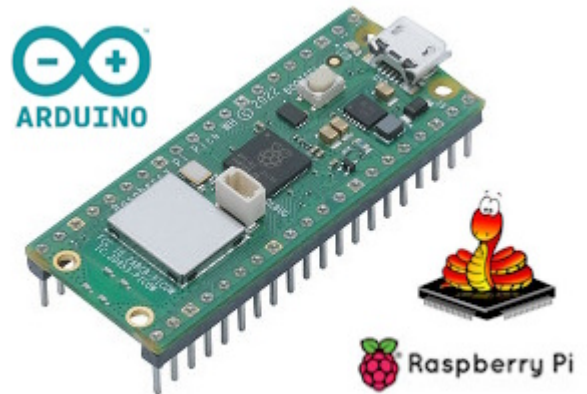




# Microcontrôleurs - Raspberry Pi Pico et Pico W

[Mise à jour le 17/2/2024]



## Ressources

- Documentation sur [raspberrypi.com](https://raspberrypi.com)
- Documentation [MicroPython](#)
- [Arduino IDE 2](#) (C,C++)
- [IDE Thonny](#) (MicroPython, CircuitPython)

## Lectures connexes

- Wiki matériels - "[Capteurs, afficheurs, préactionneurs, etc.](#)"
- [Arduino Nano RP2040 Connect](#)
- Bibliothèques - [Arduino Library List](#)

## Distributeurs

- [Go Tronic](#)

Les cartes Raspberry Pi Pico et Pico W sont basées sur un circuit [RP2040](#) conçu par Raspberry Pi. Il est très simple, a d'excellentes performances et un coût très faible (~5€). Sa programmation en MicroPython se fait via **REPL**, par **glisser-déposer** ou directement avec l'**IDE Thonny**.

