

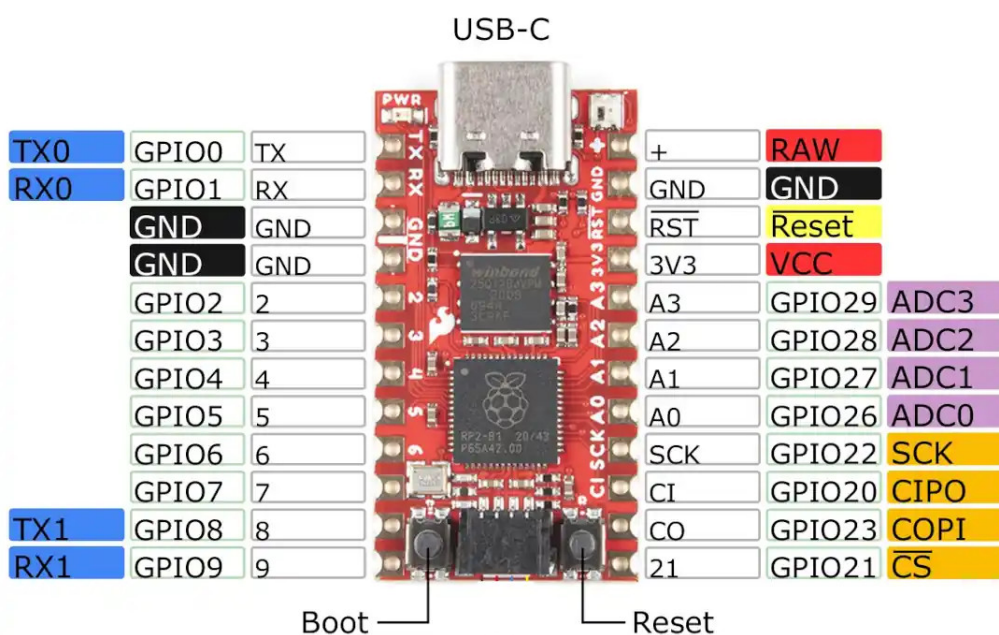
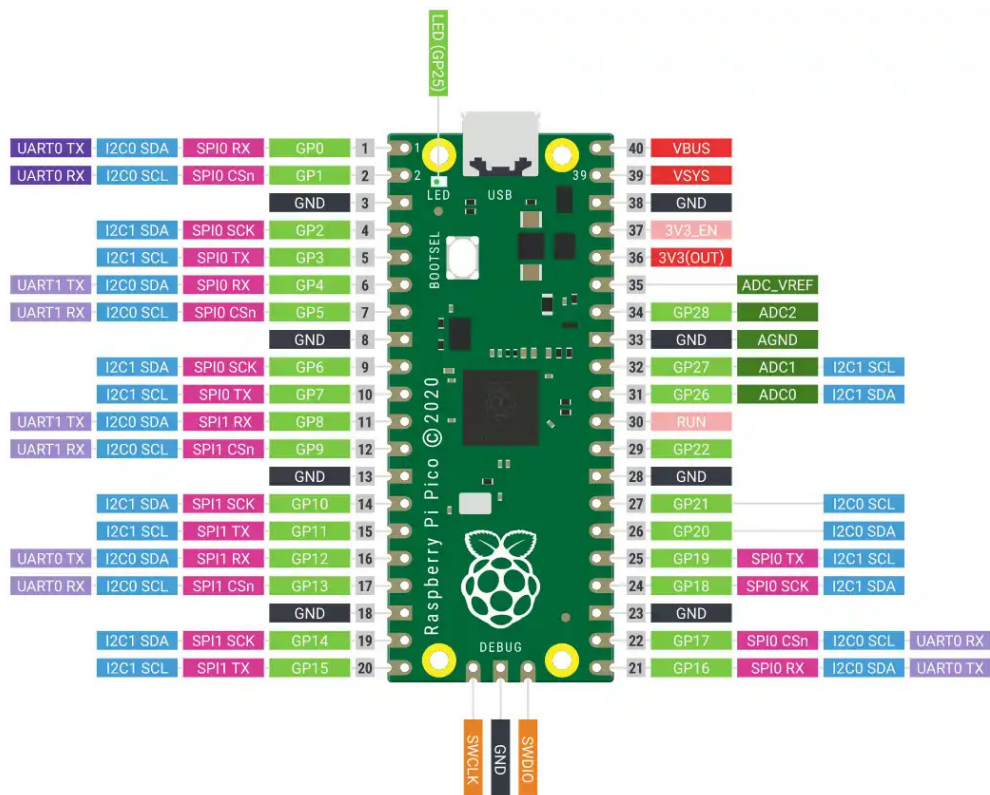
Raspberry Pi RP2040



