

ESP32-camera

zo simpel – heeft niet eens WiFi



Bera Somnath (India)

Heb je een idee voor een cameratoepassing? Probeer het eerst zelf te doen voordat je een nieuwe kant-en-klare camera koopt. Met een ESP32-board met camera en enkele extra componenten bouw je een eenvoudige maar effectieve oplossing op maat.



Alle smartphones kunnen foto's maken, opslaan en weergeven. Het bouwen van nog een camera is dan ook eigenlijk onzin, tenminste voor mij. Tenzij het echt eenvoudig is. Met een paar goedkope modules zoals een ESP32-board met camera, een klein OLED-display, een realtime-klok en een triggermechanisme kunt je je eigen batterijgevoede camera bouwen die foto's maakt die je volledig onder controle hebt. Door op een knop te drukken of door aanwezigheidsdetectie met een bewegingssensor of zelfs getriggerd door een contactloze temperatuursensor, maakt hij (onbemerkt) een foto van iemand wiens lichaamstemperatuur te hoog is. Zulke toepassingen liggen voor de hand omdat je je smartphone niet op deze manier kunt gebruiken.

ESP32-Cam

Door een PIR-sensor aan het ESP32-Cam-board toe te voegen, kun je het zo aanpassen dat het onbemerkt foto's van indringers maakt! Of je kunt met alleen de ESP32-plus-camera en het OLED-display een handige kleine webcam bouwen die verbinding maakt met een van de beschikbare netwerken in de buurt en de videostream vervolgens op het intranet publiceert. Door de port forwarding-functie van je modem of router te gebruiken, kun je de afbeelding ook op internet publiceren. Deze functie is erg handig voor diegenen die een tweede camera nodig hebben voor demodoeleinden van

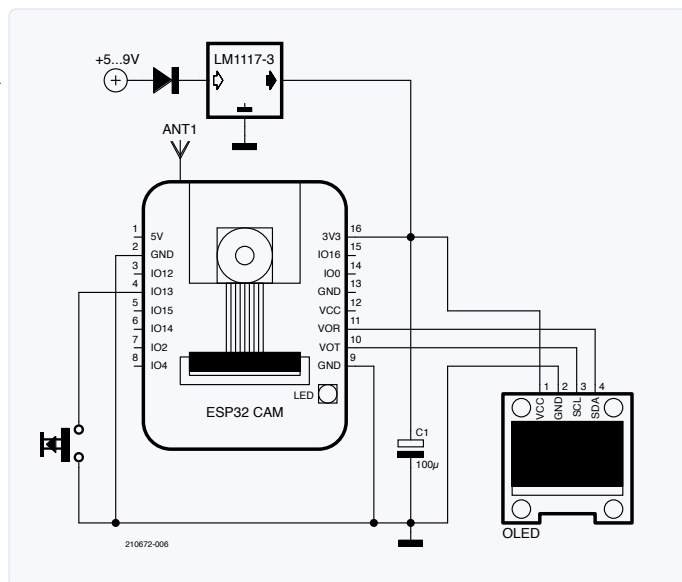
hun webinars en/of online-lessen. De kosten van de ESP32-Cam-module die we in dit artikel gebruiken bedragen ongeveer € 10.

Eerst een coole demo

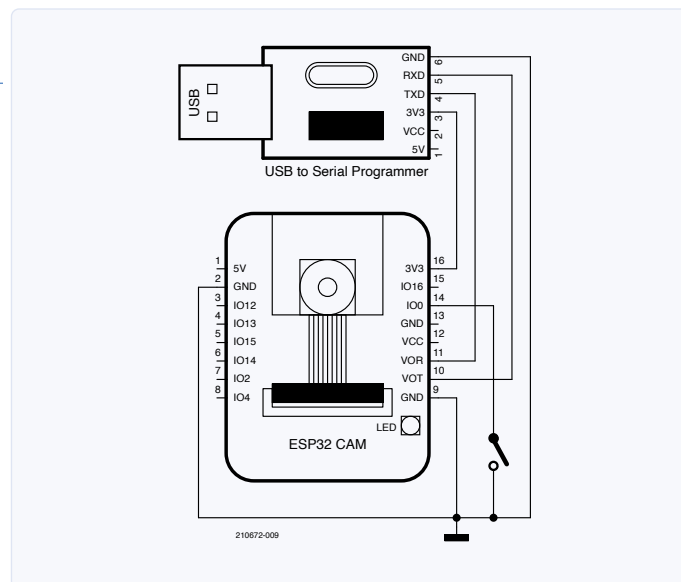
Na het installeren van de nieuwste Arduino IDE en het meest recente ESP32-package met behulp van de Boards Manager van de IDE, selecteer je als board het "AI Thinker ESP32-CAM" board (Tools -> Board -> ESP32 Arduino) en de poort waarop het is aangesloten (Tools -> Port). Open vervolgens het ESP32-webcam voorbeeldproject (File -> Examples -> ESP32 -> Camera -> Webserver). Probeer het uit; het zal je verrassen! De beide onderstaande projecten borduren hierop voort.

