Raspberry Pi Pico-series C/C++ SDK

Libraries and tools for C/C++ development on Raspberry Pi microcontrollers

Raspberry Pi Ltd

Colophon

Copyright © 2020-2024 Raspberry Pi Ltd (formerly Raspberry Pi (Trading) Ltd.)

The documentation of the RP2350 microcontroller is licensed under a Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND).

build-date: 2024-08-09
build-version: c274ff8-clean

About the SDK

Throughout the text "the SDK" refers to our Raspberry Pi Pico SDK. More details about the SDK can be found throughout this book. Source code included in the documentation is Copyright © 2023-2024 Raspberry Pi Ltd (formerly Raspberry Pi (Trading) Ltd.) and licensed under the 3-Clause BSD license.

Legal disclaimer notice

TECHNICAL AND RELIABILITY DATA FOR RASPBERRY PI PRODUCTS (INCLUDING DATASHEETS) AS MODIFIED FROM TIME TO TIME (“RESOURCES”) ARE PROVIDED BY RASPBERRY PI LTD (“RPL”) "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN NO EVENT SHALL RPL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE RESOURCES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

RPL reserves the right to make any enhancements, improvements, corrections or any other modifications to the RESOURCES or any products described in them at any time and without further notice.

The RESOURCES are intended for skilled users with suitable levels of design knowledge. Users are solely responsible for their selection and use of the RESOURCES and any application of the products described in them. User agrees to indemnify and hold RPL harmless against all liabilities, costs, damages or other losses arising out of their use of the RESOURCES.

RPL grants users permission to use the RESOURCES solely in conjunction with the Raspberry Pi products. All other use of the RESOURCES is prohibited. No licence is granted to any other RPL or other third party intellectual property right.

HIGH RISK ACTIVITIES. Raspberry Pi products are not designed, manufactured or intended for use in hazardous environments requiring fail safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, weapons systems or safety-critical applications (including life support systems and other medical devices), in which the failure of the products could lead directly to death, personal injury or severe physical or environmental damage (“High Risk Activities”). RPL specifically disclaims any express or implied warranty of fitness for High Risk Activities and accepts no liability for use or inclusions of Raspberry Pi products in High Risk Activities.

Raspberry Pi products are provided subject to RPL’s Standard Terms. RPL’s provision of the RESOURCES does not expand or otherwise modify RPL’s Standard Terms including but not limited to the disclaimers and warranties expressed in them.

Table of contents

Colophon . . . 1
    Legal disclaimer notice . . . 1
1. About the SDK . . . 8
    1.1. Introduction . . . 8
    1.2. Anatomy of a SDK Application . . . 8
2. SDK architecture . . . 11
    2.1. The Build System . . . 11
    2.2. Every Library is an INTERFACE Library . . . 12
    2.3. SDK Library Structure . . . 13
        2.3.1. Higher-level Libraries . . . 13
        2.3.2. Runtime Support Libraries . . . 13
        2.3.3. Hardware Support Libraries . . . 14
        2.3.4. Hardware Structs Library . . . 15
        2.3.5. Hardware Registers Library . . . 16
        2.3.6. TinyUSB Port . . . 17
        2.3.7. FreeRTOS Ports . . . 17
        2.3.8. Wi-Fi on Pico W . . . 18
        2.3.9. Bluetooth on Pico W . . . 18
    2.4. Directory Structure . . . 19
        2.4.1. Locations of Files . . . 19
    2.5. Conventions for Library Functions . . . 21
        2.5.1. Function Naming Conventions . . . 21
        2.5.2. Return Codes and Error Handling . . . 22
        2.5.3. Use of Inline Functions . . . 23
        2.5.4. Builder Pattern for Hardware Configuration APIs . . . 23
    2.6. Customisation and Configuration Using Preprocessor variables . . . 24
        2.6.1. Preprocessor Variables via Board Configuration File . . . 24
        2.6.2. Preprocessor Variables Per Binary or Library via CMake . . . 25
    2.7. SDK Runtime . . . 25
        2.7.1. Standard Input/Output (stdio) Support . . . 25
        2.7.2. Printf Support . . . 26
        2.7.3. Runtime Initialization and Linking . . . 26
        2.7.4. C-Library Integration . . . 26
        2.7.5. Floating-point Support . . . 26
        2.7.6. Hardware Divider . . . 27
    2.8. Multi-core support . . . 28
    2.9. Using C++ . . . 29
    2.10. Supporting both RP2040 and RP2350 . . . 29
    2.11. Next Steps . . . 30
3. Using programmable I/O (PIO) . . . 31
    3.1. What is Programmable I/O (PIO)? . . . 31
        3.1.1. Background . . . 31
        3.1.2. I/O Using dedicated hardware on your PC . . . 31
        3.1.3. I/O Using dedicated hardware on your Raspberry Pi or microcontroller . . . 31
        3.1.4. I/O Using software control of GPIOs (*"bit-banging"*) . . . 32
        3.1.5. Programmable I/O Hardware using FPGAs and CPLDs . . . 33
        3.1.6. Programmable I/O Hardware using PIO . . . 33
    3.2. Getting started with PIO . . . 34
        3.2.1. A First PIO Application . . . 34
        3.2.2. A Real Example: WS2812 LEDs . . . 38
        3.2.3. PIO and DMA (A Logic Analyser) . . . 46
        3.2.4. Further examples . . . 50
    3.3. Using PIOASM, the PIO Assembler . . . 51
        3.3.1. Usage . . . 51
        3.3.2. Directives . . . 52

