ESP-Launchpad tutorial

van nul tot flashen in een paar minuten

Dhaval Gujar, Espressif

Ontwikkel je met Espressif-chips? ESP-Launchpad is een webgebaseerde tool die het evalueren en testen van firmware vereenvoudigt. Laten we het eens van dichterbij bekijken.

Dit artikel introduceert ESP-Launchpad, een webgebaseerde tool dat het evalueren en testen van firmware voor Espressif-chips tot een fluitje van een cent maakt. Het vereenvoudigt het opzetten van de ontwikkelomgeving, maakt gebruik van webbrowsers voor het flashen van firmware en biedt voordelen voor zowel ontwikkelaars als niet-ontwikkelaars.

Waarom ESP-Launchpad?

Wanneer je een open-source project bouwt, wil je een gemakkelijke manier hebben voor gebruikers om het te evalueren. Voor embedded ontwikkelaars betekent komt dat neer op het instellen van de development host en programmeertools, het klonen van het project, compileren en vervolgens het programmeren van de hardware.
Door variaties in de development host en zijn software-omgeving kan er daarbij een heleboel misgaan. Zelfs als alles goed werkt, is het voor gebruikers die geen ontwikkelaar zijn maar gewoon het project willen uitproberen, te veel werk.
Als je ontwikkelt met een van de Espressif-chips, heb je nu een betere manier om je project eenvoudig te evalueren. ESP-Launchpad vereenvoudigt dit proces, waardoor het evalueren en testen van firmware ontworpen voor het ESP-platform snel en probleemloos verloopt.

"Op uw plaatsen, klaar, Launchpad!"

ESP-Launchpad is een webgebaseerde ervaring die gebruik maakt van de nieuwste browsermogelijkheden om op een eenvoudige manier vooraf gebouwde firmware op ESP-apparaten te flashen. Dit omzeilt de noodzaak voor de traditionele speciale software-installaties voor het instellen van de development host, en maakt gebruik van de toegankelijkheid van webplatforms. Het benut de seriële webinterface die wordt ondersteund door de browsers Chrome, Edge en Opera om vanuit de browser te communiceren met het development board om het te programmeren en toegang tot de seriële console te bieden.
Laten we eens kijken hoe je gemakkelijk je firmware kunt lanceren met ESP-Launchpad.

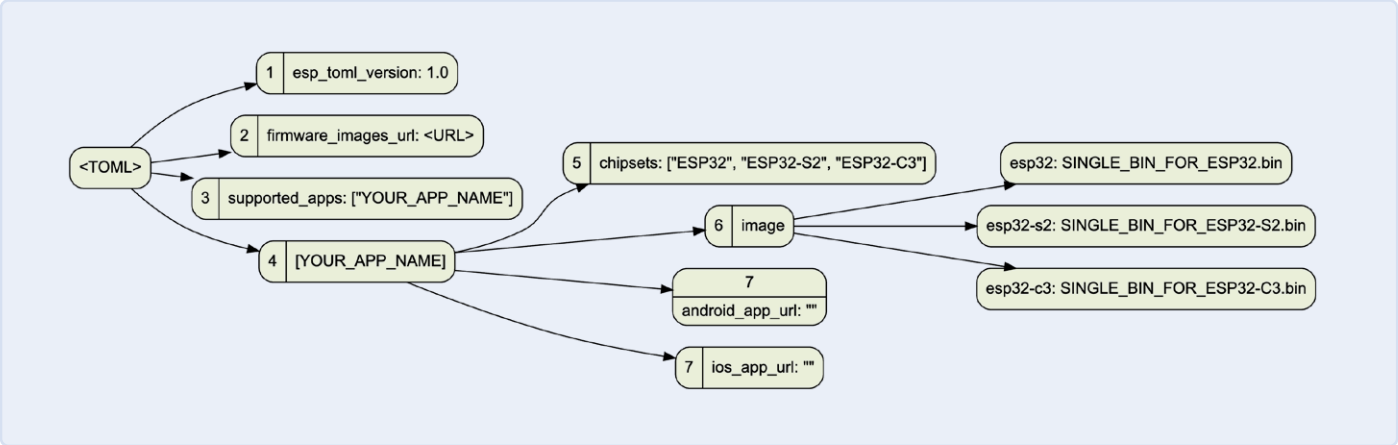
ESP-Launchpad is een webgebaseerde ervaring die gebruik maakt van de nieuwste browsermogelijkheden om op een eenvoudige manier vooraf gebouwde firmware op ESP-apparaten te flashen.

