[seeedstudio.com](https://www.seeedstudio.com)

Rust, RT-Thread OS & FreeRTOS come to the Raspberry Pi Pico - Latest open tech from seeed studio

Jonathan Tan See author's posts

6-8 minutes

Another week brings another piece of exciting news for Raspberry Pi's brand new Raspberry Pi Pico microcontroller unit (MCU)! Following its release in January 2021, the community has been abuzz with trying different programs, tools and platforms with the Pico. Today, we bring you news about support for Rust, RT-Thread OS, and FreeRTOS on the Raspberry Pi!
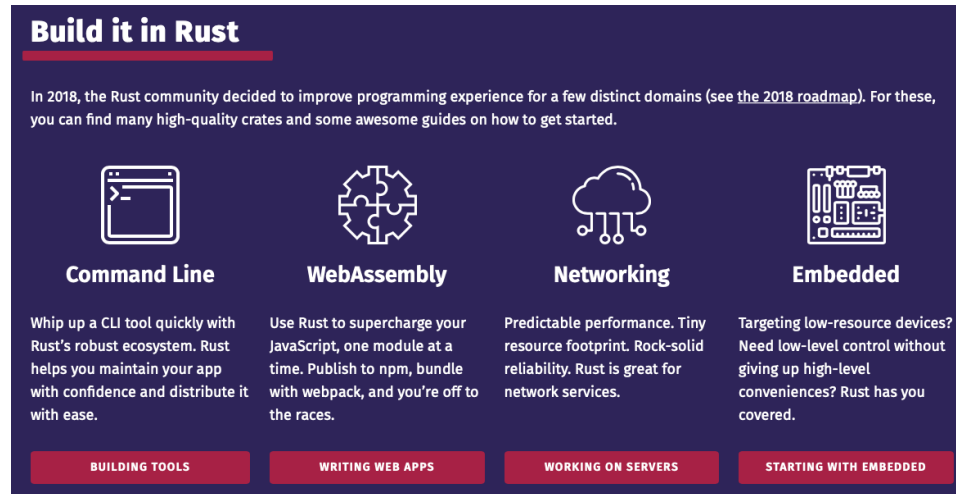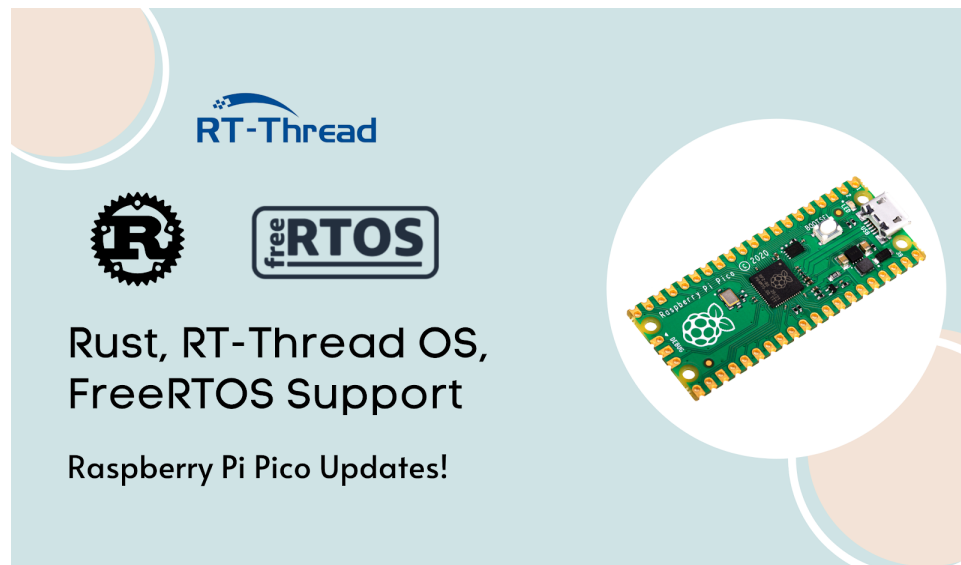

In case you missed it…

The Raspberry Pi Pico features the RP2040 dual-core Cortex-M0+, Raspberry Pi Foundation's very own silicon, and was announced to support MicroPython and C at launch. Shortly after, Arduino announced their plans to port the Arduino core to the Raspberry Pi Pico and other RP2040 boards. Read more about Arduino support

for the Raspberry Pi Pico and the RP2040 in our previous article here.

