvanische scheiding. Tijdens mijn tests was er geen noemenswaardig verschil in prestaties.

Voor elk van de ingangen voor spanningsbemonstering van IC1 is er nu dus slechts één weerstand van 100 kΩ (R28...R30). De vorige keer combineerden we alle fasespanningen op één ingang, en er kwam veel feedback van de lezers die de mogelijkheid wilden hebben om de meter te gebruiken met driedfasige of enkelfasige voeding, al naar de omstandigheden. We hebben erover nagedacht en nu kunnen we de meter in beide gevallen gebruiken. Standaard is de driedfase-modus geconfigureerd, maar als je hem enkelfasig wilt maken, moet jumper JP8 worden kortgesloten. **Figuur 3** toont de algemene bedrading voor een driedfasensysteem. Merk op dat de fasedraden worden aangesloten nadat de spanning tot 12 VAC omlaag is getransformeerd – het gebruik van een 12 VAC deurbeltransformator kan in dit geval nuttig zijn. Voor het bemonsteren en meten van de stroom wordt in plaats van een hoofdtelefoonconnector een printkroonsteen met een steek van 5,08 mm gebruikt, namelijk K1. Dit draagt bij aan de algehele robuustheid van de energiemeter. Voor de stroomsensoren is de YHDC SCT013 100 A : 50 mA geselecteerd; de weerstanden R1 tot R12 voor alle drie de stroommeetingsgangen zijn overeenkomstig gekalibreerd.

Voedingsoptimalisatie

De energiemeter wordt nu gevoed met de schakelende buck-regelaar IC3, de AP63203WU-7 van Diodes Incorporated. Voorheen werden Hi-Link HLK5M05-modules gebruikt, maar die zijn veel groter en duurder dan deze buck-converter. Dit is gedaan omdat buck-converters efficiënter zijn dan die Hi-Link modules, minder kosten en veel kleiner zijn. Door IC3 te gebruiken kunnen we de schakeling ook voeden met 12 VDC op K4 voor ontwikkelingsdoeleinden en ook met de UA, dat wil zeggen spanning fase 1 van dezelfde connector K4, voor normaal gebruik.



Figuur 4. De voor- en achterkant van de print.

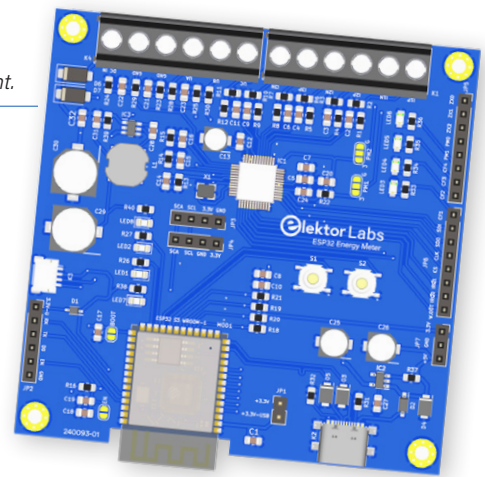
Interactieve en modulaire functies

Voor de pulsuitgangen (actieve, reactieve, schijnbare, actieve fundamentele en harmonische energie) CF1...CF4 zijn LED's toegevoegd [7] [8]. Voor de keuze van de voedingsmodus van IC1 zijn jumpers PM1 en PM2 toegevoegd. In deze versie zijn alle uitgangspennen van IC1 (ATM90E32AS) voor de MCU toegankelijk op connectoren JP5 en JP6. Hierdoor kan de energiemeter ook worden gebruikt als module met een andere MCU als de onboard ESP32-S3 niet nodig is. De ESP32-S3 heeft USB aan boord, dus het is erg handig om de MCU op deze manier te programmeren. Daarom hebben we USB-C connector K2 toegevoegd. Voor foutzoeken is aansluiting JP2 toegevoegd. Status-LED's LED1 en LED2, die kunnen worden aangestuurd door de ESP32-S3, en drukknoppen S1 en S2 zijn toegevoegd voor interactie met het OLED-display, dat kan worden aangesloten op JP3 en JP4. Waarom twee aansluitpunten? Sommige I²C-OLED-displays hebben massa als eerste pin en sommige hebben daar de 3V3-voeding. Op deze manier kan met beide OLED-pinouts worden gewerkt. Tot slot is ook de Qwiic-connector K3 toegevoegd om de functionaliteit van de energiemeter te vergroten, voor het geval men extra sensoren of modules aan dit project wil toevoegen.

De print-layout

De print-layout is zorgvuldig geoptimaliseerd voor kleine afmetingen en gemakkelijk solderen, zie **figuur 4**. Bovenaan zijn de aansluitingen voor de spannings- en stroombemonstering strategisch bij elkaar geplaatst voor integratie in DIN-rail formaat. Aan de rechterkant zijn aansluitingen voor een externe microcontroller (MCU) voorzien in de vorm van met een raster van 2,5 mm, voor gemakkelijke toegang en modulariteit. Centraal bevindt zich de aansluiting voor het OLED-display, geflankeerd

Figuur 5. 3D-model van de geassembleerde print.



door drukknoppen voor intuïtieve interactie. Naast het OLED-display zorgen stroom- en status-LED's voor directe visuele feedback, terwijl de LED's voor de energiepuls-uitgang handig in de buurt van de uitgangsaansluitingen van de MCU zijn aangebracht voor directe controle. Al in het begin van de ontwerpfase zijn de USB-C poort en de ESP32-S3 module uit de buurt van de AC-spanningvoerende delen geplaatst om de veiligheid te vergroten. Een keramische condensator naast de 3V-ingang van de ESP32-S3 zorgt voor ontkoppeling en vermindert eventuele ruis aanzienlijk. Bovendien zijn er elektrolytische condensatoren in het ontwerp opgenomen om de voeding verder te stabiliseren en de betrouwbaarheid en prestaties van de schakeling te garanderen. Deze layout stroomlijnt het assemblageproces en verbetert de functionaliteit en gebruikerservaring door een duidelijke en logische plaatsing van de componenten. In **figuur 5** zie je de rendering van de geassembleerde print.



In dit ontwerp zijn transformatoren gebruikt die netspanning voeren. Mensen die geen ervaring hebben met netspanning moeten dit project niet proberen of iemand met ervaring vragen die hen hierbij kan helpen!

Volgende stappen en een vooruitblik

Na de prototype-fase met het originele schema hebben we verschillende verbeteringen doorgevoerd om de betrouwbaarheid van de ESP32-energiemeter te vergroten. Momenteel richten we ons ook op de verdere ontwikkeling van de firmware.

Het nieuwste print-ontwerp wordt momenteel geproduceerd en we verwachten na ontvangst uitgebreide tests uit te voeren die de betrouwbaarheid van het systeem moeten garanderen. Tegelijkertijd vordert de ontwikkeling van software om de mogelijkheden van de ESP32-S3-module in onze energiemeter te maximaliseren.

Vooruitblikkend zijn we van plan om de ESP32-energiemeter te integreren met Home Assistant, met als doel een vereenvoudigd gebruik door de gebruiker. Daarnaast ontwikkelen we 'maatgesneden' firmware om het potentieel van het apparaat volledig te benutten.

Samengevat gaat het project vooruit met zowel hardware- als softwareverbeteringen. Ons doel blijft om een betrouwbare en efficiënte oplossing voor energiemeting te bieden. Dit project staat ook op het Elektor Labs platform [9], dus voel je vrij om commentaar te geven en bij te dragen! 

240093-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via saad.imtiaz@elektor.com of naar de redactie van Elektor via redactie@elektor.com.



Over de auteur

Saad Imtiaz (Senior Engineer, Elektor) is een mechatronica-ingenieur met ervaring in embedded systemen, mechatronische systemen en productontwikkeling. Hij heeft samengewerkt met talloze bedrijven, variërend van startups tot globale ondernemingen, op het gebied van prototyping en ontwikkeling. Saad heeft ook gewerkt in de luchtvaartindustrie en heeft een technologisch startup-bedrijf geleid. Bij Elektor is hij verantwoordelijk voor projectontwikkeling op het gebied van software en hardware.



