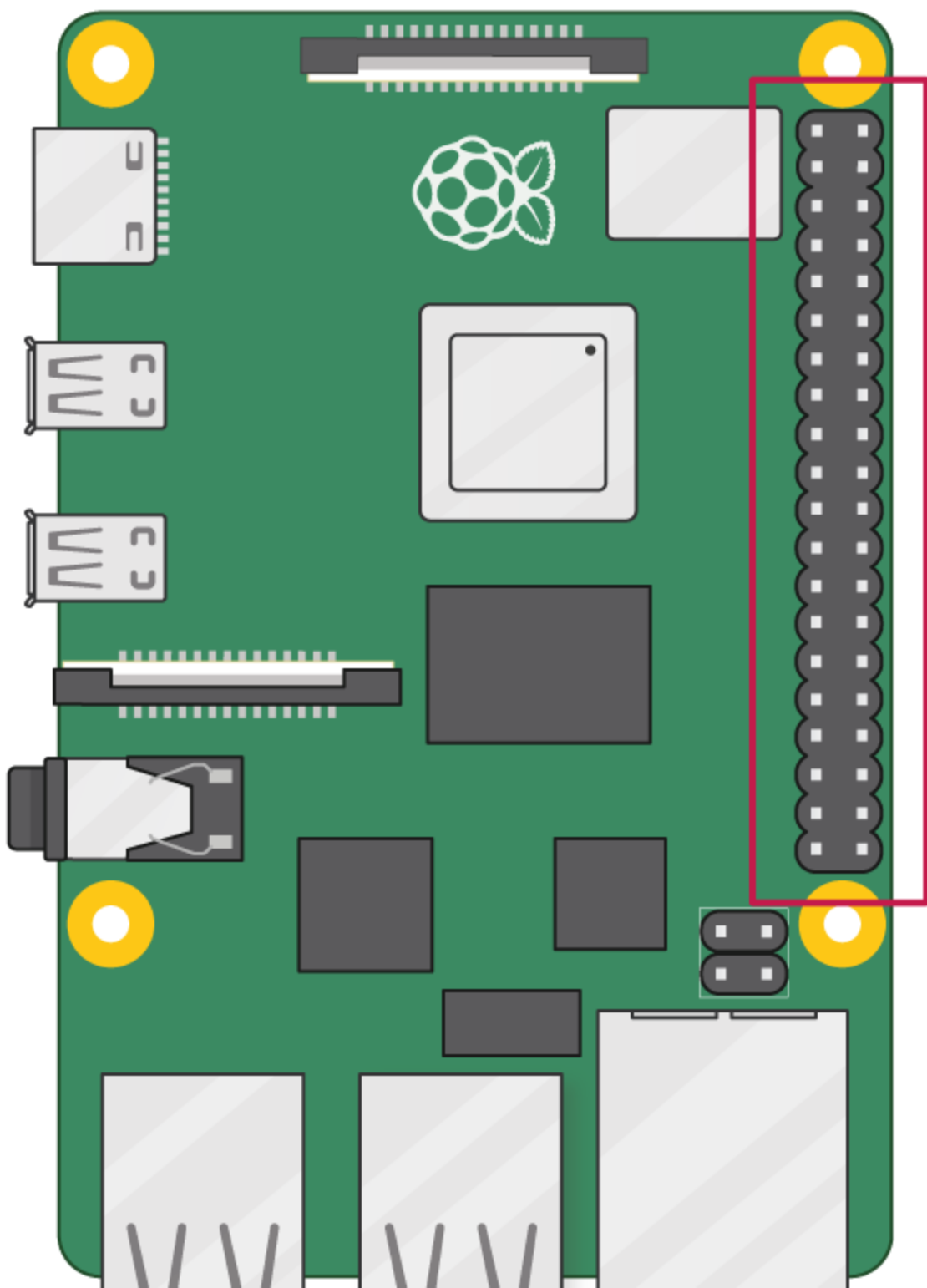
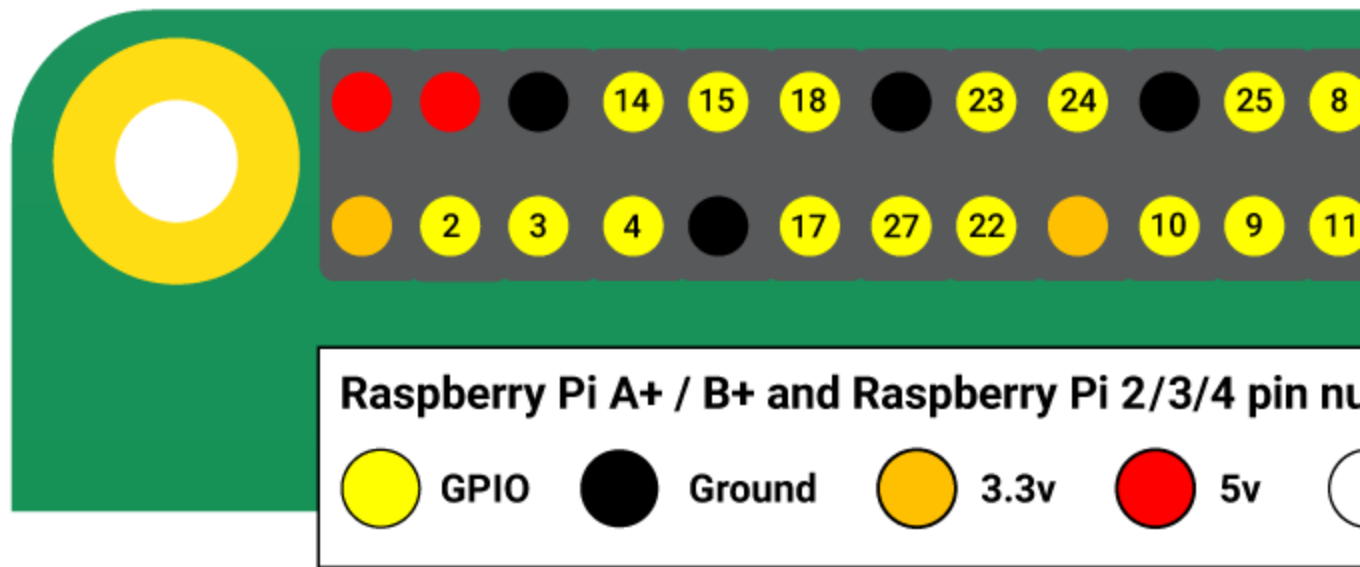


GPIO and the 40-pin Header

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards (unpopulated on Raspberry Pi Zero, Raspberry Pi Zero W and Raspberry Pi Zero 2 W). Prior to the Raspberry Pi 1 Model B+ (2014), boards comprised a shorter 26-pin header. The GPIO header on all boards (including the Raspberry Pi 400) have a 0.1" (2.54mm) pin pitch.



Any of the GPIO pins can be designated (in software) as an input or output pin and used for a wide range of purposes.



NOTE

The numbering of the GPIO pins is not in numerical order; GPIO pins 0 and 1 are present on the board (physical pins 27 and 28) but are reserved for advanced use (see below).

Voltages

Two 5V pins and two 3.3V pins are present on the board, as well as a number of ground pins (0V), which are unconfigurable. The remaining pins are all general purpose 3.3V pins, meaning outputs are set to 3.3V and inputs are 3.3V-tolerant.

Outputs

A GPIO pin designated as an output pin can be set to high (3.3V) or low (0V).

Inputs

A GPIO pin designated as an input pin can be read as high (3.3V) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software.

More

As well as simple input and output devices, the GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on specific pins.

- PWM (pulse-width modulation)
 - Software PWM available on all pins
 - Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19
- SPI
 - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
 - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C
 - Data: (GPIO2); Clock (GPIO3)
 - EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Serial
 - TX (GPIO14); RX (GPIO15)

GPIO pinout

A handy reference can be accessed on the Raspberry Pi by opening a terminal window and running the command `pinout`. This tool is provided by the [GPIO Zero](#) Python library, which is installed by default in Raspberry Pi OS.

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ pinout  
  
Revision : a02082  
SoC : BCM2837  
RAM : 1024Mb  
Storage : MicroSD  
USB ports : 4 (excluding power)  
Ethernet ports : 1  
Wi-fi : True  
Bluetooth : True  
Camera ports (CSI) : 1  
Display ports (DSI) : 1  
  
J8:  
3V3 (1) (2) 5V  
GPI02 (3) (4) 5V  
GPI03 (5) (6) GND  
GPI04 (7) (8) GPI014  
GND (9) (10) GPI015  
GPI017 (11) (12) GPI018  
GPI027 (13) (14) GND  
GPI022 (15) (16) GPI023  
3V3 (17) (18) GPI024  
GPI010 (19) (20) GND  
GPI09 (21) (22) GPI025  
GPI011 (23) (24) GPI08  
GND (25) (26) GPI07  
GPI00 (27) (28) GPI01  
GPI05 (29) (30) GND  
GPI06 (31) (32) GPI012  
GPI013 (33) (34) GND  
GPI019 (35) (36) GPI016  
GPI026 (37) (38) GPI020  
GND (39) (40) GPI021  
  
For further information, please refer to https://pinout.xyz/  
pi@raspberrypi:~ $
```

For more details on the advanced capabilities of the GPIO pins see gadgetoid's [interactive pinout diagram](#).

WARNING

While connecting up simple components to the GPIO pins is perfectly safe, it's important to be careful how you wire things up. LEDs should have resistors to limit the current passing through them. Do not use 5V for 3.3V components. Do not connect motors directly to the GPIO pins, instead use an [H-bridge circuit or a motor controller board](#).

Permissions

In order to use the GPIO ports your user must be a member of the `gpio` group. The `pi` user is a member by default, other users need to be added manually.

```
sudo usermod -a -G gpio <username>
```

GPIO in Python

Using the [GPIO Zero](#) library makes it easy to get started with controlling GPIO devices with Python. The library is comprehensively documented at [gpiozero.readthedocs.io](#).

LED

To control an LED connected to GPIO17, you can use this code:

```
from gpiozero import LED
from time import sleep

led = LED(17)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Run this in an IDE like Thonny, and the LED will blink on and off repeatedly.

LED methods include `on()`, `off()`, `toggle()`, and `blink()`.

Button

To read the state of a button connected to GPIO2, you can use this code:

```
from gpiozero import Button
from time import sleep
```

```
button = Button(2)

while True:
    if button.is_pressed:
        print("Pressed")
    else:
        print("Released")
    sleep(1)
```

Button functionality includes the properties `is_pressed` and `is_held`; callbacks `when_pressed`, `when_released`, and `when_held`; and methods `wait_for_press()` and `wait_for_release`.

Button + LED

To connect the LED and button together, you can use this code:

```
from gpiozero import LED, Button

led = LED(17)
button = Button(2)

while True:
    if button.is_pressed:
        led.on()
    else:
        led.off()
```

Alternatively:

```
from gpiozero import LED, Button

led = LED(17)
button = Button(2)

while True:
    button.wait_for_press()
    led.on()
    button.wait_for_release()
    led.off()
```

or:

```
from gpiozero import LED, Button

led = LED(17)
button = Button(2)

button.when_pressed = led.on
button.when_released = led.off
```