

# FAQ

## Frequently Asked Questions for Raspberry Pi Pico and RP2040

# Colophon

Copyright © 2020 Raspberry Pi (Trading) Ltd.

The documentation of the RP2040 microcontroller is licensed under a Creative Commons [Attribution-NoDerivatives 4.0 International](#) (CC BY-ND).

build-date: 2021-11-04

build-version: 150df05-clean

## About the SDK

Throughout the text "the SDK" refers to our [Raspberry Pi Pico SDK](#). More details about the SDK can be found in the [Raspberry Pi Pico C/C++ SDK](#) book. Source code included in the documentation is Copyright © 2020 Raspberry Pi (Trading) Ltd. and licensed under the [3-Clause BSD](#) license.

## Legal Disclaimer Notice

TECHNICAL AND RELIABILITY DATA FOR RASPBERRY PI PRODUCTS (INCLUDING DATASHEETS) AS MODIFIED FROM TIME TO TIME ("RESOURCES") ARE PROVIDED BY RASPBERRY PI (TRADING) LTD ("RPTL") "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN NO EVENT SHALL RPTL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE RESOURCES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

RPTL reserves the right to make any enhancements, improvements, corrections or any other modifications to the RESOURCES or any products described in them at any time and without further notice.

The RESOURCES are intended for skilled users with suitable levels of design knowledge. Users are solely responsible for their selection and use of the RESOURCES and any application of the products described in them. User agrees to indemnify and hold RPTL harmless against all liabilities, costs, damages or other losses arising out of their use of the RESOURCES.

RPTL grants users permission to use the RESOURCES solely in conjunction with the Raspberry Pi products. All other use of the RESOURCES is prohibited. No licence is granted to any other RPTL or other third party intellectual property right.

**HIGH RISK ACTIVITIES.** Raspberry Pi products are not designed, manufactured or intended for use in hazardous environments requiring fail safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, weapons systems or safety-critical applications (including life support systems and other medical devices), in which the failure of the products could lead directly to death, personal injury or severe physical or environmental damage ("High Risk Activities"). RPTL specifically disclaims any express or implied warranty of fitness for High Risk Activities and accepts no liability for use or inclusions of Raspberry Pi products in High Risk Activities.

Raspberry Pi products are provided subject to RPTL's [Standard Terms](#). RPTL's provision of the RESOURCES does not expand or otherwise modify RPTL's [Standard Terms](#) including but not limited to the disclaimers and warranties expressed in them.

# Table of Contents

Colophon .....	1
Legal Disclaimer Notice .....	1
Frequently Asked Questions .....	3
Is RP2040 5V tolerant? .....	3
I've accidentally connected +5V to my RP2040-based microcontroller. Is the pin dead? .....	3
Can I buy the RP2040 chip? .....	3
Can I solder it myself? .....	3
What OS does it run? .....	3
What is the maximum Flash size? .....	3
Is it open-source? .....	3
Should I buy a Raspberry Pi Pico or a Raspberry Pi Zero? .....	3
Can I overclock? .....	4
Is it supported in the Arduino or other third-party programming environments? .....	4
What low-power modes are there? .....	4
I'm using VSCode. How do I set a breakpoint on the second core? .....	4
GCC 10.2 is available. Why aren't you using it by default? .....	4
What is the maximum source impedance required for the ADC inputs to achieve the 9-bit resolution? .....	4
Where do I get technical support? .....	4
Can I run machine learning tools? .....	4
I don't want to use C or Python, can I write assembler? .....	5
Can I use an RTOS? .....	5
Is there a downloadable version of "Get started with MicroPython on Raspberry Pi Pico"? .....	5
What is a UF2 file? .....	5
It doesn't support my communications protocol, what can I do? .....	5
Why use micro-USB instead of USB-C on the Raspberry Pi Pico board? .....	5
Why is there no reset button? Constantly unplugging and plugging in is a pain! .....	5
Can I use my Raspberry Pi to develop for the Raspberry Pi Pico? .....	5
How do I completely clear the Flash on my Raspberry Pi Pico? .....	6
How can I find out when SDK or similar releases happen? .....	6
What are GPIO 23,24,25 and 29 used for on the Raspberry Pi Pico board? .....	6
What is the maximum current rating of the GPIOs? .....	6
Can I use USB serial when debugging using SWD? .....	6
What caches does the RP2040 have? .....	6
Appendix A: Documentation Release History .....	7

# Frequently Asked Questions

Some basic question and answers on frequently asked questions about Raspberry Pi Pico and RP2040.

