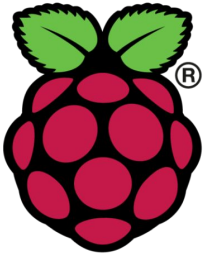


ARM Cortex M0+

José Rodrigo Solis Quiñónez 21002223

Ricardo José Farfán Miranda 21003210

Mariano José Tahay Alvarado 21001700



Raspberry Pi

Índice

- Raspberry Pi Pico
- Instalación
 - Recursos
 - Windows
 - Recomendaciones
- Debugger
 - Recursos
 - Circuito
 - Crear un nuevo proyecto
- Código
 - Conceptos clave
 - Diagramas de flujo
- Demostración
 - Retroalimentación



Raspberry Pi Pico

Raspberry Pi Pico



- RP2040
- Dual-core Arm Cortex M0+
- 133MHz
- 26 x multi-functional GPIO pins
- 2 x UART
- Temperature sensor

Raspberry Pi Documentation

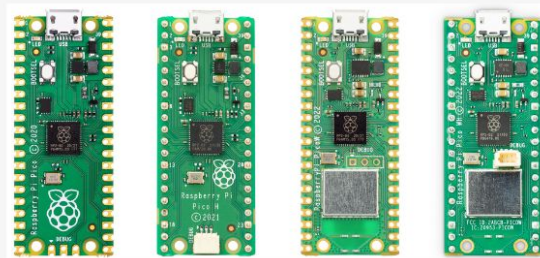
  [Computers](#)[Accessories](#)[Microcontrollers](#)[Services](#)[Pico C SDK](#)

Microcontrollers

[RP2040](#)[Raspberry Pi Pico and Pico W](#)[The family](#)[Raspberry Pi Pico and Pico H](#)[Pinout and design files](#)[Raspberry Pi Pico W and Pico WH](#)[Pinout and design files](#)[Documentation](#)[RP2040 Device](#)[Raspberry Pi Pico](#)[Raspberry Pi Pico W](#)[Software Development](#)[Software Utilities](#)[What is on your Pico?](#)[Debugging using another Raspberry Pi Pico](#)[Resetting Flash memory](#)[Raspberry Pi Debug Probe](#)

Raspberry Pi Pico and Pico W

The family

[Edit this on GitHub](#)

The Raspberry Pi Pico family currently consists of four boards; Raspberry Pi Pico (far left), Pico H (middle left), Pico W (middle right), and Pico WH (far right).

Instalación

Recurros

The image is a screenshot of a PDF viewer application. The left sidebar displays a table of contents with the following structure:

- command line
- ▼ Chapter 9. Building on other platforms
 - ▼ 9.1. Building on Apple macOS
 - 9.1.1. Installing the Toolchain
 - 9.1.2. Using Visual Studio Code
 - 9.1.3. Building with CMake Tools
 - 9.1.4. Saying "Hello World"
 - ▼ 9.2. Building on MS Windows (highlighted)
 - 9.2.1. Installing the Toolchain
 - 9.2.2. Alternative manual installation
- ▼ Chapter 10. Using other Integrated Development Environments
 - ▼ 10.1. Using Eclipse
 - 10.1.1. Setting up Eclipse for Pico on a Linux machine
 - ▼ 10.2. Using CLion
 - 10.2.1. Setting up CLion
 - ▼ 10.3. Other Environments
 - 10.3.1. Using openocd-svd
- ▼ Appendix A: Using debugprobe
 - Install OpenOCD

The main content area shows two pages from the document. The top page is titled "9.2. Building on MS Windows" and contains the text: "Installing the toolchain on Microsoft Windows 10 or Windows 11 is somewhat different to other platforms. However once installed, building code for the RP2040 is somewhat similar." The bottom page is titled "Getting started with Raspberry Pi Pico" and features a warning box that reads: "WARNING Using Raspberry Pi Pico with Windows 7 or 8 is not officially supported but can be [made to work](#)."