The following table shows the current driver status for peripherals on RP2040 MCUs:

System	Support
ADC driver	✓
Audio	✓
Backlight	✓
I2C driver	✓
SPI driver	✓
WS2812 driver	✓ using <code>PIO</code> driver
External EEPROMs	✓ using <code>I2C</code> or <code>SPI</code> driver
EEPROM emulation	✓
serial driver	✓ using <code>SIO</code> or <code>PIO</code> driver
UART driver	✓ using <code>SIO</code> driver

GPIO



The GPIO pins of the RP2040 are not 5V tolerant!

Pin nomenclature

To address individual pins on the RP2040, QMK uses the `GPx` abbreviation – where the `x` stands for the GPIO number of the pin. This number can likely be found on the official pinout diagram of your board. Note

that these GPIO numbers match the RP2040 MCU datasheet, and don't necessarily match the number you see printed on the board. For instance the Raspberry Pi Pico uses numbers from 1 to 40 for their pins, but these are not identical to the RP2040's GPIO numbers. So if you want to use the pin 11 of the Pico for your keyboard, you would refer to it as `GP8` in the config files.

Alternate functions

The RP2040 features flexible GPIO function multiplexing, this means that every pin can be connected to nearly all the internal peripherals like I2C, SPI, UART or PWM. This allows for flexible PCB designs that are much less restricted in the selection of GPIO pins. To find out which pin can use which peripheral refer to the official [Raspberry PI RP2040 datasheet](#) section 1.4.3 GPIO functions.

Selecting hardware peripherals and drivers

QMK RP2040 support builds upon ChibiOS and thus follows their convention for activating drivers and associated hardware peripherals. These tables only give a quick overview which values have to be used, please refer to the ChibiOS specific sections on the driver pages.

I2C Driver

RP2040 Peripheral `mcuconf.h` values I2C_DRIVER

I2C0	RP_I2C_USE_I2C0	I2CD0
I2C1	RP_I2C_USE_I2C1	I2CD1

To configure the I2C driver please read the [ChibiOS/ARM](#) section.

SPI Driver

RP2040 Peripheral `mcuconf.h` values SPI_DRIVER

SPI0	RP_SPI_USE_SPI0	SPID0
SPI1	RP_SPI_USE_SPI1	SPID1

To configure the SPI driver please read the [ChibiOS/ARM](#) section.

UART Driver

RP2040 Peripheral `mcuconf.h` values UART_DRIVER

UART0	RP_SIO_USE_UART0	SIOD0
UART1	RP_SIO_USE_UART1	SIOD1

Double-tap reset boot-loader entry

The double-tap reset mechanism is an alternate way in QMK to enter the embedded mass storage UF2 boot-loader of the RP2040. It enables bootloader entry by a fast double-tap of the reset pin on start up, which is similar to the behavior of AVR Pro Micros. This feature activated by default for the Pro Micro

RP2040 board, but has to be configured for other boards. To activate it, add the following options to your `keyboards config.h` file:

```
#define RP2040_BOOTLOADER_DOUBLE_TAP_RESET // Activates the double-tap
behavior
#define RP2040_BOOTLOADER_DOUBLE_TAP_RESET_TIMEOUT 200U // Timeout window in
ms in which the double tap can occur.
#define RP2040_BOOTLOADER_DOUBLE_TAP_RESET_LED GP17 // Specify a optional
status led by GPIO number which blinks when entering the bootloader
```

Pre-defined RP2040 boards

QMK defines two boards that you can choose from to base your RP2040 powered keyboard upon. These boards provide pre-configured default pins and drivers.

