Faites connaissance avec la **carte Raspberry Pi Pico** à **RP2040**

Mathias Claußen (Elektor) et **Luc Lemmens** (Elektor)

https://youtu.be/ijn-QDAgZss

Regardez « Raspberry Pi Pico Review » sur Elektor.TV!

Voilà qui s'appelle rebattre les cartes : en lançant la carte Pico, la fondation Raspberry Pi vient marcher sur les plates-bandes des Arduino, ESP32 et autres STM32 qui constituent *de facto* la référence en matière de plateformes à microcontrôleur. Écosystème logiciel, fonctions, prix, la petite nouvelle a tout pour concurrencer les anciennes. « Il est des nôtres, il a son microcontrôleur comme les autres », ont sans doute tonné *Les compagnons du comptoir de l'embarqué* à l'annonce de sa sortie.

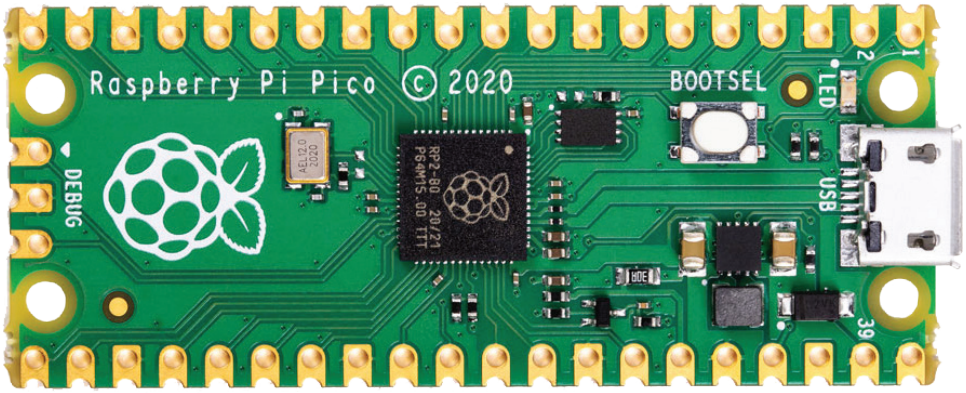

Figure 1. La carte Raspberry Pi Pico.

Encore une carte Raspberry Pi ?

Oui, mais celle-ci est d'une autre sorte (**fig. 1**) : la *Raspberry Pi Pico* se veut le complément et le compagnon de la famille Raspberry Pi. D'ailleurs, à y regarder de près, la Pico ne ressemble en rien aux cartes Raspberry Pi que nous connaissons. Elle embarque un processeur ARM à double cœur assisté de juste ce qu'il faut d'électronique pour orchestrer l'ensemble. C'est une sorte de RPi Zero modifié, alors ? Non, car le processeur en question n'est pas l'habituelle puce de Broadcom, mais le *RP2040*, un microcontrôleur conçu par la fondation RPi elle-même.

Ce RP2040 est plus précisément un microcontrôleur à double cœur ARM Cortex-M0+. Disparu, donc, le traditionnel système sur puce pour Linux : la Raspberry Pi Pico s'invite au royaume des plateformes à microcontrôleur dont les étendards les plus notoires s'appellent Arduino, Blue Pill et ESP32, sans oublier les cartes à processeurs RISC-V de popularité croissante (comme la GD32VF103). Avec un prix tournant autour de 5 €, soit moitié moins que la carte ESP32 Pico Kit, la Raspberry Pi Pico peut d'ores et déjà rivaliser avec la Blue Pill et, disons, un clone de l'Arduino Nano.

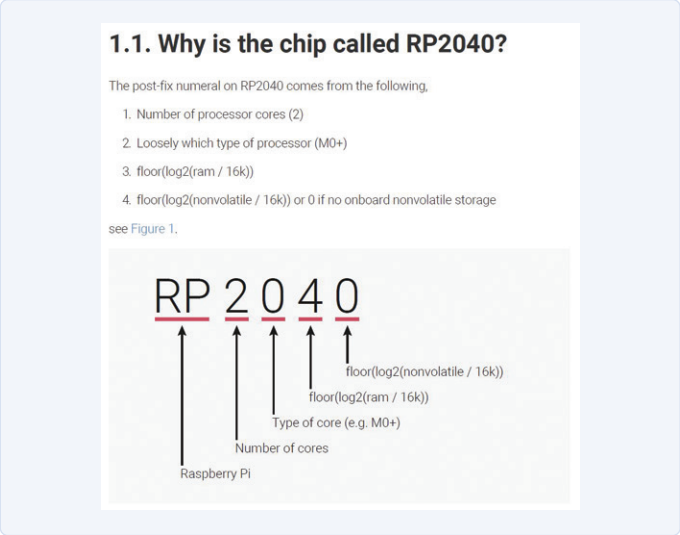

Figure 2. Schéma de nommage [9].