9.2. Building on MS Windows

Installing the toolchain on Microsoft Windows 10 or Windows 11 is somewhat different to other platforms. However once installed, building code for the RP2040 is somewhat similar.

9.2. Building on MS Windows 46

Getting started with Raspberry Pi Pico

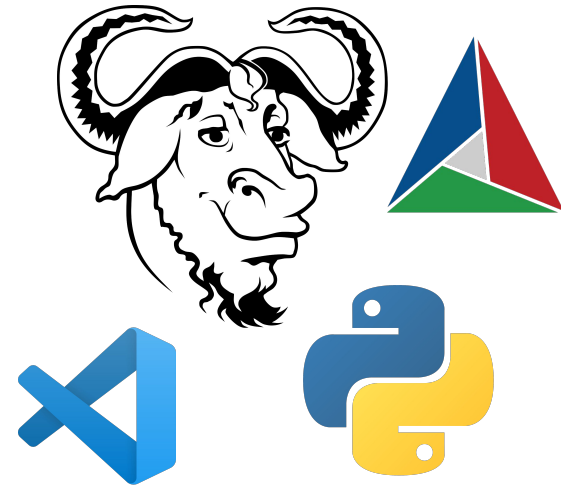
WARNING

Using Raspberry Pi Pico with Windows 7 or 8 is not officially supported but can be [made to work](#).

9.2.1. Installing the Toolchain

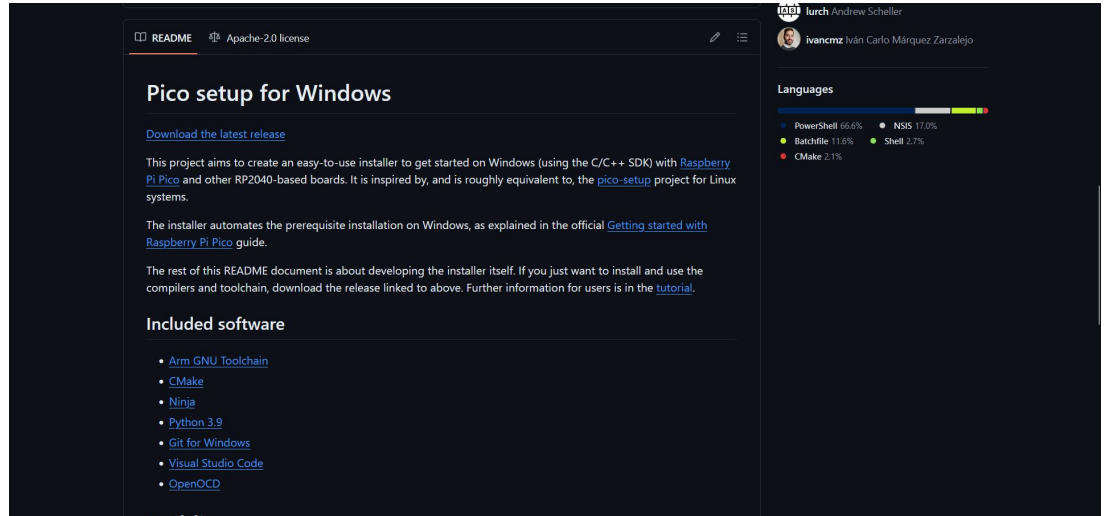
Windows

- Arm GNU Toolchain
- CMake
- Python 3.9
- Visual Studio Code
- OpenOCD



Recomendaciones

1. Usar chocolately
2. Instalar MinGW
3. Descargar el instalador del SDK
4. Siempre revisar que los PATH estén en orden



Debugger

Rekursos

69 (70 of 88)