Project 1: webcam met OLED-display

Het webcamproject maakt het niet mogelijk om per menu een WiFi-netwerk te selecteren (de netwerk-referenties zijn hardgecodeerd in het programma) en de enige manier om het IP-adres van de webcam te weten te komen, is door deze aan te sluiten op een PC met een actieve seriële terminal. Hier breiden we dit project zodat het verbinding kan maken met elk beschikbaar netwerk in de buurt. Zodra het is verbonden, wordt op het kleine OLED-display weergegeven met welk netwerk het is verbonden en wat het IP-adres is. Als de verbinding om de een of andere reden wordt verbroken,



Figuur 1. Voeg een OLED-display toe aan de ESP32-Cam voor wat extra gemak.



Figuur 2. De ESP32-Cam wordt niet geleverd met ingebouwde USB/serieel-interface. Om de sketch te laden heb je daarom een USB/serieel-adaptor nodig. Om naar de programma-laadmodus te gaan, moet de resetknop van de ESP32 worden ingedrukt (of de voeding moet worden uit- en ingeschakeld) terwijl IO0 naar massa wordt getrokken.

wordt opnieuw geprobeerd verbinding te maken met een van de bekende netwerken.

Bouw van de webcam

Figuur 1 toont hoe je het OLED-display aansluit op het ESP32-Cam-board. Er zijn maar heel weinig GPIO-pinnen over voor gebruik nadat de SD-kaart en de camera zijn aangesloten. De enige beschikbare pinnen die we kunnen gebruiken zijn IO1, IO3, IO4, IO12, IO13 (zie kader **Aantekeningen van het Elektor Lab**). IO1 ('UoT') en IO3 ('UoR') worden ook gebruikt als seriële poort voor het laden van het programma naar de ESP32 (in flash-programmeermodus). Daarom mogen deze pinnen tijdens het laden van programma's helemaal nergens mee zijn verbonden.

In dit webcamproject kunnen we probleemloos pinnen gebruiken die zijn gereserveerd voor de pinnen van de SD-kaart, omdat we de SD-kaartfunctie niet gebruiken. We hebben echter IO1 en IO3 gebruikt als I²C-pinnen voor het aansluiten van het OLED-display. Op die manier hebben we zoveel mogelijk vrije GPIO-pinnen ter beschikking. Vergeet niet de I²C-bus los te koppelen bij het laden van een programma in de ESP32 (**figuur 2**).

Gebruik

Het werkingsprincipe van de ESP32-webcam met OLED-display is heel eenvoudig gebleven. De ESP32 zoekt naar de WiFi-netwerken die als een lijst in het programma zijn opgenomen en probeert verbinding te maken met een van deze. Zodra de verbinding tot stand is gebracht, verschijnt op het OLED-display de statusmelding dat de camera gereed is, samen met het netwerk-ID en het IP-adres dat aan de webcam is toegewezen. Om toegang te krijgen tot de camera, maak je verbinding met het aangegeven netwerk en ga je in een browser naar het IP-adres van de webcam. Het poortadres is en blijft standaard (8080).