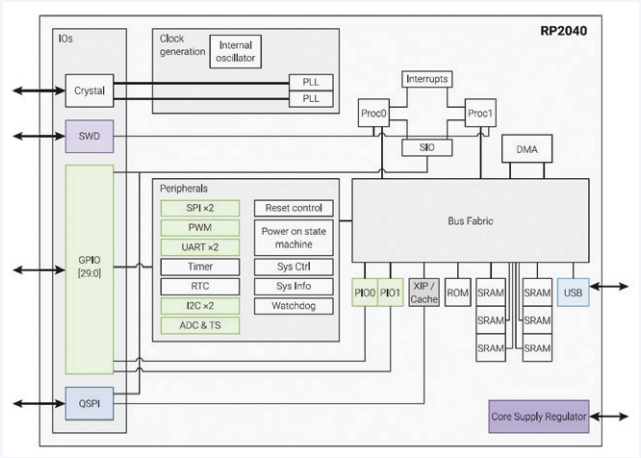

Figure 3. Diagramme fonctionnel du RP2040 [9].

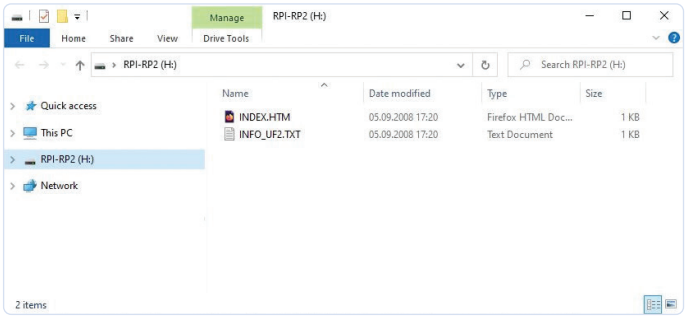

Figure 4. La carte Pico reconnue comme périphérique de stockage.

Tableau 1. Caractéristiques de la Raspberry Pi Pico

Processeur	Cortex M0+
Cœurs	2
Fréquence max.	133 MHz
SRAM	264 Ko sur 6 bancs
Flash interne	0 Ko
Flash externe	2 Mo flash QSPI (jusqu'à 16 Mo pris en charge)
GPIO	26 broches (dont 4 CA/N)
USB	1.1 hôte / esclave
CA/N	12 bits @ 500 kéch./s
Canaux CA/N	5 (dont le capteur de T°)
SPI	2
UART	2
I²C	2
PWM	16 canaux
Timer	1× 64 bits
RTC	oui
Fonctions uniques	automates finis programmables à E/S, ROM de démarrage reconnue comme mémoire USB.

Avant de passer en revue les caractéristiques de la carte Pico et du RP2040, signalons-en les deux courtes présentations signées Clemens Valens sur la chaîne YouTube d'Elektor [1] et sur le site du magazine Elektor [2].

Tour d'horizon de la carte RPi Pico

Les deux cœurs ARM Cortex-M0+ du microcontrôleur (µC) RP2040 sont cadencés à 133 MHz. Trouver un microcontrôleur à double cœur Cortex-M n'a rien d'extraordinaire en soi, mais habituellement il ne s'agit pas de Cortex-M0+ mais de cœurs plus puissants, comme le Cortex-M4, le Cortex-M7 ou le récent Cortex-M33.

Le **tableau 1** résume les caractéristiques du RP2040. À propos, d'où vient ce nom ? La réponse se trouve sur la **figure 2** (où *floor()* est la fonction « partie entière »). Cette méthode de nommage laisse suggérer que des variantes suivront, mais la fondation Raspberry Pi, que nous avons interrogée sur ce point, ne nous a pas encore répondu.

L'agencement des périphériques et blocs de fonctions du RP2040 est schématisé sur la **figure 3**. L'absence de mémoire flash interne y est notable. Les concepteurs de la Pico ont en effet choisi d'implanter cette mémoire à l'extérieur du µC, plus précisément dans le petit W25Q16JUXIQ, une mémoire flash NOR de 2 Mo. Le RP2040 prenant en charge 16 Mo de mémoire, il suffit de remplacer cette puce par une autre pour disposer d'un espace de stockage plus grand. Le port USB de la carte peut être configuré en tant qu'hôte ou esclave en mode USB 1.1 (12 Mo/s). Autrement dit on peut y relier des périphériques USB ou l'utiliser comme périphérique USB. La ROM de démarrage permet de charger du code depuis un ordinateur. Nul besoin ici d'outils de programmation spéciaux, la mémoire est reconnue par le PC comme mémoire externe. Le RP2040 peut donc être programmé par copie de fichiers dans le dossier associé au nom de volume de la carte (**fig. 4**). Simple et pratique.

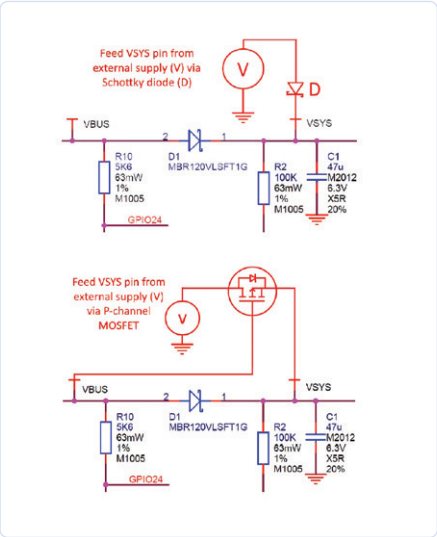

Figure 5. Deux façons d'alimenter la carte [3].

