

Aan de slag met de ESP32-C3 RISC-V MCU

Mathias Claußen (Elektor)

Benieuwd naar de ESP32-C3 van Espressif? Het is een single-core alternatief voor de ESP8266 en werkt met de open RISC-V instructieset-architectuur. Laten we die eens bekijken.

Op de ESP32-C3 van Espressif is met spanning gewacht. Hij heeft maar één core in plaats van de gebruikelijke twee processorkernen in andere ESP32-controllers. Maar deze kern werkt met de gratis open RISC-V instructieset-architectuur die concurreert met de op ARM gebaseerde controllers die veel worden ingezet voor IoT-toepassingen. Ik heb onlangs de ESP32-C3 en de ESP32-C3-DevKitC-02 aan de tand gevoeld. Dit zijn mijn ervaringen.

Wie mij volgt op Twitter (@ElektorMathias) [1] heeft misschien een aantal ESP32-C3 DevKits op mijn bureau zien liggen. En wie onze *Lab Notes* in juni 2021 [2] heeft gelezen weet ook dat er aan de software-ondersteuning voor de ESP32-C3 nog wordt gewerkt. De ontwikkelkit die we hier bespreken is uitgerust met de revision 2-versie van de ESP32-C3-chip. Deze voorlopige versies komen met een (dubbelzijdig bedrukt) A4'tje met een overzicht van de problemen die tot nu toe bekend zijn.

Revisie 2 lijdt onder meer aan slapeloosheid. Dat wil zeggen: hij trekt een hoop stroom in de deep sleep-modus, en de USB/JTAG seriële adapter in de chip werkt niet. Bij de recentere revision 3 zouden deze problemen opgelost moeten zijn, zoals ik al meldde op Twitter [3].

Ook aan Espressif's ESP-IDF-repository voor ondersteuning van de ESP32-C3 wordt nog gewerkt. Die kan dus ook nog bugs bevatten. Omdat daar de Arduino-ondersteuning op is gebouwd, zullen die bugs ook op dat platform terechtkomen. Op het moment dat ik dit schrijf, moet de Development Release van het Arduino Support Package worden geïnstalleerd in de Arduino-IDE. Dat gaat door de volgende link in te vullen bij *Voorkeuren* in het veld *Meer Board Managers*:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

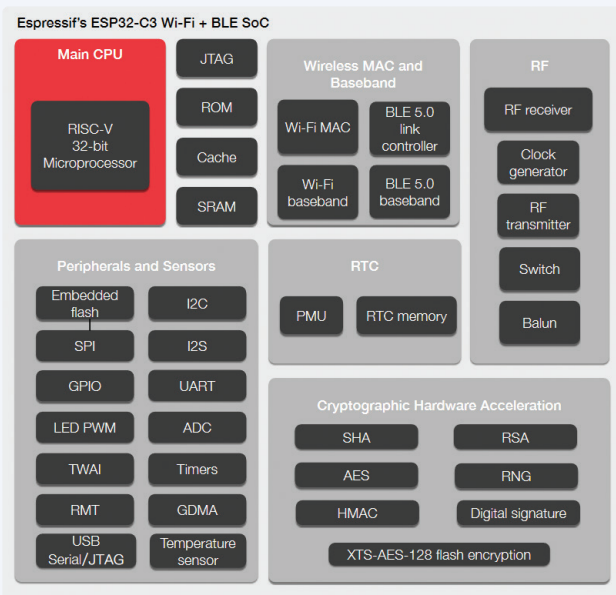
Dit installeert versie 2.0.0-rc1. Kopers van ESP32-C3-ontwikkelkits zouden de laatste rev.3-versie van de chip moeten krijgen, zoals te lezen is op het forum bij ESP32.com [4]. De voorlopige rev.2 was alleen op een beperkt aantal van de eerste ontwikkelkits gemonteerd.

