

4. Buzzer

1. Learning objectives

1. Learn to experiment with an external buzzer on the Raspberry Pi Pico2.
2. Understand the use of active buzzers.

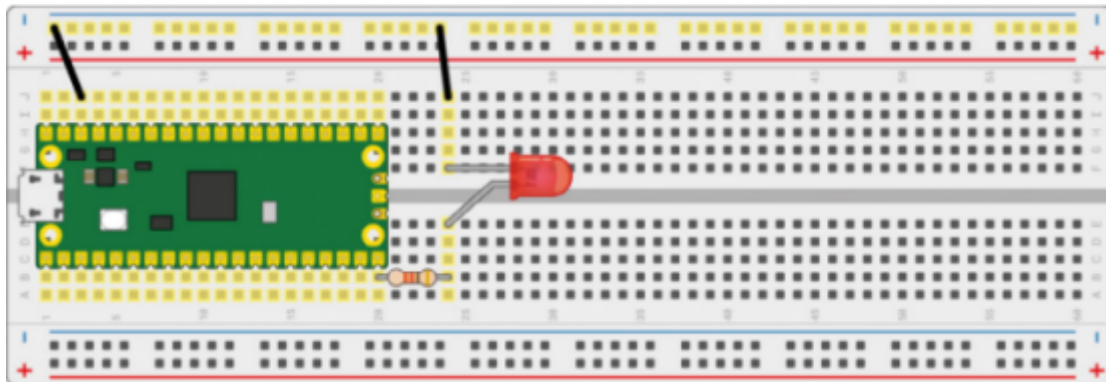
2. Hardware construction

This course.

Active buzzer*1

220Ω resistor*1

The circuit wiring diagram is shown below: (Replace the LED light with a buzzer, and connect the + sign of the buzzer to the resistor end)



3. Program analysis

Thonny programming

For the use of ThonnyIDE, please refer to the previous environment construction related courses.

```
import machine
import utime

buzzer = machine.Pin(15, machine.Pin.OUT)

while True:
    for i in range(80):
        buzzer.value(1)
        utime.sleep(0.001)
        buzzer.value(0)
        utime.sleep(0.001)

    for i in range(100):
        buzzer.value(1)
        utime.sleep(0.002)
        buzzer.value(0)
        utime.sleep(0.002)
```

import machine

The machine library contains all the instructions MicroPython needs to communicate with Pico and other MicroPython-compatible devices, extending the language of physical computing.

import utime

The "utime" library. This library handles everything to do with time, from measuring it to inserting delays into your program. The units are in seconds.

buzzer = machine.Pin(15, machine.Pin.OUT)

Set IO15 as the output pin.

Through the While loop, two different for loops make the buzzer emit sounds of different frequencies

4. Experimental phenomenon

After the program is downloaded, we can hear the buzzer playing sounds of different frequencies.