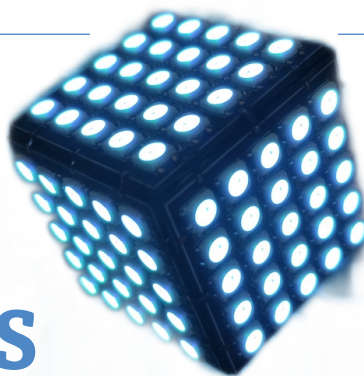


Magische RGB LED-kubus

hardware ontwerpen rond een RP2040



Mohamed Belkhir Sayari (Tunesië)

Zin in een RGB LED-kubus als blikvanger voor op je bureau of werktafel? Zou je het leuk vinden om je eigen ‘magische’ kubus te bouwen op basis van een Raspberry Pi RP2040? Zo ja, dan is deze kubus met 6 zijden met elk 25 LED's iets voor jou.



Figuur 1. De RGB LED-kubus.

Je kunt je eigen gadget maken op basis van je eigen printontwerp. Je hoeft alleen maar een paar tips te volgen om een professioneel ogend apparaat te krijgen. Laten we gelijk naar het hoofdonderwerp gaan. Ik zal vertellen hoe ik een RGB LED-kubus heb gemaakt die wordt aangestuurd door de Raspberry Pi RP2040-microcontroller (**figuur 1**). Ik zal ook uitleggen hoe je een RP2040 aan het werk zet, en ik zal een basisopzet presenteren om deze geweldige MCU in andere apparaten te gebruiken.

Ontwerp van de LED-kubus op basis van een RP2040

Sinds begin 2021 heeft het wereldwijde tekort aan elektronica-componenten het moeilijk gemaakt om aan goedkope, efficiënte MCU's te komen. Maar ik heb ontdekt dat de RP2040 een uitkomst kan zijn voor ontwerpers in deze moeilijke tijden, vanwege zijn compacte formaat, gebruiksgemak, goede prestaties en lage prijs. Toen ik aan de LED-kubus begon, was deze MCU vrij nieuw en waren er nog niet veel gedetailleerde projecten mee gebouwd, dus dit project bood me mijn eerste kans om met deze MCU aan de slag te gaan [1].

