Raspberry Pi Pico RP2040 PIO Emulator

A Useful Development & Debugging Aid

Jürgen Reuter

soundpaint.org

GPN Karlsruhe, 2022/5/20

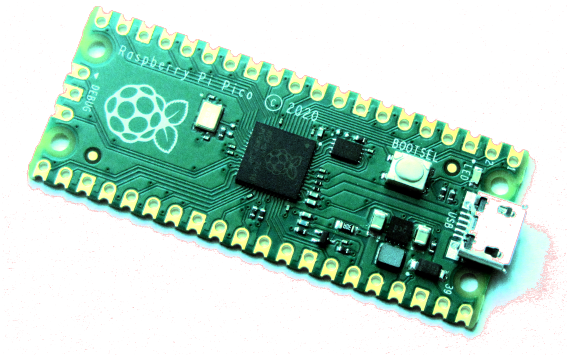
Overview

1. Pico, RP2040, PIO
2. Developing PIO Programs & RP2040 Emulator
3. Demo
4. Q & A

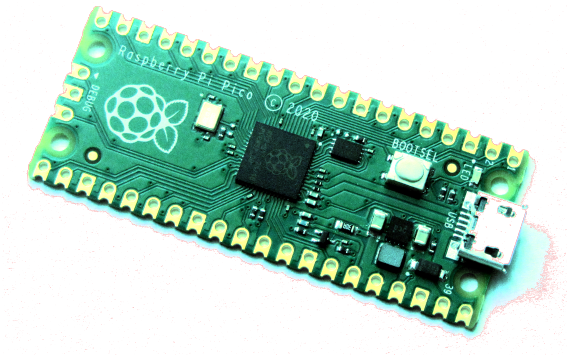
Overview

1. Pico, RP2040, PIO
2. Developing PIO Programs & RP2040 Emulator
3. Demo
4. Q & A

Pico

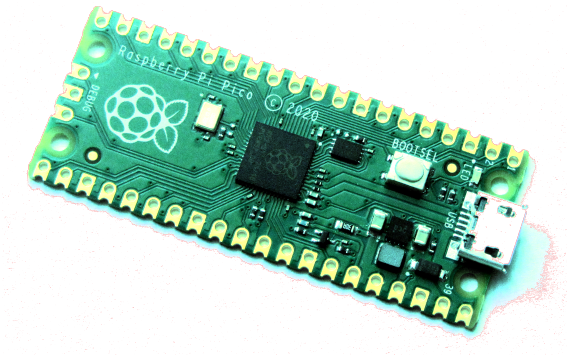


Pico



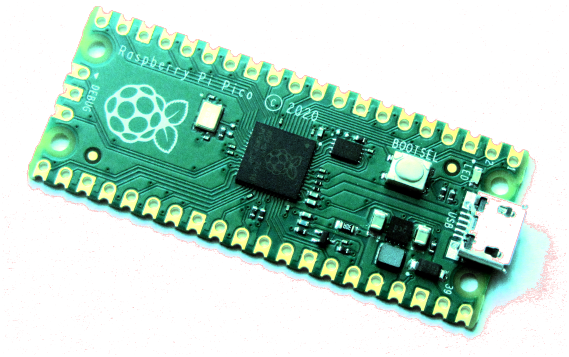
- microcontroller board

Pico



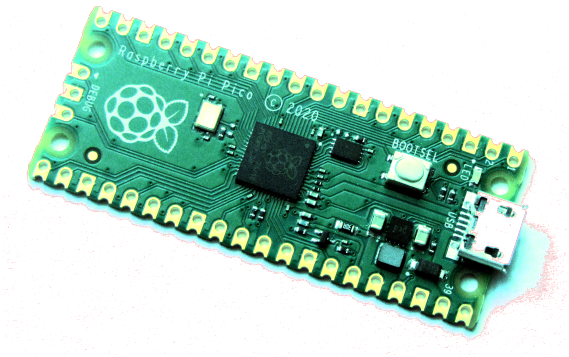
- microcontroller board
- far superior to prior Arduino boards (performance, memory)