De ESP32-C3 DevKitC-02

Eerst kijken we naar de ESP32-C3 en de DevKitC-02 (figuur 1). De ESP32-C3-chip is in feite een opvolger van de Espressif ESP8266. Net als de ESP8266 heeft hij één core die geschikt is voor klokfrequenties tot 160 MHz, en een 2,4 GHz BGN WiFi-chip met een volledige TCP/IP-stack. Maar daar houdt de overeenkomst met de ESP8266 op. De ESP32-C3 bevat veel meer van de periferie die we kennen van de ESP32 van Espressif, de opvolger van de ESP8266. Naast WiFi beschikt



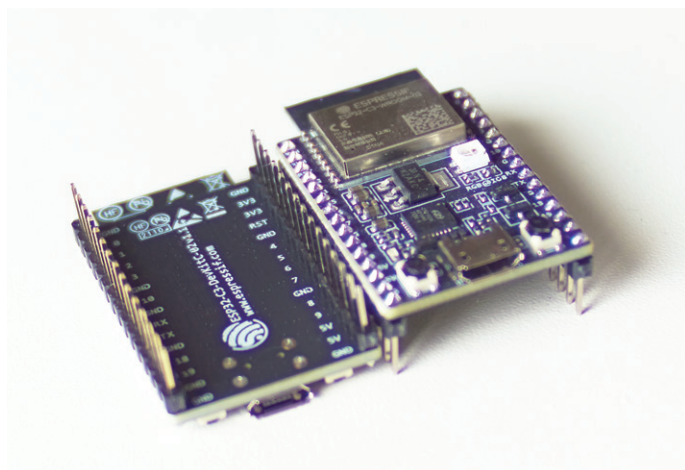
Figuur 1. De ESP32-C3 DevKitC-02.



Figuur 2. Blokschema van de ESP32-C3.

- > RISC-V CPU geklokt op 160 MHz
- > 400 KB SRAM (16 KB Flash-cache)
- > Geïntegreerde 2,4 GHz Wi-Fi (BGN)
- > Bluetooth LE 5.0
- > Crypto Hardware Accelerator
- > 22 programmeerbare GPIO's
- > 2 x 12-bit SAR ADC
- > 3 x SPI (ondersteunt SPI, dual SPI, Quad SPI en QPI)
- > 2 x UART (ondersteunt RS232, RS485 IrDA tot 5 MBd)
- > 1 x I2C (maximaal 800 kbit/s)
- > 1 x I2S
- > RMT (afstandsbesturing van periferie)
- > TWAI (CAN 2.0 b compatibel/ISO 11898-1)
- > PWM
- > Interne USB/JTAG-adapter

Tabel 1. Specificaties van de ESP32-C3.



Figuur 3. Beide kanten van de ESP32-C3-DevKitC-02.

de ESP32-C3 over BLE 5.0 en Bluetooth Mesh-communicatie. Hij heeft ook de nuttige GPIO-matrix, zodat bijna elke functie aan bijna elke pin kan worden toegewezen. In **figuur 2** ziet u het blokschema van de ESP32-C3, compleet met USB/serieel- en JTAG-adapter. In het blokschema van de controller in figuur 2 is duidelijk te zien, dat de ESP32-C3 een afstammeling van de ESP32 is (zie ook het overzicht van

de specificaties in **tabel 1**). Met zijn 384 kB RAM heeft de ESP32-C3 bijna vijf keer zoveel RAM als de ESP8266 (80 kB). Het belangrijkste verschil tussen de ESP32-C3 en alle andere ESP32- en ESP8266-chips is de processorkern. Terwijl zijn voorgangers werkten met de Tensilica L106 of LX6/LX7 [5] RISC-processor heeft de ESP32-C3 een RISC-V CPU. Dat betekent dat u compilers en andere programma's uit de RISC-V-toolchain kunt gebruiken met deze kern. Verbeteringen aan deze compilers en andere gereedschappen komen dus ten goede aan gebruikers van de ESP32-C3. De community hoeft geen toolchain op te zetten, want dat is al gebeurd bij de introductie van de ESP8266. Het DevKitC-02-board bevat een USB/serieel-converter en een WS2812-compatibele RGB-LED. Een gedetailleerd schema van het board vindt u onder [6]. In **figuur 3** ziet u boven- en onderzijde van het board.

Blinky - maar dan anders

De WS2812-compatibele RGB-LED op de kaart wordt bestuurd met behulp van een serieel communicatieprotocol. We kunnen hem dus niet in- en uitschakelen via een I/O-pen. Gelukkig bevat de Adafruit NeoPixel-bibliotheek routines om dit type LED aan te sturen. Als u die bibliotheek nog niet hebt geïnstalleerd, kunt u hem op de normale manier toevoegen aan uw Arduino-IDE. Met de code in **listing 1** laat u de RGB-LED rood knipperen. U hoeft de code niet aan te passen voor uitvoering op de ESP32-C3. Uploaden van de code gaat net zo gemakkelijk als bij een ESP32. Dankzij de I/O-matrix kunt u ook WS2812-LED's via andere pinnen aansturen.