3.3.3. Values . . . 54
3.3.4. Expressions . . . 54
3.3.5. Comments . . . 54
3.3.6. Labels . . . 55
3.3.7. Instructions . . . 55
3.3.8. Pseudoinstructions . . . 56
3.3.9. Output pass through . . . 56
3.3.10. Language generators . . . 56
3.4. PIO Instruction Set Reference . . . 62
3.4.1. Encoding (version 0, RP2040) . . . 62
3.4.2. Encoding (version 1, RP2350) . . . 62
3.4.3. Summary . . . 63
3.4.4. JMP . . . 63
3.4.5. WAIT . . . 64
3.4.6. IN . . . 65
3.4.7. OUT . . . 66
3.4.8. PUSH . . . 67
3.4.9. PULL . . . 68
3.4.10. MOV (to RX) . . . 69
3.4.11. MOV (from RX) . . . 70
3.4.12. MOV . . . 71
3.4.13. IRQ . . . 72
3.4.14. SET . . . 73
4. Library documentation . . . 75
4.1. Hardware APIs . . . 76
4.1.1. hardware_adc . . . 77
4.1.2. hardware_base . . . 82
4.1.3. hardware_claim . . . 84
4.1.4. hardware_clocks . . . 86
4.1.5. hardware_divider . . . 100
4.1.6. hardware_dcp RP2350 . . . 109
4.1.7. hardware_dma . . . 110
4.1.8. hardware_exception . . . 131
4.1.9. hardware_flash . . . 135
4.1.10. hardware_gpio . . . 138
4.1.11. hardware_hazard3 RP2350 . . . 169
4.1.12. hardware_i2c . . . 169
4.1.13. hardware_interp . . . 178
4.1.14. hardware_irq . . . 187
4.1.15. hardware_pio . . . 200
4.1.16. hardware_pll . . . 241
4.1.17. hardware_powman RP2350 . . . 242
4.1.18. hardware_pwm . . . 249
4.1.19. hardware_resets . . . 263
4.1.20. hardware_riscv RP2350 . . . 268
4.1.21. hardware_riscv_platform_timer RP2350 . . . 268
4.1.22. hardware_rtc RP2040 . . . 270
4.1.23. hardware_rcp RP2350 . . . 273
4.1.24. hardware_spi . . . 273
4.1.25. hardware_sha256 RP2350 . . . 281
4.1.26. hardware_sync . . . 285
4.1.27. hardware_ticks . . . 293
4.1.28. hardware_timer . . . 295
4.1.29. hardware_uart . . . 310
4.1.30. hardware_vreg . . . 320
4.1.31. hardware_watchdog . . . 320
4.1.32. hardware_xosc . . . 324
4.2. High Level APIs . . . 324
4.2.1. pico_aon_timer . . . 325
4.2.2. pico_async_context . . . 328