Automatic Zoom

Chapter 10. Using other Integrated Development Environments

10.1. Using Eclipse

10.1.1. Setting up Eclipse for Pico on a Linux machine

10.2. Using CLion

10.2.1. Setting up CLion

10.3. Other Environments

10.3.1. Using openocd-svd

Appendix A: Using debugprobe

Install OpenOCD

Debug Probe

Debug Probe wiring

Debug with a second Pico

Install debugprobe

debugprobe wiring

Debug Probe interfaces

Use the UART

Linux

Windows

macOS

Debug with OpenOCD

Debug with SWD

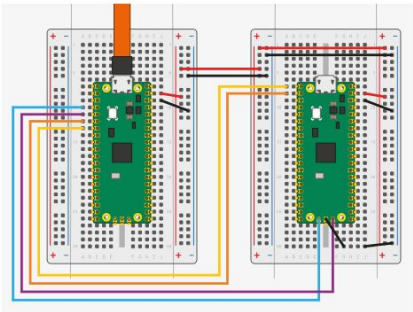
Appendix B: Using Picotool

Getting picotool

Debug with a second Pico

One Pico can reprogram and debug another using the `debugprobe` firmware, which transforms a Pico into a USB → SWD and UART bridge.

Figure 43. Wiring between Pico A (left) and Pico B (right) with Pico A acting as a debugger and Pico B as a system under test. You must connect at least the ground and the two SWD wires. Connect the UART serial port to provide access to the UART serial output of Pico B. You can also bridge the power supply to power both boards with one USB cable. For more information, see [debugprobe wiring](#).



Install `debugprobe`

You can download a UF2 binary of `debugprobe` from [the Pico documentation](#).

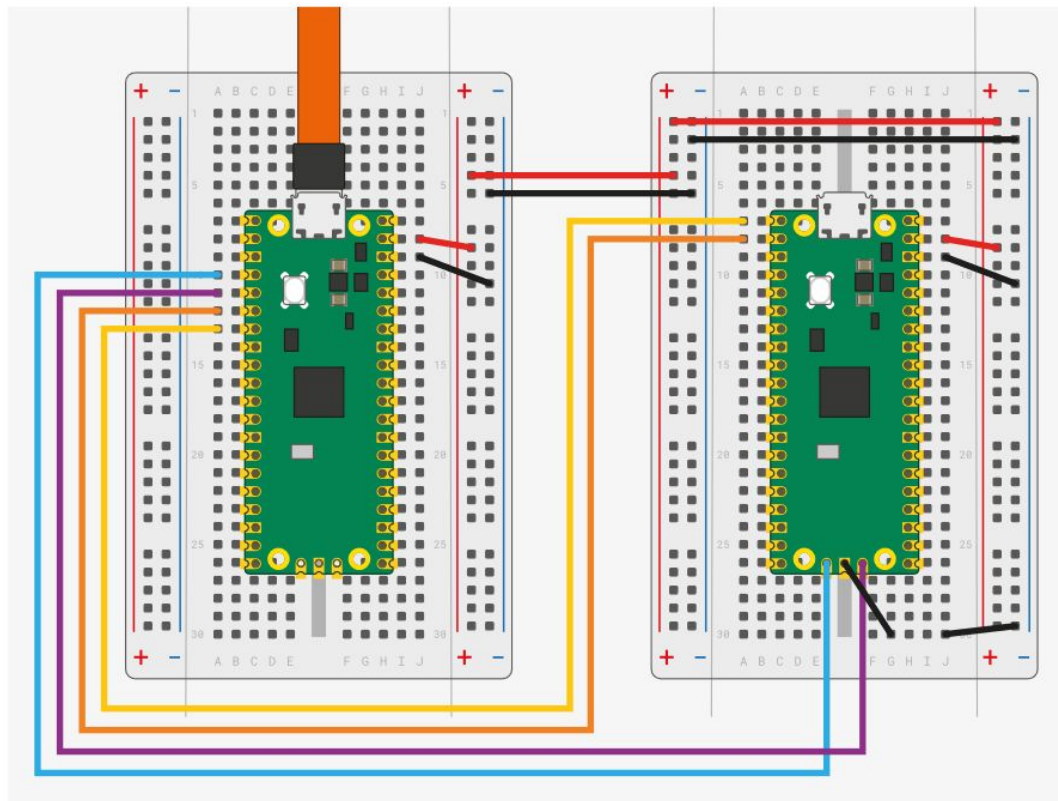
Boot the debugger Pico with the `B00TSEL` button pressed. Copy `debugprobe_on_pico.uf2` onto the device to begin debugging.