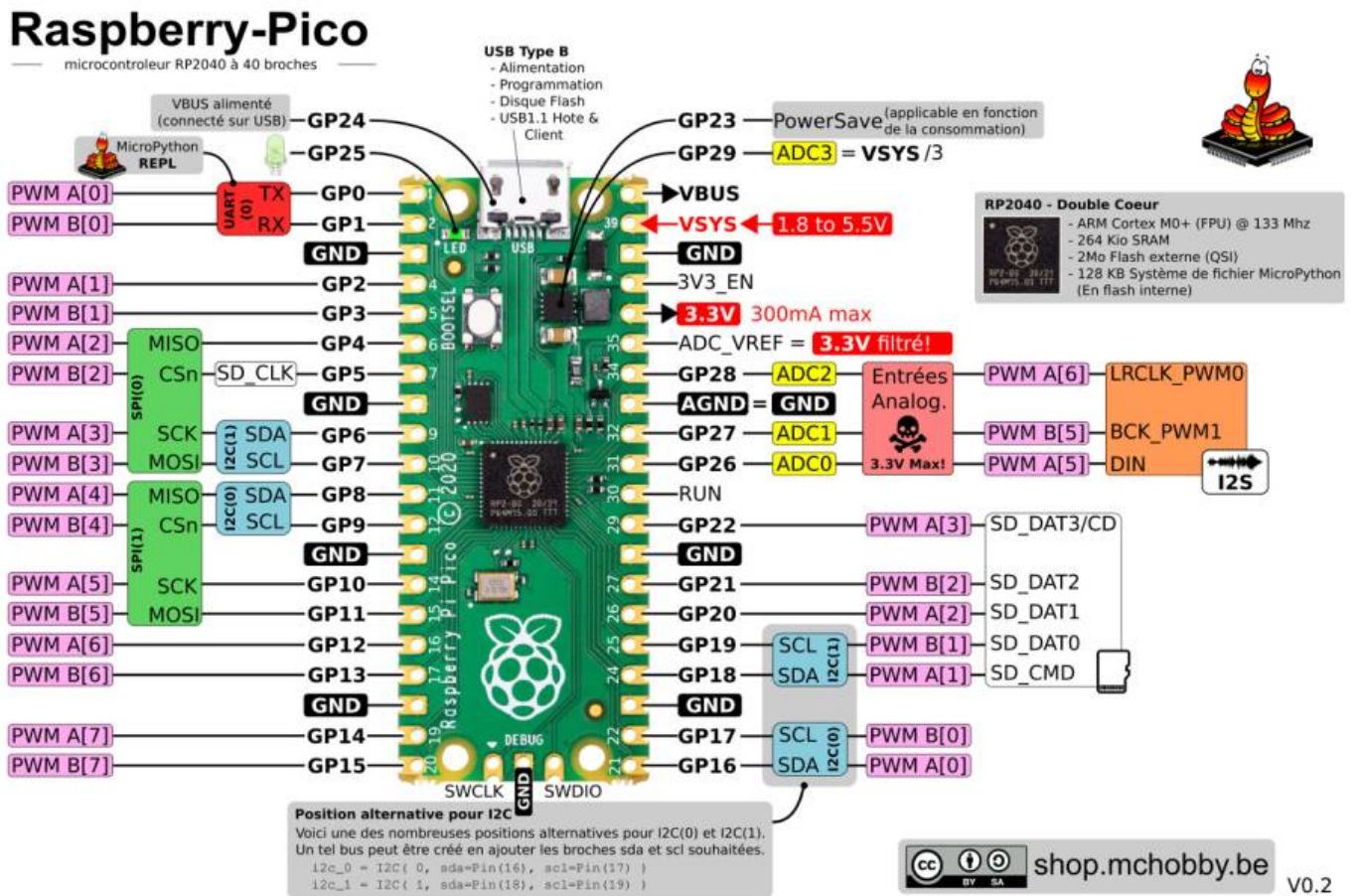
# A. Les Matériels

## A1. Raspberry Pi Pico

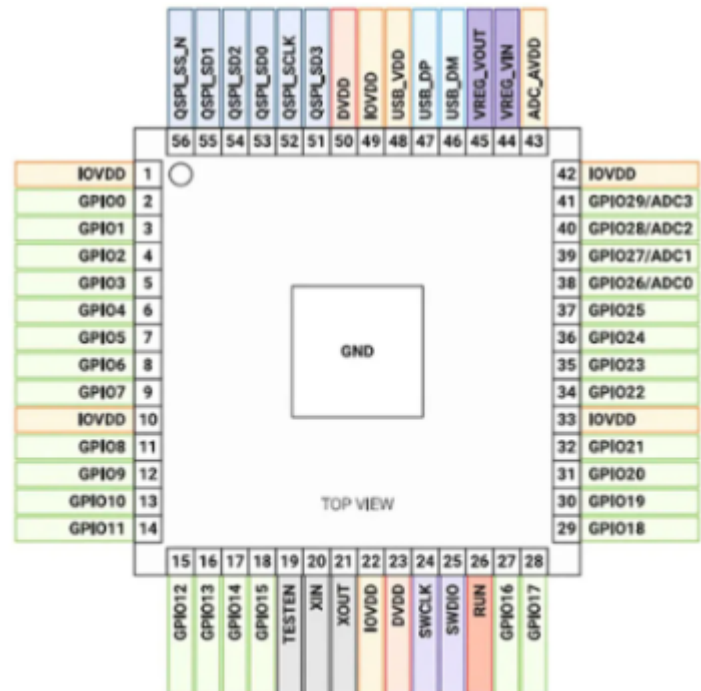
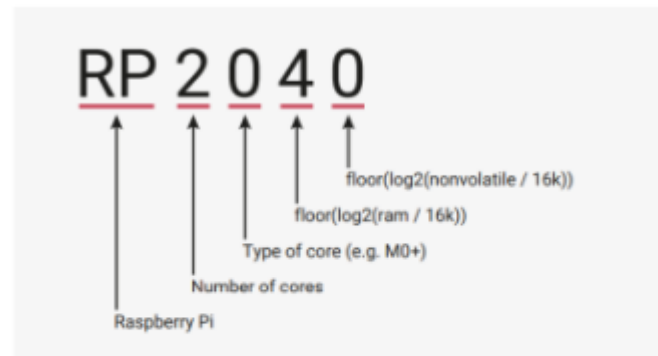
### 1.1 Généralités