4.2.3. pico_bootsel_via_double_reset . . . 339
4.2.4. pico_flash . . . 339
4.2.5. pico_i2c_slave . . . 341
4.2.6. pico_multicore . . . 343
4.2.7. pico_rand . . . 354
4.2.8. pico_sha256 RP2350 . . . 356
4.2.9. pico_stdlib . . . 361
4.2.10. pico_sync . . . 363
4.2.11. pico_time . . . 377
4.2.12. pico_unique_id . . . 398
4.2.13. pico_util . . . 399
4.3. Third-party Libraries . . . 403
4.3.1. tinyusb_device . . . 403
4.3.2. tinyusb_host . . . 403
4.4. Networking Libraries . . . 403
4.4.1. pico_btstack . . . 404
4.4.2. pico_lwip . . . 405
4.4.3. pico_cyw43_driver . . . 408
4.4.4. pico_cyw43_arch . . . 409
4.5. Runtime Infrastructure . . . 447
4.5.1. boot_stage2 . . . 448
4.5.2. pico_atomic . . . 448
4.5.3. pico_base . . . 449
4.5.4. pico_binary_info . . . 450
4.5.5. pico_bootrom . . . 451
4.5.6. pico_bit_ops . . . 456
4.5.7. pico_cxx_options . . . 457
4.5.8. pico_clib_interface . . . 457
4.5.9. pico_crt0 . . . 457
4.5.10. pico_divider . . . 457
4.5.11. pico_double . . . 465
4.5.12. pico_float . . . 465
4.5.13. pico_int64_ops . . . 466
4.5.14. pico_malloc . . . 466
4.5.15. pico_mem_ops . . . 466
4.5.16. pico_platform . . . 467
4.5.17. pico_printf . . . 475
4.5.18. pico_runtime . . . 475
4.5.19. pico_runtime_init . . . 476
4.5.20. pico_stdio . . . 477
4.5.21. pico_standard_binary_info . . . 486
4.5.22. pico_standard_link . . . 487
4.6. External API Headers . . . 487
4.6.1. boot_picobin_headers . . . 487
4.6.2. boot_picoboot_headers . . . 488
4.6.3. boot_uf2_headers . . . 488
4.6.4. pico_usb_reset_interface_headers . . . 488
5. SDK configuration . . . 489
5.1. Full List of SDK Configuration Defines . . . 490
6. CMake build configuration . . . 503
6.1. Full List of SDK Configuration Variables . . . 503
6.2. Platform and Board Configuration . . . 505
6.3. Compiler and Toolchain configuration . . . 506
6.4. Binary Type configuration . . . 507
7. CMake build functions . . . 508
7.1. Control of picotool post-processing (not available on RP2040) . . . 508
8. Board configuration . . . 509
8.1. The Configuration files . . . 509
8.2. Building applications with a custom board configuration . . . 511
8.3. Available configuration parameters . . . 511

9. Embedded Binary Information . . . 512
9.1. Basic information . . . 512
9.2. Pins . . . 512
9.3. Full Information . . . 513
9.4. Including Binary Information . . . 513
9.5. Setting Common Information from CMake . . . 515
Appendix A: App Notes . . . 517
Attaching a 7 segment LED via GPIO . . . 517
Wiring information . . . 517
List of Files . . . 517
Bill of Materials . . . 519
DHT-11, DHT-22, and AM2302 Sensors . . . 520
Wiring information . . . 520
List of Files . . . 521
Bill of Materials . . . 523
Attaching a 16x2 LCD via TTL . . . 523
Wiring information . . . 524
List of Files . . . 524
Bill of Materials . . . 527
Attaching a microphone using the ADC . . . 528
Wiring information . . . 528
List of Files . . . 529
Bill of Materials . . . 530
Attaching a BME280 temperature/humidity/pressure sensor via SPI . . . 531
Wiring information . . . 531
List of Files . . . 531
Bill of Materials . . . 536
Attaching a MPU9250 accelerometer/gyroscope via SPI . . . 536
Wiring information . . . 536
List of Files . . . 537
Bill of Materials . . . 540
Attaching a MPU6050 accelerometer/gyroscope via I2C . . . 540
Wiring information . . . 540
List of Files . . . 541
Bill of Materials . . . 543
Attaching a 16x2 LCD via I2C . . . 543
Wiring information . . . 544
List of Files . . . 544
Bill of Materials . . . 547
Attaching a BMP280 temp/pressure sensor via I2C . . . 548
Wiring information . . . 548
List of Files . . . 548
Bill of Materials . . . 553
Attaching a LIS3DH Nano Accelerometer via i2c. . . . 553
Wiring information . . . 554
List of Files . . . 554
Bill of Materials . . . 557
Attaching a MCP9808 digital temperature sensor via I2C . . . 557
Wiring information . . . 557
List of Files . . . 558
Bill of Materials . . . 560
Attaching a MMA8451 3-axis digital accelerometer via I2C . . . 561
Wiring information . . . 561
List of Files . . . 561
Bill of Materials . . . 564
Attaching an MPL3115A2 altimeter via I2C . . . 564
Wiring information . . . 565
List of Files . . . 565
Bill of Materials . . . 569
Attaching an OLED display via I2C . . . 569