Pico



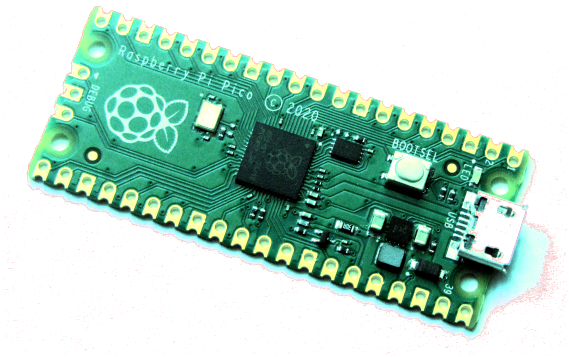
- microcontroller board
- far superior to prior Arduino boards (performance, memory)
- only moderately more power consumption

Pico



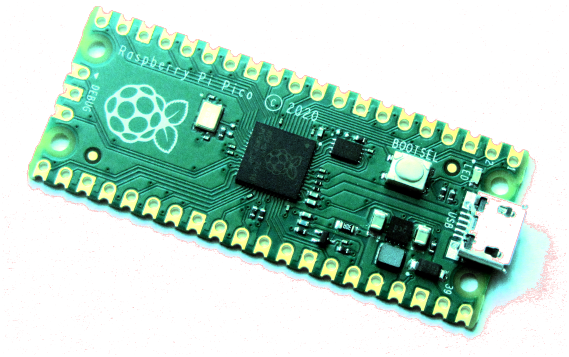
- microcontroller board
- far superior to prior Arduino boards (performance, memory)
- only moderately more power consumption
- expect success similar to previous RPi boards

Pico



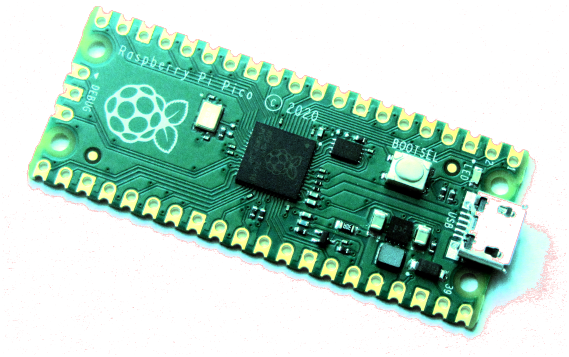
- microcontroller board
- far superior to prior Arduino boards (performance, memory)
- only moderately more power consumption
- expect success similar to previous RPi boards
- excellent software support

Pico



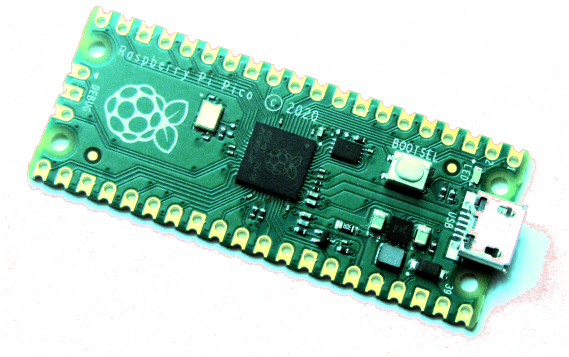
- microcontroller board
- far superior to prior Arduino boards (performance, memory)
- only moderately more power consumption
- expect success similar to previous RPi boards
- excellent software support
- based on RP2040 microcontroller chip

Pico



- microcontroller board
- far superior to prior Arduino boards (performance, memory)
- only moderately more power consumption
- expect success similar to previous RPi boards
- excellent software support
- based on RP2040 microcontroller chip
- price: board ≈ 4US$, RP2040 chip ≈ 1US$

Pico



- microcontroller board
- far superior to prior Arduino boards (performance, memory)
- only moderately more power consumption
- expect success similar to previous RPi boards
- excellent software support
- based on RP2040 microcontroller chip
- price: board ≈ 4US$, RP2040 chip ≈ 1US$
- further reading: https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

RP2040

RP2040

- microcontroller chip

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores
- 264 KB RAM

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores
- 264 KB RAM
- DMA controller

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores
- 264 KB RAM
- DMA controller
- 30 GPIO pins

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores
- 264 KB RAM
- DMA controller
- 30 GPIO pins
- 2 PIO units for programming GPIOs

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores
- 264 KB RAM
- DMA controller
- 30 GPIO pins
- 2 PIO units for programming GPIOs
- system frequency up to 133 MHz (w/o overclocking)

RP2040

- microcontroller chip
- designed by Raspberry Pi foundation
- 2 ARM Cortex-M0+ cores
- 264 KB RAM
- DMA controller
- 30 GPIO pins
- 2 PIO units for programming GPIOs
- system frequency up to 133 MHz (w/o overclocking)
- further reading: https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf

PIO

PIO

- **P**rogrammable **I**nput / **O**utput

PIO

- **P**rogrammable **I**nput / **O**utput
- 2 PIO units per RP2040

PIO

- **P**rogrammable **I**nput / **O**utput
- 2 PIO units per RP2040
- 4 I/O processors per PIO unit

PIO

- **P**rogrammable **I**nput / **O**utput
- 2 PIO units per RP2040
- 4 I/O processors per PIO unit
- 4 I/O processors share 32 words of program memory

PIO

- **P**rogrammable **I**nput / **O**utput
- 2 PIO units per RP2040
- 4 I/O processors per PIO unit
- 4 I/O processors share 32 words of program memory
- very compact op-codes for short programs

PIO

- **P**rogrammable **I**nput / **O**utput
- 2 PIO units per RP2040
- 4 I/O processors per PIO unit
- 4 I/O processors share 32 words of program memory
- very compact op-codes for short programs
- further reading: Chap. 3 *PIO* of `https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf`

PIO Processor Instruction Set (1)

PIO Processor Instruction Set (1)

- data transfer instructions

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops
- no logic ops

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops
- no logic ops
- that's it (almost)

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops
- no logic ops
- that's it (almost)
- exactly one instruction per cycle for exact timing

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops
- no logic ops
- that's it (almost)
- exactly one instruction per cycle for exact timing
- single-word op-codes only, each 32 bits of size

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops
- no logic ops
- that's it (almost)
- exactly one instruction per cycle for exact timing
- single-word op-codes only, each 32 bits of size
- optional delay cycles following each instruction

PIO Processor Instruction Set (1)

- data transfer instructions
- conditional jump / wait instruction
- no arithmetic ops
- no logic ops
- that's it (almost)
- exactly one instruction per cycle for exact timing
- single-word op-codes only, each 32 bits of size
- optional delay cycles following each instruction
- further reading: https://raspberrypi.github.io/pico-sdk-doxygen/weblinks_page.html

PIO Processor Instruction Set (2)

Mnemonic	Short Description
JMP	Set program counter to specified address if specified condition is true. Otherwise, no operation.
WAIT	Stall until some condition is met.
IN	Shift specified number of bits from specified source into the Input Shift Register.
OUT	Shift specified amount of bits out of the Output Shift Register and write those bits to specified destination.
PUSH	Push the contents of the Input Shift Register into the RX FIFO as a single 32-bit word. Clear Input Shift Register to all-zeroes.
PULL	Load a 32-bit word from the TX FIFO into the Output Shift Register.
MOV	Copy data from the specified source to the specified destination.
IRQ	Set or clear the IRQ flag selected by the specified index argument.
SET	Write specified immediate value to specified destination.

Example Program

```
.program squarewave
        set pindirs, 1      ; Set pin to output
again:
        set pins, 1 [1]     ; Drive pin high, then delay for one cycle
        set pins, 0         ; Drive pin low
        jmp again           ; Set PC to label 'again'
```

(Source: RP2040 datasheet)

Typical Applications of PIO Programs

Typical Applications of PIO Programs

- simple signal generator

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface
- I2C interface

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface
- I2C interface
- VGA driver (somewhat challenging)

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface
- I2C interface
- VGA driver (somewhat challenging)
- maybe driver for LCD panels?

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface
- I2C interface
- VGA driver (somewhat challenging)
- maybe driver for LCD panels?
- custom Pico interconnect (e.g. token ring?)

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface
- I2C interface
- VGA driver (somewhat challenging)
- maybe driver for LCD panels?
- custom Pico interconnect (e.g. token ring?)
- ...

Typical Applications of PIO Programs

- simple signal generator
- simple toys like LED blinking
- SPI interface
- USB interface
- UART interface
- I2C interface
- VGA driver (somewhat challenging)
- maybe driver for LCD panels?
- custom Pico interconnect (e.g. token ring?)
- ...
- further examples: https://github.com/raspberrypi/pico-examples

Overview

1. Pico, RP2040, PIO
2. Developing PIO Programs & RP2040 Emulator
3. Demo
4. Q & A

Typical Development Cycle with RP2040 Hardware

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook
- 2: upload program

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook
- 2: upload program
- 3: test & debug program

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook
- 2: upload program
- 3: test & debug program
- 4: if ok, go to 7

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook
- 2: upload program
- 3: test & debug program
- 4: if ok, go to 7
- 5: otherwise, fix program