- Documentation du [Raspberry Pi Pico](https://www.raspberrypi.com) sur le site [raspberrypi.com](https://www.raspberrypi.com)

### 1.2 Brochage



### 1.3 Caractéristiques de la carte



- **Alimentation:**
  - 5 Vcc via micro USB
  - 1,8 à 5,5 Vcc via la broche VSYS
- **Microcontrôleur:** [RP2040](#)
  - **Microprocesseur:** ARM Cortex-M0+ Dual Core à 133 MHz
  - **Mémoire SRAM:** 264 KB on-chip
  - **Module RTC** intégré
- **Mémoire Flash:** 2 MB externes
- **26 broches GPIO** comprenant:
  - 23 x E/S digitales
  - 3 x entrées analogiques (via ADC 12 bit)
  - 2 x interfaces UART
  - 2 x bus I2C
  - 16 x sorties PWM
  - 1 x port micro-USB 1.1: alimentation, programmation, hôte USB et périphérique de stockage USB
  - 1 x interface SWD de debug
- **LED** programmable sur GP25
- **Capteur** de température intégré

### 1.4 PDF

- [Raspberry Pi Pico Datasheet](#)
- [Getting started with Raspberry Pi Pico](#)
- [Raspberry Pi Pico Python SDK](#)

### TENSION ENTREE GPIO

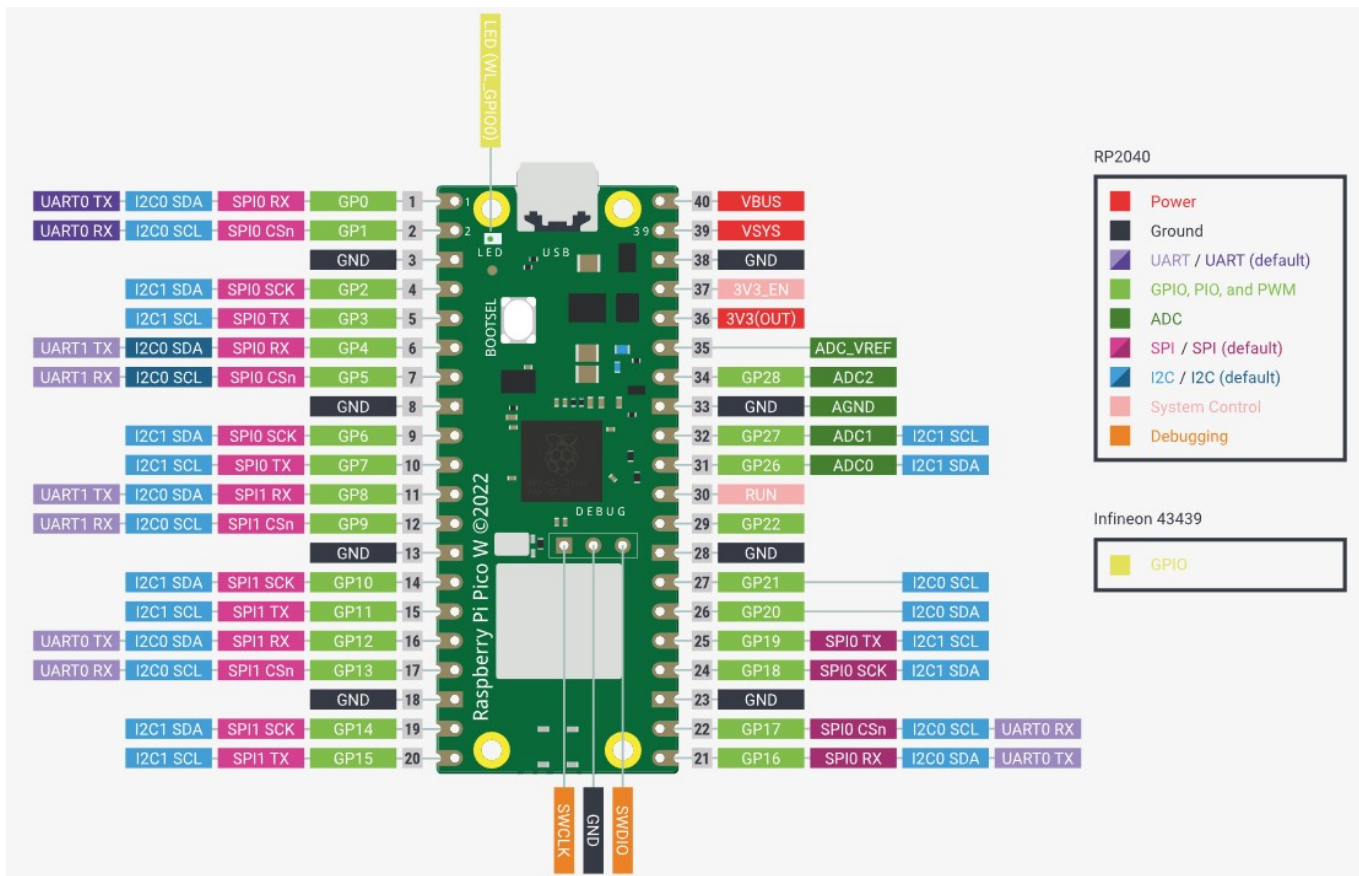
Les entrées et sorties logiques de cette carte sont uniquement compatibles **3,3Vcc**. Une tension supérieure endommagerait irrémédiablement la Raspberry Pi Pico.