## Is RP2040 5V tolerant?

No. RP2040 microcontrollers use 3.3V for GPIO.

## I've accidentally connected +5V to my RP2040-based microcontroller. Is the pin dead?

It depends; often RP2040 will survive, but it is not recommended, and can reduce the life of the microcontroller.

## Can I buy the RP2040 chip?

Yes. See [the product page](#) for details.

## Can I solder it myself?

For the RP2040 itself, yes; it's not easy, and you will need magnification of some sort and a very good soldering iron to go with a very steady hand, but it can be hand soldered. The Raspberry Pi Pico board on the other hand has been designed so that it can be easily soldered down to another PCB; it's why it has castellated contacts on each edge.

## What OS does it run?

Like other microcontrollers, it doesn't run an OS by default, but we supply an [SDK](#) which provides a rich library of functions for using the hardware/features of the chip, along with higher-level functionality often found in an OS. You can use this to write "bare metal" C/C++ applications easily, or you can use [MicroPython](#), which is even simpler for beginners.

## What is the maximum Flash size?

16 MiB (2<sup>24</sup> bytes) of Flash memory.

## Is it open-source?

The Raspberry Pi Pico board design files are open-source, along with all the provided software and examples, see the links [in the documentation](#). The internal design of the RP2040 microcontroller itself is not open-source.

## Should I buy a Raspberry Pi Pico or a Raspberry Pi Zero?

These are two very different devices, a microcontroller and a microcomputer, intended for different purposes. Raspberry Pi Pico is a great device for dedicating to a very specific task, whereas Raspberry Pi Zero is a multipurpose device. Raspberry Pi Zero has HDMI out, a camera interface, etc; Raspberry Pi Pico does not. However, Raspberry Pi Pico has an on-board ADC as well as other peripherals not present on Raspberry Pi Zero, and consumes considerably less power; it is therefore much more suited to embedded applications than Raspberry Pi Zero.

## Can I overclock?

The maximum `clk_sys` clock speed is 133MHz at normal core voltage (1.1V  $\pm$ 10%), but it can be overclocked. However, this requires more power and a higher core voltage, and may affect the life expectancy of your device. No warranties are given when running outside of the limits specified in the [RP2040 Datasheet](#).

## Is it supported in the Arduino or other third-party programming environments?

Yes. Alongside the very comprehensive [C/C++ SDK](#) both Raspberry Pi Pico and other RP2040-based boards are supported by an official port of [MicroPython](#), while Adafruit also ported [CircuitPython](#) to RP2040. There has been both an [unofficial](#) and an [official](#) port of the Arduino environment.

## What low-power modes are there?

There is no specific low-power mode, but you can do various things to reduce power consumption: slow the system clock, power down some of the memories, gate the clock to peripherals, reduce the core supply voltage, etc.

There are two specific modes for turning off parts of the device when not being used:

SLEEP mode is when processors are in wfi/wfe and DMA is inactive, so you can shut off most system clocks, including things like bus fabric.

DORMANT mode is when you shut down all oscillators, and so is lower power still, but then you have fewer options for waking.

## I'm using VSCode. How do I set a breakpoint on the second core?

At the moment you need to use the command window in VSCode to set a GDB breakpoint on the second core.

## GCC 10.2 is available. Why aren't you using it by default?

Our development work has been using the more common versions of GCC as supplied with various distributions. We have checked that everything builds using major GCC versions from 6.3.1 to 10.2, so all of those should work.

## What is the maximum source impedance required for the ADC inputs to achieve the 9-bit resolution?

We are currently working on characterising this, and will update the [RP2040 Datasheet](#) when it's done.

## Where do I get technical support?

For help on the Raspberry Pi Pico, C/C++ SDK or RP2040, you can use [our forums](#). For all other things, for example [CircuitPython](#), [Arduino](#) or [MicroPython](#) support, please contact the appropriate vendors.

There is also an [unofficial Discord server](#) run by Pimoroni.

## Can I run machine learning tools?

Yes. TensorFlow Lite for Microcontrollers has been [ported to Raspberry Pi Pico](#) by Google. It is an early version so there is lots of optimisation still to do. However there are some [example projects](#) starting to appear.