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook
- 2: upload program
- 3: test & debug program
- 4: if ok, go to 7
- 5: otherwise, fix program
- 6: go to 2

Typical Development Cycle with RP2040 Hardware

- 1: write program e.g. on PC / Notebook
- 2: upload program
- 3: test & debug program
- 4: if ok, go to 7
- 5: otherwise, fix program
- 6: go to 2
- 7: done; can eat gulasch now

Testing: Impossible to Debug While Running on RP2040

Testing: Impossible to Debug While Running on RP2040

- no single-step support

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter
- no direct access to instruction delay status

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter
- no direct access to instruction delay status
- no direct access to FIFOs

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter
- no direct access to instruction delay status
- no direct access to FIFOs
- no direct access to PIO's GPIO state

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter
- no direct access to instruction delay status
- no direct access to FIFOs
- no direct access to PIO's GPIO state
- no direct access to various status bits

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter
- no direct access to instruction delay status
- no direct access to FIFOs
- no direct access to PIO's GPIO state
- no direct access to various status bits
- need additional hardware for recording GPIOs

Testing: Impossible to Debug While Running on RP2040

- no single-step support
- no logging support
- no direct access to program counter
- no direct access to instruction delay status
- no direct access to FIFOs
- no direct access to PIO's GPIO state
- no direct access to various status bits
- need additional hardware for recording GPIOs
- ...

So, what to do?

So, what to do?

- implement an emulator

So, what to do?

- implement an emulator
- focus not on realtime, but on debugging

So, what to do?

- implement an emulator
- focus not on realtime, but on debugging
- access to full internal state in emulator

So, what to do?

- implement an emulator
- focus not on realtime, but on debugging
- access to full internal state in emulator
- found & reported various bugs & flaws in official RP2040 reference manual

So, what to do?

- implement an emulator
- focus not on realtime, but on debugging
- access to full internal state in emulator
- found & reported various bugs & flaws in official RP2040 reference manual
- fun fact: my ordered Pico arrived *after* implementation of emulator

So, what to do?

- implement an emulator
- focus not on realtime, but on debugging
- access to full internal state in emulator
- found & reported various bugs & flaws in official RP2040 reference manual
- fun fact: my ordered Pico arrived *after* implementation of emulator
- source code (license: GPL v2): `https://github.com/soundpaint/rp2040pio`

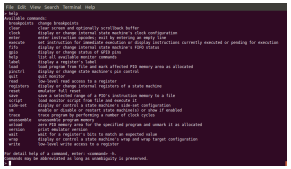
So, what to do?

- implement an emulator
- focus not on realtime, but on debugging
- access to full internal state in emulator
- found & reported various bugs & flaws in official RP2040 reference manual
- fun fact: my ordered Pico arrived *after* implementation of emulator
- source code (license: GPL v2): `https://github.com/soundpaint/rp2040pio`
- full documentation: `https://rp2040pio-docs.readthedocs.io/en/latest/`