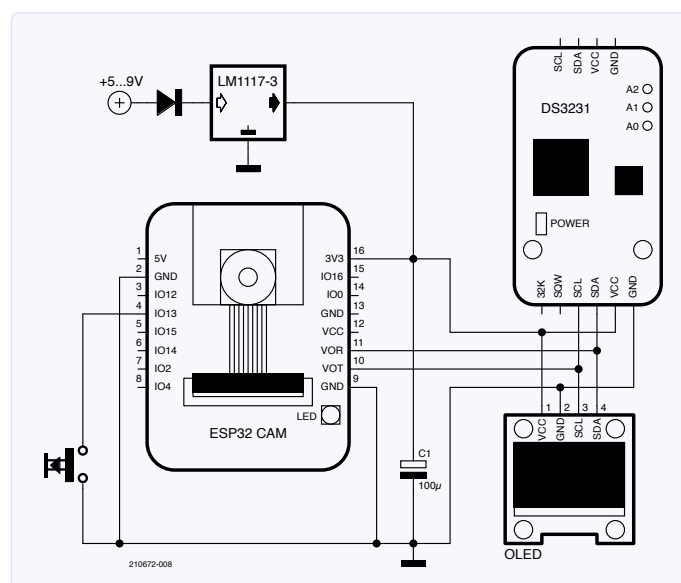
De software voor de ESP32-webcam plus OLED-display kan worden gedownload van [1]. Merk op dat de bibliotheek "ESP8266 and ESP32 OLED driver for SSD1306 displays" vereist is (we gebruikten versie

4.1.0) [2]. Deze kan worden geïnstalleerd met de bibliotheekmanager van de Arduino IDE. Vergeet niet de I²C-bus los te koppelen voordat je een sketch laadt.

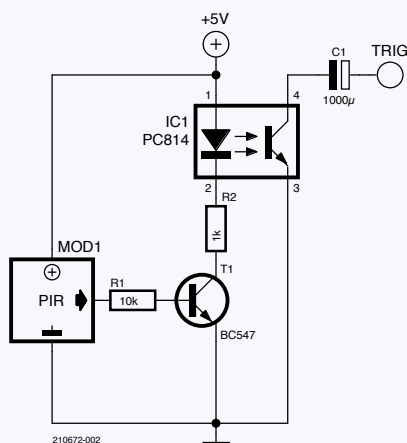
Project 2: getriggerde camera met OLED en RTC

In dit project wordt telkens wanneer er iets gebeurt een foto gemaakt. Foto's worden met een volgnummer en timestamp opgeslagen op een SD-kaart.

De schakeling in kwestie (**figuur 3**) is een uitbreiding van de vorige schakeling. Een DS3231 (of DS1307) realtime-klokmodule (RTC) wordt aan de I²C-bus toegevoegd en een triggersignaal wordt



Figuur 3. Een drukknop en een realtime-klokmodule veranderen de webcam van figuur 1 in een getriggerde bewakingscamera die ook timestamps genereert.



Figuur 4. We kunnen een 5V-PIR-sensor gebruiken – sommige werken zelfs op 3,3 V – om de ESP32-camera te activeren. Voor volledige galvanische scheiding raden we echter aan om een optocoupler (of een relais) te gebruiken.

aangesloten op IO13. Dit kan een drukknop zijn of een PIR-sensor (figuur 4) of iets anders dat een korte laag-actieve puls genereert wanneer een foto moet worden gemaakt.

IO4 wordt gebruikt als lamp: de on-board superheldere witte LED van de ESP32-Cam wordt gebruikt als extra lichtbron voor de camera. IO13 is de trigger-ingang voor het maken van foto's. Het programma voorziet de bestandsnaam van datum en tijd en slaat deze op de SD-kaart op. Vervolgens gaat hij in de deep sleep-modus om de batterij te sparen wanneer er niets te doen is en geen foto's worden gemaakt. Een laag-actieve puls op IO13 wekt de camera om een nieuwe foto te maken.

Dit programma bestaat alleen uit de functie **setup**; de functie **loop** wordt niet gebruikt. De reden hiervoor is het gebruik van de deep sleep-modus. Wanneer het apparaat uit deze modus ontwaakt, start het opnieuw op en voert het de functie **setup** uit en maakt het een foto voordat het terugkeert naar deep sleep, zodat de functie **loop** nooit wordt bereikt.