Wiring information . . . 570
List of Files . . . 571
Bill of Materials . . . 581
Attaching a PA1010D Mini GPS module via I2C . . . 581
Wiring information . . . 581
List of Files . . . 582
Bill of Materials . . . 585
Attaching a PCF8523 Real Time Clock via I2C . . . 585
Wiring information . . . 585
List of Files . . . 586
Bill of Materials . . . 589
Interfacing 1-Wire devices to the Pico . . . 589
Wiring information . . . 590
Bill of materials . . . 590
List of files . . . 591
Communicating as master and slave via SPI . . . 598
Wiring information . . . 598
Outputs . . . 598
List of Files . . . 600
Bill of Materials . . . 604
Appendix B: Building the SDK API documentation . . . 605
Appendix C: SDK release history . . . 607
Release 1.0.0 (20/Jan/2021) . . . 607
Release 1.0.1 (01/Feb/2021) . . . 607
Boot Stage 2 . . . 607
Release 1.1.0 (05/Mar/2021) . . . 607
Backwards incompatibility . . . 608
Release 1.1.1 (01/Apr/2021) . . . 608
Release 1.1.2 (07/Apr/2021) . . . 608
Release 1.2.0 (03/Jun/2021) . . . 608
New/improved Board headers . . . 608
Updated TinyUSB to 0.10.1 . . . 608
Added CMSIS core headers . . . 609
API improvements . . . 609
General code improvements . . . 611
SVD . . . 611
pioasm . . . 611
RTOS interoperability . . . 611
CMake build changes . . . 611
Boot Stage 2 . . . 611
Release 1.3.0 (02/Nov/2021) . . . 611
Updated TinyUSB to 0.12.0 . . . 611
New Board Support . . . 612
Updated SVD, `hardware_regs`, `hardware_structs` . . . 612
Behavioural Changes . . . 613
Other Notable Improvements . . . 613
CMake build . . . 615
pioasm . . . 615
elf2uf2 . . . 615
Release 1.3.1 (18/May/2022) . . . 615
New Board Support . . . 615
Notable Library Changes/Improvements . . . 616
Build . . . 617
pioasm . . . 617
elf2uf2 . . . 617
Release 1.4.0 (30/Jun/2022) . . . 617
New Board Support . . . 617
Wireless Support . . . 617
Notable Library Changes/Improvements . . . 618
Build . . . 620

Release 1.5.0 (11/Feb/2023) . . . 620
New Board Support . . . 620
Library Changes/Improvements . . . 620
New Libraries . . . 623
Build . . . 624
Bluetooth Support for Pico W (BETA) . . . 624
Release 1.5.1 (14/Jun/2023) . . . 625
Board Support . . . 625
Library Changes/Improvements . . . 625
New Libraries . . . 626
Miscellaneous . . . 627
Build . . . 627
Bluetooth Support for Pico W . . . 627
Release 2.0.0 (08/Aug/2024) . . . 628
Notices . . . 628
Major New Features . . . 628
Security and Code Signing . . . 629
Board Support . . . 629
New Libraries . . . 630
Library Changes / Improvements . . . 633
pico_bt_stack . . . 639
FreeRTOS integration . . . 643
Backwards Incompatibilities . . . 644
Build . . . 644
Building Documentation . . . 645
Fixed Issues . . . 645
Appendix D: Documentation release history . . . 646