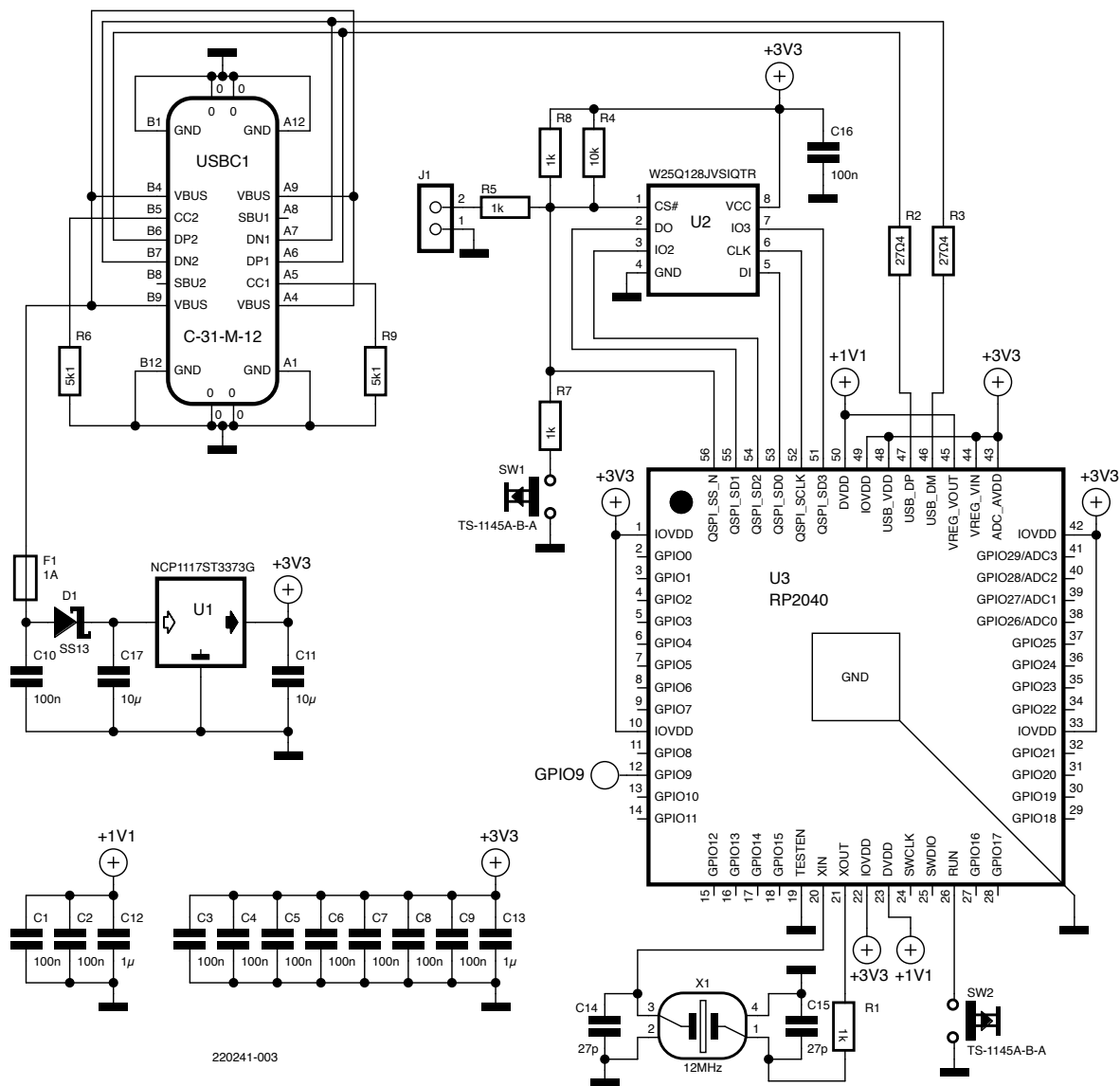
Ik wilde de meest geschikte opzet wat betreft voeding en code voor mijn RP2040 vinden; het was een min of meer toevallige keuze om een door de MCU aangestuurde 5x5-LED-matrix te bouwen. Maar al snel veranderde het idee van één matrix in een zesdelige gekoppelde matrix die een kubus vormde.

Om te beginnen is het belangrijk te weten hoe je met deze MCU moet omgaan, en moet je begrijpen wat de vereisten zijn. Raadpleeg daarvoor het uitgewerkte schema (**figuur 2**). Het schema is verdeeld in vier blokken: de MCU, voeding, USB-C en geheugen. We bekijken die blokken nu afzonderlijk.

Rondom de microcontroller

Het eerste blok met het label RP2040 in figuur 2 toont de componenten die de MCU nodig heeft, en je ziet dat daar veel condensatoren bij zitten! Ik raad aan om elk onderdeel in deze sectie te monteren om verrassingen te voorkomen wanneer je je hardware inschakelt. De condensatoren zijn hoofdzakelijk 100n-ontkoppelcondensatoren, en deze moeten bij elke voedingsaansluiting van de MCU worden gemonteerd.

Deze ontkoppelcondensatoren filteren de voedingsruis uit die wordt veroorzaakt door de andere elementen van de schakeling; deze ruis wordt door de condensatoren kortgesloten. De condensatoren doen daarnaast ook dienst als lokale voedingsbuffer en ondersteunen de interne schakelingen van de RP2040 door spanningsdalingen op te



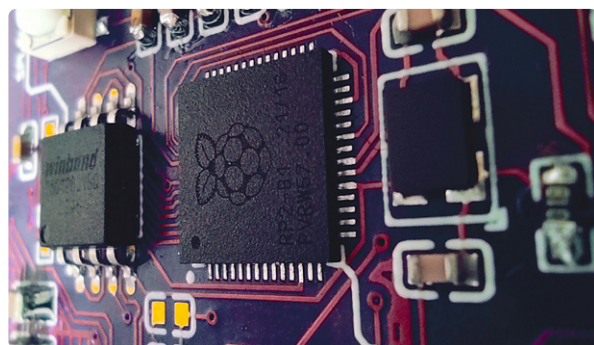
Figuur 2. Schema van de basis-hardware rond de RP2040.

vangen als plotselinge meer stroom nodig is.

