

Introduction of Raspberry_Pi_Pico Kit

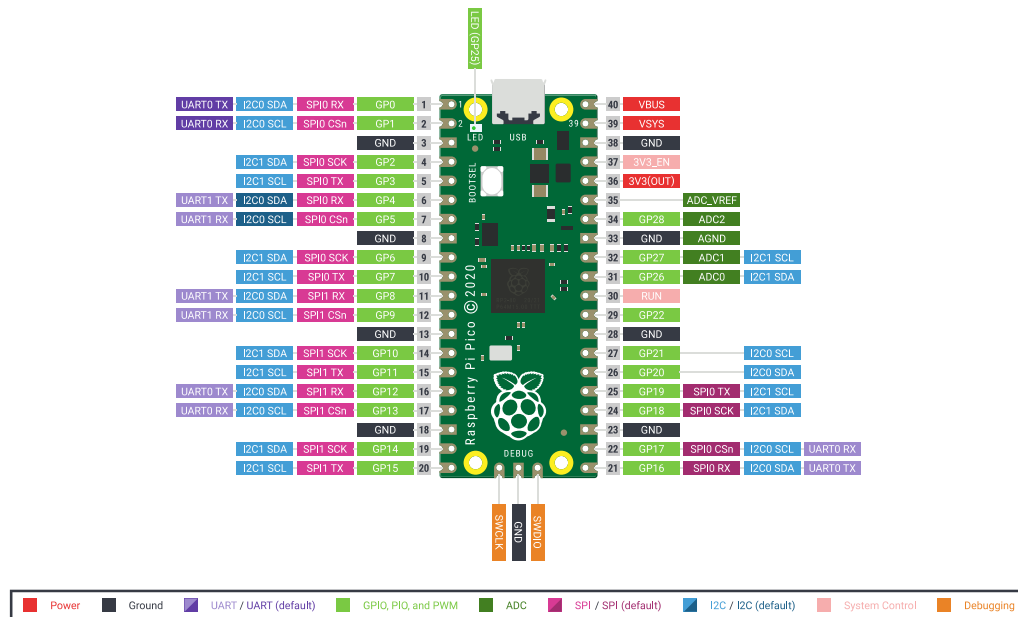
1. Raspberry Pi Pico

1.1 Technical Specification

Raspberry Pi Pico is a low-cost, high-performance microcontroller board with flexible digital interfaces. Key features include:

- ✧ [RP2040](#) microcontroller chip designed by Raspberry Pi in the United Kingdom
- ✧ Dual-core Arm Cortex M0+ processor, flexible clock running up to 133 MHz
- ✧ 264KB of SRAM, and 2MB of on-board Flash memory
- ✧ Castellated module allows soldering direct to carrier boards
- ✧ USB 1.1 with device and host support
- ✧ Low-power sleep and dormant modes
- ✧ Drag-and-drop programming using mass storage over USB
- ✧ 26 × multi-function GPIO pins
- ✧ 2 × SPI, 2 × I2C, 2 × UART, 3 × 12-bit ADC, 16 × controllable PWM channels
- ✧ Accurate clock and timer on-chip
- ✧ Temperature sensor
- ✧ Accelerated floating-point libraries on-chip
- ✧ 8 × Programmable I/O (PIO) state machines for custom peripheral support

1.1.1 Pinout and Design Files



- Download the [Pinout Diagram](#) (PDF)

1.1.2 Design Files

- Download [Design Files](#) (Cadence Allegro)
- Download [STEP File](#)
- Download [Fritzing Part](#)

NOTE: More information on Fritzing is available on the fritzing.org web site.

1.2 Documentation

Documentation for Raspberry Pi Pico and other RP2040-based boards.

1.2.1 RP2040 Device

RP2040 Datasheet

A microcontroller by Raspberry Pi

Hardware design with RP2040

Using RP2040 microcontrollers to build boards and products

1.2.2 Raspberry Pi Pico

[Raspberry Pi Pico Datasheet](#)

An RP2040-based microcontroller board.

[Getting started with Raspberry Pi Pico](#)

C/C++ development with Raspberry Pi Pico and other RP2040-based microcontroller boards

NOTE: While it is not officially supported there is a [Pico Setup for Windows](#) installation tool which automates installation of the C/C++ SDK on Windows 10.

1.2.3 Software Development

[Raspberry Pi Pico C/C++ SDK](#)

Libraries and tools for C/C++ development on RP2040 microcontrollers

[Raspberry Pi Pico Python SDK](#)

A MicroPython environment for RP2040 microcontrollers

The API level Doxygen documentation for the Raspberry Pi Pico C/C++ SDK is also available [as a micro-site](#).

NOTE: If you are building applications with the C/C++ SDK and targeting boards other than the Raspberry Pi Pico, you will need to pass `-DPICO_BOARD=boardname` to CMake. Here boardname is the name of your board, e.g. for the Adafruit Feather RP2040 you should pass `-DPICO_BOARD=adafruit_feather_rp2040`. See the [boards/ directory](#) in the Pico SDK, and the [forums](#), for more information.

1.3 Software Utilities

1.3.1 What is on your Pico?

If you have forgotten what has been programmed into your Raspberry Pi Pico, and the program was built using our Pico C/C++ SDK, it will usually have a name and other useful information embedded into the binary. You can use the [Picotool](#) command line utility to find out these details. Full instructions on how to use Picotool to do this are available in our ['getting started'](#) documentation.

- Go to the [Picotool Github repository](#).

1.3.2 Debugging using another Raspberry Pi Pico

It is possible to use one Raspberry Pi Pico to debug another Pico. This is possible via picoprobe, an application that allows a Pico to act as a USB → SWD and UART converter. This makes it easy to use a Pico on non-Raspberry Pi platforms such as Windows, Mac, and Linux computers where you don't have GPIOs to connect directly to your Pico. Full instructions on how to use Picoprobe to do this are available in our ['getting started'](#) documentation.

- Download the [UF2 file](#)
- Go to the [Picoprobe Github repository](#)

1.3.3 Resetting Flash memory

Pico's BOOTSEL mode lives in read-only memory inside the RP2040 chip, and can't be overwritten accidentally. No matter what, if you hold down the BOOTSEL button when you plug in your Pico, it will appear as a drive onto which you can drag a new UF2 file. There is no way to brick the board through software. However, there are some circumstances where you might want to make sure your Flash memory is empty. You can do this by dragging and dropping a special UF2 binary onto your Pico when it is in mass storage mode.

- Download the [UF2 file](#)
- See the [code on Github](#)