Tool Set

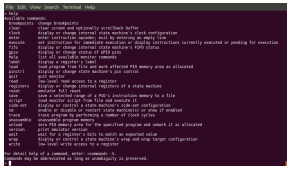
Tool Set

- monitor & control program

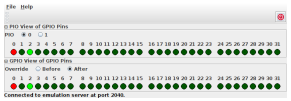


Tool Set

- monitor & control program

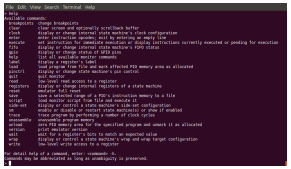


- GPIO observer

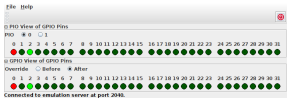


Tool Set

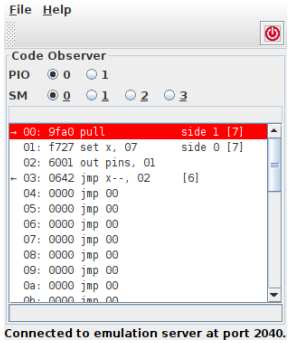
- monitor & control program



- GPIO observer

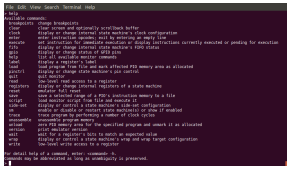


- code observer

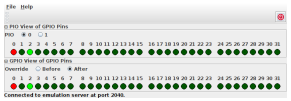


Tool Set

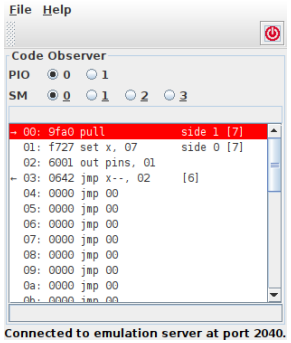
- monitor & control program



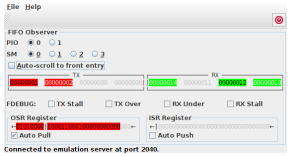
- GPIO observer



- code observer

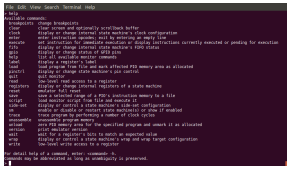


- FIFO observer

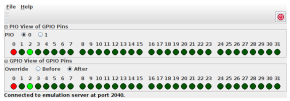


Tool Set

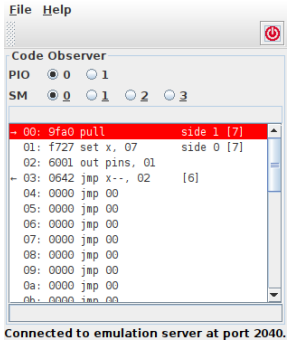
- monitor & control program



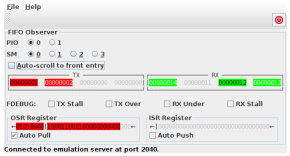
- GPIO observer



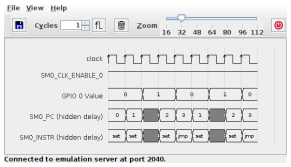
- code observer



- FIFO observer

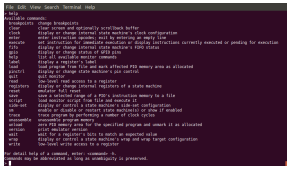


- diagram creator

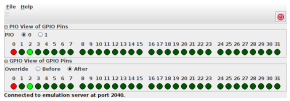


Tool Set

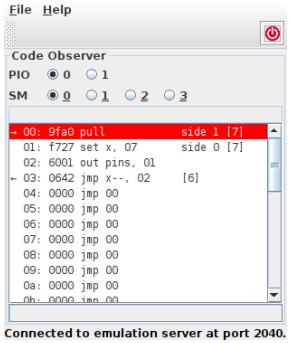
- monitor & control program



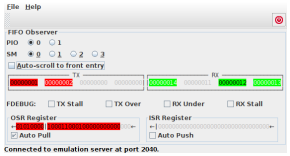
- GPIO observer



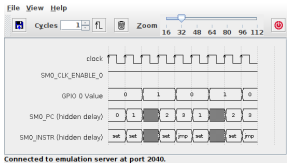
- code observer



- FIFO observer



- diagram creator



- . . . (maybe more to come)

Overview

1. Pico, RP2040, PIO
2. Developing PIO Programs & RP2040 Emulator
3. Demo
4. Q & A

Live Demo

Overview

1. Pico, RP2040, PIO
2. Developing PIO Programs & RP2040 Emulator
3. Demo
4. Q & A

Questions

Questions?

References

Raspberry Pi Foundation.
2021a.
raspberrypi / pico-examples.
https://github.com/raspberrypi/pico-examples, February.

Raspberry Pi Foundation.
2021b.
raspberrypi / pico-sdk.
https://github.com/raspberrypi/pico-sdk, February.

Raspberry Pi (Trading) Ltd.
2021a.
Raspberry Pi Pico C/C++ SDK.
November.

Raspberry Pi (Trading) Ltd.
2021b.
Raspberry Pi Pico Datasheet.
Raspberry Pi (Trading) Ltd., November.

Raspberry Pi (Trading) Ltd.
2021c.
Raspberry Pi Pico SDK: Documentation and datasheets.
https://raspberrypi.github.io/pico-sdk-doxygen/weblinks_page.html.

Raspberry Pi (Trading) Ltd.
2021d.
RP2040 Datasheet.
November.

Jürgen Reuter.
2021a.
Rp2040 pio emulator documentation.
https://rp2040pio-docs.readthedocs.io/en/latest/, May.

Jürgen Reuter.
2021b.
soundpaint / rp2040pio.
https://github.com/soundpaint/rp2040pio, February.