## A2. Raspberry Pi Pico W

### 2.1 Généralités

- Documentation du [Raspberry Pi Pico W](#) sur le site [raspberrypi.com](https://raspberrypi.com).

### 2.2 Brochage



## 2.3 Caractéristiques

Raspberry Pi Pico W ajoute des interfaces sans fil monobande 2,4 GHz intégrées (802.11n) à l'aide de l'**Infineon CYW43439** tout en conservant le facteur de forme Pico. L'interface sans fil 2,4 GHz intégrée présente les caractéristiques suivantes :

- [Wifi 4](#) (802.11n) + [Bluetooth 5.2](#), bande unique (2,4 GHz)
- Sécurité : [WPA3](#)
- Point d'accès logiciel prenant en charge jusqu'à **quatre clients**

L'antenne est une antenne embarquée sous licence d'ABRACON (anciennement ProAnt).

## 2.4 PDF

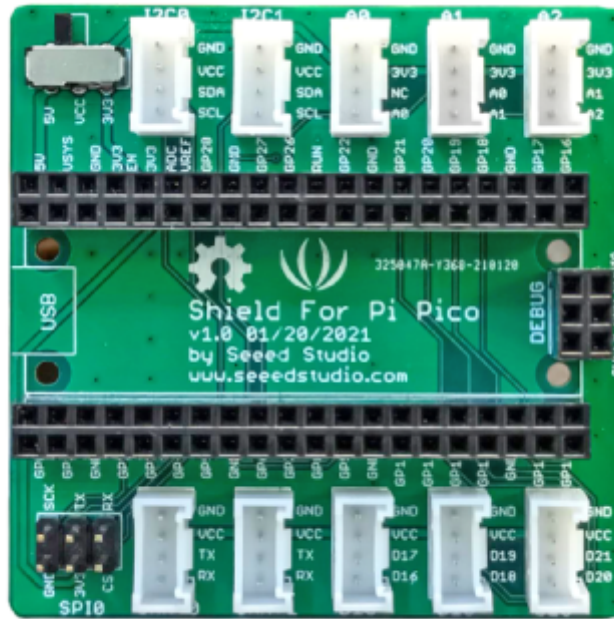
- [Raspberry Pi Pico W Datasheet](#)
- [Connecting to the Internet with Raspberry Pi Pico W](#)
- [Raspberry Pi Pico Python SDK](#)

