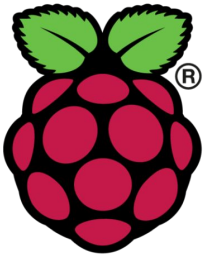# ARM Cortex M0+

José Rodrigo Solis Quiñónez 21002223

Ricardo José Farfán Miranda 21002223

Mariano José Tahay Alvarado 21002223

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
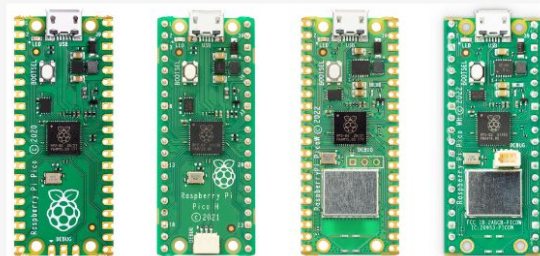- 2 x UART
- Temperature sensor

Raspberry Pi Documentation

🔍 Search CTRL K

| Computers | Accessories | **Microcontrollers** | Services | Pico C SDK |

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

Recursos

Automatic Zoom

command line

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.
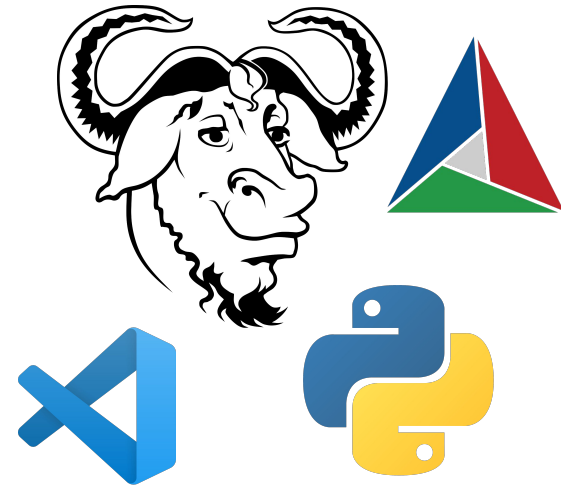
⊖ **WARNING**

Using Raspberry Pi Pico with Windows 7 or 8 is not officially supported but can be made to work.
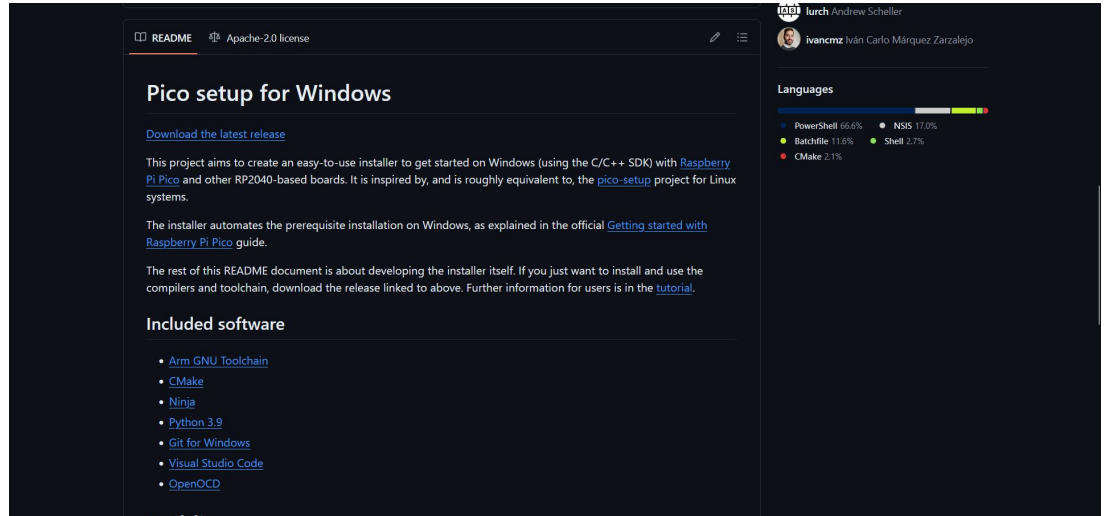
9.2.1. Installing the Toolchain