De software voor de getriggerde camera kan worden gedownload van [1]. Naast de OLED-driverbibliotheek die nodig is voor het eerste project [2], heeft dit project ook de bibliotheek "Rtc by Makuna" nodig (we gebruikten versie 2.3.5) [3]. Ook deze bibliotheek kan worden geïnstalleerd met de bibliotheekmanager van de IDE. Vergeet niet de I²C-bus los te koppelen voordat je een sketch uploadt (figuur 2).

Conclusie

De ESP32-Cam is erg goedkoop, maar biedt toch veel mogelijkheden. Door meerdere van deze camera's op een rij te installeren, kunnen ze worden gebruikt als een vast opgestelde scanner voor grotere objecten, bijvoorbeeld om foto's van de kentekenplaten van

naderende auto's te maken bij snelheidsovertredingen. Een enkele camera met PIR-sensor kan worden gebruikt voor 'snapshots' van vogels of andere dieren. Er zijn ook uitgebreidere toepassingen denkbaar. Door bijvoorbeeld gezichtsdetectie en een contactloze temperatuurscanner toe te voegen, kan het systeem enige beveiliging tegen Covid bieden. ◀

210672-03

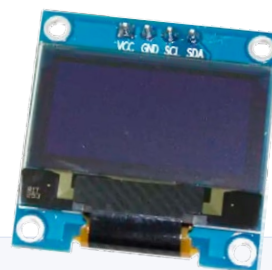
Vragen of opmerkingen?

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



ONDERDELENLIJST

- ESP32-Cam board
- SD-kaart
- 0,96" I²C SSD1306-compatibel OLED-display
- DS3231/DS1307 I²C RTC-module
- USB/serieel-adaptor (kabel)
- LM1117 3,3V-regelaar
- PIR-sensor of drukknop
- Optioneel: batterij, condensatoren enzovoort



GERELATEERDE PRODUCTEN

- ▶ **ESP32-Cam-CH340 Development Board (SKU 19333)**
www.elektor.nl/19333
- ▶ **0.96" I²C OLED Display (SKU 18747)**
www.elektor.nl/18747
- ▶ **SparkFun Real Time Clock Module - RV-8803 (Qwiic) (SKU 19646)**
www.elektor.nl/sparkfun-real-time-clock-module-rv-8803-qwiic

WEB LINKS

- [1] Downloads voor dit artikel op Elektor Labs: <https://www.elektormagazine.com/labs/esp32-camera>
- [2] ESP8266 en ESP32 OLED-drivers voor SSD1306-displays: <https://github.com/ThingPulse/esp8266-oled-ssd1306>
- [3] Rtc van Makuna: <https://github.com/Makuna/Rtc>
- [4] Getriggerde camera (versie van Elektor Labs): <https://www.elektormagazine.com/210672-03>

Aantekeningen van het Elektor Lab

Veel ESP32-Cam-kaarten worden samen met een praktische USB/serieel-converter dochterprint verkocht.* (Misschien moeten we het een moederbord noemen omdat de ESP32-module erop wordt aangesloten en van daaruit gevoed wordt?). Deze combinatie (figuur 5) maakt het programmeren van de ESP32 heel eenvoudig, maar het 'botst' met de I²C-interface op poorten IO1 en IO3 zoals door de auteur in zijn project gebruikt. Wanneer deze pinnen worden gebruikt, moet de I²C-poort telkens worden losgekoppeld telkens wanneer je de ESP32 wilt herprogrammeren; daarom zochten we naar een andere manier om de I²C-bus aan te sluiten.

In theorie kan de I²C-bus op de ESP32 worden geconfigureerd voor gebruik van bijna alle IO-pinnen, maar op het ESP32-Cam-board resulteert in allerlei opstartproblemen en PSRAM-geheugenfouten. Dit komt door de pull-up weerstanden op de I²C-bus.

Het ESP32-Cam-board heeft twee 8-polige uitbreidingsconnectoren. Op het eerste gezicht lijkt dat misschien in orde, maar bij nadere beschouwing is het niet zo eenvoudig. In de eerste plaats worden zes van de zestien pinnen gebruikt voor de voeding, waardoor er slechts tien overblijven voor IO. Zes hiervan (IO2, IO4, IO12, IO13, IO14 en IO15) worden gedeeld met de SD-kaarthouder. Dan blijven nog slechts de poorten IO0, IO1, IO3 en IO16 over.