Gerelateerde producten

- **Qoitech Otii Arc - Power Supply, Power Meter and Data Acquisition**
www.elektor.nl/19270
- **ESP Terminal**
www.elektor.nl/20526
- **Arduino Nano ESP32**
www.elektor.nl/20562

WEBLINKS

- [1] Saad Imtiaz, "Project-update: ESP32-gebaseerde energiemeter", Elektor januari/februari 2024:
<http://www.elektormagazine.nl/magazine/elektor-326/62603>
- [2] ESP32 S3 DevKit-C Schematic: https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf
- [3] ESP32 S3 Pinout Help Guide: <https://luisllamas.es/en/which-pins-can-i-use-on-esp32-s3>
- [4] SCH_ESP32-S3-USB-Bridge-MB_V2.1 Schematic: <https://tinyurl.com/usbbridgeschematic>
- [5] ESP32-S3 Pin Reference: http://wiki.fluidnc.com/en/hardware/ESP32-S3_Pin_Reference
- [6] ESP32-S3: Which Pins Should I Use?: <https://atomic14.com/2023/11/21/esp32-s3-pins.html>
- [7] Application Note Poly-Phase Energy Metering IC M90E32AS: <https://tinyurl.com/polyphasemetering>
- [8] Atmel M90E32AS | datasheet: https://eu.mouser.com/datasheet/2/268/Atmel_46003_SE_M90E32AS_Datasheet-1368788.pdf
- [9] ESP32-energiemeter op Elektor Labs:
<https://elektormagazine.com/labs/esp32-energy-meter-an-open-source-solution-for-real-time-energy-monitoring>



Project-update #3: ESP32- gebaseerde energie- meter

integratie en test met
Home Assistant

Saad Intiaz (Elektor)

In de vorige project-update hebben we verteld over verbeteringen aan het schema en de print van de ESP32-energiemeter.

Dit artikel richt zich op de praktische implementatie en integratie van de nieuwe versie. Het biedt een stap-voor-stap handleiding voor het integreren van de meter met ESPHome en Home Assistant voor effectieve energiemonitoring. Verder behandelen we de kalibratie van het apparaat.

In de vorige aflevering [1] richtten we ons op het elektronische ontwerp en de print van de ESP32 Energy Meter, met verbeteringen in modulariteit en veiligheidsfuncties. Voordat we naar de volgende project-update gaan, eerst een kort overzicht.

In de meest recente incarnatie van het ESP32 Energy Meter-project hebben we een upgrade uitgevoerd naar de ESP32-S3 micro-controller, met meer rekenkracht en een grotere functionaliteit. In het nieuwe ontwerp is de print afgeslankt en is een transformator-



Figuur 1. De volttooide ESP32 Energy Meter met OLED-display en live statusindicatie.

gebaseerd voedingssysteem opgenomen, dat gebruik maakt van een 230V/12V-transformator voor het bemonsteren van de spanning en de werking van het systeem. Dit verbetert de veiligheid aanzienlijk terwijl de flexibiliteit voor zowel één- als driefase-installaties behouden blijft. Andere verbeteringen waren de integratie van een efficiëntere AP63203WU-7 buck-converter, de modulaire print, kalibratie van de stroomtrafo-samplingschakeling en nog veel meer. Dit optimaliseert niet alleen de prestaties en functionaliteit van de energiemeter, maar vermindert ook de kosten en de omvang.

In dit artikel bespreken we de stappen die zijn genomen om deze energiemeter aan de praat te krijgen en het traject van de labtafel naar de meterkast. Verder zullen we ook bespreken hoe hij wordt ingesteld en gekalibreerd, en ten slotte hoe hij wordt geïntegreerd met Home Assistant en ESP Home om de verzamelde gegevens van de energiemeter te tonen en te controleren. In **figuur 1** ziet u het ESP32 Energy Meter-project in actie, ondergebracht in een 3D-geprinte behuizing met een OLED-display. De afbeelding toont live-statusindicatoren die continu het realtime-stroomverbruik bijhouden en weergeven.

Bouw

De nieuwe print was compacter en eenvoudiger te solderen en de layout bood voldoende ruimte voor elke component om de montage te vergemakkelijken. Om de nabouw van het project en eventuele aanpassingen door zowel enthousiastelingen als professionals te vergemakkelijken, zijn de volledige onderdelenlijst (BOM) in Mouser-formaat en de productiebestanden beschikbaar op de GitHub-repository van Elektor Lab [2].

Voor de aansluitingen voor de spannings- en stroombemonstering werden printkroonstenen van CUI Devices gebruikt. De kwaliteit van deze kroonstenen is veel beter dan van de goedkope blauwe exemplaren die je op de meeste sensormodules ziet. Omdat we te maken hebben met wisselspanningen en energiemeting, is het van vitaal belang om veilige en betrouwbare aansluitingen te hebben.

Ruisonderdrukking is een kritisch aspect van het printontwerp, dat wordt aangepakt door zowel elektrolytische als keramische condensatoren te integreren rond de ATM90E32S-energiemeetchip. Deze opstelling helpt bij het uifilteren van zowel laag- als hoogfrequente ruis, wat zorgt voor een nauwkeurigere en stabielere energiemeting. De print is afgebeeld in **figuur 2**.

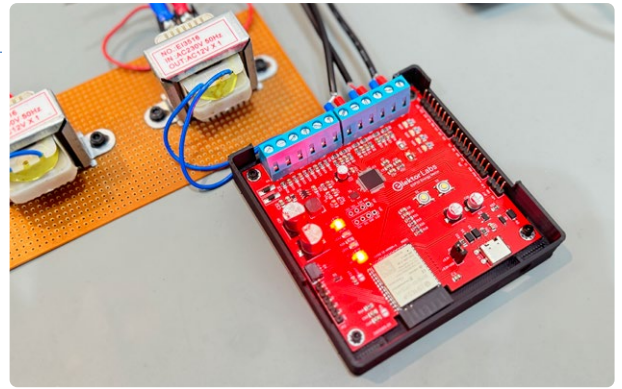
Zoals eerder opgemerkt, hebben we besloten om een stepdown-transformator te gebruiken voor de spanningsbemonstering en als hoofdbron voor de voeding van het hele systeem. Zo'n transformator is goedkoop en gemakkelijk te vinden, maar de meeste van deze transformatoren nemen veel ruimte in beslag wanneer ze gebruikt worden in een op maat gemaakte behuizing, zoals te zien in **figuur 3**. Het is daarom het beste om hier DIN-rail beltransformatoren te gebruiken om de opstelling netter en veiliger te maken; dergelijke transformatoren zijn gemakkelijk online te vinden. Bovendien hangt de nauwkeurigheid van spanningsmetingen af van de karakteristieken van de transformatoren, waaronder de nauwkeurigheid van de spanningsverhouding, de faseverschuiving en lineariteit.

Het is u misschien opgevallen dat er in de foto's maar één transformator aan de energiemeter is bevestigd. De energiemeter is geconfigureerd voor gebruik in éénfase-modus door jumper JP8 op de achterkant van de print te overbruggen (kort te sluiten), zoals te zien in **figuur 4**. Om de energiemeter in driefase-modus te gebruiken of om de spanning van elke fase in een driefasensysteem te bemonsteren met behulp van drie stepdown-transformatoren, moet u de primaire wikkelingen van de drie transformatoren verbinden met de respectieve fasen (L1, L2, L3). Aan de secundaire zijde sluit u een uiteinde van de wikkeling van elke transformator aan op een gemeenschappelijk neutraal punt, waardoor u een sterschakeling (Y) vormt. De vrije uiteinden van de secundaire wikkelingen (V1, V2, V3) leveren dan de spanningsuitgangen (UA, UB en UC) op de print voor elke fase. Belangrijke overwegingen zijn dat de transformatoren de juiste nominale waarde hebben voor de spanning en stroom van het systeem, dat de primaire en secundaire zijden om veiligheidsredenen strikt geïsoleerd blijven en dat er een stabiele en goed gebalanceerde nulverbinding is om meetonnauwkeurigheden te voorkomen.