## A3. Les shields

### 3.1 Shield Grove

#### 3.1.1 Description

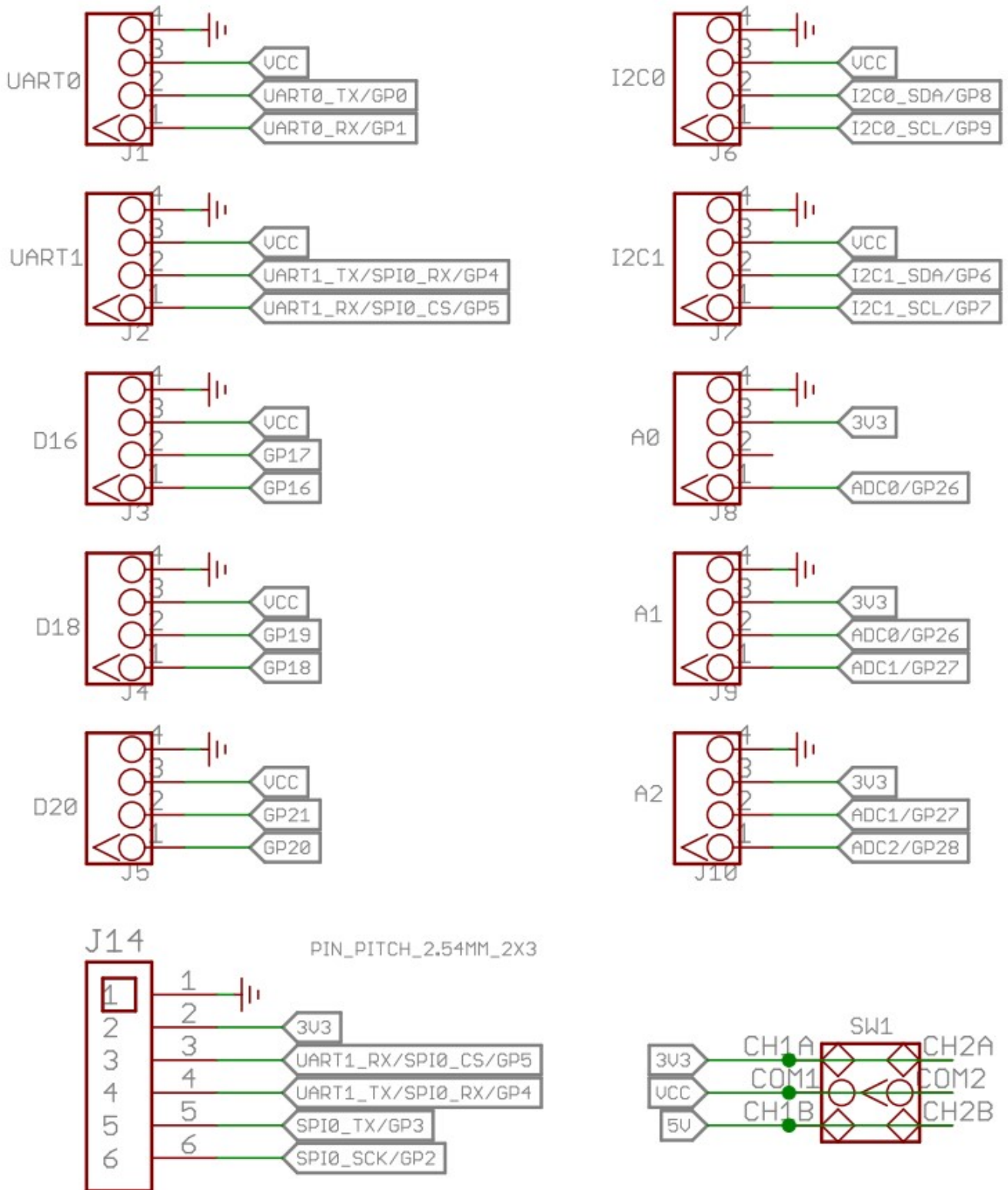
Le **Grove Shield Pico** est une carte d'interface permettant de raccorder sans souder les **capteurs**, les afficheurs et les préactionneurs Grove sur une carte Raspberry Pi Pico (W). [\[Schéma\]](#)