## I don't want to use C or Python, can I write assembler?

Of course; check out the [SDK](#) which has some library functions written in assembler.

The processor (dual-core ARM Cortex M0+) implements the ARMv6-M Thumb instruction set, including a number of 32-bit instructions that use Thumb-2 technology. The ARMv6-M instruction set comprises all of the 16-bit Thumb instructions from ARMv7-M excluding CBZ, CBNZ and IT and the 32-bit Thumb instructions BL, DMB, DSB, ISB, MRS and MSR.

## Can I use an RTOS?

Yes, we will be working on porting an RTOS soon.

## Is there a downloadable version of "Get started with MicroPython on Raspberry Pi Pico"?

Yes there is a [downloadable PDF](#) of "Get started with MicroPython on Raspberry Pi Pico" by Gareth Halfacree and Ben Everard.

## What is an UF2 file?

It's a file format for flashing microcontrollers, developed by Microsoft. More information can be found at <https://github.com/microsoft/uf2>.

## It doesn't support my communications protocol, what can I do?

There are inbuilt hardware blocks to support UART, SPI and I2C, but there is also a unique and very specialised hardware block called the PIO (Programmable I/O) which can be programmed to implement all sorts of different digital protocols, both input and output. For example you can program it to be a UART, (in fact, with 8 channels, it can implement another 4 TX/RX UARTS, or 8 RX or 8 TX's, or combinations thereof!), RS485, and almost any other protocol you can think of. See the [RP2040 Datasheet](#) for more information on the PIO, and [pico-examples](#) for some examples of the PIO in action.

## Why use micro-USB instead of USB-C on the Raspberry Pi Pico board?

Cost! USB-C connectors are still more expensive than micro-USB, and when you are making millions of devices, even a few cents more per connector really adds up. We also believe that, especially in the educational arena, micro-USB is still much more common than USB-C. If you must have it, RP2040 boards with USB-C are available from our partners.

## Why is there no reset button? Constantly unplugging and plugging in is a pain!

Cost is the main reason, adding an extra button can make a big difference. But we also expect that if you are making a lot of changes during development, you will have set up the SWD (Serial Wire Debug) system, which allows you to download code to the board via the debugger (as described in [Getting started with Raspberry Pi Pico](#)). This means you don't need to power off or use the BOOTSEL button at all. There are other options as well, such as [adding your own reset button](#).

## Can I use my Raspberry Pi to develop for the Raspberry Pi Pico?

Of course. Although any model of Raspberry Pi would be able to communicate with the Raspberry Pi Pico, we recommended more recent models for development, especially if using the VSCode IDE as that requires a strong CPU

and at least 2GB of RAM. So the Raspberry Pi 4 and Raspberry Pi 400 are best when using an IDE, but if you are using the command line, the Raspberry Pi 3 or even Raspberry Pi 2 should be OK. Compile times will be a bit longer though, and we do recommend a 16GB card to ensure you have enough space for all the code trees.

## How do I completely clear the Flash on my Raspberry Pi Pico?

If you are moving between MicroPython and C programs, you might find you have problems with certain applications, and need to clear the Flash completely to start again. We provide a special uf2 program that can do this for you called `nuke.uf2`. It can be found [here](#) and the source code for it can be found at <https://github.com/raspberrypi/pico-examples/tree/master/flash/nuke/nuke.c>.

## How can I find out when SDK or similar releases happen?

Every now and then we will update the SDK or any of the other Raspberry Pi Pico [repositories](#). You can easily be informed when this happens using GitHub. Login to your GitHub account, and subscribe to the relevant repository - Watch (top right) → Custom → Releases. You can also Star a repository if you like it!

## What are GPIO 23,24,25 and 29 used for on the Raspberry Pi Pico board?

These GPIOs are used on the Raspberry Pi Pico board for the following purposes:

- GPIO23 goes to the PS pin of the voltage regulator, to adjust it between PFM and PWM mode.
- GPIO24 is the centre of a 5k6/10k voltage divider off the VBUS from USB, on the USB side of a diode, so it detects if USB is supplying voltage
- GPIO 25 is the on-board LED
- GPIO29/ADC3 is another voltage divider to measure VSYS, but with an extra FET in place so the divider can't leak into the ADC and 3.3V rail, when the 3.3V rail is off

See the [Raspberry Pi Pico Datasheet](#) for more information.

