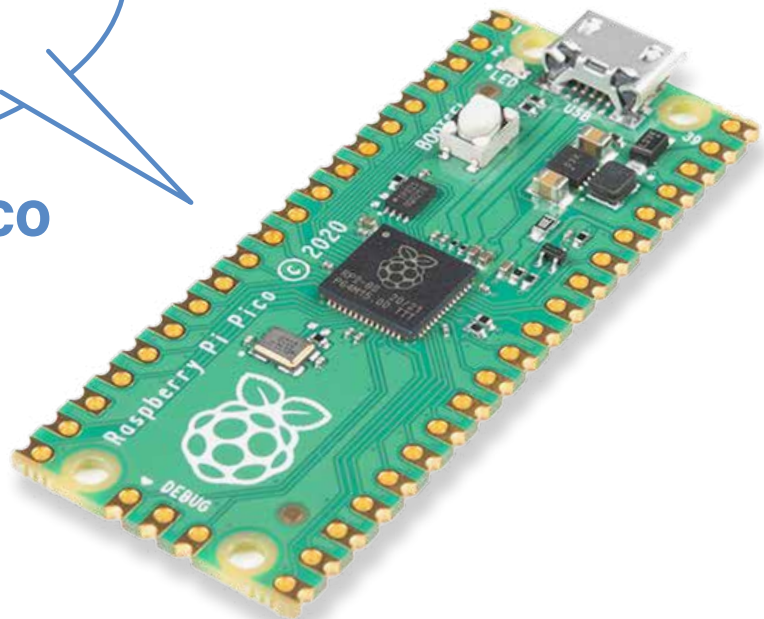


Hello
World

Raspberry Pi Pico et RP2040

Pleins feux sur les premiers
microcontrôleur et micro-
processeur de Raspberry Pi



Avra Saslow (États-Unis)

Confiante dans son rôle à jouer dans l'avenir de l'informatique, la Fondation Raspberry Pi s'est lancée dans le développement de produits révolutionnaires : un microcontrôleur et un microprocesseur RPi parfaitement compatibles avec MicroPython et C/C++. Bienvenue dans le nouveau monde de Raspberry Pi ... le RPi Pico et le microprocesseur RPi RP2040.

Le RPi Pico (**fig. 1**) est idéal pour les projets de moindre complexité qui ne font tourner qu'un programme à la fois – situation où vous n'avez pas besoin de la taille et de la puissance d'un RPi complet - mais pour lesquels une plate-forme Arduino ne convient pas. Il s'accompagne d'une documentation incroyablement complète pour ses kits de développement logiciel (SDK) MicroPython et C/C++. Si vous êtes plus familier avec la programmation en Python, ou si avec votre connaissance d'Arduino vous pouvez passer au C++ (dont Arduino est une variante), alors le Pico pourrait bien devenir votre nouveau microcontrôleur préféré.

Présentation de RPi Pico

Allons au fait et examinons les spécifications techniques !

Le microprocesseur est le nouveau RP2040 conçu par Raspberry Pi. Le nom reprend

les initiales de la fondation, suivies du nombre de cœurs du processeur (2), du type de processeur (Mo+), de $\text{floor}(\log_2(\text{ram}/16k))$, et enfin de $\text{floor}(\log_2(\text{nonvolatile}/16k))$. La **figure 2** illustre l'algorithme de nommage !

La puce intègre une amorce UF2 qui permet de flasher le microcontrôleur avec le micrologiciel via USB (par glisser-déposer), ainsi que des routines de calcul en virgule flottante. Elle est équipée de cœurs de processeur Dual Cortex Mo+ avec une horloge variable jusqu'à 133 MHz. Couplée à une matrice de bus à haute performance, elle peut obtenir la pleine puissance sur les deux cœurs en même temps. Ce niveau de performance pourrait autoriser à terme des modèles d'apprentissage machine avec TensorFlow pour MicroPython. Elle dispose d'une vaste RAM interne (264 Ko de SRAM), mais utilise une mémoire flash externe (2 Mo de mémoire flash embarquée), ce qui

permet à l'utilisateur de choisir la mémoire qu'il lui faut.

- Modes de fonctionnement à faible puissance (veille et veille profonde).
- Un USB intégré qui peut servir à la fois de dispositif et d'hôte.
- Divers périphériques numériques dont 2x UART, 2x I²C, 2x SPI, jusqu'à 16 canaux PWM, un minuteur avec quatre alarmes et un compteur en temps réel.
- 30 GPIO multifonctions dont quatre utilisables en CA/N (**fig. 3**).
- Un capteur de température.
- 8x automates finis d'E/S programmables (PIO) pour la prise en charge de périphériques personnalisés. Sur la plupart des microcontrôleurs, il faut faire du *bit-bang*, c'est-à-dire utiliser l'unité centrale et du logiciel pour activer et désactiver directement des broches. Cela fonctionne mais prend



Listage 1. Commandes de terminal pour l'installation de MicroPython.

```

echo „Obtaining MicroPython“;
cd ~/;
mkdir pico;
cd pico;
git clone -b pico git@github.com:raspberrypi/micropython.git;

echo „Obtain additional tools“;
sudo apt update;
sudo apt install cmake gcc-arm-non-eabi;
echo „Building MicroPython“;
cd micropython;
git submodule update --init --recursive;
make -C mpy-cross;
cd ports/rp2;
make

```

du temps, surtout si vous utilisez des interruptions, et cela consomme de précieuses ressources de calcul dont vous pourriez avoir besoin pour autre chose. Pour traiter les données qui entrent et sortent, on peut utiliser les E/S programmables ou les 2 blocs PIO de quatre automates finis du Pico, et le décharger ainsi des contraintes de traitement pour la gestion des protocoles de communication.

➤ Et bien davantage !

En ce qui concerne les logiciels, le RP2040 est compatible avec les environnements de développement multiplateforme C/C++ et MicroPython, y compris un accès facile au débogage. Même si le RPi Pico ne bouleverse pas forcément les références informatiques, les documentations du RP2040 et du Pico élaborées par la Fondation Raspberry Pi sont d'une profondeur technique inégalée. Avec de telles caractéristiques, le RPi Pico est puissant, incroyablement extensible et sa taille est (presque) à la hauteur de son nom "Pico". Il s'aligne sur tous les produits Raspberry Pi tant par sa taille que par son accessibilité en termes de caractéristiques techniques et de coût. La Fondation Raspberry Pi a réussi son entrée sur le marché des μ C !

MicroPython sur le Pico

Vous avez peut-être remarqué que le langage pris en charge en passant du RPi à RPi Pico a changé de Python à MicroPython. C'est parce que Python est trop gourmand en ressources pour fonctionner sur de petits microcontrôleurs. Pour les microcontrôleurs comme le Pico, on utilise donc un portage de Python, appelé MicroPython. Presque identique à Python, il ne contient qu'un sous-ensemble des modules de la bibliothèque standard, de sorte qu'il se contente de peu de RAM (environ 16 K selon MicroPython).

Il faut quelques étapes pour faire fonctionner MicroPython sur le RPi Pico. Le processus commence par le clonage du dépôt Github de MicroPython et l'installation

de CMake et de GNU Embedded Toolchain pour Arm afin d'aider à la construction du logiciel. Les commandes du Terminal (écrites en script shell) construisent les programmes exécutables et les bibliothèques à partir du code source de MicroPython (**listage 1**).

À partir de là, l'installation du logiciel MicroPython se fait par glisser-déposer, où vous faites glisser le microprogramme .uf2 sur la carte, vue comme périphérique de stockage de masse USB. Il faudra maintenir le bouton BOOTSEL enfoncé pour la faire passer dans ce mode de stockage de masse USB. Une fois le microprogramme chargé sur la carte, vous pourrez vous connecter au REPL (Read Evaluate Print Loop) de MicroPython, qui est un moyen simple de tester

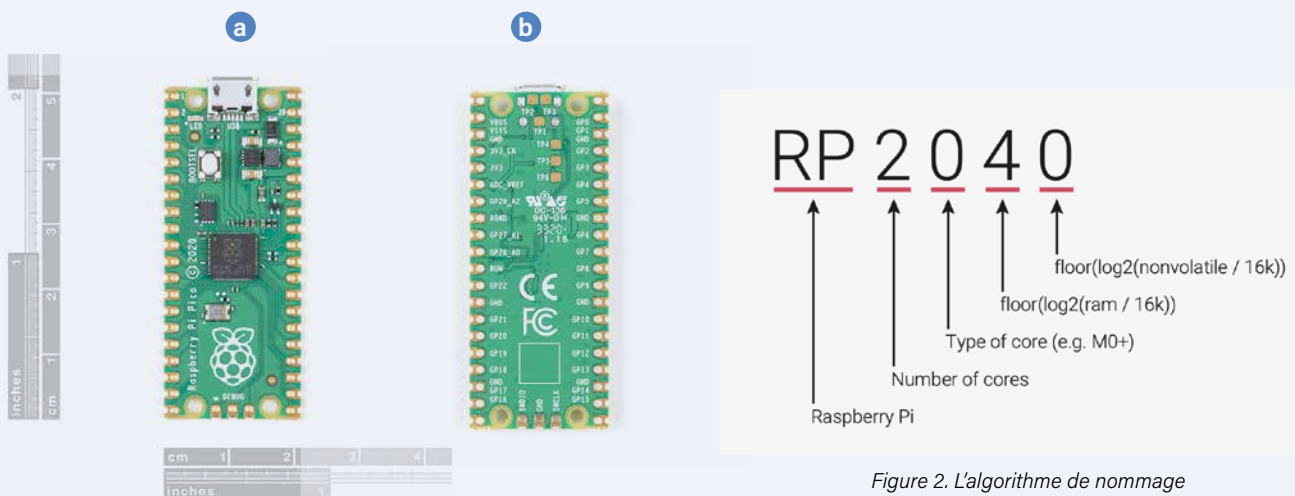


Figure 2. L'algorithme de nommage

Figure 1a. Le Raspberry Pi Pico est une excellente solution lorsque vous n'avez besoin ni de la taille ni de la puissance d'un RPi complet. Fig.1 b : Regardez le dessous du Pico.

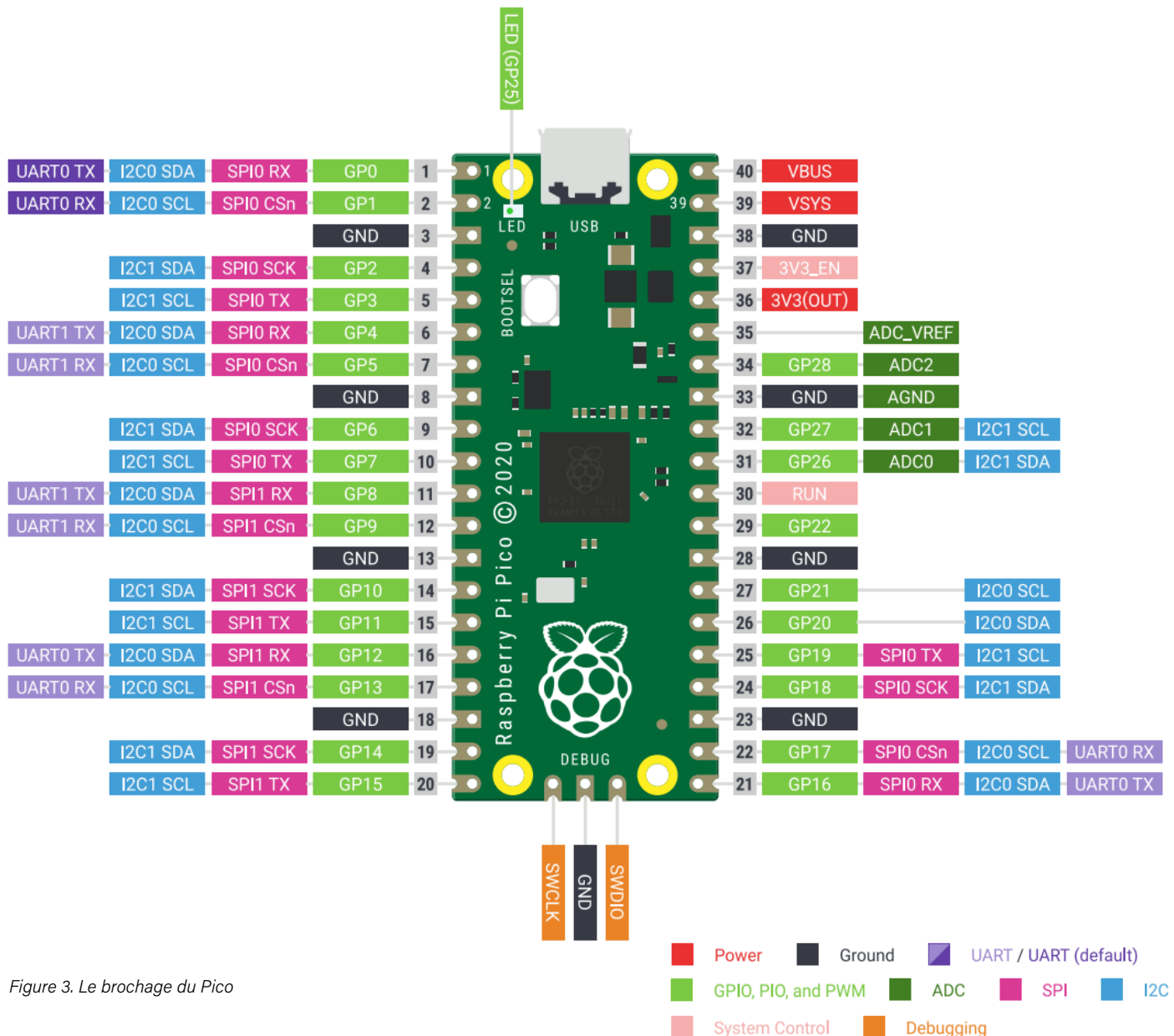


Figure 3. Le brochage du Pico

le code et d'exécuter des commandes. En utilisant l'exemple de code de Raspberry Pi, j'ai pu facilement mettre en place un code qui utilise un minuteur pour faire clignoter la LED embarquée (**listage 2**).

Un peu de morse ?

Comme je travaillais déjà sur le Pico avant sa sortie alors qu'il était encore *top secret*, je voulais communiquer avec lui de manière un peu cryptée. Aussi, à partir du code « *Blink an LED* », j'ai fait un script MicroPython qui compose en morse le message *Hello World* (**listage 3**). Comme il est difficile d'illustrer sur papier le clignotement d'une LED, je lui ai aussi fait imprimer la traduction du code morse sur le terminal (**fig. 4**). Et ça a donné : Hello World, RPi Pico !



Listage 2. Une minuterie pour faire clignoter une LED.

```
##credit of the Raspberry Pi Foundation
from machine import Pin, Timer
led = Pin(25, Pin.OUT)
tim = Timer()
def tick(timer):
    global led
    led.toggle()

tim.init(freq=2.5, mode=Timer.PERIODIC,
callback=tick)
```



Figure 4. "Hello World" en morse

Listage 3. Un script MicroPython qui produit „Hello World“ en morse.

```
#-----
# Importation des bibliothèques nécessaires, connexion à la LED intégrée,
# réglage de la fréquence de la LED
#-----

import time
from machine import Pin
led=Pin(25,Pin.OUT)# la LED du Pico est sur la broche 25
BlinkRate=0.25

#-----
# Fonctions pour la durée du signal en code Morse et code lui-même
#-----

def dash():
    led.value(1)
    time.sleep(4*BlinkRate)
    led.value(0)
    time.sleep(BlinkRate)

def dot():
    led.value(1)
    time.sleep(BlinkRate)
    led.value(0)
    time.sleep(BlinkRate)

def pause():
    time.sleep(BlinkRate)

code = {'A':'.-','B':'-...','C':'-.-.','D':'-..','E':'.','F':'.-..','G':'--.',
'H':'....','I': '..','J':'.---','K': '-.-','L':'.-..','M': '--','N': '-.',
'O': '---','P': '-.-.','Q': '--.-','R': '.-.','S': '...','T': '-','U': '..-',
'V': '...-','W': '-.-','X': '-.-.','Y': '-.-.','Z': '--.',
'0': '-----','1': '.-----','2': '..---','3': '...--','4': '....-',
'5': '.....','6': '-....','7': '--...','8': '---..','9': '----.',
',': '-.-.-.',
'.': '-.-.-.',
'?': '..-.-.',
'/': '-.-.-.',
'@': '-.-.-.',
' ': ' / ',
}

#-----
# Fonction qui renvoie simplement la phrase en morse à partir de la phrase
# anglaise en majuscules
#-----

def convertToMorseCode(sentence):
    sentence = sentence.upper() #transformation en majuscule pour que le
    dictionnaire reconnaisse le caractère
    secretSentence = "" #effacement de la phrase à ajouter
    for i in sentence: #pour chaque caractère de la phrase, le chercher dans
    le dictionnaire et le remplacer par le code Morse
        secretSentence += code[i] + " "
    return secretSentence

#-----
# Fonction principale qui fait clignoter la LED selon la phrase en morse
#-----

while True:
    sentence = "Hello World"
    secretSentence = convertToMorseCode(sentence)
    for i in secretSentence:
        if i == ".":
            dot()
        elif i == "-":
            dash()
        else:
            pause()
```

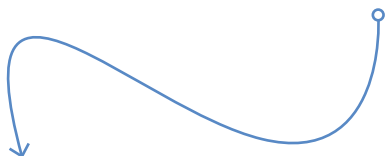
Le futur antérieur de Raspberry Pi

Avec son premier ordinateur monocarte Raspberry Pi a ouvert en 2012 un nouveau monde informatique. Jamais auparavant un PC polyvalent aussi petit et à un coût aussi faible n'avait été produit. La Fondation Raspberry Pi aura su démontrer que ses capacités et fonctions interactives étendues permettaient de l'utiliser pour autre chose que la simple navigation sur le web. Avec l'informatique physique permise par un GPIO à 40 broches et Python préchargé, le RPi a littéralement brisé la barrière des coûts et des compétences liés à la conception de projets automatisés basés sur l'IdO. Aujourd'hui n'importe qui, quelles que soient ses compétences techniques, peut connecter des capteurs et des affichages au RPi et le programmer en Python pour faire du calcul, effectuer des tâches d'entrée/sortie multiples ou afficher des données sur un serveur web.

La liaison durable avec Python

Python est resté un pilier de Raspberry Pi au cours de la dernière décennie ; malgré la mise à niveau de composants physiques comme la puissance de calcul, le graphique et les E/S par rapport aux modèles précédents, le support du langage lui-même n'a pas changé. Je pense que c'est en partie parce que Python se marie si bien, par nature, avec le prototypage et l'open source du RPi. Il existe des douzaines de bibliothèques Python qui permettent le traitement rapide des données, l'analyse et l'apprentissage machine. Tout comme l'extensibilité offerte par le GPIO au RPi, Python est extensible, modulaire et flexible. Les applications web comme Jupyter Notebooks ouvrent une vaste gamme de flux de travail dans les domaines de la science des données, du calcul scientifique et de l'apprentissage machine, avec des greffons qui ajoutent de nouveaux composants et s'intègrent aux composants existants. Il n'est donc pas surprenant que Python s'associe si bien avec RPi.

Présentation des produits SparkFun RP2040 !



Le RPi Pico est construit autour du puissant RP2040, et donc beaucoup des qualités indéniables du microcontrôleur proviennent des caractéristiques techniques de la puce. Pour le RP2040 Sparkfun propose trois nouvelles cartes qui utilisent toutes les caractéristiques de calcul offertes par la puce, ainsi que les environnements de développement multiplateforme C/C++ et MicroPython.

Nous avons ajouté à nos cartes pour le RP2040 une petite touche d'innovation propre à SparkFun :

- SparkFun RP2040 Thing Plus (fig. 5)
- Pro Micro RP2040 (fig. 6)
- carte processeur MicroMod RP2040 (fig. 7)

Toutes ces cartes ont huit fois plus de flash que le Pico. 16 Mo de mémoire flash, c'est bien sûr beaucoup pour sauvegarder des données hors ligne ! Avec autant de mémoire flash sur chaque carte, des modèles d'apprentissage machine avec TensorFlow utilisant le RP2040 sont parfaitement réalisables. Les cartes partagent également toutes les caractéristiques techniques du RP2040, notamment :

- Deux processeurs ARM Cortex-M0+ (jusqu'à 133 MHz)
- 264 ko de SRAM intégrée dans six banques
- Six E/S dédiées pour SPI Flash (avec XIP)
- 30 GPIO multifonctions : matériel dédié pour les périphériques les plus courants ; E/S programmable pour prise en charge de périphériques supplémentaires ; et quatre canaux CN/A 12 bits avec un capteur de température interne
- Développement multiplateforme C/C++ et MicroPython avec accès facile au débogage et à l'amorçage UF2

Chacun de ces microcontrôleurs RP2040 construits par SparkFun a cependant ses propres avantages, alors observons-les de plus près.

SparkFun RP2040 Thing Plus

Au facteur de forme *feather*, avec 18 broches GPIO, la RP2040 Thing Plus de SparkFun est une excellente carte pour les projets qui peuvent nécessiter de la mobilité ou qui intègrent un jeu complet de périphériques. Elle est équipée d'un connecteur JST pour batterie LiPo à cellule unique, et sert à la fois de circuit de charge et de jauge. Elle comporte aussi un emplacement pour carte micro-SD pour l'enregistrement de données ou un stockage supplémentaire. La

Thing Plus RP2040 comprend une LED WS2812 RGB adressable, des broches JTAG PTH, quatre trous de fixation et, bien sûr, un connecteur Qwiic qui facilite grandement le prototypage avec des capteurs et des affichages.

Pro Micro RP2040

La Pro Micro RP2040 est au format Pro Micro et offre une compatibilité USB-C complète. Comme la Thing Plus RP2040, elle comprend une LED WS2812 RGB adressable, un connecteur Qwiic, ainsi que des pastilles crénelées (*castellated pads*), et un bouton de démarrage et de réinitialisation. Les vias crénelés font de ce modèle un candidat idéal pour des projets tels que la création d'un contrôleur pour des programmes ou jeux informatiques ou la construction d'un clavier. Plus précisément, cette carte pourrait vous permettre de construire un clavier entièrement personnalisable pour l'adapter à vos besoins spécifiques. Si vous cherchez de l'inspiration pour un clavier qui permet des actions particulières, je recommande le projet de Jason Rudolph : <https://github.com/jasonrudolph/keyboard>. En fait, tout périphérique avec des E/S dédiées s'intégrerait parfaitement avec la Pro Micro RP2040. De plus, on peut construire tout cela sans rien perdre du confort de MicroPython.