- Il est équipé de 10 connecteurs 4 broches dont :
  - **3 entrées analogiques,**
  - **3 entrées-sorties logiques,**
  - **2 interfaces I2C** et
  - **2 interfaces UART.**
- Le **bus SPI** est accessible sur un connecteur 6 broches mâles.

### 3.1.2 Schéma de la connectique du shield

- Brochage des connecteurs: **GND - Vcc - signal 2 - signal 1**

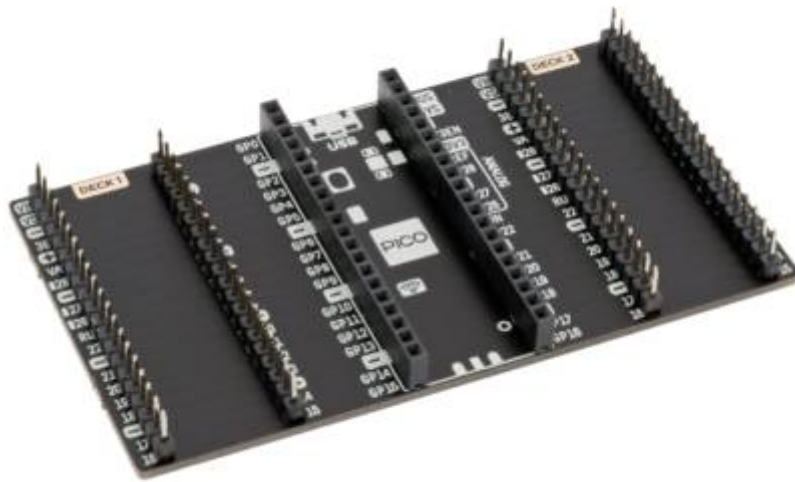


**Vérifier la position du commutateur** permettant de régler **Vcc=5V** ou **Vcc=3,3V** sur les connecteurs lors de l'utilisation des E/S numériques et des circuits I2C et UART.

### 3.2 Pico OMNIBUS

Doublez les broches GPIO et attachez deux modules complémentaires à un seul Raspberry Pi Pico

avec [Pico Omnibus](#).