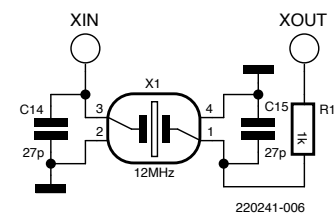
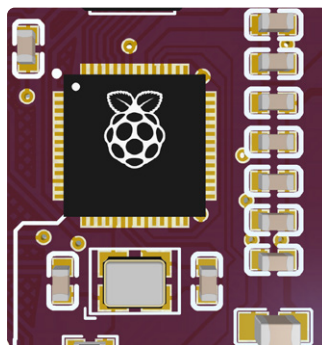
Nu begrijp je het belang van de ontkoppelcondensatoren. Maar dat is nog niet alles! Zorg er bij het printontwerp (**figuur 3**) voor dat de condensatoren zo dicht mogelijk bij de voedingspinnen van de MCU zitten. Dit maakt de filtering veel effectiever.

De RP2040 heeft zijn eigen interne oscillator, dus in principe heeft hij niet per se een externe oscillator nodig. Omdat deze interne oscillator

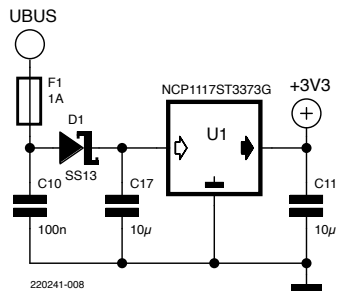
niet bijzonder stabiel is, verdient een externe 12MHz-oscillator aanbeveling (**figuur 4**). Wij hebben de X322512MSB4SI-oscillator gebruikt. (Vergeet niet dat we voor makkelijk verkrijgbare onderdelen hebben gekozen; je kunt natuurlijk gebruiken wat je wilt). De twee belangrijkste factoren waarmee bij kristallen rekening moet worden gehouden, zijn de lastcapaciteit (LC) en de equivalente serieweerstand (ESR). Je kunt de benodigde gegevens vinden in de datasheet van de compo-



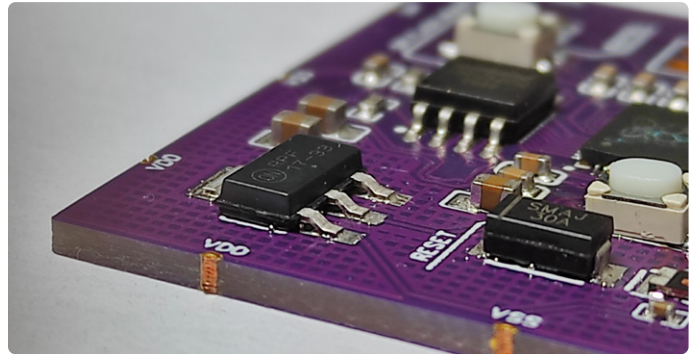
Figuur 3. Ontkoppelcondensatoren en positie van het kristal.



Figuur 4. De kristal-oscillator.



Figuur 5. De 3,3V-spanningsregelaar.