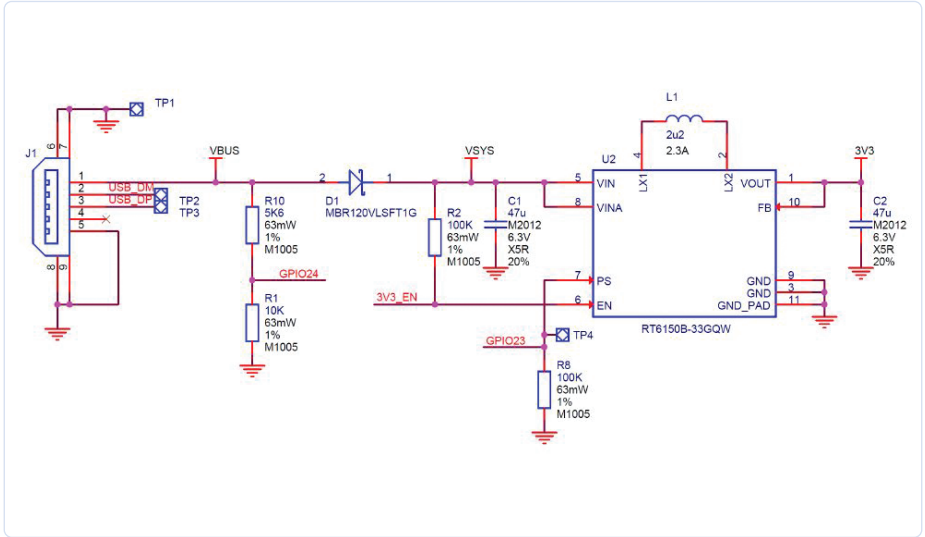

Figure 6. Schéma du convertisseur Buck-Boost embarqué [3].

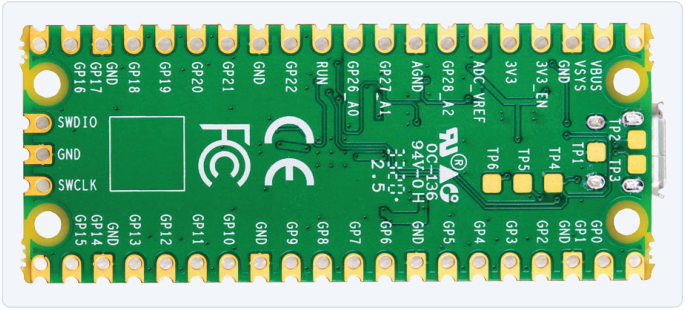

Figure 7. Le nom des broches est indiqué sur la face cuivre.

Pour ce qui est de l'alimentation, pas de tourment non plus en vue. La carte peut être alimentée en 5 V par micro-USB, mais accepte aussi des tensions de 1,8 V à 5,5 V délivrées par un convertisseur CC/CC Buck-Boost relié à sa broche VSYS. On peut l'alimenter avec p. ex. une batterie au lithium ou un jeu d'accus NiMH. La **figure 5** (extraite de la fiche technique) illustre le câblage de deux sources d'alimentation. Le convertisseur CC/CC est une puce RT6150 qui ne nécessite que quelques composants externes (**fig. 6**, fiche technique en [4]).

Parmi les 40 broches réparties en vis-à-vis le long de la carte figurent 26 broches GPIO, dont trois entrées CA/N, le reste étant pour l'alimentation et la masse. Le nom des broches est indiqué côté cuivre (**fig. 7**). La tension maximale d'E/S est de 3,3 V pour l'ensemble du port GPIO. Les trois broches placées sur un des petits côtés de la carte (SWCLK, SWIO et GND) forment le port *Serial Wire Debug*, c'est-à-dire le connecteur de programmation et de débogage du RP2040.

Périphériques et blocs d'entrée/sortie programmables

La liste des périphériques offerts par la carte Pico est classique : deux UART, deux contrôleurs I²C, deux contrôleurs SPI, 16 canaux MLI (PWM), un CA/N à 12 bits et 500 kéch./s, un capteur de température intégré, une horloge à temps réel, un temporisateur/compteur, et les fonctions GPIO de base. Moins commune est l'interface USB hôte et esclave, encore plus rares sont les deux blocs à E/S programmables (PIO) comprenant chacun quatre automates finis.

Ces automates finis permettent d'implanter des interfaces UART, I²C et SPI supplémentaires, mais aussi des interfaces matérielles telles que VGA, DPI, SD-Card et bien d'autres. Leur usage est décrit dans la fiche technique et illustré par des programmes d'exemple. La **figure 8** montre un bloc PIO et ses quatre automates finis. Neuf instructions servent à les commander : JMP, WAIT, IN, OUT, PUSH,