The Rust Programming Language is big in the embedded world. It's known to be fast and incredibly memory-efficient, which makes it great for microcontrollers where the optimisation of processing capabilities and power consumption are critical.

Source: Rust-Lang.org

In particular, Rust is widely used for IoT applications and servers because of its tiny resource footprint and reliable performance. It also features a great collection of libraries that helps developers build their applications quickly and easily.

Read more about Rust on their official website.

Rust Code Running on Raspberry Pi Pico!

Recently, Jonathan Pallant shared about his success to run some simple LED blinking code in Rust on the Raspberry Pi Pico!

The process is fairly complex and experimental at the moment, but I'll do my best to run it through in simple terms.

1. The RP2040 chip has external QSPI (Quad Serial Peripheral Interface) flash memory.

2. First, the manufacturer pre-programmed internal bootloader reads the first 256 bytes of this external flash and copies it to the top of the SRAM (static RAM).

3. Pallant has used this first 256 bytes as a 'second stage bootloader',

which is used to reconfigure the XIP_SSI (Execute in Place Synchronous Serial Interface) to the right values for the external QSPI flash.

4. Finally, the user application is loaded by reading a vector table at the QSPI flash memory address, which points to the remaining contents of the QSPI flash (excluding the second stage bootloader).

In a nutshell, Pallant's code has reconfigured the Raspberry Pi Pico to be able to read and execute the user application written in Rust!

To compile the code, he explains that he had to first build the Rust code with "cargo build -release", then use the elf2uf2 tool from the pico-sdk to create a UF2 file. This UF2 file can then be copied to the ROM bootloader, similar to how we flash MicroPython code to the Pico.

If you want to take a look at the specifics, he's also released his code and documentation on GitHub.

Rust Development is Underway!

Rust is being developed actively for use on the Raspberry Pi Pico! As of writing, the latest developments include a Peripheral Access Crate (PAC) for the Raspberry Pi Pico, which will allow Rust to be used for interfacing with the Pico's peripheral input output interfaces.

If you want to follow the latest projects and progress, be sure to visit the rp-rs Github.

The exciting news doesn't stop there – RT-Thread OS has announced their official support for the Raspberry Pi Pico, now available in the RT-Thread Studio IDE!

RT-Thread OS on the Raspberry Pi Pico

RT-Thread is an open source embedded real-time operating system specialising in IoT purposes. For those unfamiliar, a real-time operating system (or RTOS) is an operating system that serves real-time applications in processing data as it comes in, without buffer delays.

RT-Thread has been around since 2006, and has since been strongly supported with community development for more than a decade. The advantages of RT-Thread include:

- Extreme resource efficiency, with the minimum kernel only requiring 1.2KB RAM and 3KB Flash

- Integrated power management components with low-power modes to minimise system power consumption

- Complete yet still-growing package ecosystem includes middle-tier component supports such as file system, network protocols, GUI, audio frameworks etc.

- Easy to use coding style with console shell debugging and other development tools

- High scalability from minimal, tiny boards like the Raspberry Pi Pico up to feature-rich systems on Cortex-A7, A0, or even A55 dual cores
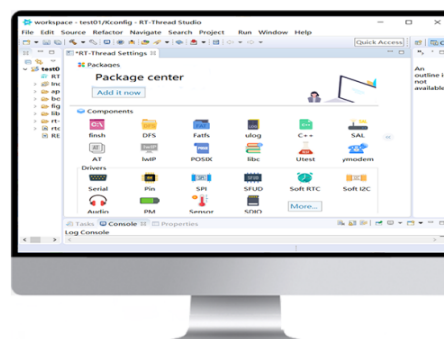
RT-Thread Studio Showcase, Source: RT-Thread.io

You can begin developing with RT-Thread OS on the Raspberry Pi Pico right now.