Carte processeur MicroMod RP2040

Enfin, SparkFun sort une carte à processeur MicroMod pour le RP2040. L'écosystème MicroMod, est un système modulaire qui permet d'échanger les cartes porteuses (ou cartes de support) et les cartes à processeur. Informez-vous sur www.sparkfun.com/micromod. Ainsi, au lieu d'acheter des cartes qui ont un processeur prédéfini ou des types spécifiques d'entrées et de sorties, MicroMod vous permet de changer d'avis en cours de projet, soit avec le processeur, soit avec la carte porteuse.

SparkFun propose diverses cartes de support, équipées chacune d'un connecteur Qwiic, ce qui vous permet d'ajouter toujours plus de capteurs et d'afficheurs à votre projet. Dès à présent, les cartes porteuses offrent de multiples options pour tirer parti du RP2040.

- ATP (*all the pins* = liaisons à toutes les broches)
- Saisie et affichage
- Enregistrement de données
- Apprentissage machine
- Météo

Désormais, comme avec les autres cartes processeurs (avec les puces ESP32, Artemis, nRF52840 et SAMD51), vous pouvez utiliser

le RP2040 avec la carte à processeur MicroMod RP2040. Vous tirez ainsi pleinement parti de la puce RP2040 et pouvez ajouter les périphériques nécessaires pour votre projet. Si vous souhaitez vous familiariser rapidement avec la puce RP2040 et voir comment elle effectue diverses tâches, la carte processeur MicroMod RP2040 est aussi simple que possible pour l'expérimentation et le prototypage.

Dernières réflexions

Plus que jamais, la Fondation Raspberry Pi poursuit la création de produits performants et accessibles, et encourage les projets créatifs et open source. Le Raspberry Pi Pico, le RP2040, et tous les produits associés de SparkFun disposent non seulement d'une vaste documentation inégalée, mais ils sont à l'origine d'un environnement de développement avec MicroPython innovant. La seule question est de savoir comment commencer vos créations avec

le RP2040. Est-ce par le biais du Pico, très avantageux ? Ou la méthode de prototypage modulaire de MicroMod ? Ou peut-être la petite taille du Pro Micro. Ou bien encore grâce à un projet portable utilisant la carte RP2040 Thing Plus. Peu importe votre choix, allez-y ! Bâissez des projets en utilisant toutes les capacités de MicroPython et des logiciels *open source*. Dites *hello* à votre nouveau microcontrôleur favori à la sauce Raspberry Pi !

200711-04 – VF : Denis Lafourcade



Revendeurs agréés Raspberry Pi

Les produits à base de Raspberry et les accessoires RPi vous intéressent-ils ? SparkFun et Elektor sont tous deux revendeurs agréés de Raspberry Pi.

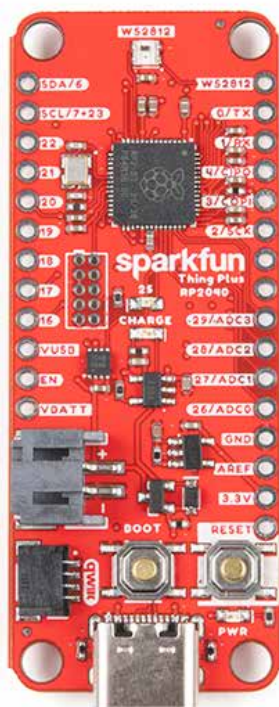


Figure 5. SparkFun Thing Plus



Figure 6. SparkFun Pro Micro



Figure 7. Carte processeur MicroMod RP2040



Accessoires

Vous trouverez chez SparkFun et Elektor les principaux accessoires mentionnés dans cet article :

- > Carte à processeur MicroMod RP2040
www.elektormagazine.fr/esfe-en-rpipo1



- > Carte à microcontrôleur Raspberry Pi Pico
www.elektormagazine.fr/esfe-en-rpipo3



- > Pro Micro RP2040
www.elektormagazine.fr/esfe-en-rpipo2



- > RP2040 Thing Plus de SparkFun
www.elektormagazine.fr/esfe-en-rpipo4