Voorbereiden van de firmware

Zodra de firmware is voltooid, is de volgende stap om deze te bouwen voor alle targets die je wilt ondersteunen en ze te converteren naar enkele binaries. Wees gerust, *esptool.py* (het one-stop-shop Python-hulpprogramma van Espressif voor alles wat met flashen te maken heeft) biedt een handige `merge_bin` optie die je kan helpen om op eenvoudige wijze zo'n binary te maken. Je kunt hier meer over vinden op onze documentatiepagina [1]. Deze enkele binaries moeten worden overgebracht naar een publiek toegankelijke URL die we later zullen gebruiken.
Opmerking: deze tutorial gaat niet verder in hoe je dat moet doen.
Leuk om te weten: het geheime ingrediënt dat ESP-Launchpad aandrijft is in feite een JavaScript-port van `esptool` met de naam `esptool-js`, die onder de motorkap gebruik maakt van de WebSerial API.

Begrip begint bij de basis: de kracht van TOML

Voordat we in de praktijk duiken, maken we ons eerst vertrouwd met de TOML-structuur van ESP-Launchpad, een centraal aspect van ESP-Launchpad. De TOML-configuratiebestanden van ESP-Launchpad dienen als eenvoudige configuratie-blauwdruk die je firmware-componenten, ondersteunde hardware en aanvullende telefoon-apps in detail beschrijven.



Figuur 1. Diagram van de TOML-structuur.

Hieronder een voorbeeld:

```
esp_toml_version = 1.0
firmware_images_url = "<URL to your firmware images
directory>"
supported_apps = ["YOUR_APP_NAME"]

[YOUR_APP_NAME]
chipsets = ["ESP32", "ESP32-S2", "ESP32-C3"]
image.esp32 = "SINGLE_BIN_FOR_ESP32.bin"
image.esp32-s2 = "SINGLE_BIN_FOR_ESP32-S2.bin"
image.esp32-c3 = "SINGLE_BIN_FOR_ESP32-C3.bin"
android_app_url = ""
ios_app_url = ""
```

Werp een blik op **figuur 1** om dit beter te begrijpen.

- `esp_toml_version` specificeert de versie van het TOML-schema.
- `firmware_images_url` is een publiek toegankelijke URL van de bestandsserver waar je firmware-binaries beschikbaar zijn om te downloaden. Pro tip: wij hosten onze TOML-bestanden en binaries beide op GitHub, met behulp van GitHub Pages.
- `supported_apps` is een array van de lijst met apps die je ondersteunt en waarvoor de binaire bestanden beschikbaar zijn. Je kunt meerdere apps hebben en de ESP-Launchpad UI zal deze apps tonen in de apps-dropdown. De drie waarden die we zojuist hebben bekeken, waren de key value-paren op het hoogste niveau.
- `[" YOUR_APP_NAME"]` – dit is een tabel die in feite een verzameling is van key value-paren. Deze bevat:
- `.......chipsets` – een matrix met een lijst van ESP-chipsets waarvoor je vooraf gebouwde firmware zult leveren. Let op de schrijfwijze hier, met uitsluitend hoofdletters, *ESP32-C3*.
- `.......image.<chip-naam>` – dit zijn sleutels die de naam van het binaire image associëren met elk van de ondersteunde targets. Dit wordt toegevoegd aan de `firmware_images_url` om de uiteindelijke URL voor de binary te verkrijgen. Let nogmaals op de schrijfwijze hier, met kleine letters, *esp32-c3*.
- `.......ios_app_url` en `android_app_url` zijn optionele maar speciale key value-paren onder dezelfde app-tabel, waarmee je links kunt weergeven in de vorm van QR-codes die gebruikers eenvoudig kunnen scannen nadat je app succesvol is geflashed.

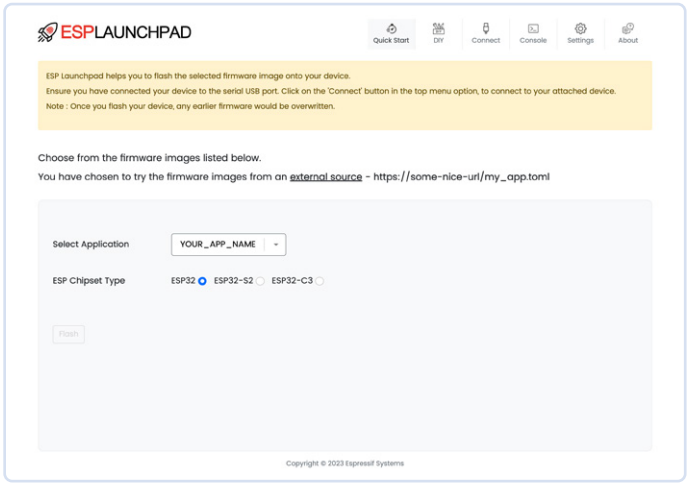
Laten we eens kijken hoe ESP-Launchpad eruit ziet (**figuur 2**) als we de voorbeeld-TOML geven die we zojuist hebben gemaakt, met de volgende denkbeeldige URL:

https://espressif.github.io/esp-launchpad/?flashConfigURL=https://some-nice-url/my_app.toml

