

19

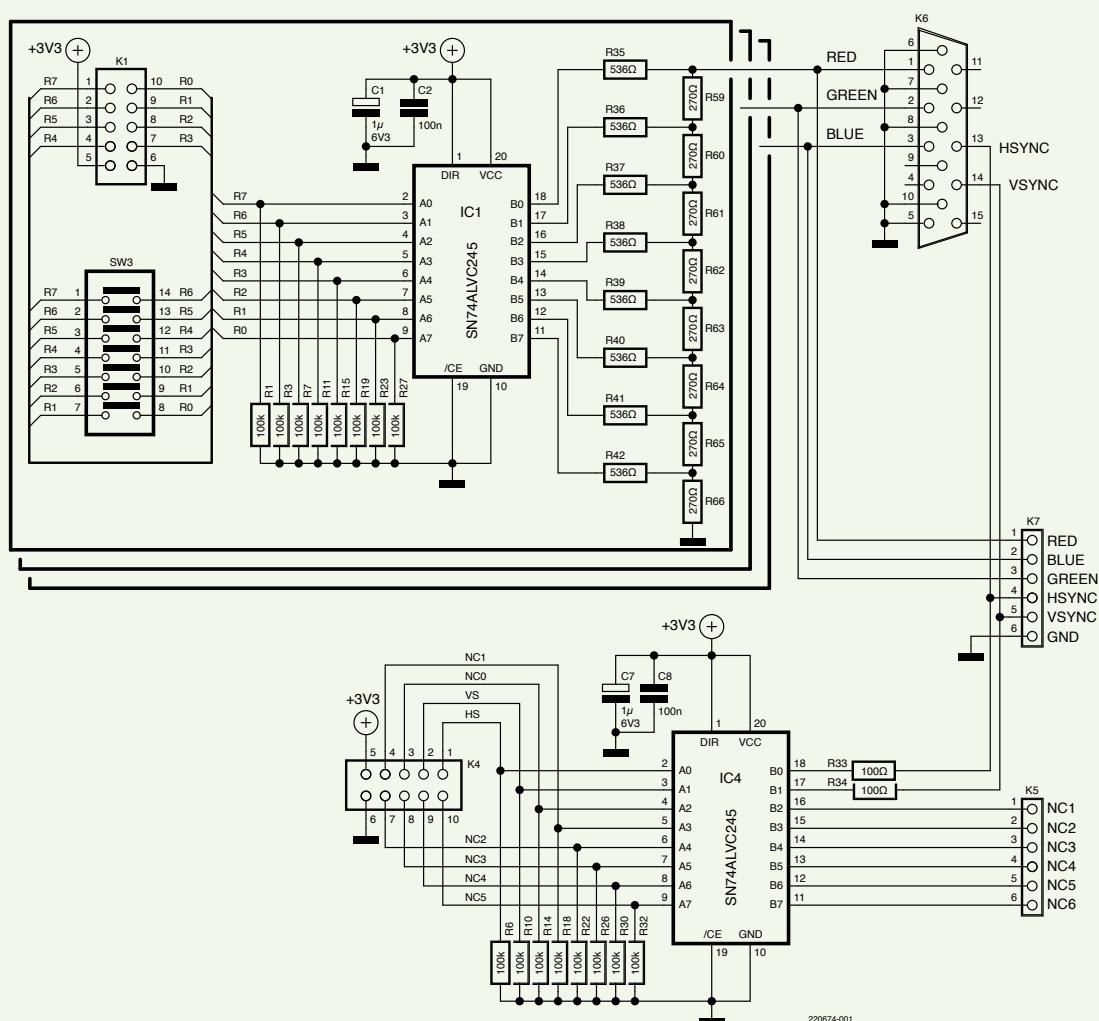
Programmeerbare video-DAC

voor elk formaat tot RGB888

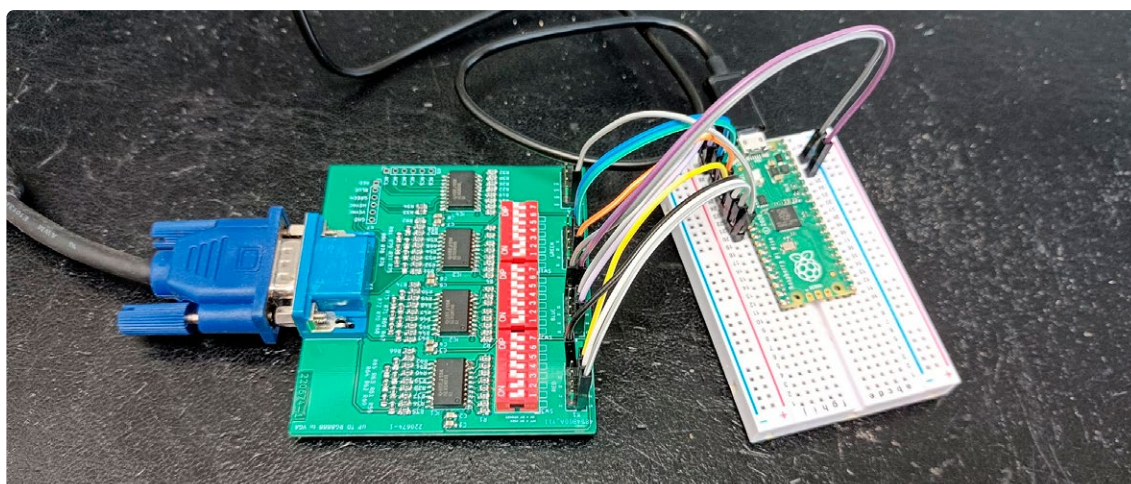


Mathias Claussen (ontwerp) en Clemens Valens (tekst)

Wanneer je experimenteert met microcontrollers en video in VGA-kwaliteit voor retro-gaming of een andere toepassing, heb je een digitaal/analoog-converter nodig. Met dit flexibele ontwerp kun je allerlei videoformaten uitproberen voordat je de uiteindelijke resolutie vastlegt.



Figuur 1. Hier is maar één kleurkanaal weergegeven; de beide andere zijn identiek.



Figuur 2. Het prototype in RGB332-modus, aangestuurd door een Raspberry Pi Pico.

In een niet zo grijs verleden had je speciale video-IC's nodig om een video-uitgang aan je computersysteem toe te voegen. Maar tegenwoordig kan bijna elke micro-controller een VGA-videosignaal 'on the fly' produceren. De resolutie en kleurdiepte hangen af van de hoeveelheid beschikbaar RAM en de snelheid van de MCU. Een eenvoudige digitaal/analoog-converter (DAC) is alles wat nodig is om digitale bitstromen om te zetten in analoge spanningen voor de rode, groene en blauwe kanalen van de VGA-monitor.

De vereiste woordbreedte van de DAC wordt bepaald door de resolutie en kleurdiepte van het gewenste video-signaal. Deze twee parameters zijn op hun beurt weer afhankelijk van de beschikbare rekenkracht in de MCU die het videosignaal genereert. Het is mogelijk dat aan het begin van het project nog niet duidelijk is waartoe je systeem in staat is, zodat je misschien liever begint met een DAC met geringe woordbreedte. Aan de andere kant is het mogelijk dat je, als een project bijna klaar is, merkt dat er nog ruimte is om de kwaliteit van het video-signaal te verbeteren. In dat geval zou je een DAC met een grotere woordbreedte willen.

Schakeling

De hier gepresenteerde schakeling kan in beide situaties worden gebruikt, en in alle andere situaties daartussen. Het is een driekanaals video DAC met programmeerbare woordbreedte, waarbij elk kanaal onafhankelijk van de andere kan worden ingesteld. Daarom kan hij worden gebruikt voor alles van RGB111 tot RGB888 (8bit-resolutie voor alle kleuren). Elk cijfer kan elke waarde hebben in een bereik van 1 tot 8; zo zijn RGB535 en RGB221 bijvoorbeeld mogelijke video-kleurformaten.

De schakeling is getekend in **figuur 1**. We zien hier slechts één kleurkanaal (rood), omdat de kanalen voor elke kleur hetzelfde zijn. De digitale video komt binnen op K1 met maximaal acht bit per kanaal. De bits worden gebufferd door IC1, die een R2R ladder-DAC aanstuurt. De DAC-uitgang is een videosignaal in het bereik van 0 V tot 0,7 V. Pull-down weerstanden aan de ingangen zorgen ervoor dat niet-aangesloten bits als nul worden gezien.

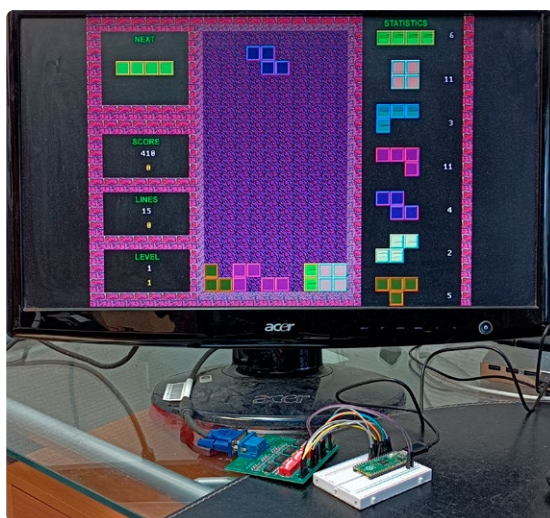
Het aantal bits per kleur wordt ingesteld met de DIP-schakelaars van SW3. Voor een 8bit-signaal moeten alle schakelaars uit staan. Om de woordbreedte te verkleinen, sluit je de schakelaars die onderaan beginnen met R1 en R0. Voor een woordbreedte van 4 bit sluit je bijvoorbeeld de vier onderste schakelaars. Sluit de bits van boven naar beneden aan, dat wil zeggen het meest significante bit aan respectievelijk R7, G7 of B7 en werk dan omlaag. Op die manier kan de video-output altijd zijn maximale waarde (0,7 V) bereiken in plaats van zwakker te worden. IC4 buffert de horizontale en verticale synchronisatie-signalen. De overige zes buffers kunnen voor iets anders gebruikt worden. Ze zijn beschikbaar op K5.

Om de video-DAC snel uit te proberen zonder uren te hoeven programmeren, sluit je hem aan op een Raspberry Pi Pico-board en een VGA-monitor (zie **figuur 2** en **tabel 1**). Gebruik HSYNC als synchronisatiesignaal.

Tabel 1. Raspberry Pi Pico pin map voor de RGB332 videodemo's van [1].

Raspberry Pi Pico	Signaal
GP0	B0
GP1	B1
GP2	G0
GP3	G1
GP4	G2
GP5	R0
GP6	R1
GP7	R2
GP8	HS
3V3	3V3
GND	GND
(GP19)	(Audio)

Figuur 3.
Pico speelt Tetris!



Download een demo van [1], programmeer deze in de Pico en geniet van de graphics (**figuur 3**). De demo's op [1] zijn in RGB332-formaat. Daarom moet je de onderste vijf schakelaars voor rood en groen en de onderste zes schakelaars voor blauw sluiten:

$$\text{aantal te sluiten schakelaars} = 8 - \text{woordbreedte voor de kleur}$$

Sommige demo's voeren audio uit op GPIO19 van de Pico. Zet een weerstand van 1,5 kΩ in serie met de uitgang en voeg een condensator van 10 nF naar massa toe om een eenvoudig laagdoorlaatfilter voor het audiosignaal te maken.

De ontwerpbestanden van het project zijn beschikbaar op [2].

220674-01

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.

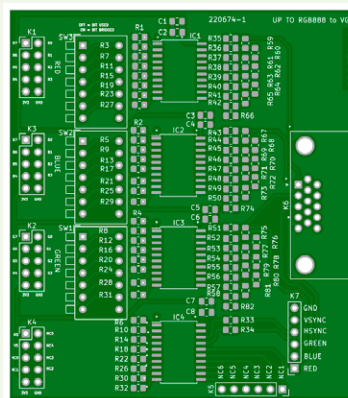
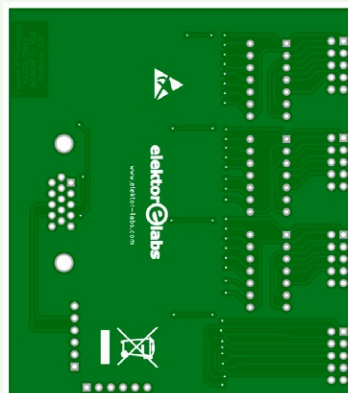


Gerelateerde producten

- > Joy-IT 10" Touchscreen for RPi incl. metal case and stand (HDMI/VGA/BNC/AV) <https://elektor.nl/18266>
- > Raspberry Pi Pico RP2040 H <https://elektor.nl/20223>



Onderdelenlijst



Weerstanden (0805):

R1...R32 = 100 k
R33,R34 = 100 Ω
R35...R58,R66,R74,R82 = 536 Ω
R59...R65,R67...R73,R75...R81 = 270 Ω

Condensatoren (0805):

C1,C3 C5,C7 = 1 μ
C2,C4,C6,C8 = 100 n

Halfgeleiders:

IC1,IC2,IC3,IC4 = SN74ALVC245 (SOIC-20W)

Diversen:

K1,K2,K3,K4 = 10-polige 2-rijige pinheader (raster 2,54 mm)
K5,K7 = 6-polige busstrip (raster 2,54 mm)
K6 = DB15 high-density connector
SW1,SW2,SW3 = 7-polige DIP-schakelaar (SPST)

WEBLINKS

[1] Voorbeeldprogramma's: <https://github.com/Panda381/PicoVGA>

[2] Projectpagina op Elektor Labs: <http://www.elektormagazine.com/labs/up-to-rgb888-to-vga>