## What is the maximum current rating of the GPIOs?

In total (adding up all the individual GPIO demands), the current for the IOVDD must be  $\leq 50\text{mA}$ .

## Can I use USB serial when debugging using SWD?

No, you need to use a UART based console when debugging using SWD. This is because the USB stack will be paused when the RP2040 cores are stopped during debugging, which will cause any attached devices to disconnect.

## What caches does the RP2040 have?

There is just one, an execute-in-place (XIP) cache on the Flash (because Flash memory is quite slow). There are otherwise no instruction or data caches.

# Appendix A: Documentation Release History

Table 1.  
Documentation  
Release History

Release	Date	Description
1.0	21/Jan/2021	<ul style="list-style-type: none"> <li>Initial release</li> </ul>
1.1	26/Jan/2021	<ul style="list-style-type: none"> <li>Minor corrections</li> <li>Extra information about using DMA with ADC</li> <li>Clarified M0+ and SIO CPUID registers</li> <li>Added more discussion of Timers</li> <li>Update Windows and macOS build instructions</li> <li>Renamed books and optimised size of output PDFs</li> </ul>
1.2	01/Feb/2021	<ul style="list-style-type: none"> <li>Minor corrections</li> <li>Small improvements to PIO documentation</li> <li>Added missing TIMER2 and TIMER3 registers to DMA</li> <li>Explained how to get MicroPython REPL on UART</li> <li>To accompany the V1.0.1 release of the C SDK</li> </ul>
1.3	23/Feb/2021	<ul style="list-style-type: none"> <li>Minor corrections</li> <li>Changed font</li> <li>Additional documentation on sink/source limits for RP2040</li> <li>Major improvements to SWD documentation</li> <li>Updated MicroPython build instructions</li> <li>MicroPython UART example code</li> <li>Updated Thonny instructions</li> <li>Updated Project Generator instructions</li> <li>Added a FAQ document</li> <li>Added errata <a href="#">E7</a>, <a href="#">E8</a> and <a href="#">E9</a></li> </ul>
1.3.1	05/Mar/2021	<ul style="list-style-type: none"> <li>Minor corrections</li> <li>To accompany the V1.1.0 release of the C SDK</li> <li>Improved MicroPython UART example</li> <li>Improved Pinout diagram</li> </ul>
1.4	07/Apr/2021	<ul style="list-style-type: none"> <li>Minor corrections</li> <li>Added errata <a href="#">E10</a></li> <li>Note about how to update the C SDK from Github</li> <li>To accompany the V1.1.2 release of the C SDK</li> </ul>



Release	Date	Description
1.4.1	13/Apr/2021	<ul style="list-style-type: none"><li>• Minor corrections</li><li>• Clarified that all source code in the documentation is under the <a href="#">3-Clause BSD</a> license.</li></ul>
1.5	07/Jun/2021	<ul style="list-style-type: none"><li>• Minor updates and corrections</li><li>• Updated FAQ</li><li>• Added SDK release history</li><li>• To accompany the V1.2.0 release of the C SDK</li></ul>
1.6	23/Jun/2021	<ul style="list-style-type: none"><li>• Minor updates and corrections</li><li>• ADC information updated</li><li>• Added errata <a href="#">E11</a></li></ul>
1.6.1	30/Sep/2021	<ul style="list-style-type: none"><li>• Minor updates and corrections</li><li>• Information about B2 release</li><li>• Updated errata for B2 release</li></ul>
1.7.0	03/Nov/2021	<ul style="list-style-type: none"><li>• Minor updates and corrections</li><li>• Fixed some register access types and descriptions</li><li>• Added core 1 launch sequence info</li><li>• Described SDK "panic" handling</li><li>• Updated picotool documentation</li><li>• Additional examples added to <b>Appendix A: App Notes</b> appendix in the <a href="#">Raspberry Pi Pico C/C++ SDK</a> book</li><li>• To accompany the V1.3.0 release of the C SDK</li></ul>
1.7.1	04/Nov/2021	<ul style="list-style-type: none"><li>• Minor updates and corrections</li><li>• Better documentation of USB double buffering</li><li>• Picoprobe branch changes</li><li>• Updated links to documentation</li></ul>

The latest release can be found at <https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-faq.pdf>.



**Raspberry Pi**

Raspberry Pi is a trademark of the Raspberry Pi Foundation

---

**Raspberry Pi Trading Ltd**