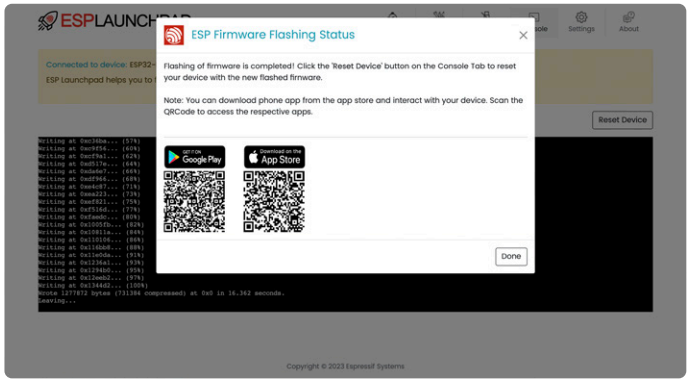
Oké, dat ziet er goed uit! En nu?

Slechts drie eenvoudige stappen voor de gebruiker!

- Aansluiten: sluit het ESP-apparaat aan op de seriële USB-poort van je computer.
- Verbinden: gebruik het tool-menu om verbinding te maken met het apparaat.
- Kiezen en flashen: selecteer de vooraf gebouwde firmware en flash deze.



Figuur 2. ESP-Launchpad met voorbeeld-configuratie geladen.



Figuur 3. Het laatste scherm van ESP-Launchpad.

Na het flashen wordt de gebruiker begroet met een console die het apparaatlog toont en een popup met de QR-codes van de telefoon-apps, als je die hebt toegevoegd aan de TOML (**figuur 3**). Gefeliciteerd, je hebt zojuist je app gepubliceerd met ESP-Launchpad! Pro-tip: we hebben een badge gemaakt (**figuur 4**) die je kunt toevoegen aan de *README* of webpagina van je project en die een hyperlink toevoegt naar de flash config-URL van je app! Je kunt meer informatie vinden in de *README.md* van het project [2].

Figuur 4: Voeg deze badge toe!

ESP-Launchpad: waar firmware en community elkaar ontmoeten

Uniek aan ESP-Launchpad is de mogelijkheid om gebruikers hun firmware-apps te laten delen zodat anderen ze kunnen proberen. Door te verwijzen naar een eenvoudig configuratiebestand kunnen ontwikkelaars bepalen waar de vooraf gebouwde binaire bestanden van hun firmware vandaan komen, welke hardware wordt ondersteund en zelfs linken naar aanvullende telefoon-apps. Deze holistische benadering zorgt ervoor dat de firmware niet alleen wordt gedeeld als een binair bestand, maar als de ervaring die de ontwikkelaar voor ogen heeft.

Tot slot

Het ware potentieel van elke tool wordt pas gerealiseerd wanneer het in handen is van de gebruikers. We raden je daarom aan om ESP-Launchpad te verkennen, te integreren in je workflow en je ervaringen te delen. Je inzichten en bijdragen zijn van onschatbare waarde bij het verfijnen van ons aanbod. Laten we samen de open-source evaluatie-ervaring naar een hoger plan tillen.

230596-03

Vragen of opmerkingen?

Stuur een e-mail naar de auteur via dhaval.gujar@espressif.com of naar de redactie van Elektor via redactie@elektor.com.

Over de auteur

Dhaval Gujar is engineer bij Espressif Systems, met een focus op embedded systemen. Zijn passie is de dynamische combinatie van technologieën zoals cloud, IoT en meer. Gefascineerd door het zich steeds ontwikkelende technologielandschap verdiept Dhaval zich met enthousiasme in moderne technologische veranderingen.

WEBLINKS

[1] Basiscommando's – binaries combineren voor flashen met merge_bin: https://tinyurl.com/esp32mergebinaries
[2] De GitHub-repository van dit project: https://github.com/espressif/esp-launchpad

Gerelateerde producten

- **ESP32-DevKitC-32E**
  www.elektor.nl/20518
- **LILYGO T-Display-S3 ESP32-S3 Development Board (met Headers)**
  www.elektor.nl/20299