NOTE

Use `debugprobe_on_pico.uf2` to use a Pico for debugging. Use `debugprobe.uf2` for the Debug Probe accessory hardware.

Build `debugprobe`

Circuito



Crear un nuevo proyecto

The screenshot shows the GitHub interface for the repository 'raspberrypi / pico-setup-windows'. The left sidebar displays the file tree with folders like '.github', 'config', 'docs', 'packages', 'resources', and 'tests', and files like '.editorconfig', '.gitignore', 'LICENSE', 'README.md', 'build.ps1', 'common.ps1', 'update.ps1', and 'version.txt'. The 'docs' folder is expanded, and 'tutorial.md' is selected. The main content area shows the 'tutorial.md' file, which is a document titled 'Pico Setup for Windows'. It contains sections for 'Installing the tools', 'Starting Visual Studio Code', and 'Opening the examples'. The 'Installing the tools' section mentions downloading the latest release. The 'Starting Visual Studio Code' section describes how to find the Pico - Visual Studio Code shortcut in the Raspberry Pi Pico SDK folder. The 'Opening the examples' section explains how to launch Visual Studio Code using the Start Menu shortcut and how to re-open the examples repository later.

raspberrypi / pico-setup-windows

Type to search

<> Code Issues 7 Pull requests Projects Wiki Security Insights

Files

master

Go to file

> .github

> config

▼ docs

tutorial.md

> packages

> resources

> tests

.editorconfig

.gitignore

LICENSE

README.md

build.ps1

common.ps1

update.ps1

version.txt

pico-setup-windows / docs / tutorial.md

ndabas Fix vscode config copy command 9dbdfa3 · last year History

Preview Code Blame 150 lines (95 loc) · 6.79 KB

Raw Copy Download Edit

Pico Setup for Windows

Installing the tools

Download [the latest release](#) and run it.

Starting Visual Studio Code

In your Start Menu, look for the *Pico - Visual Studio Code* shortcut, in the *Raspberry Pi Pico SDK <version>* folder. The shortcut sets up the needed environment variables and then launches Visual Studio Code.

Opening the examples

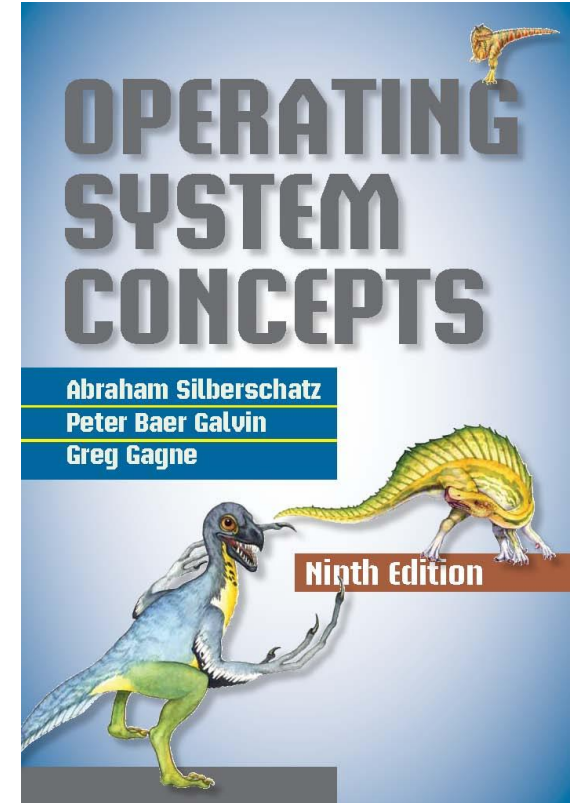
The first time you launch Visual Studio Code using the Start Menu shortcut, it will open the [pico-examples](#) repository.