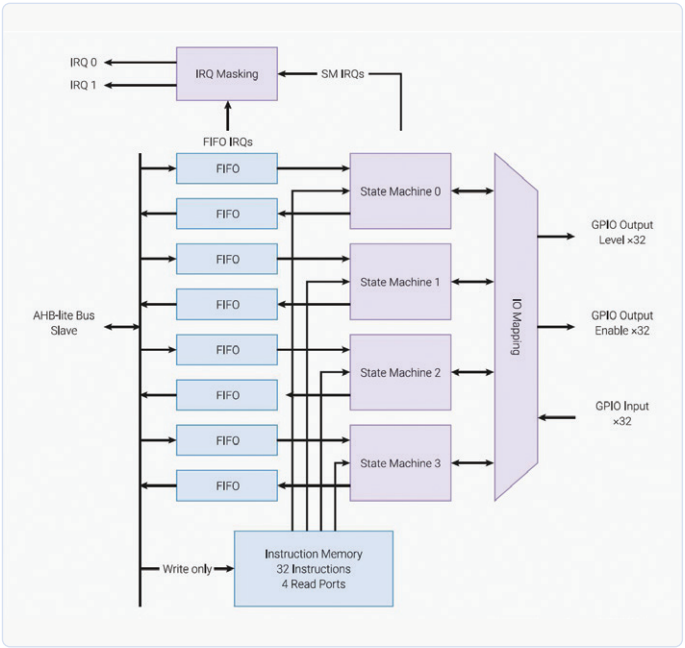

Figure 8. Bloc PIO à automates finis programmables [9].

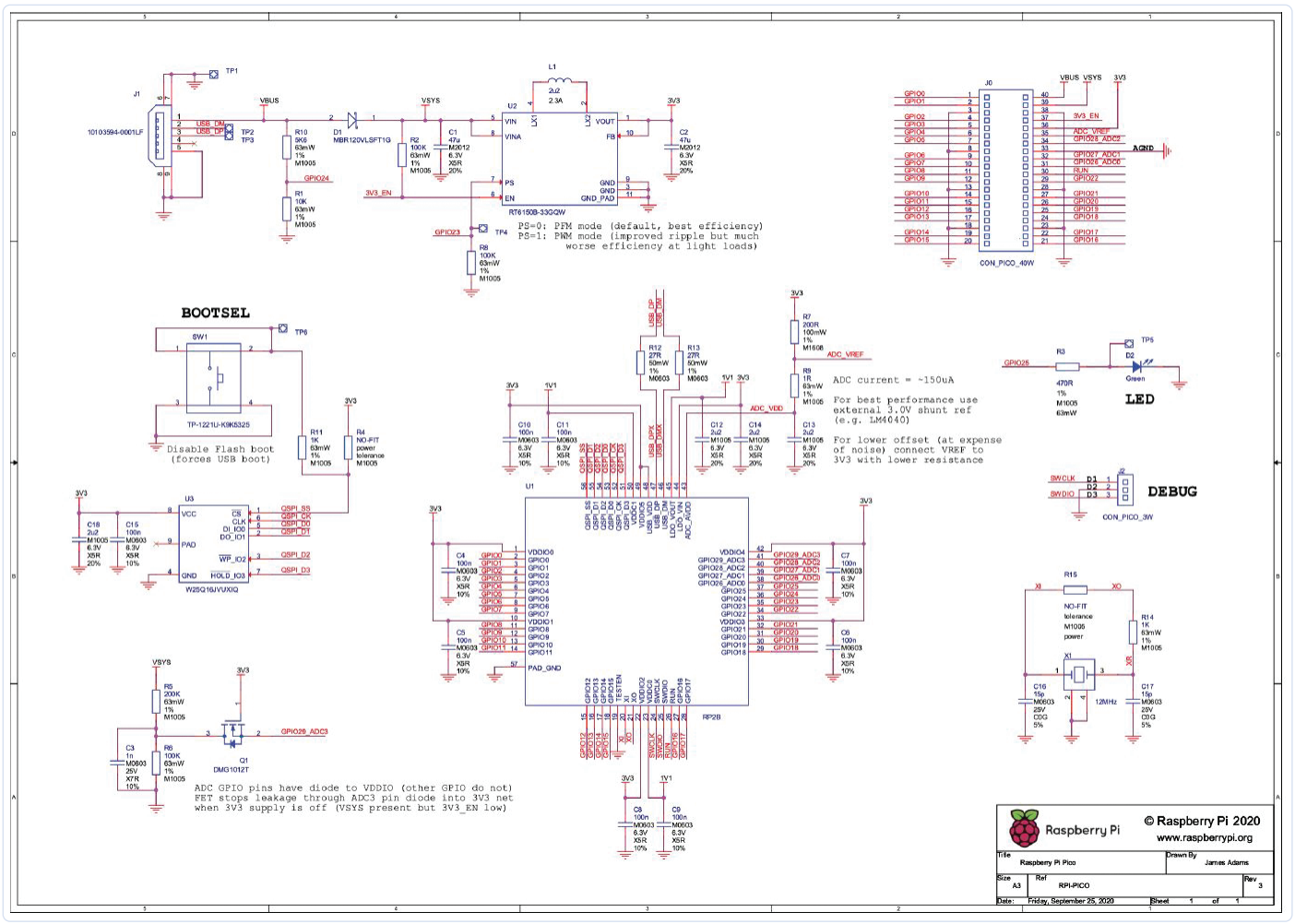

Figure 9. Un bloc MLI (PWM) du RP2040 [9].

PULL, MOV, IRQ, et SET. Ils peuvent fonctionner indépendamment des deux cœurs, et ainsi servir à la mise en œuvre d'interfaces absentes de la carte sans solliciter le processeur.

Les UART du RP2040 reposent sur l'UART PrimeCell d'ARM (PL011) présent sur d'autres cartes à RPi. Il autorise un débit de 961,6 kbauds. C'est encore ARM qui est à la manœuvre dans les deux interfaces SPI avec son port SSP PrimeCell (PL022), lui aussi exploité sur diverses cartes à RPi. En mode maître, le débit maximal vaut 1,843 Mbauds/s pour une fréquence d'horloge SPI d'au moins 3,686 MHz. L'horloge du contrôleur I²C peut opérer selon trois modes : *Standard* (100 kHz), *Fast* (400 kHz) et *Fast Plus* (1 MHz), avec un adressage sur 7 et 10 bits en maître ou esclave. Ces trois interfaces permettent de connecter la plupart des matériels en offrant à leur égard un bon équilibre entre fonctions et complexité.

Le CA/N du RP2040 est un simple SAR offrant 500 kéch./s à une résolution de 12 bits. Il possède quatre entrées et un capteur de température. Sur un AVR (p. ex.), un CA/N fonctionne avec une tension de référence interne de 2,6 V ou 1,1 V. Cette référence est externe sur le RP2040 et s'applique sur la broche ADC_AVDD ; le schéma de principe de la carte Pico (**fig. 9**) la montre reliée à 3,3 V et filtrée par un circuit RC (R7/R9/C13).

L'interface USB est également intéressante. Nous l'avons dit plus haut, elle peut être configurée en hôte ou esclave. Son mode *Full Speed* (USB 1.1, soit 12 Mo/s) permet l'utilisation de périphériques USB (p. ex. un clavier et une souris), mais aussi de relier la carte à un PC ou à un autre RPi. En mode hôte, le contrôleur prend en charge les concentrateurs USB, autrement dit il est possible de relier simultanément plusieurs périphériques à la Pico.

Le temporisateur/compteur (*timer*) de 64 bits du RP2040 repose sur une base de temps de 1 µs. Il permet de programmer jusqu'à quatre alarmes et des temporisations de l'ordre de la µs. Pour les

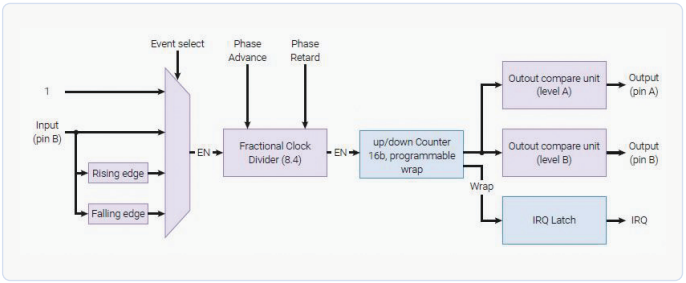

Figure 10. Un bloc MLI (PWM) du RP2040 [9].

interruptions ou évènements à déclencher périodiquement, vous pouvez utiliser une des huit unités MLI (PWM) à 16 bits et 2 canaux. Vous pouvez aussi mesurer des fréquences ou des rapports cycliques avec leurs broches GPIO afin de produire des interruptions et des requêtes DMA. Un diviseur d'horloge « fractionnaire » permet un réglage plus précis des fréquences.
L'unité DMA (à accès mémoire direct) permet de transférer des données dans le système sans le recours du processeur. Vous pouvez ainsi, sans processeur donc, effectuer des conversions A/N et placer les valeurs obtenues dans un tampon prédéfini de la mémoire, ou encore transférer des données de la mémoire à un UART. Copier et déplacer le contenu de plusieurs tampons d'un endroit à l'autre avec cette méthode peut même s'avérer plus rapide qu'avec le processeur. L'unité DMA se montre tout aussi pratique pour envoyer des données sur un écran externe. Elle peut de même être reliée aux automates finis programmables (PIO) pour échanger rapidement des données avec des périphériques externes.

Schémas, manuels et fichiers de conception

La carte Pico est la première à exhiber un RP2040. Sur le schéma de la **figure 9**, on le voit entouré du convertisseur CC/CC, de la mémoire flash NOR QSPI, de l'interface USB, et d'un circuit chargé de mesurer la tension d'alimentation et de mettre le convertisseur CC/CC en mode basse consommation.
La fondation RPi fournit des fichiers KiCad et Fritzing à l'intention des concepteurs désireux de se familiariser avec le RP2040. Notons ici avec un certain intérêt que la conception de référence n'est pas la carte Pico, mais une carte à RP2040 à assembler et configurer soi-même. Alors que le brochage de la plupart des microcontrôleurs semble avoir été organisé par un amateur de Mikado, celui du RP2040 (**fig. 11**) est parfaitement ordonné. Les broches sont regroupées par fonctions, d'où un routage aisé vers le matériel externe. Tous les concepteurs de matériel apprécieront.
La documentation et le logiciel n'étant (à ce jour) pas encore dans leurs versions finales, quelques modifications mineures sont à attendre. Il n'en reste pas moins que la documentation est à la hauteur de celles auxquelles nous a habitués la fondation RPi : soignée et complète. La carte Pico est à matériel ouvert et accompagnée d'un SDK *open source* que vous pourrez exploiter aussi bien sur un RPi que sur un PC ou un Mac. Pour ce qui est de l'écriture de vos applications, vous aurez le choix entre MicroPython et C/C++.