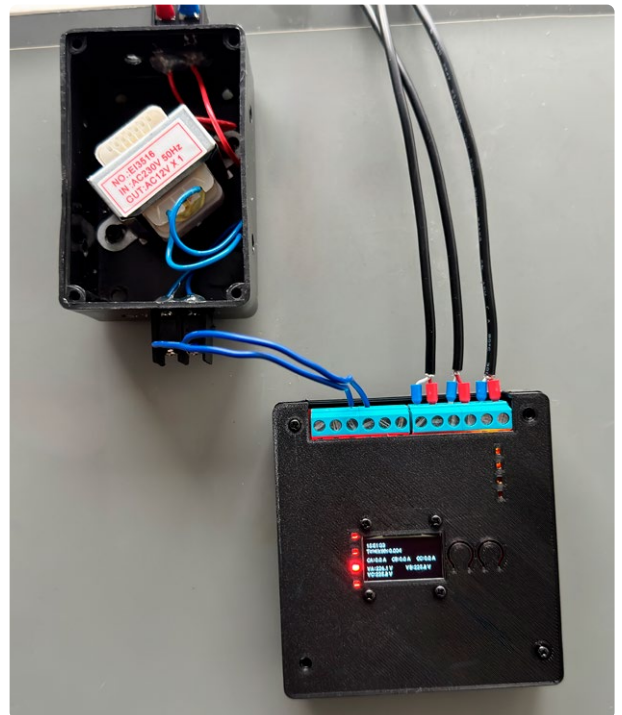
Installeren met ESPHome en Home Assistant

Als onderdeel van het ontwikkelingsconcept wordt een specifieke firmware geschreven om de mogelijkheden van de energiemeteetchip en de geavanceerde AI-functies van de ESP32-S3 te benutten. Hoewel de ontwikkeling van dergelijke op maat gemaakte firmware veel tijd vergt en nog steeds aan de gang is, beperkt dit de bruikbaarheid van de energiemeter niet. Het apparaat is volledig functioneel met bestaande platforms zoals Home Assistant, wat een directe oplossing biedt voor energiemonitoring. Daarom ligt in dit artikel de focus op de integratie van de ESP32-energiemeter met Home Assistant [3] en ESPHome [4]. In deze paragrafen leiden we u door het installeren van de energiemeter in de Home Assistant-omgeving waarbij de volledige functionaliteit beschikbaar komt.

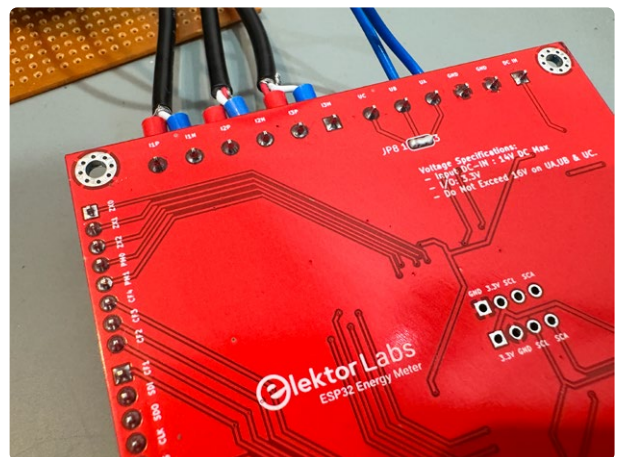
Om de ESP32 Energy Meter te installeren met de ESPHome-firmware en te integreren in Home Assistant, moet u eerst Home Assistant installeren. (Een uitgebreide handleiding is te vinden in een artikel van mijn collega Clemens Valens [5].) Integreer daarna ESPHome vanuit de *Add-on Store*; maak vervolgens een nieuw project aan in



Figuur 2. De geassembleerde print van de ESP32 Energy Meter.



Figuur 3. 220V/12V-transformator, ondergebracht in een op maat gemaakte behuizing.



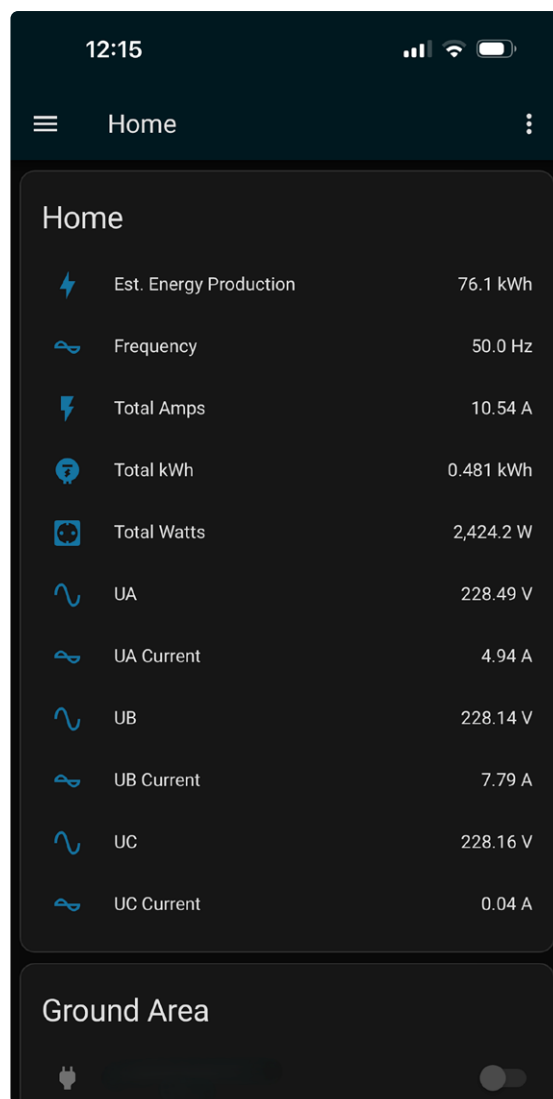
Figuur 4. Jumperconfiguratie voor éénfase-gebruik op de print van de ESP32 Energy Meter.

ESPHome voor uw ESP32-apparaat. Dit genereert automatisch een basis-YAML-configuratiebestand (zo'n YAML-bestand specificeert een apart ESPHome-project met alle gebruikte sensoren en vele andere opties). U moet dit standaardbestand downloaden voordat u de volgende stappen uitvoert.

Sluit de ESP32 aan op uw computer en selecteer de juiste COM-poort voor de ESP32-S3. Klik in het ESPHome-dashboard op *Install* en kies het *.bin* bestand om de firmware te flashen. Zodra de firmware succesvol is geladen en uw ESP32 Energy Meter wordt herkend door Home Assistant, kunt u de initiële YAML-configuratie bewerken. Open hiervoor het ESPHome-dashboard in Home Assistant, zoek het apparaat en klik op de *Edit*-optie op de kaart van de energiemeter. Vervang de bestaande configuratie door de YAML-inhoud in de GitHub-repository [2]. Zorg ervoor dat u uw API-, OTA- en WiFi-referenties correct configureert in deze nieuwe YAML-setup.

Installeer de nieuwe configuratie draadloos op uw ESP32 Energy Meter. Daarna is het apparaat actief en verbonden. Om de gegevens van de energiemeter weer te geven op het dashboard van uw Home Assistant, moet u het ESPHome-apparaat toewijzen aan een specifiek gebied in Home Assistant. Dit helpt bij het organiseren van het dashboard door apparaten te groeperen op basis van hun fysieke of logische locatie in huis. Om een idee te krijgen van wat mogelijk is, verwijzen we naar **figuur 5**, waar de gegevens van de energiemeter op het Home Assistant-dashboard zijn weergegeven.

De integratie van de ESP32 energiemeter met Home Assistant vereenvoudigt niet alleen het monitoren van energieverbruik, maar ontsluit ook een reeks krachtige functies van het platform. Home Assistant biedt een intuïtieve interface voor realtime-datavisualisatie, automatische besturing en naadloze integratie met andere slimme apparaten in uw huis.



Figuur 5. Home Assistant-dashboard met realtime-energiegegevens van de ESP32-energiemeter.



Figuur 6. Gedetailleerd verloop van het energieverbruik, grafisch weergegeven in Home Assistant.

Deze integratie maakt het mogelijk om gedetailleerde historische grafieken en analyses te maken binnen Home Assistant, waardoor u een goed inzicht krijgt in energieverbruikpatronen in de loop van de tijd; **figuur 6** toont een voorbeeld. Aan de hand daarvan kunnen gebruikers weloverwogen beslissingen nemen over hun energieverbruik, potentiële besparingen identificeren en de energie-efficiëntie van het huis optimaliseren.

Door het hierboven beschreven installatieproces te volgen, kunnen gebruikers ten volle profiteren van deze mogelijkheden, waardoor de ESP32 Energy Meter een centraal onderdeel wordt van hun smart home-ecosysteem. Deze integratie verbetert niet alleen de functionaliteit van de energiemeter, maar verrijkt ook de algehele smart home-ervaring met uitgebreide energiemonitoring en beheermogelijkheden.

YAML-configuratie

De meegeleverde YAML-configuratie stelt de ESP32 Energy Meter in met ESPHome, waardoor essentiële elektrische parameters zoals spanning, stroom en vermogen voor alle drie fasen kunnen worden bewaakt. Het maakt gebruik van de mogelijkheden van de ATM90E32-sensor, met gedetailleerde definities voor SPI-communicatie en individuele sensoren voor elke fase. Deze opstelling meet niet alleen het totale verbruik, maar berekent dit ook, met een dagteller voor kWh en een OLED-display voor realtime-visualisatie van de gegevens. Deze configuraties zijn gemaakt volgens de instructies op de ESPHome-pagina voor de ATM90E32-sensor [6].

Om de nauwkeurigheid van de gegevens die de ESP32 Energy Meter rapporteert te garanderen, is kalibratie een cruciale stap. De details om de versterkingsinstellingen voor stroomtransformatoren en spanningsingangen aan te passen wordt in de volgende paragrafen beschreven.

Testen en kalibreren

De testopstelling gebruikte een haardroger met verschillende snelheden en warmtestanden als belasting binnen een bereik van 0,7 A tot 8 A. Het netsnoer van een stekkerdoos werd gestript om de transforma-



Figuur 7. Opstelling voor het testen en kalibreren van de ESP32-energiemeter met een variabele belasting.



Figuur 8. Stroommeetang voor de kalibratie.

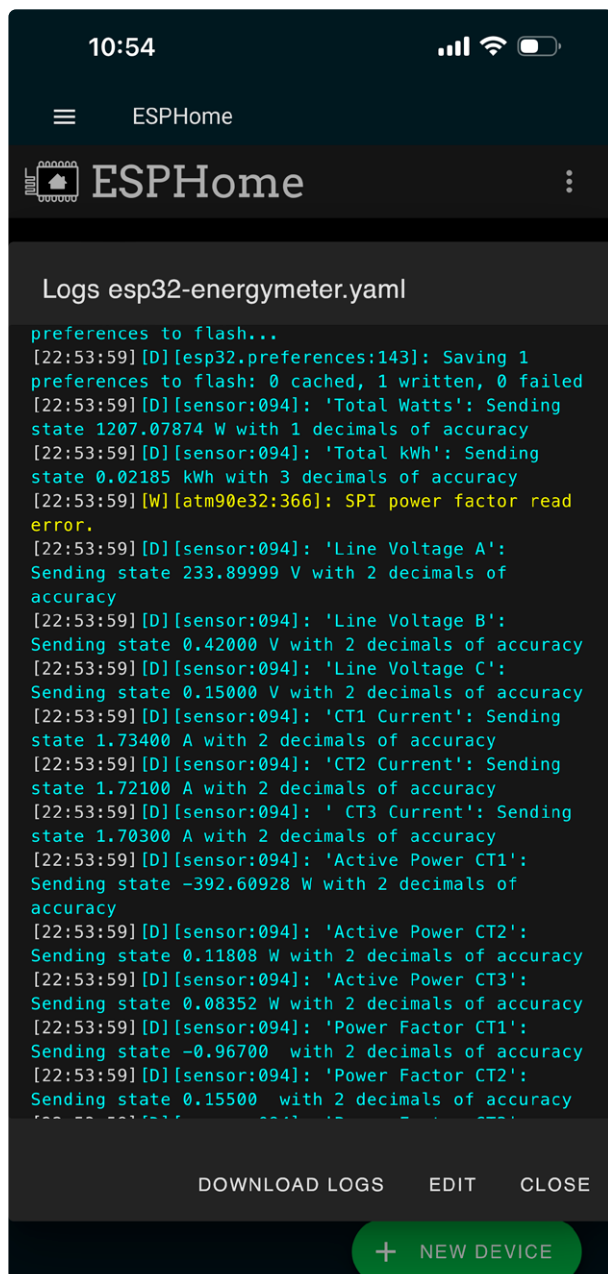
toren met deelbare kern op de stroomvoerende of neutrale draad te kunnen plaatsen, zodat directe controle onder verschillende omstandigheden mogelijk was (**figuur 7**).

Ik heb de stroom- en spanningskalibratie van de ESP32 Energy Meter uitgevoerd met mijn UT201+ multimeter. Deze biedt een resolutie van 0,001 A terwijl de nauwkeurigheid is gespecificeerd als $\pm 4\%$ +10 digits voor stroom en een resolutie van 0,001 V met een nauwkeurigheid van $\pm 1\%$ +5 digits voor spanning. Dit is nauwkeurig genoeg voor de meeste projecten, maar is iets minder nauwkeurig dan professionele meters. Tijdens de kalibratie zijn de stroomaflezingen vergeleken: de aflezing van de stroomtang was 1,692 A (zie **figuur 8**), terwijl de aflezingen berekend door de energiemeter na kalibratie 1,70...1,73 A aangaven, zoals te zien in **figuur 9**. Gezien de specificaties van de UT201+ en de SCT-013-000, een klasse 1 split core-transformator die een nauwkeurigheid binnen 1% van de werkelijke waarde garandeert, valt deze kleine afwijking binnen de verwachte foutmarge. Voor een nog grotere nauwkeurigheid kan echter een betere stroomtang worden gebruikt. Om de nauwkeurigheid van de ESP32-energiemeter verder te verfijnen, werden aanpassingen gedaan aan de versterkingsinstellingen voor zowel spannings- als stroommetingen. Voor spanning werd de sensor gekalibreerd met de formule:

$$\text{new gain_voltage} = (\text{your voltage reading} / \text{ESPHome voltage reading}) * \text{existing gain_voltage value}$$

Hetzelfde geldt voor de stroom:

$$\text{new gain_ct} = (\text{your current reading} / \text{ESPHome current reading}) * \text{existing gain_ct value}$$



Figuur 9. Uiteindelijke kalibratieresultaten die een verbeterde meetnauwkeurigheid tonen.



WAARSCHUWING: werkzaamheden in de meterkast brengen risico's met zich mee, waaronder het gevaar van elektrische schokken of brand. Het is van vitaal belang dat u de stroom volledig uitschakelt voordat u met de installatie begint. In de meeste landen mag dit werk alleen worden uitgevoerd door een gekwalificeerde elektricien!

Deze nieuwe versterkingswaarden werden vervolgens bijgewerkt in het ESPHome YAML-configuratiebestand, gevolgd door het opnieuw compileren en laden van de firmware. Dit proces kan indien nodig worden herhaald voor een optimale nauwkeurigheid. Deze gekalibreerde waarden helpen bij het verfijnen van de metingen en zijn cruciaal voor nauwkeurige rapportage en analyse in elke energiemonitoringopstelling.

Installatie van de ESP32-meter in de meterkast

Het installeren van de ESP32-energiemeter in mijn meterkast bleek goed te doen, maar om zowel de veiligheid als de functionaliteit te garanderen moet daarbij op de kleinste details gelet worden. Ik begon met het selecteren van een groep met de kleinste stroombegrenzing. Deze keuze was belangrijk, omdat het een veiligheidsbuffer bood; de zekering zal dan tijds afgeschakelen in het geval van onverwachte pieken of transformatorstoringen en zo het systeem beschermen. Het gebruik van stroomtransformatoren met een deelbare kern kwam vooral van pas vanwege hun eenvoudige installatie. Deze transformatoren kunnen snel op elke belasting worden geklemd, maar het was cruciaal om de stroomrichting correct te kiezen om nauwkeurige metingen te garanderen. Het is belangrijk op te merken dat als stroomrichting en stroomtransformator niet correct zijn uitgelijnd, er negatieve vermogenswaarden zullen verschijnen; dat duidt dan op een onjuiste installatie.



Figuur 10. De ESP32 Energy Meter geïnstalleerd in de meterkast; hij bewaakt hier het stroomverbruik in realtime.

In **figuur 10** ziet u de ESP32-energiemeter in actie, ingebouwd in de meterkast. Deze afbeelding toont de energiemeter met realtime stroom- en spanningsmetingen en de bijbehorende belasting in kilowatt, ter illustratie van de functionaliteit in een 'echte' situatie.

De ontwikkeling gaat door

Hoewel de actuele softwareconfiguratie draait op ESPHome, zijn er ontwikkelingen gaande om de mogelijkheden van de ESP32-energiemeter uit te breiden. In de nabije toekomst willen we het project integreren met nieuwe firmware die speciaal is ontworpen om het volledige potentieel van de ESP32-S3 chip te benutten. Deze toekomstige firmware zal naar verwachting geavanceerde functies bevatten, zoals gedetailleerde energieanalyses en mogelijk baanbrekende AI/ML-functionaliteiten die energieverbruik patronen kunnen voorspellen en verbruikers kunnen identificeren op basis van hun belastings-footprint. Hoewel de kern van het ontwerp en de operationele aspecten van het project zijn voltooid, is de ontwikkeling van deze geavanceerde functies een complexe en tijdrovende onderneming. Ik ben enthousiast over de mogelijkheden en toegewijd aan het verleggen van de grenzen van wat deze energiemeter kan bereiken.

Het ESP32 Energy Meter-project is continu in ontwikkeling en voegt bij elke update meer functionaliteit toe. Leden van de community die geïnteresseerd zijn in de aankomende AI- en ML-functionaliteiten of die willen bijdragen aan de ontwikkeling, worden aangemoedigd om mee te doen. Samenwerking zal de voortgang versnellen en resulteren in een robuustere oplossing voor energiemonitoring met meer functies. Houd de vinger aan de pols voor verdere ontwikkelingen, want het project is erop gericht om deze veelzijdige tool voor energiebeheer te verfijnen en naar nieuwe hoogten te tillen. ◀

240244-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via saad.imtiaz@elektor.com of naar de redactie van Elektor via redactie@elektor.com.



Over de auteur

Saad Imtiaz (Senior Engineer, Elektor) is een mechatronica-ingenieur met ervaring in embedded systemen, mechatronische systemen en productontwikkeling. Hij heeft samengewerkt met talloze bedrijven, variërend van startups tot globale ondernemingen op het gebied van prototyping en ontwikkeling. Saad heeft ook gewerkt in de luchtvaartindustrie en heeft een technologische startup geleid. Bij Elektor is hij verantwoordelijk voor projectontwikkeling op het gebied van software en hardware.



Gerelateerde producten

> **PeakTech 4350 Clamp Meter**
www.elektor.nl/18161

> **Siglent SDM3045X Multimeter**
www.elektor.nl/17892



WEBLINKS

- [1] Saad Imtiaz, "Project-update #2: ESP32-gebaseerde energiemeter", Elektor mei/juni 2024:
<http://www.elektormagazine.nl/magazine/elektor-343/62866>
- [2] ESP32-energiemeter Github-repository: <https://github.com/ElektorLabs/esp32-energymeter>
- [3] Home Assistant: <https://home-assistant.io/>
- [4] ESPHome: <http://esphome.io>
- [5] Clemens Valens, "Domotica helemaal niet moeilijk", Elektor september/oktober 2020:
<http://www.elektormagazine.nl/magazine/elektor-157/59027>
- [6] ATM90E32 Power Sensor: <https://esphome.io/components/sensor/atm90e32.html>

Project-update #4: ESP32-gebaseerde energiemeter

energiemonitoring met MQTT



Saad Imtiaz (Elektor)

In een eerdere update hebben we het configureren van de energiemeter met een ESP32 en de integratie ervan in Home Assistant uitgelegd. We bespraken ook de toekomstige mogelijkheden om de ESP32-S3-chip te gebruiken voor AI- en ML-functionaliteit om patronen in energieverbruik te voorspellen en apparaten te identificeren. In deze update zetten we een stap vooruit door firmware te introduceren die real-time energiemonitoring mogelijk maakt met behulp van MQTT, wat de weg vrijmaakt voor geavanceerde functies.

Eerder bespraken we de hele ontwikkeling van de ESP32 energiemeter – van werkbank naar groepenkast – waarbij we zijn begonnen met het verzamelen van de componenten, het instellen van de energiemeter met ESPHome en Home Assistant, kalibreren en testen, en uiteindelijk het installeren in de groepenkast. In de laatste update [1] hebben we de meter geïntegreerd in Home Assistant, met het ambitieuze doel om AI- en ML-mogelijkheden toe te voegen om patronen in energieverbruik te voorspellen

en apparaten te identificeren op basis van hun specifieke energieverbruik.

Terwijl de AI/ML-integratie nog in volle gang is – gezien de noodzakelijke omvangrijke voorbereiding van de data – is deze kleine update gericht op een cruciale tussenstap: real-time energiemonitoring met behulp van MQTT. MQTT is een lichtgewicht berichtenprotocol dat is ontworpen voor efficiënte communicatie. Lees het kader **Wat is MQTT?** voor meer informatie.

Aangepaste firmware en MQTT

In dit artikel bespreken we de volgende fase van het project – het gebruik van MQTT en de Arduino IDE om real-time energiemonitoring mogelijk te maken. Deze update behandelt de ontwikkeling van de firmware via welke de ESP32 kan communiceren met een MQTT-broker, waardoor energiegegevens naar een Home Assistant-server of een ander MQTT-compatibel platform kunnen worden verzonden.

Het voordeel van het gebruik van MQTT met aangepaste firmware ten opzichte van ESPHome is dat de eigen firmware veel meer flexibiliteit biedt op het gebied van integratie en aanpassing. Met aangepaste firmware hebt u volledige controle over hoe de gegevens worden verzameld, verwerkt en verzonden, zodat u het systeem

Wat is MQTT?

MQTT is een eenvoudig berichtenprotocol dat is ontworpen voor efficiënte communicatie tussen apparaten, vooral in IoT-omgevingen. Centraal in dit systeem staat de MQTT-broker, een server die fungeert als een hub voor berichtenuitwisseling. De broker ontvangt berichten van apparaten, ook wel publishers genoemd, en stuurt ze door naar de juiste ontvangers of subscribers, gebaseerd op een systeem van topics.

Een topic in MQTT is een string die berichten categoriseert en fungeert als een kanaal waar informatie wordt gepubliceerd, terwijl een subscriber een apparaat of applicatie is die luistert naar een specifiek topic om deze berichten te ontvangen. Bijvoorbeeld, in een smart home-setup kan een topic als `home/energy/voltage` spanningsmetingen bevatten, en een dashboard (subscriber) dat zich abonneert op dit topic zou deze metingen in real-time ontvangen en weergeven.

De broker zorgt ervoor dat berichten efficiënt en veilig worden afgeleverd, zelfs over onbetrouwbare netwerken. In IoT-toepassingen is de MQTT-broker cruciaal voor het beheer van gegevensuitwisseling tussen sensoren, apparaten en systemen (die de MQTT-clients zijn), waardoor real-time bewaking, controle en automatisering mogelijk wordt.



Listing 1: De firmware (uittreksel).

```
#include <WiFi.h>
#include <SPI.h>
#include <ATM90E32.h>
#include <MQTTPubSubClient.h>
#include <config.ino> // Include configuration file for WiFi and MQTT details

// WiFi Credentials
const char* ssid = WIFISSID;      // Your WiFi SSID
const char* pass = WIFIPASSWORD;  // Your WiFi Password

WiFiClient client;
MQTTPubSubClient mqtt;

ATM90E32 energymeter{};

void setup() {
    ...

    /* Initialize the ATM90E32 energy meter with the specified parameters */
    energymeter.begin(CS_PIN, LINEFREQ, PGA_GAIN, VOLTAGE_GAIN, GAIN_CT_A, GAIN_CT_B, GAIN_CT_C);
    ...

    /* Begin the WiFi connection using the provided SSID and password */
    WiFi.begin(ssid, pass);

    /* Initialize the MQTT client */
    mqtt.begin(client);

    /* Connect to WiFi, MQTT broker, and Home Assistant */
    connect();
    ...
}

void loop() {
    /* Keep the MQTT client updated */
    mqtt.update();

    /* Reconnect to the MQTT broker if the connection is lost */
    if (!mqtt.isConnected()) {
        connect();
    }

    /* Check and send energy data to Home Assistant every 3 seconds */
    static uint32_t prev_ms = millis();
    if (millis() > prev_ms + 3000) {
        prev_ms = millis();
        getEnergyData(); // Retrieve energy data and send via MQTT
    }
}

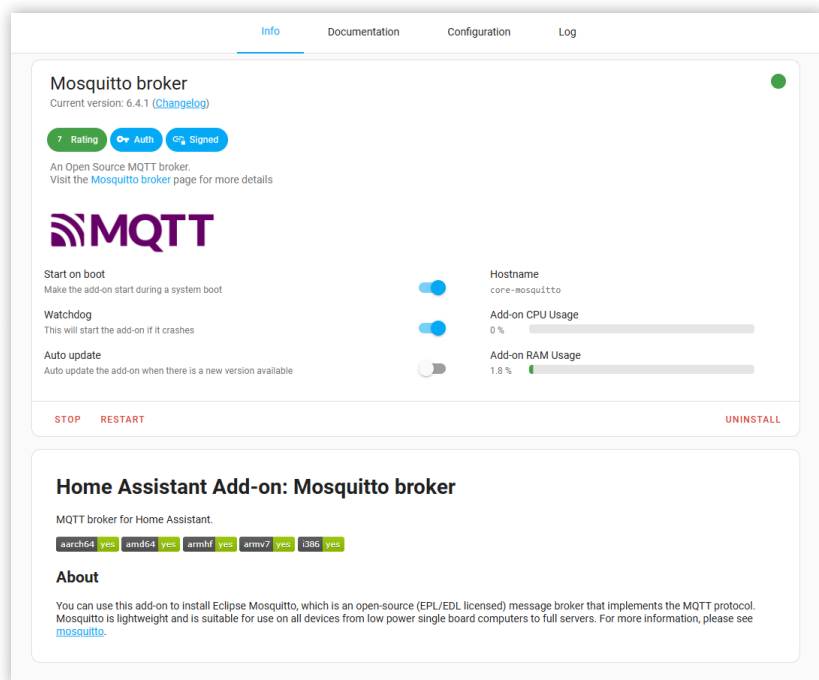
void getEnergyData() {
    // Retrieve system status from the ATM90E32
    unsigned short sys0 = energymeter.GetSysStatus0(); //EMMState0
    unsigned short sys1 = energymeter.GetSysStatus1(); //EMMState1
    unsigned short en0 = energymeter.GetMeterStatus0(); //EMMIntState0
    unsigned short en1 = energymeter.GetMeterStatus1(); //EMMIntState1

    // Print system and meter status for debugging
    Serial.println("Sys Status: S0:0x" + String(sys0, HEX) + " S1:0x" + String(sys1, HEX));
    Serial.println("Meter Status: E0:0x" + String(en0, HEX) + " E1:0x" + String(en1, HEX));
    delay(10);

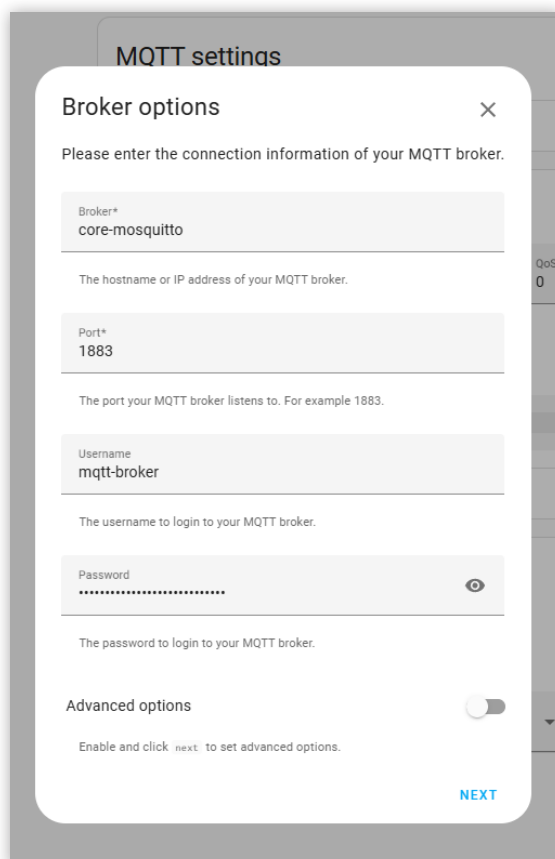
    // Check if the MCU is not receiving data from the energy meter
    if (sys0 == 65535 || sys0 == 0) Serial.println("Error: Not receiving data
                                                from energy meter - check your connections");

    // Retrieve all parameters from the ATM90E32
    float lineVoltageA = energymeter.GetLineVoltageA();
    float lineVoltageB = energymeter.GetLineVoltageB();
    float lineVoltageC = energymeter.GetLineVoltageC();
    ...

    // Send all the collected energy data via MQTT to Home Assistant
    mqtt.publish("esp32energymeter/lineCurrentA", String(lineCurrentA).c_str());
    mqtt.publish("esp32energymeter/lineCurrentB", String(lineCurrentB).c_str());
    mqtt.publish("esp32energymeter/lineCurrentC", String(lineCurrentC).c_str());
    mqtt.publish("esp32energymeter/totalCurrent", String(totalCurrent).c_str());
    ...
}
```



▲
 Figuur 1.
 Configuratieopties
 voor de MQTT-broker
 in Home Assistant,
 inclusief de optie
 Start on boot om automatisch
 opstarten te garanderen.



Figuur 2. MQTT-
 integratie inrichten in
 Home Assistant met
 IP-adres, poort en
 gebruikersgegevens.

kunt aanpassen aan specifieke projectvereisten. Deze mate van controle is vooral handig voor complexe toepassingen waarbij u de prestaties moet optimaliseren, moet integreren met niet-standaard hardware of geavanceerde functies zoals AI en machine learning-algoritmen moet implementeren.

Bovendien zorgt aangepaste firmware voor een eenvoudigere integratie met een groot aantal platformen buiten Home Assistant, zoals cloud-gebaseerde diensten, aangepaste dashboards en andere IoT-systemen. U kunt ook meer verfijnde beveiligingsmaatregelen implementeren, zoals aangepaste encryptieprotocollen of geavanceerde verificatiemechanismen, om ervoor te zorgen dat uw gegevens over het hele netwerk veilig zijn. Bovendien kan aangepaste firmware worden geoptimaliseerd voor specifieke gebruiksscenario's, waardoor de overhead wordt verminderd en de systeemefficiëntie wordt verbeterd, wat vooral belangrijk is in omgevingen met beperkte resources.

De software

De firmware is geschreven om de ESP32 te verbinden met een WiFi-netwerk en MQTT te gebruiken voor communicatie, zodat de energiegelgegevens naar een Home Assistant-server kunnen worden gestuurd voor monitoring- en automatiseringsdoeleinden. In **listing 1** ziet u een voorbeeldversie van de code; de volledige code en alle hardwarebestanden zijn beschikbaar op de GitHub-repository van dit project [2].

Het project berust op twee belangrijke bibliotheken: de *ATM90E32*-bibliotheek, geleverd door CircuitSetup [3], die de communicatie met het energiemeet-IC afhandelt, en de *MQTTPubSubClient*-bibliotheek [4], die de communicatie beheert als een MQTT-client. De *ATM90E32*-bibliotheek is essentieel voor het verzamelen van gegevens van de chip van de energiemeter, inclusief meetgegevens zoals spanning, stroom en vermogen. Tijdens het ontwikkelproces kwam een belangrijke uitdaging naar voren bij het integreren van MQTT met gebruikersnaam- en wachtwoordverificatie. Hoewel veel MQTT-bibliotheken de basistaken aankunnen, viel *MQTTPubSubClient* op als een van de weinige die compatibel was met de ESP32 en de noodzakelijke verificatiefuncties ondersteunde.

De software begint met het opnemen van de noodzakelijke bibliotheken voor de netwerkverbinding, SPI-communicatie, interfacing met het energiemeet-IC en het beheren van de MQTT-communicatie. Configuratiegegevens voor WiFi en MQTT worden opgeslagen in een apart configuratiebestand. De `setup()`-functie initialiseert de seriële poort voor debugging, stelt de ATM90E32-energieremeter in met de gespecificeerde parameters en maakt verbinding met het WiFi-netwerk en de MQTT-broker. Deze initiële setup zorgt ervoor dat de ESP32 klaar is om te communiceren met Home Assistant en andere MQTT-compatibele platformen.

In de hoofdloop `loop()` wordt de MQTT-client continu bijgewerkt om de verbinding met de broker in stand te houden.

De MQTT-data als sensoren definiëren in Home Assistant

Om de gegevens te controleren die uw ESP32-energiemeter via MQTT verstuurt, moet u deze meetwaarden definiëren als sensoren in Home Assistant. Dat gaat als volgt:

Het configuratiebestand openen:

Open uw Home Assistant-configuratiebestand (*configuration.yaml*) met File Editor of een andere teksteditor.

MQTT-sensoren definiëren:

Voeg in het *configuration.yaml*-bestand de volgende configuratie toe om uw MQTT-sensoren te definiëren:

```
mqtt:
  sensor:
    - name: Line Voltage A
      unique_id: esp32_voltage_a
      state_topic: "esp32energymeter/lineVoltageA"
      unit_of_measurement: "V"

    - name: Line Current A
      unique_id: esp32_current_a
      state_topic: "esp32energymeter/lineCurrentA"
      unit_of_measurement: "A"
```

Uw sensoren aanpassen:

Vervang *Line Voltage A*, *Line Current A* enzovoort door benamingen naar eigen keuze.

Zorg ervoor dat het *state_topic* overeenkomt met het topic dat wordt gebruikt in je ESP32-firmware voor het publiceren van de gegevens. De *unique_id* moet een unieke identificatie zijn voor elke sensor, zodat Home Assistant ze goed kan volgen en beheren.

Instellingen opslaan en Home Assistant herstarten:

Sla na het toevoegen van de sensordefinities het bestand *configuration.yaml* op en start Home Assistant weer op om de wijzigingen toe te passen.

De sensoren weergeven:

Zodra Home Assistant opnieuw is opgestart, moeten uw MQTT-sensoren verschijnen op het dashboard, zodat u de energiegegevens in realtime kunt controleren.

Als de verbinding op enig moment wordt verbroken, probeert de code automatisch opnieuw verbinding te maken. Daarnaast is de software geprogrammeerd om energiegegevens op te halen van de ATM90E32-chip en deze gegevens naar de MQTT-broker te sturen. Dankzij deze opzet kan Home Assistant het energieverbruik in bijna realtime monitoren, wat waardevolle inzichten oplevert voor domotica-doeleinden.

De `getEnergyData()`-functie speelt een cruciale rol in de software door verschillende energiegegevens te verzamelen van de ATM90E32-chip. Deze gegevens omvatten netspanning, stroom, actief, reactief en schijnbaar vermogen, vermogensfactor, fasehoek, frequentie en temperatuur. De verzamelde gegevens worden vervolgens gepubliceerd naar specifieke MQTT-topics, waardoor ze toegankelijk zijn voor monitoring en analyse binnen Home Assistant. Voor ontwikkelaars en gebruikers die debugging mogelijk maken, worden de energiegegevens ook weergegeven op de seriële monitor, zodat de gegevens eenvoudig kunnen worden gecontroleerd en gevalideerd. Om ervoor te zorgen dat de ESP32 een stabiele verbinding met het netwerk en de MQTT-broker behoudt, is de functie `connect()` opgenomen. Deze functie handelt het herverbindingsproces af als de WiFi- of MQTT-verbinding

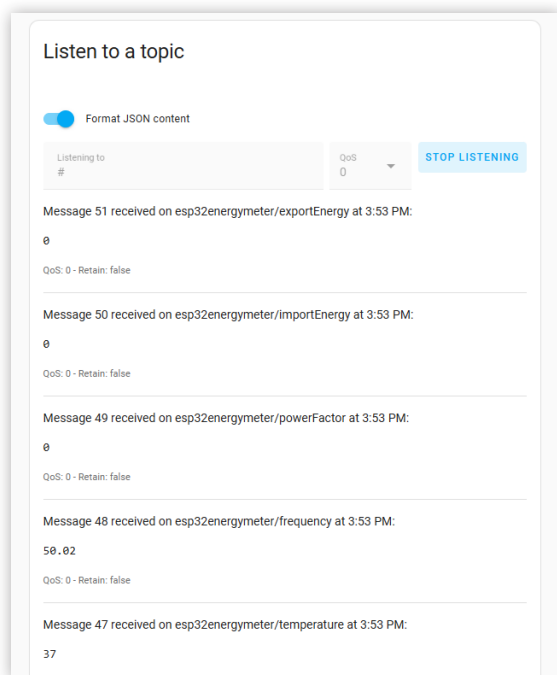
wordt verbroken. Debugberichten geven real-time feedback over de verbindingstatus en een LED wordt gebruikt om een geslaagde verbinding te signaleren.

Een MQTT-broker inrichten in Home Assistant

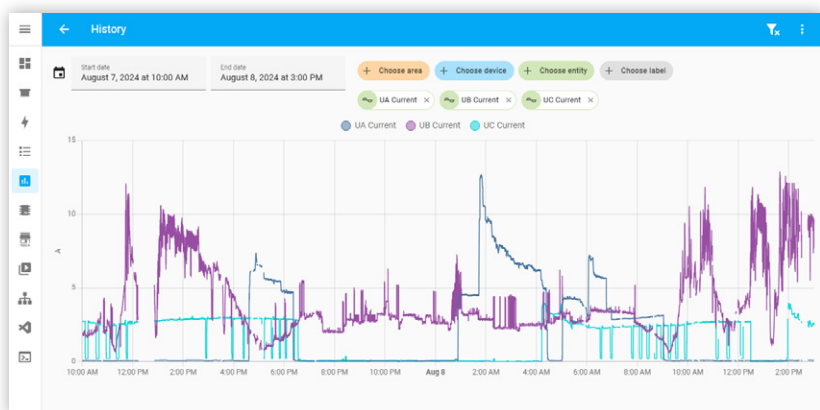
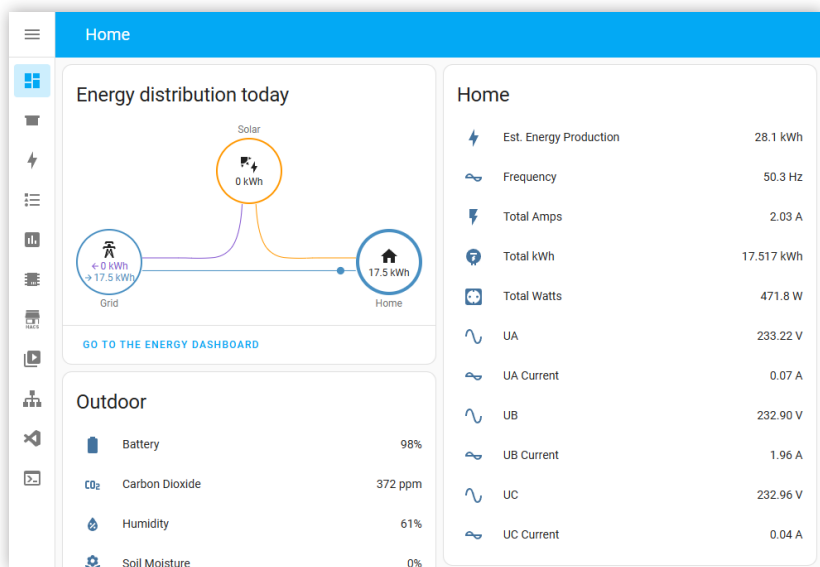
Om de ESP32-energiemeter het MQTT-protocol te laten gebruiken voor het verzenden van de meterstanden, moet u eerst een MQTT-broker inrichten. Een MQTT-broker kan worden ingericht op bijna elke computer die verbonden is met uw thuisnetwerk. Het kan geconfigureerd worden op uw PC met behulp van een Docker-container, direct geïnstalleerd worden op een Raspberry Pi, of zelfs gehost worden op een cloudserver voor toegang op afstand. Om het echter eenvoudig te houden en naadloos te integreren met uw smart home-setup, gaan we het inrichten op Home Assistant.

Om de MQTT Add-on in Home Assistant te installeren, gaat u eerst naar uw Home Assistant-dashboard door te navigeren naar de URL ervan. Ga vanuit de zijbalk naar *Settings*, dan *Add-ons* en open de *Add-on Store*. Zoek naar *MQTT*, waar u de officiële Mosquitto-broker in de resultaten zou moeten vinden. Klik erop en selecteer *Install*; de installatie kan enkele ogenblikken duren.

Figuur 3. Testen van de MQTT-verbinding door u te abonneren op alle onderwerpen om er zeker van te zijn dat berichten worden ontvangen door Home Assistant.



Figuur 4. Aangepast dashboard in Home Assistant; dit visualiseert real-time energiegegevens van de ESP32-energiemeter.



Figuur 5. Voorbeeld van grafieken die in Home Assistant zijn gemaakt om het energieverbruik te monitoren in het History-dashboard.

Eenmaal geïnstalleerd kunt u de MQTT-broker configureren door de instellingen aan te passen – meestal zijn de standaardinstellingen voldoende, maar u kunt en moet een specifieke gebruikersnaam en wachtwoord instellen. Na de configuratie klikt u op de **Start**-knop om de MQTT-broker te activeren, en activeert u de **Start on Boot**-optie om ervoor te zorgen dat hij automatisch start wanneer Home Assistant wordt herstart, zoals getoond in **figuur 1**.

Vervolgens moet u MQTT-integratie configureren binnen Home Assistant. Ga naar **Settings**, dan **Devices & Services**, en selecteer **Integrations**. Klik op **Add Integration**, zoek naar **MQTT** en selecteer het. Home Assistant zal automatisch de draaiende MQTT-broker detecteren. Indien gevraagd, configureer de MQTT-instellingen, zoals het IP-adres van de broker – meestal localhost als het op hetzelfde apparaat draait als Home Assistant (u kunt ook *core-mosquitto* invoeren om hetzelfde resultaat te krijgen). Blijf voor de poort bij de standaard 1833, en zorg ervoor dat de gebruikersnaam en wachtwoord die van een huidige gebruiker in Home Assistant zijn, zoals in **figuur 2**. U kunt ook een aparte gebruiker maken in Home Assistant, alleen voor MQTT in dit geval.

Nu moeten we ervoor zorgen dat Home Assistant is ingesteld om berichten te ontvangen van uw ESP32-energiemeter. U kunt de verbinding testen door u te abonneren op een topic “#” om alle berichten te ontvangen die naar uw MQTT-broker worden gestuurd (**figuur 3**). Deze pagina is toegankelijk door te klikken op **Configure** in **Integration entities** in het MQTT integration menu-item.

U kunt nu uw ESP32 Energy Meter verbinden met de MQTT-broker op Home Assistant. In het bestand *config.ino* van de ESP32-firmware stelt u **HOMEASSISTANT_IP** in op het IP-adres van uw Home Assistant-instantie. Configureer vervolgens de **DEVICE_NAME**, **USER_ID**, en **PASSWORD** als u authenticatie hebt ingesteld op de MQTT-broker. Nadat u deze instellingen hebt geconfigureerd, flasht u de ESP32 met de firmware en zorgt u ervoor dat hij verbinding maakt met de MQTT-broker. Eenmaal verbonden zal de ESP32 Energy Meter beginnen met het publiceren van energiegegevens naar de MQTT-broker, die u kunt monitoren in Home Assistant door u te abonneren op de juiste MQTT-topics.

Om tenslotte energiegegevens zichtbaar te maken in Home Assistant, worden de MQTT-entiteiten automatisch aangemaakt voor elk onderwerp dat uw ESP32 Energy Meter publiceert. Zo niet, wat soms kan gebeuren, dan moet u de MQTT-entiteiten definiëren in het *configuration.yaml* bestand in Home Assistant. U kunt de instructies volgen die worden vermeld in het kader **De MQTT-data als sensoren in Home Assistant definiëren**.

Daarna kunt u de sensor-entiteiten vinden onder **Settings** in Home Assistant, dan **Devices & Services** en **Entities**. Gebruik deze entiteiten om aangepaste dashboards te

maken in Home Assistant, zodat u real-time energiegegevens kunt visualiseren, grafieken kunt maken en waarschuwingen kunt instellen op basis van verbruiksdrempels zoals getoond in **figuur 4** en **figuur 5**. Met MQTT en Home Assistant kunt u ook acties automatiseren op basis van energiegegevens zoals getoond in **figuur 6**, integreren met andere slimme apparaten en krijgt u waardevolle informatie over het energieverbruik van uw huis.

Beginners raad ik aan om de *Getting Started*-handleiding van Home Assistant [5] te lezen, die een uitgebreide inleiding geeft voor het inrichten en gebruiken van Home Assistant. Daarnaast kunt u de *MQTT Integration*-documentatie van Home Assistant [6][7] raadplegen voor gedetailleerde instructies over het configureren van MQTT en het effectief integreren van uw apparaten. Deze bronnen zullen u helpen om aan de slag te gaan met Home Assistant en MQTT, waardoor uw smart home efficiënter en gebruiksvriendelijker wordt. ◀

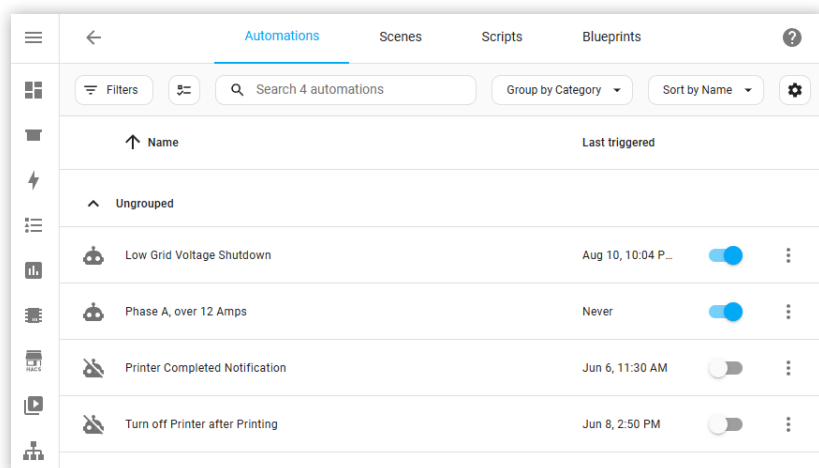
vertaling: Hans Adams — 240349-03

Over de auteur

Saad Imtiaz, Senior Engineer bij Elektor, is een mechatronica-ingenieur met uitgebreide ervaring in embedded systemen en productontwikkeling. Tijdens zijn carrière heeft hij samengewerkt met een breed scala aan bedrijven, van innovatieve startups tot gevestigde globale ondernemingen, om geavanceerde prototypes en projecten te ontwikkelen. Saad heeft een rijke achtergrond, waaronder een periode in de luchtvaartindustrie en het leiden van een technologische startup, en brengt een unieke mix van technische expertise en ondernemersgeest met zich mee in zijn rol bij Elektor. Hier draagt hij bij aan projectontwikkeling in zowel software als hardware.

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via saad.imtiaz@elektor.com of naar de redactie van Elektor via redactie@elektor.com.



Figuur 6. Geautomatiseerde acties in Home Assistant gebaseerd op energiegegevens die zijn ontvangen van de ESP32-energiemeter via MQTT.



Gerelateerde producten

- > **Home Assistant Green**
www.elektor.nl/20725
- > **Raspberry Pi 5 (2 GB RAM)**
www.elektor.nl/20951

WEBLINKS

- [1] Saad Imtiaz, "Project Update #3: ESP32-gebaseerde energiemeter", Elektor juli/augustus 2024: <https://elektormagazine.nl/240244-03>
- [2] ESP32 Energy Meter Github-repository: <https://github.com/ElektorLabs/esp32-energymeter>
- [3] ATM90E32 Arduino-bibliotheek door CircuitSetup: <https://github.com/CircuitSetup/ATM90E32>
- [4] MQTTPubSubClient-bibliotheek door hideakitai: <https://github.com/hideakitai/MQTTPubSubClient>
- [5] Getting started, Home Assistant: <https://www.home-assistant.io/getting-started/>
- [6] MQTT Integration, Home Assistant: <https://www.home-assistant.io/integrations/mqtt/>
- [7] MQTT Sensor, Home Assistant : <https://www.home-assistant.io/integrations/sensor.mqtt/>