Chip-type en -versie uitlezen

Bij ESP32's kunnen we op de chip opgeslagen informatie over de basiseigenschappen uitlezen. In **listing 2** ziet u deze informatie, uitgelezen via de seriële interface op 115200 baud (de communicatie op 9600 baud van chip-revisie 2 met de ingebouwde USB/serieel-converter levert onleesbare tekens op). Hier kunt u zien welke versie van de chip er op uw ESP32-C3-kit zit. Hopelijk ziet u dat u revisie 3 hebt, dan hebt u geen last van de bugs die bekend zijn van revisie 2.

Porten van ESP32-projecten

Heel nuttig bij de ESP32-C3 zijn de ingebouwde WiFi- en BLE-communicatie. Samen met SPIFFS of het LittleFS-bestandssysteem voor het beheren van webpagina's en andere data op de ESP32, vormt dat een ideaal platform voor allerlei WiFi-toepassingen. Wie nog werkt met een oudere (1.X-)versie van de Arduino-IDE zal een gepatchte plugin [7] moeten installeren om files naar het ESP32-bestandssysteem te kunnen uploaden. De originele versie van *me-no-dev* [8] werkt niet met de ESP32-C3. Als voorbeeld hebben we de code [9] gebruikt voor een ESP32 Mini-NTP-server [10]. Deze kan in het eigen netwerk de kloktijd beschikbaar maken via NTP en is geschreven voor het Arduino-framework. Na enkele aanpassingen met betrekking tot de pintoewijzing kan de code zonder problemen worden gecompileerd en in de ESP32-C3 worden geladen. Dit eenvoudige voorbeeld laat zien dat veel bestaande code en kennis van de ESP32 rechtstreeks bruikbaar is voor de ESP32-C3, dus zelfs nieuwelingen in de RISC-V-omgeving moeten die overstap kunnen maken. De meeste bestaande ESP32-voorbeelden draaien op de ESP32-C3, en omdat FreeRTOS ook hier in de achtergrond aanwezig is, kunt u gebruik maken van alle voordelen en nadelen die



Listing 1. Blinky .

```
/* This sample needs the Adafruit NeoPixel library */
#include <Adafruit_NeoPixel.h>
#define PIN      8
#define NUMPIXELS 1
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  pixels.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
  pixels.clear(); // Set all pixel colors to 'off'
}

void loop() {
  delay(1000); // wait for a second
  pixels.setPixelColor(0, pixels.Color(255, 0, 0));
  pixels.show();
  delay(1000); // wait for a second
  pixels.setPixelColor(0, pixels.Color(0, 0, 0));
  pixels.show();
}
```

dat met zich mee brengt, net als op de ESP32.

Omdat de ESP32-C3 op dit moment alleen wordt ondersteund in een ontwikkeltak van de ESP32-chips in de Arduino-omgeving, ondersteunen nog niet alle IDE's de chip volledig. Naarmate het framework meer volwassen wordt, is betere ondersteuning te verwachten.

Een single-core oplossing

De ESP32-C3 is een voordelig alternatief voor de ESP8266 met heel veel van de periferie van de ESP32 en maar één processorkern. De geïntegreerde USB/serieel- en JTAG-adapters maken het uitwisselen van bestanden en data via USB gemakkelijk. U kunt zelfs code gaan debuggen (als er niets gekks gebeurt in revisie 3 van de chip).

Dankzij het Arduino-framework kan bestaande code worden

WEBLINKS

- [1] @ElektorMathias op Twitter: <https://twitter.com/ElektorMathias/status/1386623477604626433>
- [2] Elektor lab notes juni 2021: <https://www.elektormagazine.com/news/elektor-lab-notes-june-2021>
- [3] Tweet over USB serieel/JTAG-ondersteuning op de ESP32-C3: <https://twitter.com/ElektorMathias/status/1392475706647584776>
- [4] Forum Thread betreffend ESP32 C3-revisies: <https://esp32.com/viewtopic.php?t=21040>
- [5] Wikipedia - Cadence Tensilica Products: <https://en.wikipedia.org/wiki/Tensilica>
- [6] ESP32 C3 DevKitC-02 schema: https://dl.espressif.com/dl/schematics/SCH_ESP32-C3-DEVKITC-02_V1_1_20210126A.pdf
- [7] ESP32 data-uploader voor Arduino IDE (patched versie): <https://github.com/lorol/arduino-esp32fs-plugin/releases>
- [8] ESP32 data-uploader voor Arduino IDE: <https://github.com/me-no-dev/arduino-esp32fs-plugin>
- [9] Github Repository Mini NTP Server met GPS: <https://github.com/ElektorLabs/180662-mini-NTP-ESP32>
- [10] Mini NTP Server met GPS in het Elektor Lab: <https://www.elektormagazine.com/labs/mini-ntp-server-with-gps>
- [11] Materiaal bij dit artikel: <https://bit.ly/3CD6iNs>