Développement en MicroPython

Les codeurs aguerris choisiront sans doute C/C++ pour programmer le RP2040 (ou utiliseront l'EDI Arduino, dont la compatibilité prochaine avec la carte Pico a été annoncée). MicroPython a l'avantage d'être plus simple. La plateforme de développement peut être un RPi 4 avec Raspberry Pi OS ou – d'après le manuel – un PC avec une distribution Linux basée sur Debian. De là il vous faudra : soit compiler vous-même un portage de MicroPython pour RPi OS en suivant les instructions du manuel [5] ; soit télécharger le fichier binaire UF2 d'installation depuis la page [6].
La dernière méthode est la plus simple : on maintient enfoncé le bouton BOOTSEL, on relie la Pico à un RPi (ou à un PC), puis on relâche le bouton une fois la carte connectée. Celle-ci sera reconnue comme nom de volume RPI-RP2. Ne reste alors plus qu'à y glisser le fichier UF2 pour flasher la mémoire du RP2040 et installer MicroPython.

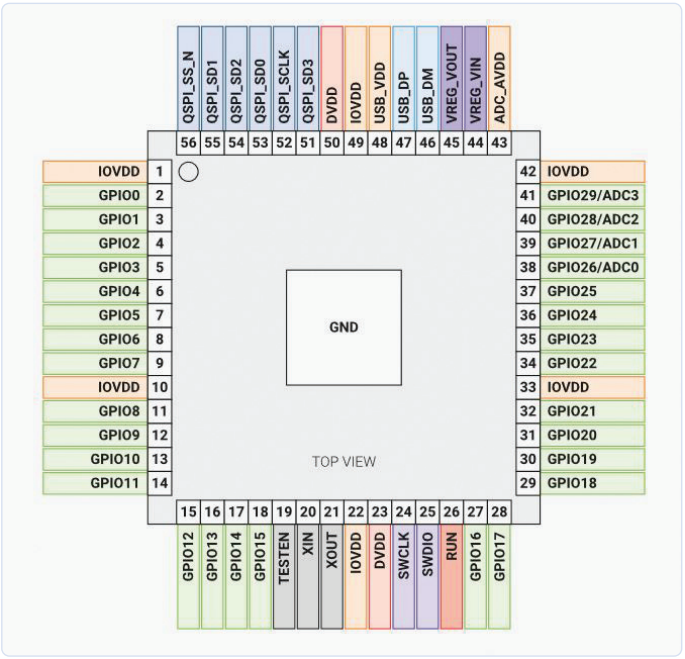

Figure 11. Brochage du RP2040 [9].

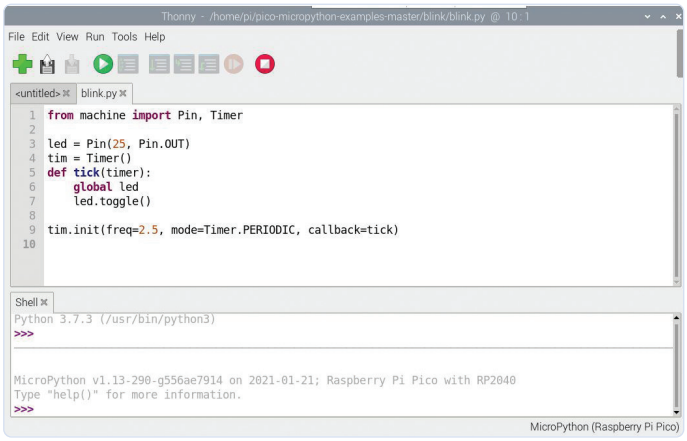

Figure 12. L'EDI Thonny sur un RPi 4.

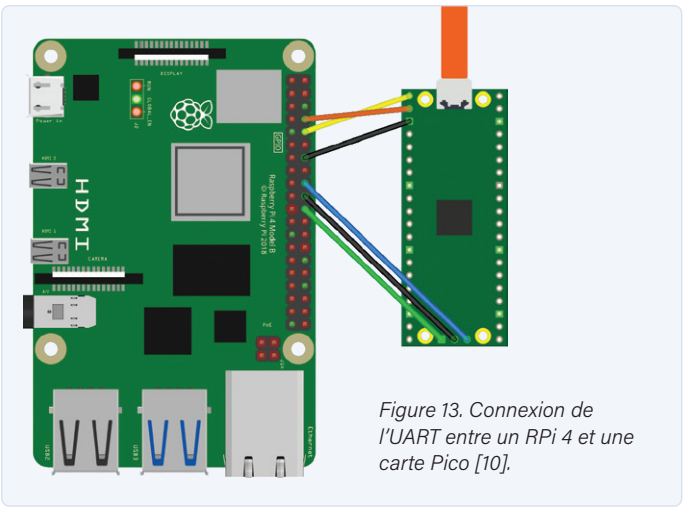

Figure 13. Connexion de l'UART entre un RPi 4 et une carte Pico [10].

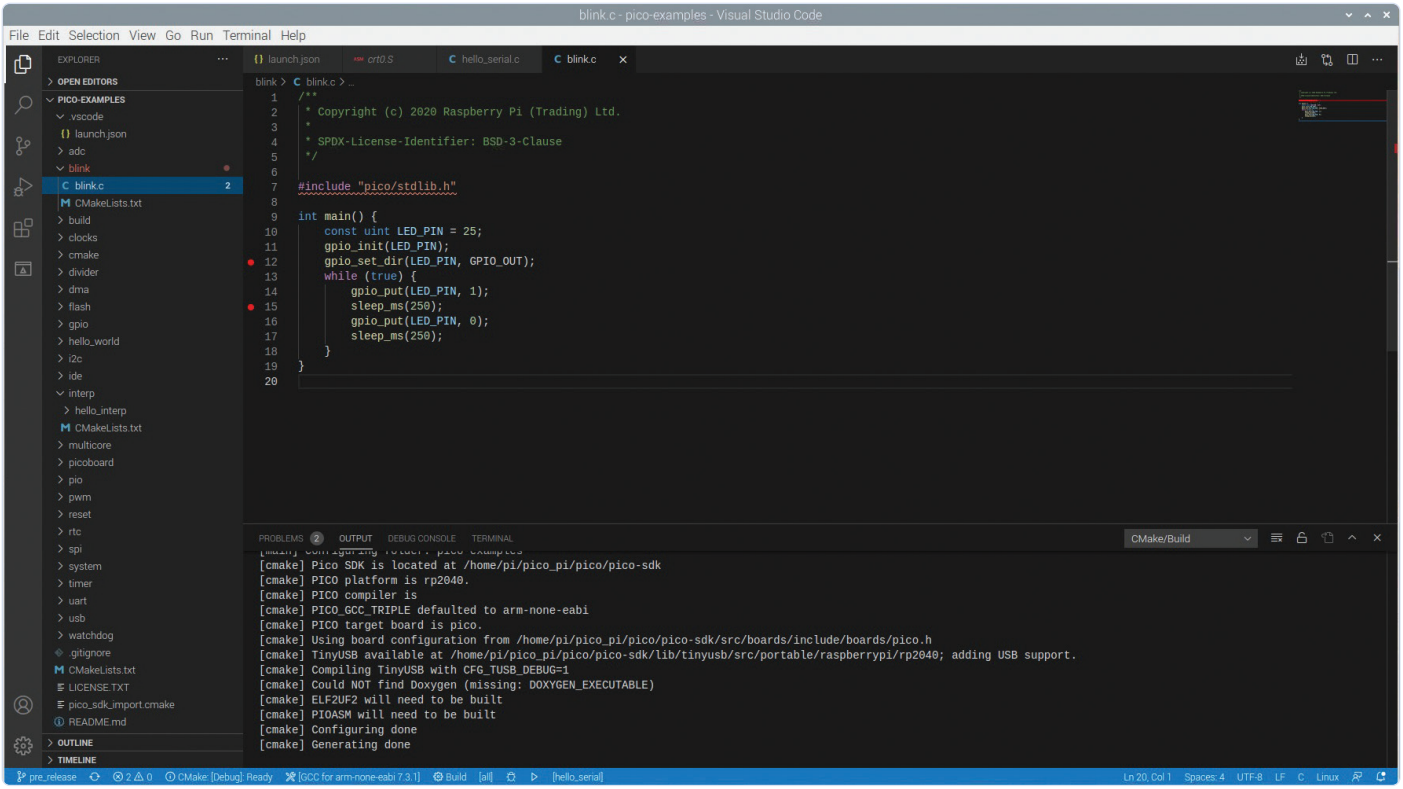

Figure 14. L'EDI Visual Studio Code sur un RPi 4.

Le manuel explique comment accéder à l'interpréteur interactif Python (aussi appelé REPL) via l'interface USB ou l'UART de la carte Pico. Ce REPL n'est pas toujours des plus pratique. L'interface MicroPython pour la Pico et d'autres cartes à RP2040 est heureusement reconnue par divers EDI, dont Thonny (**fig. 12**). Là encore le manuel explique comment l'installer et le configurer.

La fondation RPi a publié un livre d'initiation à MicroPython appelé *Get Started with MicroPython on Raspberry Pi Pico* (Halfacree & Everard, 2021). Les auteurs expliquent notamment l'usage des blocs PIO et comment interagir avec le monde extérieur (lien dans l'encadré **Produits**).

Développement en C/C++

La plateforme privilégiée est le RPi 4, mais la mise en place d'autres environnements de développement est couverte par le manuel [7]. Un script à télécharger depuis [8] se charge de toutes les étapes nécessaires à l'installation sur le RPi. Le schéma de la **figure 13** illustre la connexion physique à établir entre le nano-ordinateur et la carte Pico. Dans cette configuration le RPi 4 sert également

POURQUOI NE PAS AVOIR ÉVOQUÉ LE MODE BASSE CONSOMMATION DE LA PICO ?