Figuur 6. De spanningsregelaar NCP111733T3G.

nent. De door ons gebruikte oscillator heeft een lastcapaciteit van 20 pF nodig. Daarom hebben we twee condensatoren (C15 en C14) parallel aan XIN en XOUT van het kristal geplaatst. De voor de hand liggende vraag hierbij is, hoe je de juiste waarde van de condensator kunt bepalen. De gewenste waarde van de lastcapaciteit (LC) via een eenvoudige formule uit C14 en C15:

$$LC = \frac{C14 \times C15}{C14 + C15}$$

Aangezien C14 en C15 dezelfde waarde hebben, is $LC = C14/2$, daarom hebben we gekozen voor 27 pF voor C14 en C15. Ja, ik weet dat $27/2 = 13,5$ pF minder is dan 20 pF, maar vergeet niet dat je een print gebruikt die extra capaciteit toevoegt via zide kopersporen; deze kan oplopen tot 7 pF. In totaal komen we uit op een capaciteit van 20,5 pF – dicht genoeg bij 20 pF.

De ESR-factor is ook te vinden in de datasheet van de fabrikant. In ons geval bedraagt deze 80 Ω; ik heb gemerkt dat die waarde samen met een weerstand van 1 kΩ (R1) voldoende is om oversturing van het kristal te voorkomen.

Voeding

Dit was de eerste keer dat ik te maken kreeg met een microcontroller die een spanning van 1,1 V nodig heeft. Maar ik maakte me daar geen zorgen over omdat de MCU al een interne low-dropout regelaar aan boord heeft die speciaal is gebouwd om het IC van 1,1 V te voorzien. Natuurlijk is een externe spanningsbron nodig; ik voed de RP2040 met 3,3 V via een externe spanningsregelaar (figuur 5).

De NCP1117 is een veel gebruikte spanningsregelaar, en hij is verkrijgbaar met verschillende uitgangsspanningen, dus let erop dat je de 3V3-versie gebruikt, met het typenummer 33T3G. Ik heb een zelfherstellende zekering (PTC) in de voedingsingang (VBUS) gebruikt om de schakeling te beschermen tegen overstroom.

De compacte afmetingen van deze component (figuur 6) maken hem geschikt voor ons kleine printontwerp. Vergeet niet dat we met deze print een kleine LED-kubus willen maken, dus eigenlijk moeten we al onze elektronica-componenten verdelen over een compacte print-layout van maximaal 45 × 45 mm. Dit is de belangrijkste reden waarom

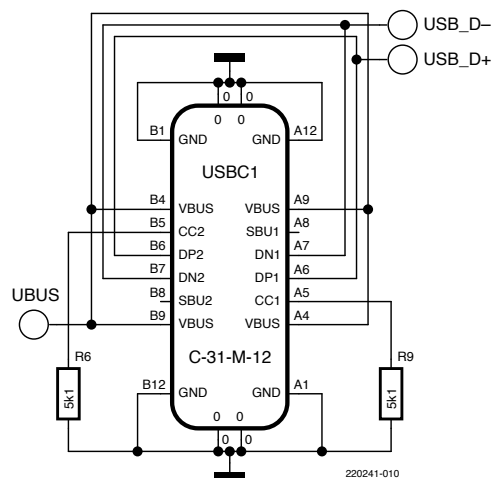
we kiezen voor componenten in kleine behuizing. Maar als je andere hardware op basis van de RP2040 wilt ontwerpen, ben je natuurlijk vrij om grotere componenten te gebruiken.

USB C-connector

We hebben geen USB/TTL-converter, omdat de RP2040 MCU een interne full-speed USB-interface heeft. We kunnen dus gewoon een USB-connector gebruiken en de USB_DP en USB_DM van de RP2040 aan te sluiten op de USB D+ en D- van de connector. Ik heb een USB type-C-connector gebruikt (figuur 7).

Je kunt de pull-down weerstanden (R9 en R6) weglaten omdat de MCU geen externe pull-up of pull-down weerstanden nodig heeft, maar omdat dit de meest geschikte USB-C-opzet is die we voor al onze projecten kunnen bedenken, monteren we ze zolang ze maar klein zijn. Dus heb ik een 0402-weerstand genomen om de print-layout overzichtelijk te houden. Het zorgde in elk geval voor een gemakkelijkere routing van de sporen. Met deze connector kan de print fungeren als Host of Device, afhankelijk van de bootmodus.