## B. Les logiciels

### 1. Programmer les Paspberry Pi Pico en MicroPython

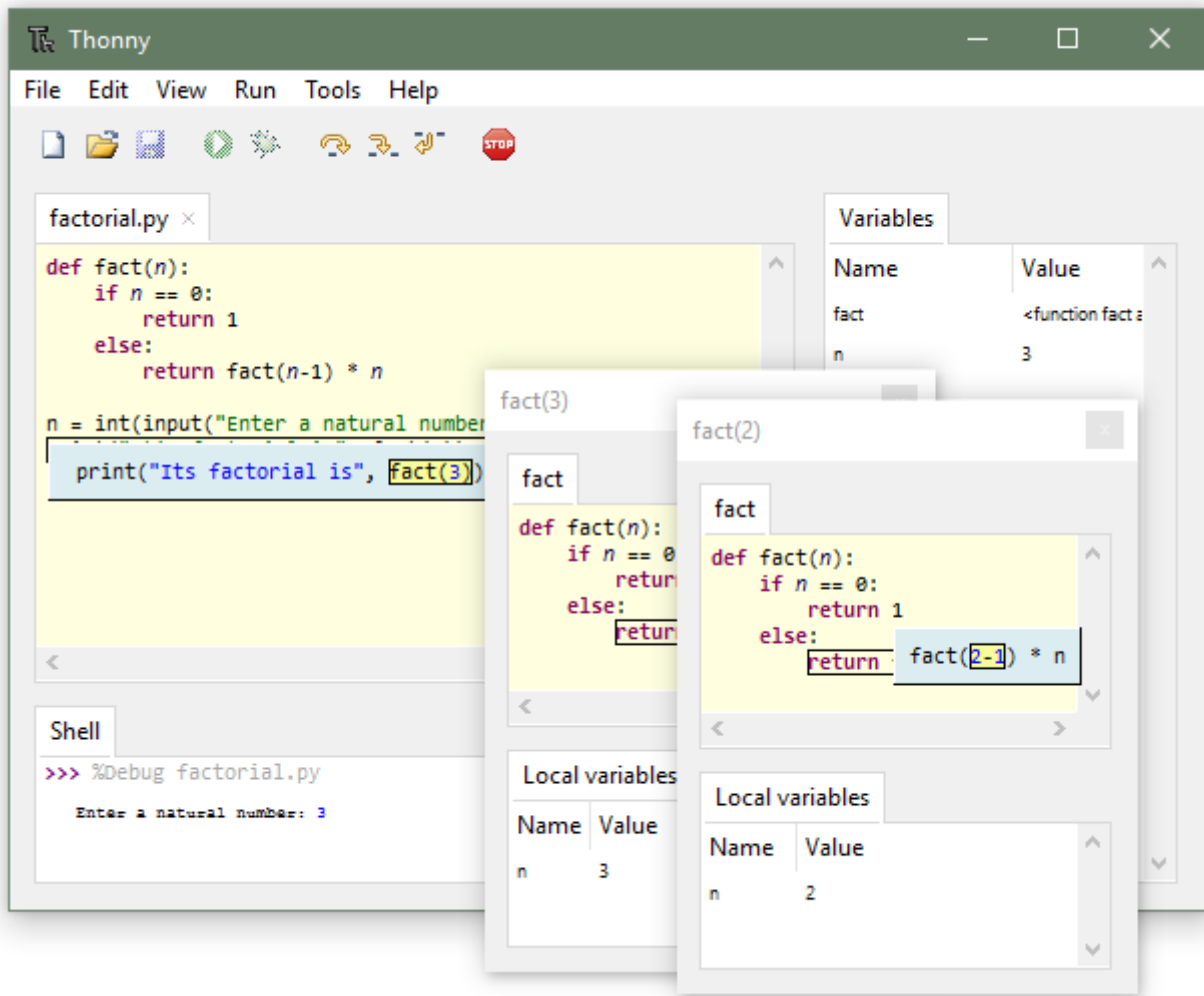
#### 1.1 Installer MicroPython

- **Voir le tutoriel** ["Getting Started with Raspberry Pi Pico W using MicroPython"](#)

#### 1.2 IDE Thonny

- **Installer** l'éditeur : [Thonny](#)





### 1.3 Exemples de programme en MicroPython

- [Getting started with Raspberry Pi Pico](#)
  - Blink the onboard LED
  - Use digital inputs and outputs
  - Control LED brightness with PWM
  - Control an LED with an analogue input
- [How to Connect Raspberry Pi Pico W to the Internet](#)
- [How to Use Raspberry Pi Pico W With Node-RED](#)
- [Raspberry Pi Pico W Web Server Tutorial with MicroPython](#)
- [MQTT and Raspberry Pi Pico W: Start with Mosquitto \(MicroPython\)](#)
- [Utiliser la PWM sur la Raspberry Pi Pico](#)

### 1.4 Kit de développement Windows (SDK) et VSCode

- [La fondation Raspberry Pi officialise un kit de développement Windows pour le Raspberry Pico](#)

## 2. Programmer les Paspberry Pi Pico (en C/C++) sous Arduino

- **Ressources**

- [Documentation](#) officielle Arduino.

- **Tutoriels**

- [Utiliser la Pi Pico depuis l'Arduino IDE](#)
- [Basic Multicore Pico Project](#)

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=microc:uc:rp2&rev=1739784728>

Last update: **2025/02/17 10:32**