Follow the official tutorial by RT-Thread for a step-by-step guide on getting started.
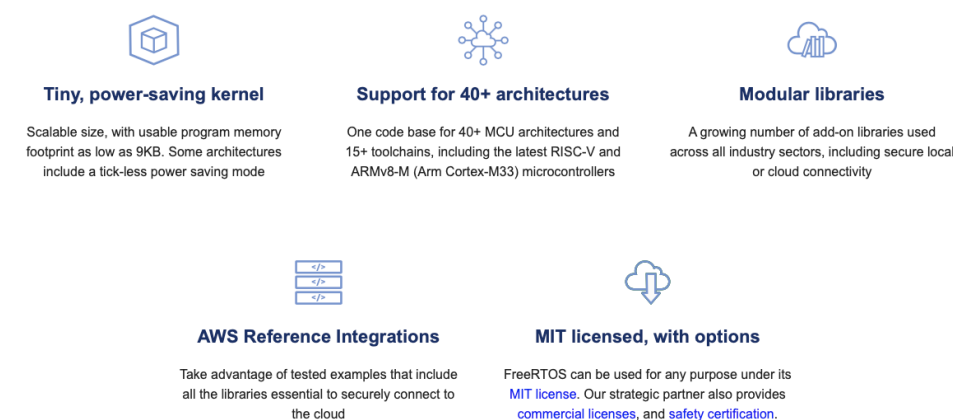
FreeRTOS on the Raspberry Pi Pico

What is FreeRTOS?

FreeRTOS is another well known operating system in the IoT RTOS

scene, similarly extensively developed for over a decade.

FreeRTOS is offered as a kernel with modular IoT libraries, and is specially developed for microcontrollers only. Hence, it also features a low memory footprint and power optimisation features for microcontrollers.

FreeRTOS also supports over 40 MCU architectures, including the latest RISC-V and ARMv8-M microcontrollers, along with a number of modular libraries for local or cloud connectivity support. FreeRTOS releases come in LTS (long term support) versions, so the longevity of your projects on FreeRTOS will be guaranteed.

Tiny, power-saving kernel

Scalable size, with usable program memory footprint as low as 9KB. Some architectures include a tick-less power saving mode

Support for 40+ architectures

One code base for 40+ MCU architectures and 15+ toolchains, including the latest RISC-V and ARMv8-M (Arm Cortex-M33) microcontrollers

Modular libraries

A growing number of add-on libraries used across all industry sectors, including secure local or cloud connectivity

AWS Reference Integrations

Take advantage of tested examples that include all the libraries essential to securely connect to the cloud

MIT licensed, with options

FreeRTOS can be used for any purpose under its MIT license. Our strategic partner also provides commercial licenses, and safety certification.

Source: FreeRTOS.org

FreeRTOS Development for the Pico

While the news has been quieter on official development for FreeRTOS, PicoCPP has recently shared their example Blinky project with FreeRTOS on GitHub. It uses a basic GPIO C++ class, and can be built by following the instructions for C projects on the Raspberry Pi Pico with CMake.

While the support for FreeRTOS on the Raspberry Pi is still a step behind RT-Thread OS, we can be hopeful for developments in the future!

Summary

We're seeing many more platforms being developed for the Raspberry Pi Pico and its RP2040 chip. As more communities come together on this single platform, the future for the Pico and

the microcontroller world is looking to be very promising and exciting!

For our previous coverage on the Raspberry Pi Pico, please visit:

- [Raspberry Pi Pico, the first microcontroller released by Raspberry Pi Foundation, based on the new RP2040 MCU](#)
- [Programmable I/O with Raspberry Pi Pico](#)
- [Arduino IDE Support Announced for the Raspberry Pi Pico!](#)

Purchase a Raspberry Pi Pico!

The Raspberry Pi Pico is now available for preorder on the Seeed Online Store. [Purchase yours now!](#)

Please follow and like us: