

19 CNA vidéo programmable

traite tous les formats jusqu'à RGB888



Mathias Claussen (Conception) et Clemens Valens (Texte)

Lorsque vous expérimentez les microcontrôleurs et la vidéo de qualité VGA pour le rétro gaming ou une autre application, vous avez besoin d'un convertisseur numérique-analogique. Cette conception flexible vous permet d'essayer toutes sortes de formats vidéo avant de fixer la résolution finale.

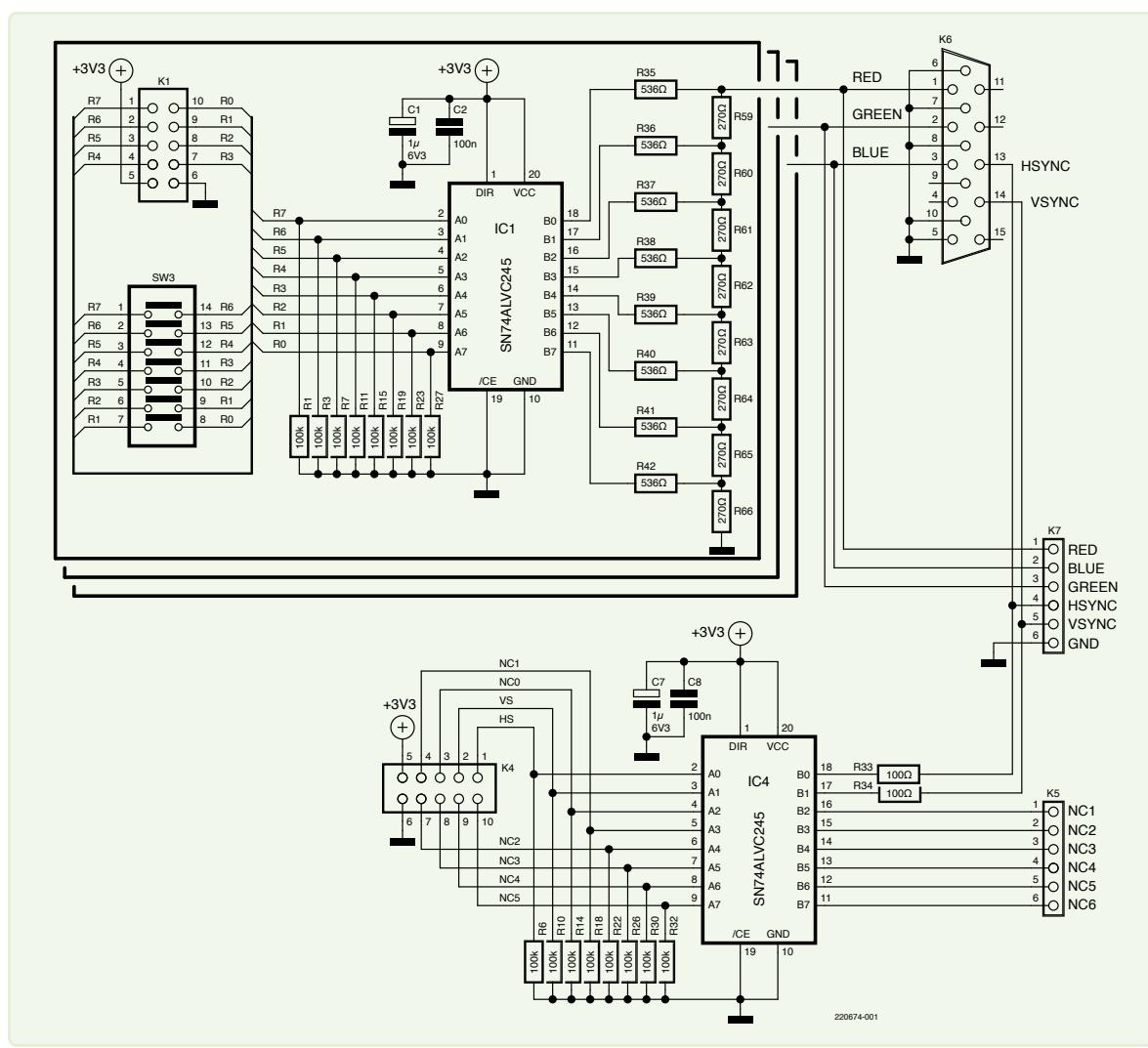


Figure 1. Un seul canal de couleur est représenté, les deux autres sont identiques.