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

Recursos

Automatic Zoom
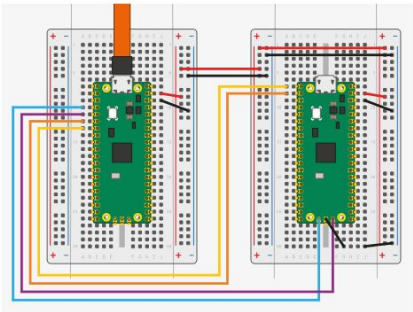
Debug with a second Pico

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.
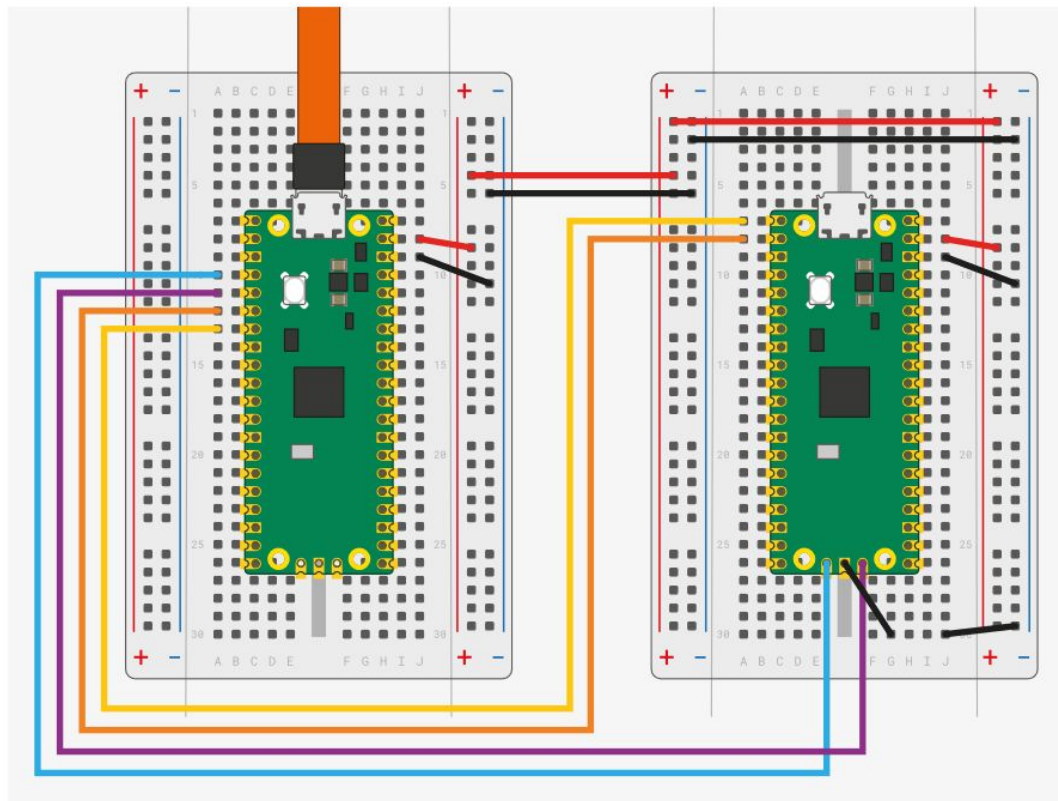
Install debugprobe

You can download a UF2 binary of debugprobe from the Pico documentation.

Boot the debugger Pico with the BOOTSEL 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

Código

Conceptos clave

- Concepto de un thread
- Calendarizar un thread
- Algoritmo de "Round Robin"
- Context Switch
- Timers
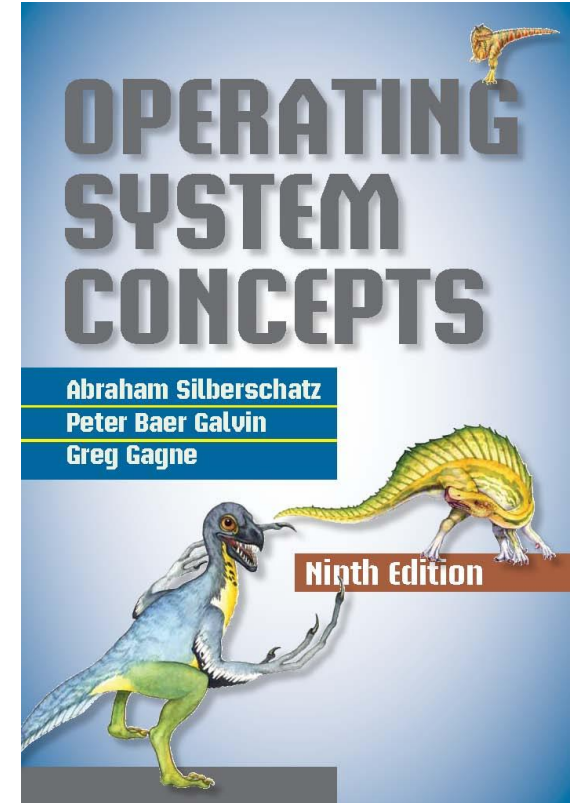
Diagramas de flujo

```
┌─────────────────────────┐
│      Nombre del test     │
└─────────────────────────┘
             │
        ┌─────────┐
        │  main() │
        └─────────┘
             │
      ┌───────────────┐
      │ stdio_init_all()│
      └───────────────┘
             │
      ┌───────────────┐
      │  thread_init() │
      └───────────────┘
             │
      ┌───────────────┐
      │  timer_init()  │
      └───────────────┘
             │
      ┌───────────────┐
      │   run_task()   │
      └───────────────┘
```

main.c

```
                          ┌─────────────────┐
                          │   timer_init()  │
                          └────────┬────────┘
                                   │
┌─────────────────┐      ┌─────────┴────────┐      ┌──────────────────────────┐
│  thread_tick()  ├──────┤ timer_callback() ├──────┤  thread_check_blocked()  │
└────────┬────────┘      └─────────┬────────┘      └──────────────────────────┘
         │                         │
┌────────┴────────┐      ┌─────────┴────────┐
│  thread_yield() │      │      ticks++     │
└────────┬────────┘      └──────────────────┘
         │
┌────────┴────────┐
│    schedule()   │
└─────────────────┘
```

timer.c

Muchas Gracias