hergebruikt met de ESP32-C3; de ruime hoeveelheid RAM- en flash-geheugen maken grotere projecten mogelijk. We kunnen in de toekomst zeker vele ESP32-C3-boards van derden verwachten op de Europese markt en het zal interessant zijn om te kijken welke functionaliteit die dan bieden. Iedereen die thuis een ESP32-DevKitC-02 heeft, kan al aan de slag met het schrijven en testen van code voor deze omgeving. Listings van de hier gebruikte Arduino sketches zijn te vinden op onze GitHub-pagina [11]. 🚩

210466-03

Vragen of opmerkingen?

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



Listing 2. Uitlezen van de chip-informatie.

```
/* From the IDF documentation at https://github.com/espressif/esp-idf/components/esp_hw_support/include/
esp_chip_info.h */
/*
typedef enum {
    CHIP_ESP32    = 1, //!< ESP32
    CHIP_ESP32S2  = 2, //!< ESP32-S2
    CHIP_ESP32S3  = 4, //!< ESP32-S3
    CHIP_ESP32C3  = 5, //!< ESP32-C3
    CHIP_ESP32H2  = 6, //!< ESP32-H2
} esp_chip_model_t;
// Chip feature flags, used in esp_chip_info_t
#define CHIP_FEATURE_EMB_FLASH    BIT(0)    //!< Chip has embedded flash memory
#define CHIP_FEATURE_WIFI_BGN     BIT(1)    //!< Chip has 2.4GHz WiFi
#define CHIP_FEATURE_BLE          BIT(4)    //!< Chip has Bluetooth LE
#define CHIP_FEATURE_BT           BIT(5)    //!< Chip has Bluetooth Classic
#define CHIP_FEATURE_IEEE802154   BIT(6)    //!< Chip has IEEE 802.15.4
typedef struct {
    esp_chip_model_t model; //!< chip model, one of esp_chip_model_t
    uint32_t features;      //!< bit mask of CHIP_FEATURE_x feature flags
    uint8_t cores;          //!< number of CPU cores
    uint8_t revision;       //!< chip revision number
} esp_chip_info_t;
*/

void setup() {
    Serial.begin(115200);
}

void loop() {
    delay(5000);
    Serial.println("esp_chip_info()");
    Serial.println("-----");
    esp_chip_info_t info;
    esp_chip_info(&info);
    Serial.print("Chip Model: ");
    switch(info.model){
        case 1:{
            Serial.println("ESP32");
        }break;
        case 2:{
            Serial.println("ESP32-S2");
        }break;
```

Een bijdrage van

Tekst en illustraties: Mathias Claußen

Redactie: Jens Nickel, C.J. Abate

Vertaling: Evelien Snel

Layout: Giel Dols



GERELATEERDE PRODUCTEN

- **ESP-C3-12F-Kit Development Board with Built-in 4 MB Flash (SKU 19855)**
www.elektor.nl/19855

```
        case 4:{
            Serial.println("ESP32-S3");
        }break;
        case 5:{
            Serial.println("ESP32-C3");
        }break;
        case 6:{
            Serial.println("ESP32-H2");
        } break;
        default:{
            Serial.print("Unknown Chipmodel");
            Serial.println(info.model);
        }
    }
}

Serial.print("Features :");
if(info.features&CHIP_FEATURE_EMB_FLASH){
    Serial.print(" Embedded Flash ");
}
if(info.features&CHIP_FEATURE_WIFI_BGN){
    Serial.print(" WiFi (BGN) ");
}
if(info.features&CHIP_FEATURE_BLE){
    Serial.print(" BLE ");
}

if(info.features&CHIP_FEATURE_BT){
    Serial.print(" BT Classic ");
}
if(info.features&CHIP_FEATURE_IEEE802154){
    Serial.print(" IEEE802.155.4 ");
}
Serial.println("");
Serial.print("Cores: ");
Serial.println(info.cores);
Serial.print("Chip Revision: ");
Serial.println(info.revision);
Serial.println("-----");
Serial.println();
}
```