C'est effectivement une des premières caractéristiques à étudier lorsqu'on acquiert un nouveau microcontrôleur. Un microcontrôleur en sommeil consomme généralement quelques centaines de nA ou quelques µA. D'après sa fiche technique, le RP2040 affiche une consommation moyenne de 0,18 mA dans son état de sommeil le plus profond appelé *dormant*. Cela semble de prime abord tout à fait acceptable, mais sur la carte Pico les conditions changent. Le RP2040 n'est en effet plus isolé, mais flanqué de la mémoire flash externe et du convertisseur CC/CC. Alors que le courant de veille de la mémoire flash ne vaut que 50 µA et peut même descendre à 15 µA, le convertisseur CC/CC doit encore fonctionner en présence de charges très basses. Si on prend ces éléments en compte, on trouve que la Pico en veille consomme 0,8 mA à 25 °C. Cette valeur est à garder à l'esprit si l'on projette d'alimenter la carte avec des piles ou accus durant une longue période. Dans ce cas, il faudra trouver un compromis entre les fonctions exploitées, la consommation de la carte et l'usage qui en est prévu. Le microcontrôleur parfait n'existe pas, mais cette première puce de la fondation RPi est prometteuse. L'avenir nous dira si ses concepteurs auront su gagner quelques µA en mode basse consommation, et comment ils y sont parvenus. On peut faire ici le parallèle avec le développement logiciel, où l'optimisation du code consiste à le rendre à la fois plus rapide et moins énergivore sur les appareils alimentés par piles ou accus.

d'interface de débogage grâce à une version modifiée d'OpenOCD. Avec un PC ou un Mac, évidemment dépourvus de broches GPIO, il faut utiliser une seconde carte Pico exécutant Picoprobe. La procédure est décrite dans le manuel.

Parmi les outils installés par le script figure l'EDI Visual Studio (**fig. 14**), a priori plus commode que l'interpréteur de commandes du RPi. Les programmes d'exemple que nous avons reproduits fonctionnaient comme attendu, mais le générateur de projets (conçu pour vous épargner l'instanciation en ligne de commande d'un nouveau projet) n'était pas encore tout à fait au point. Ce genre de petits défauts est le propre des nouveautés. Il en a été de même avec le RPi 4, mais tout cela sera corrigé à mesure que les utilisateurs signaleront les bogues et problèmes rencontrés.

Le RP2040 bénéficie d'un ensemble fourni de bibliothèques et d'exemples, suffisamment étayé pour que vous n'ayez pas à écrire vous-même les pilotes de chaque élément de la puce. La bibliothèque TinyUSB a été complétée pour prendre en charge le RP2040, donc pour l'USB aussi vous aurez sous la main tout ce qu'il faut pour démarrer rapidement. De plus le code est ouvert. Libre à vous donc de fouiller dans ses dossiers et de l'étudier à profit. Les EDI comme Wiring ou Arduino prennent en charge toutes les cartes intéressantes, donc gageons que la brillante Pico n'échappera pas à leur radar et que nous aurons bientôt au moins un EDI de plus à disposition.

Sur un PC ou un Mac, privilégiez une machine virtuelle pour la mise en place de votre environnement de développement, et créez régulièrement des instantanés du système. En cas de problème, il vous suffira de revenir à un état de la machine où le monde était encore merveilleux.

Laissez-vous Picoter

Si jusque-là vous fuyiez les microcontrôleurs parce que : 1) trop chers à vos yeux et, 2) moins bien documentés que votre machine à laver, laissez-nous résumer cet article : une carte Pico ne coûte que 5 €, et la documentation de la fondation RPi explique tout, dit tout, sait tout, a réponse à tout. La question ne devrait pas être de savoir si vous devez acheter une Pico, mais combien en acheter. Les outils de développement, les exemples fournis et la pédagogie des manuels rendent la découverte de la Pico amusante. Même l'aridité apparente de la fiche technique peut vous happer ! La Pico se veut accessible à tous, et à ce titre saura aussi initier les plus jeunes aux microcontrôleurs. Partagez avec nous vos réalisations, nous sommes impatients de voir ce que vous tirerez de la Pico !

210045-04

Votre avis, s'il vous plait ?
Veuillez adresser vos questions et vos commentaires par courriel à mathias.claussen@elektor.com ou à redaction@elektor.fr.

Ont contribué à cet article
Auteurs : Mathias Claußen et Luc Lemmens
Rédaction : Jens Nickel
Maquette : Giel Dols
Traducteur : Hervé Moreau

LIENS

[1] **C. Valens, « Raspberry Pi Pico Review », Elektor.TV, 1/21/2021** : https://youtu.be/ijn-QDAgZss
[2] **C. Valens, Place au µC RP2040. Hue Pico, ElektorMagazine.fr, 22/01/2021** : http://elektormagazine.fr/news/microcontroleur-raspberry-pi-rp2040-pico-board
[3] **Fiche technique du RT6150** : https://datasheets.raspberrypi.org/pico/pico_datasheet.pdf
[4] **RT6150 datasheet** : www.richtek.com/assets/product_file/RT6150A=RT6150B/DS6150AB-04.pdf
[5] **Manuel pour utiliser MicroPython** : https://datasheets.raspberrypi.org/pico/sdk/pico_python_sdk.pdf
[6] **Démarrer avec la Pico** : www.raspberrypi.org/documentation/pico/getting-started/
[7] **Configuration C/C++** : https://datasheets.raspberrypi.org/pico/sdk/pico_c_sdk.pdf
[8] **Script d'installation** : https://github.com/raspberrypi/pico-setup/blob/master/pico_setup.sh
[9] **Fiche technique du RP2040** : https://datasheets.raspberrypi.org/rp2040/rp2040_datasheet.pdf
[10] **Getting started with Pico** : https://datasheets.raspberrypi.org/pico/getting_started_with_pico.pdf