Generic Pro Micro RP2040

This is the default board that is chosen, unless any other RP2040 board is selected in your `keyboards rules.mk` file. It assumes a pin layout for the I2C, SPI and Serial drivers which is identical to the Sparkfun Pro Micro RP2040, however all values can be overwritten by defining them in your `keyboards config.h` file. The [double-tap](#) reset to enter boot-loader behavior is activated by default.

Driver configuration define	Value
I2C driver	
I2C_DRIVER	I2CD1
I2C1_SDA_PIN	GP2
I2C1_SCL_PIN	GP3
SPI driver	
SPI_DRIVER	SPID0
SPI_SCK_PIN	GP18
SPI_MISO_PIN	GP20
SPI_MOSI_PIN	GP19
Serial driver	
SERIAL_USART_DRIVER (SIO Driver only)	SIOD0
SOFT_SERIAL_PIN	undefined, use SERIAL_USART_TX_PIN
SERIAL_USART_TX_PIN	GP0
SERIAL_USART_RX_PIN	GP1
UART driver	
UART_DRIVER	SIOD0
UART_TX_PIN	GP0
UART_RX_PIN	GP1

The pin-outs of Adafruit's KB2040 and Boardsource's Blok both deviate from the Sparkfun Pro Micro RP2040. Lookup the pin-out of these boards and adjust your keyboards pin definition accordingly if you want

to use these boards.

Generic RP2040 board

This board can be chosen as a base for RP2040 keyboards which configure all necessary pins and drivers themselves and do not wish to leverage the configuration matching the Generic Pro Micro RP2040 board. Thus it doesn't provide any pre-configured pins or drivers. To select this board add the following line to your keyboards `rules.mk` file.

```
BOARD = GENERIC_RP_RP2040
```

Split keyboard support

Split keyboards are fully supported using the [serial driver](#) in both full-duplex and half-duplex configurations. Two driver subsystems are supported by the RP2040, the hardware UART based `SIO` and the Programmable IO based `PIO` driver.

Feature	SIO Driver	PIO Driver
Half-Duplex operation		✓
Full-Duplex operation	✓	✓
TX and RX pin swapping		✓
Any GPIO as TX and RX pin	Only UART capable pins	✓
Simple configuration		✓

The `PIO` driver is much more flexible then the `SIO` driver, the only “downside” is the usage of `PIO` resources which in turn are not available for advanced user programs. Under normal circumstances, this resource allocation will be a non-issue.

RP2040 second stage bootloader selection

As the RP2040 does not have any internal flash memory it depends on an external SPI flash memory chip to store and execute instructions from. To successfully interact with a wide variety of these chips a second stage bootloader that is compatible with the chosen external flash memory has to be supplied with each firmware image. By default an `W25Q080` compatible bootloader is assumed, but others can be chosen by adding one of the defines listed in the table below to your keyboards `config.h` file.

Compatible with flash chip	Selection
W25Q080	Selected by default
AT25SF128A	<code>#define RP2040_FLASH_AT25SF128A</code>
GD25Q64CS	<code>#define RP2040_FLASH_GD25Q64CS</code>
W25X10CL	<code>#define RP2040_FLASH_W25X10CL</code>
IS25LP080	<code>#define RP2040_FLASH_IS25LP080</code>
Generic 03H flash	<code>#define RP2040_FLASH_GENERIC_03H</code>

RP2040 Community Edition

The “RP2040 Community Edition” standard is a pinout that was defined by a committee of designers on the BastardKB Discord server.

These boards are designed to be a drop-in replacement for keyboards wanting an upgrade from ATmega32u4 based pro micros (eg. Elite-C).

Pinout Compatible Controllers

[0xB2 Splinky](#)

[Elite-Pi](#)

[Sea-Picro EXT](#)

[0xCB Helios](#)

[Frood](#)

[Liatris](#)



[Edit this page](#)