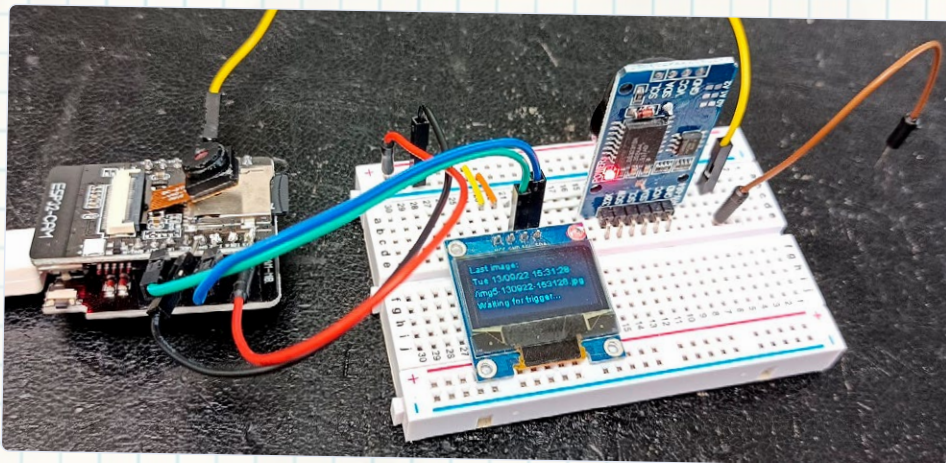
Voor het laden van een programma naar de ESP32 met aangesloten dochterprint zijn de poorten IO0, IO1 en IO3 nodig.

IO16 wordt gedeeld met de chip select-pen van de PSRAM-chip, dus deze kan alleen veilig worden gebruikt wanneer PSRAM niet nodig is.

De SD-kaart bibliotheek is standaard ingesteld op 4-bit datamodus, maar kan met

```
SD_MMC.begin("/sdcard",true);
```

in de 1-bit-modus worden gebracht. Hierdoor komen de poorten IO4, IO12 en IO13 vrij. Poort 4 wordt gedeeld met de (helder)witte flits-LED. Deze heeft een 47-kΩ pull-up weerstand en



Figuur 5. Het prototype van Elektor Labs is opgebouwd op breadboard. De ESP32-Cam met dochterprint is links te zien, de verticale module rechts is de RTC.

een 10+1 kΩ pull-down weerstand die een transistor aanstuurt. Poort 13 heeft ook een pull-up weerstand van 47 kΩ en wordt in dit project gebruikt als camera-triggeringang.

IO12 is een van die ESP32-pinnen die bij het opstarten met zorg moeten worden behandeld, omdat deze de waarde van VDD_ SDIO bepaalt, de spanning van de flash-geheugeninterface. Hij moet bij het opstarten op het ESP32-Cam-board omlaag worden getrokken.

Na wat experimenteren bleek dat de I²C-bus veilig kan worden verplaatst naar de poorten IO0 (SDA) en IO3 (SCL – of andersom, dat maakt niet uit, zolang de software maar correct is geconfigureerd). Door deze poorten te gebruiken botst de I²C-bus niet met de seriële programmeerpoort of met de opstartmodus, aangezien zowel IO0 als IO3 geacht worden hoog te zijn bij het opstarten.

Tot slot (en dat staat los van het bovenstaande) moet je beseffen dat IO33 is aangesloten op een rode LED aan de ESP32-modulezijde van het board en die je op elke gewenste manier kunt gebruiken (zolang je er rekening mee houdt dat deze laag-actief is).

We lieten ons een beetje meeslepen door het tweede project en pasten het programma aan (figuur 6). Het gebruikt niet langer geëmuleerd EEPROM voor het opslaan van het afbeeldingsnummer, maar in plaats daarvan een bestand op de SD-kaart. Wis gewoon het bestand "counter.txt" om de teller op nul te zetten. Ons programma kan worden gedownload van [4].



Figuur 6. Het werkt!

* Zie het kader Gerelateerde producten voor nog een andere versie van de ESP32-Cam-module.