De connector moet op enige afstand van de rand van de print worden



Figuur 7. De USB-C aansluiting van de RP2040.



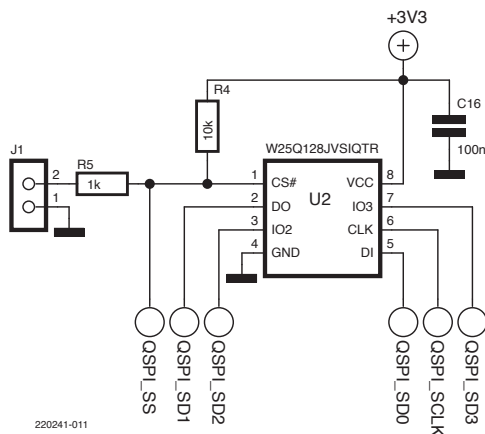
geplaatst, omdat we de zijanten van de kubus aan de rand met elkaar verbinden. In principe hebben we een vrije ruimte van ongeveer 1...2 mm langs de randen van de print nodig.

Flashgeheugen (voor het programma)

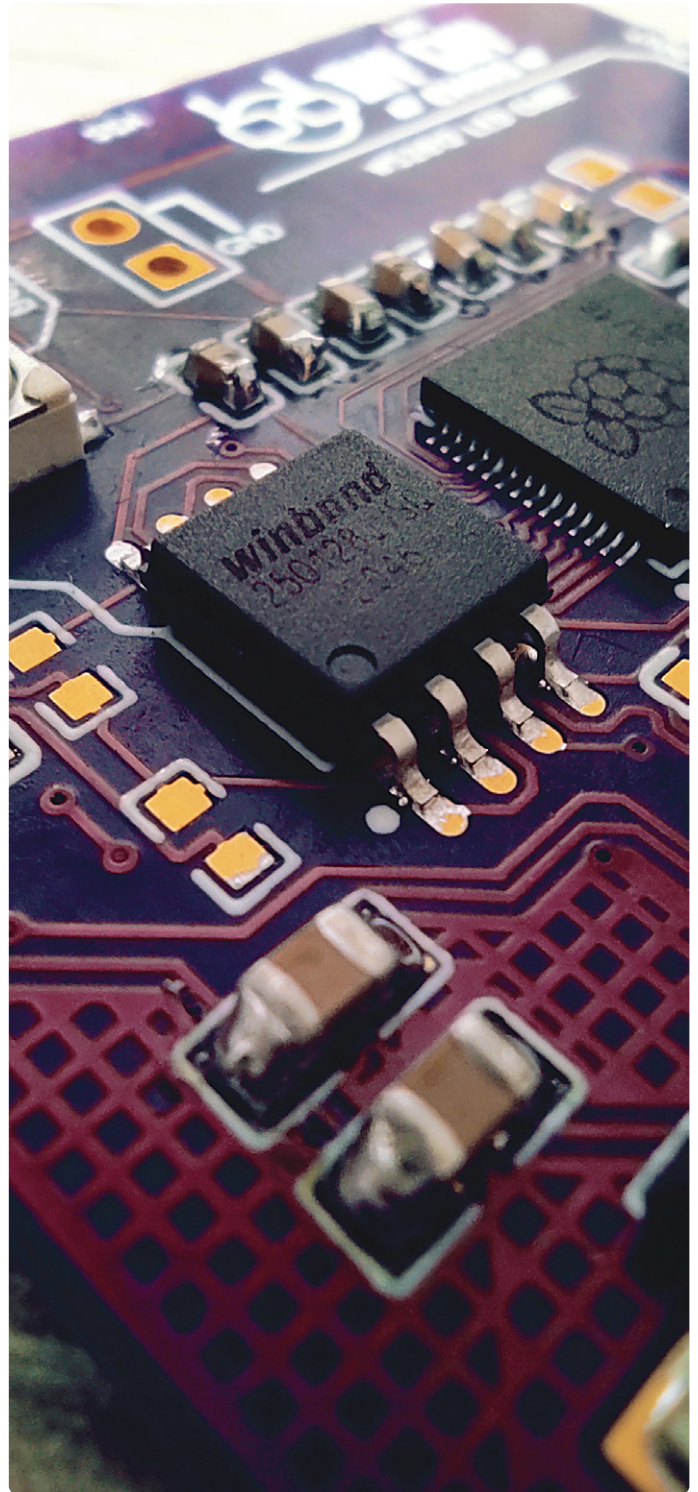
Vergeleken met de klassieke AVR- en ARM-microcontrollers die we gewend zijn, heeft de RP2040 een extern geheugen nodig voor de programmacode, zodat de RP2040 met die code opstart en die dan uitvoert. Over booten en het uitvoeren van code gesproken, we hebben het dan over snelle data-overdracht van de geheugenchip naar de MCU; dat is de reden voor onze keuze van een Quad Serial Peripheral Interface-flashgeheugen (QSPI) om de programmacode in op te slaan (**figuur 8**). Deze flashgeheugens zijn met verschillende opslagcapaciteit verkrijgbaar. In dit geval gebruiken we de W25Q128JVSQTR (**figuur 9**) met een capaciteit van 128 Mbit (16 MB); dat is de maximale geheugenruimte die de RP2040 via zijn databus kan adresseren.

De QSPI-pinnen moeten rechtstreeks worden aangesloten op de RP2040, en deze twee onderdelen moesten dicht bij elkaar op de print worden geplaatst om de verbindingen zo kort mogelijk te houden en om overdrachtsfouten door storing van dichtbij geplaatste componenten te voorkomen.

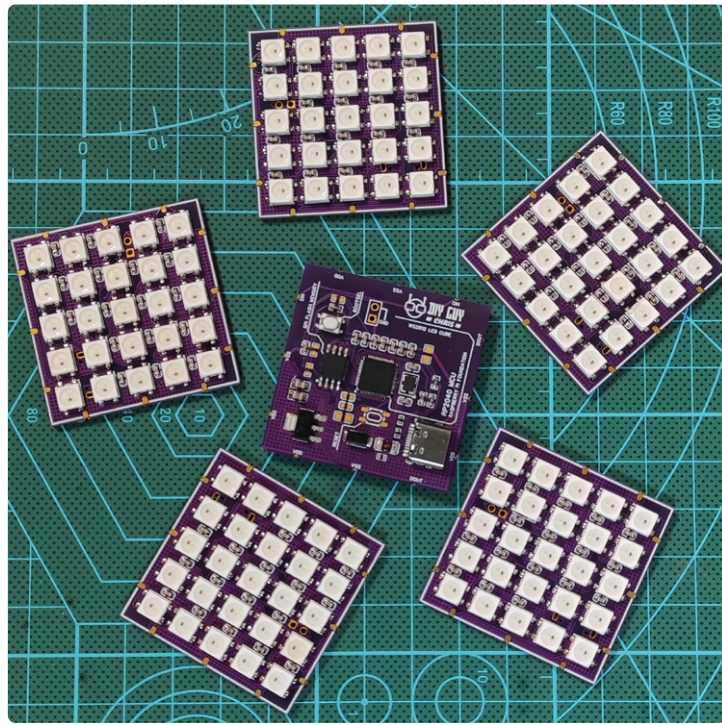
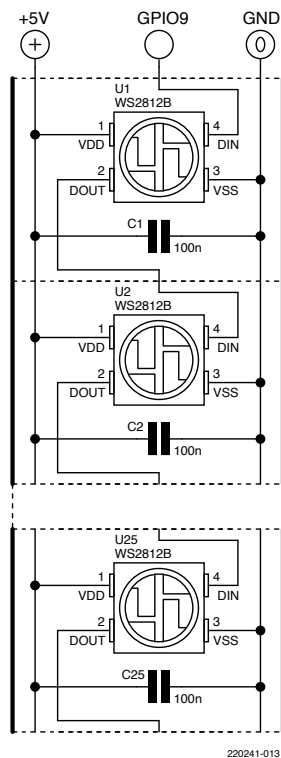
Bij de bespreking van de USB-C-opzet had ik al opgemerkt dat de RP2040 als Host of als Device kan worden gebruikt, afhankelijk van de bootmodus. Dit is afhankelijk van het niveau op de QSPI_SS-pin tijdens het booten van het geheugen (figuur 8). Hier zorgt weerstand R4 als pull-up naar 3,3 V voor de instelling van chip select (CS#) op hetzelfde niveau als zijn eigen 3,3V-voedingsspin bij het inschakelen. Bij het inschakelen zal de QSPI_SS-pin van de RP2040 automatisch hoog worden getrokken, maar de toestand van de pin blijft onbekend tijdens het omschakelen van het niveau. Weerstand (R4) zorgt voor een gedefinieerd niveau en garandeert dat het geheugen goed werkt en de programmacode naar de RP2040 wordt gestuurd.



Figuur 8. Het QSPI-flashgeheugen.



Figuur 9. De W25Q128 op ware grootte.



Figuur 10. De LED-schakeling en de opgebouwde printen.

Wat betreft de tweede bootmodus (storage device mode): dan moet jumper (J1) de QSPI_SS-pin naar massa trekken als we spanning op de print zetten. Het geheugen wordt dan herkend als USB-opslag en het programma kan er direct in worden gekopieerd. Een reset van de schakeling zet de bootmodus om en dan zal de RP2040 het gekopieerde programma uitvoeren.

En zo gaat het verder

De onderzijde van deze print bevat een in serie geschakelde 5×5 RGB LED-matrix (figuur 10). Dus als we overwegen om zes exemplaren van deze printplaat te gebruiken, komen we uit op een totaal van 150 LED's. Hoe kunnen we die allemaal aansturen met deze RP2040? Simpel: we gebruiken maar één pin van de processor om deze adresseerbare LED's aan te sturen. Heb je al eens gehoord van adresseerbare LED's (in ons project hebben we de WS2812B gebruikt)? Dat zijn LED's met een geïntegreerde controllerchip en een stuurschakeling voor het omzetten van het datasignaal, en dat alles in een behuizing van 5×5 mm. De waarden voor helderheid en kleur gaan van de RP2040-uitgang langs alle leds om de controller van elke LED te actualiseren.

We hebben deze print ontworpen om de mogelijkheden van een RP2040-MCU uit te proberen. Aangezien het bij dit kubusproject werkt, zal deze hardwareopzet zeker ook werken voor andere projecten. Volg gewoon de aanbevelingen en tips over de benodigde componenten en je zult je schakeling zonder grote verrassingen aan de praat krijgen. Alle bestanden bij dit project (schema's, Gerber-bestanden, RP2040-software en STL-bestanden voor 3D-printen) kunnen worden gedownload van [1].

220241-03

Over de auteur

Mohamed Belkhir Sayari – 'DIY GUY Chris' op YouTube (www.youtube.com/MEGADAS/) – is een Tunesische elektronica-R&D-ingenieur met het hart van een echte innovator. Hij beschouwt elk project als een leerzaam avontuur en een kans om zijn vaardigheden te tonen.

Vragen of opmerkingen?

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



GERELATEERDE PRODUCTEN

- > D. Ibrahim, *Raspberry Pi Pico for Radio Amateurs* (Elektor 2021) (SKU 20041) www.elektor.nl/20041
- > Raspberry Pi RP2040 Microcontroller (SKU 19742) www.elektor.nl/19742

WEBLINK

[1] DIY GUY Chris, "How to Build Magic RGB LED Cube (With Code and Files)," March 23, 2022: www.youtube.com/watch?v=A0OtHySzadk