To re-open the examples repository later, you can open the copy installed at `C:\Users\<user>\Documents\Pico-<version>\pico-examples`.

Código

Conceptos clave

- Concepto de un thread
- Calendarizar un thread
- Algoritmo de “Round Robin”
- Context Switch
- Timers



Diagramas de flujo

Nombre del test

main()

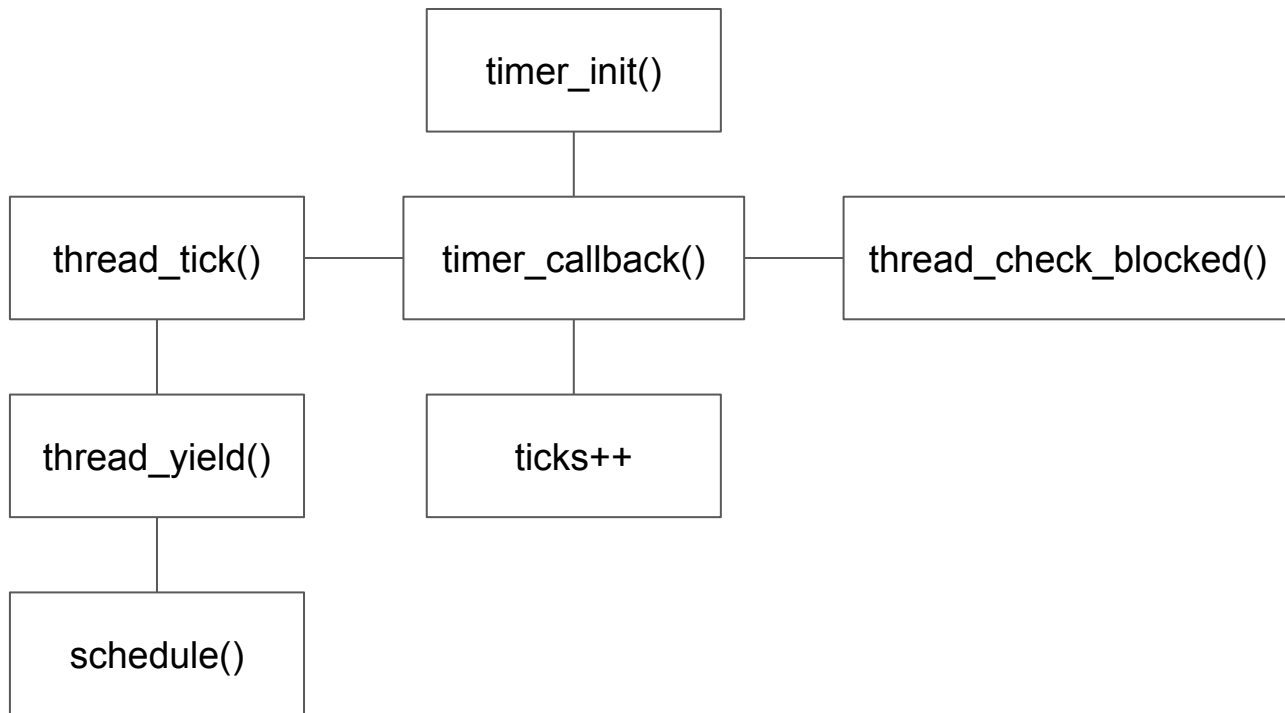
stdio_init_all()

thread_init()

timer_init()

run_task()

main.c



timer.c

Muchas Gracias

<https://github.com/MarianoTahay/raspberry>