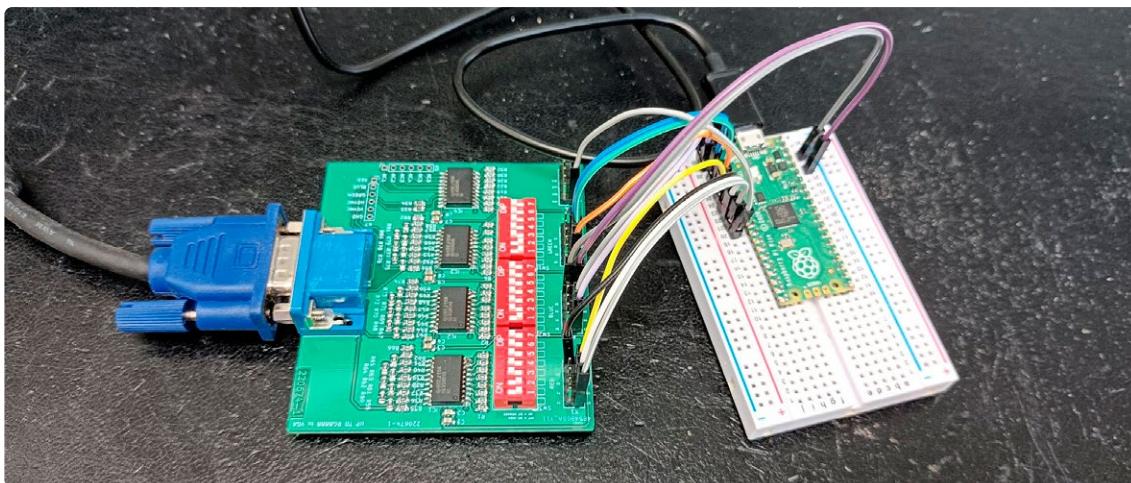


Figure 2. Le prototype en mode RVB332, piloté par un Raspberry Pi Pico.

Dans un passé pas si lointain, si vous vouliez ajouter une sortie vidéo à votre système informatique, vous aviez besoin de circuits intégrés vidéo spéciaux. Mais aujourd'hui, presque n'importe quel microcontrôleur peut produire un signal vidéo VGA à la volée. La résolution et la profondeur des couleurs dépendent de la quantité de mémoire vive disponible et de la vitesse du micro. Un simple convertisseur numérique-analogique (CNA) suffit à transformer les flux de bits numériques en tensions analogiques pour les canaux rouge, vert et bleu du moniteur VGA.

La largeur de mot requise du CNA sera déterminée par la résolution et la profondeur de couleur du signal vidéo cible. Ces deux paramètres dépendent à leur tour des ressources informatiques disponibles dans le micro qui génère le signal vidéo. Les capacités de votre système peuvent ne pas être claires au début du projet, et vous pouvez donc préférer commencer avec un CNA à faible résolution. D'un autre côté, lorsque le projet est quasiment terminé, vous pouvez constater qu'il reste de la place pour améliorer la qualité du signal vidéo. Dans ce cas, vous voudrez un CNA avec une plus grande résolution.

Circuit

Le circuit présenté ici peut être utilisé dans les deux situations, et tout ce qui se trouve entre les deux. Il s'agit d'un CNA vidéo à trois canaux avec une résolution programmable, où chaque canal peut être configuré indépendamment des autres. Par conséquent, il peut être utilisé pour tout ce qui va de RVB111 à RVB888 (résolution de 8 bits pour toutes les couleurs). Chaque chiffre peut prendre n'importe quelle valeur dans la plage de 1 à 8, par exemple RVB535 et RVB221 sont des formats de couleur vidéo possibles.

Le circuit est illustré à la **figure 1**. Il ne montre qu'un seul canal de couleur (rouge), car les autres canaux sont identiques. La vidéo numérique entre au niveau de K1 avec jusqu'à huit bits par canal. Les bits sont tamponnés par IC1, qui pilote un convertisseur numérique-analogique à échelle de résistance R2R. La sortie du CNA est un signal vidéo dans la gamme de 0 V à 0,7 V. Des résistances de rappel aux entrées garantissent que les bits non connectés sont vus comme des zéros.

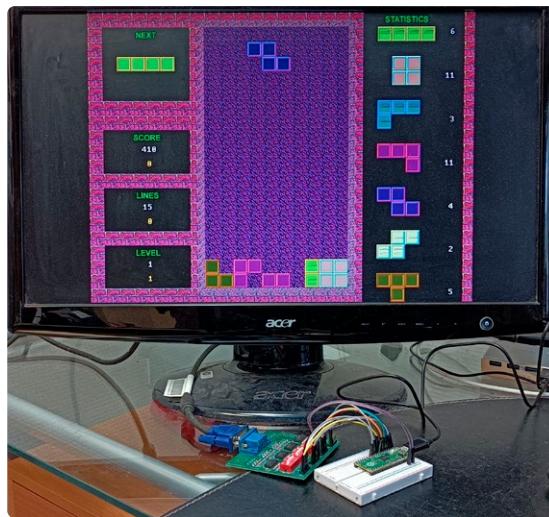
Le nombre de bits par couleur est réglé à l'aide des interrupteurs DIP SW3. Pour un signal de 8 bits, tous les interrupteurs doivent être désactivés. Pour réduire la largeur du mot, fermez les interrupteurs en commençant en bas par celui qui est étiqueté R1 et R0. Par exemple, pour une largeur de mot de 4 bits, fermez les quatre interrupteurs du bas. Connectez les bits de haut en bas, c'est-à-dire connectez le bit le plus significatif à R7, G7 ou B7, respectivement, puis descendez. En procédant ainsi, la sortie vidéo peut toujours atteindre sa valeur maximale (0,7 V) au lieu de s'affaiblir.

IC4 fournit des tampons pour les signaux de synchronisation horizontale et verticale. Les six tampons restants peuvent être utilisés pour autre chose. Ils sont exposés sur K5.

Tableau 1 : plan des broches du Raspberry Pi Pico pour les démonstrations vidéo RVB332 de [1].

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

Figure 3. Regardez le Pico jouer à Tetris !



Pour essayer rapidement le CNA vidéo sans passer des heures à le programmer, connectez-le à une carte Raspberry Pi Pico et à un moniteur VGA (voir **figure 2** et **tableau 1**). Utilisez HSYNC comme signal de synchronisation. Téléchargez une démo à partir de [1], programmez-la dans le Pico et appréciez les graphiques (**figure 3**). Les démos sur [1] sont au format RVB332. Vous devez donc fermer les cinq commutateurs inférieurs pour le rouge et le vert, et les six commutateurs inférieurs pour le bleu :

*nombre d'interrupteurs à fermer =
8 - nombre de bits pour la couleur*

Certaines démos ont une sortie son sur le GPIO19 du Pico. Mettez une résistance de 1,5 kΩ en série avec la sortie et ajoutez un condensateur de 10 nF vers la masse pour créer un simple filtre passe-bas pour le signal audio. Les fichiers de conception du projet sont disponibles sur [2].

VF : Maxime Valens — 220674-04

Des questions, des commentaires ?
Contactez Elektor (redaction@elektor.fr).



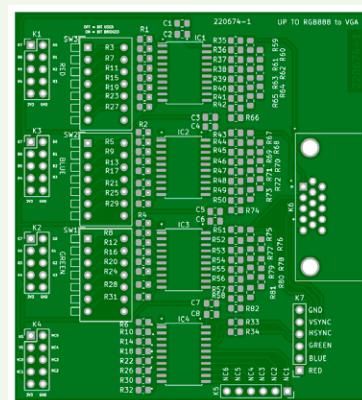
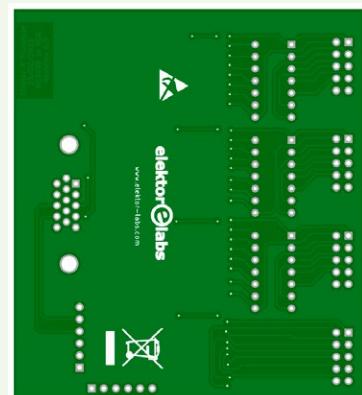
Produits

- **Écran tactile de 25 cm pour RPi (HDMI/VGA/BNC/AV)**
<https://www.elektor.fr/joy-it-10-touchscreen-for-rpi-incl-metal-case-and-stand-hdmi-vga-bnc-av>
- **Raspberry Pi Pico RP2040 H**
<https://www.elektor.fr/raspberry-pi-pico-rp2040-h>

LIENS

- [1] Exemples de programmes : <https://github.com/Panda381/PicoVGA>
- [2] Dossiers de projets chez Elektor Labs : <https://elektormagazine.fr/labs/up-to-rgb888-to-vga>

Liste des composants



Résistances (0805)

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

Condensateurs (0805)

C1, C3, C5, C7 = 1 µF
C2, C4, C6, C8 = 100 nF

Semi-conducteurs

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

Divers

K1, K2, K3, K4 = Connecteur mâle, 2 rangées, 10 voies (pas de 2,54 mm)
K5, K7 = Embase à 6 voies (pas de 2,54 mm)
K6 = Connecteur DB15 haute densité
SW1, SW2, SW3 = Interrupteur DIP à 7 voies (SPST)