**Project 13：Passive Buzzer**

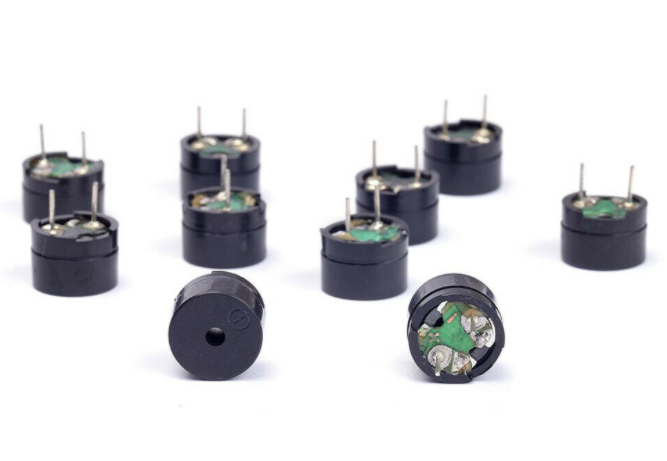
# **Introduction**

In a previous project, we have learned an active buzzer, which can only produce one sound and may let you feel monotonous. In this project, we will learn a passive buzzer and use the Raspberry Pi Pico to control the passive buzzer to sound an alarm. Unlike the active buzzer, the passive buzzer can emit sounds of different frequencies.

1. **Components Required**

|  |  |  |  |
| --- | --- | --- | --- |
| 17a6d7f241a04d4e932cb06b758197c4 | | _DSC2552 | |
| Raspberry Pi Pico\*1 | | Raspberry Pi PicoExpansion Board\*1 | |
| H4QG0GNSDN2S4]TSS)6UP4J |  |  |  |
| Passive Buzzer\*1 | Breadboard\*1 | Jumper Wires | USB Cable\*1 |

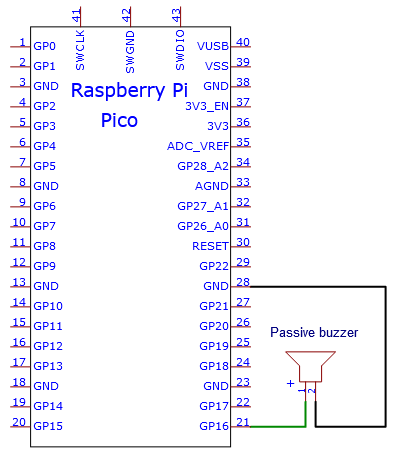
**3.Component Knowledge**

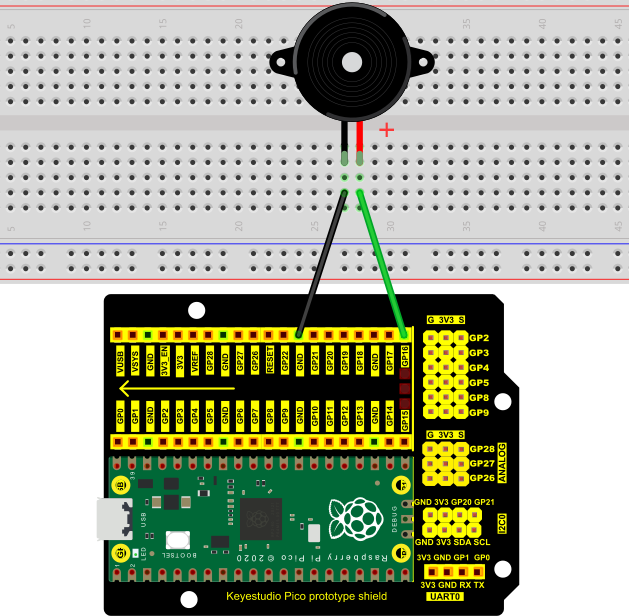


A passive buzzer is an integrated electronic buzzer with no internal vibration source. It must be driven by 2K to 5K square wave, not a DC signal. The two buzzers are very similar in appearance, but one buzzer with a green circuit board is a passive buzzer, while the other with black tape is an active buzzer. Passive buzzers cannot distinguish between positive polarity while active buzzers can.



**4. Circuit Diagram and Wiring Diagram**



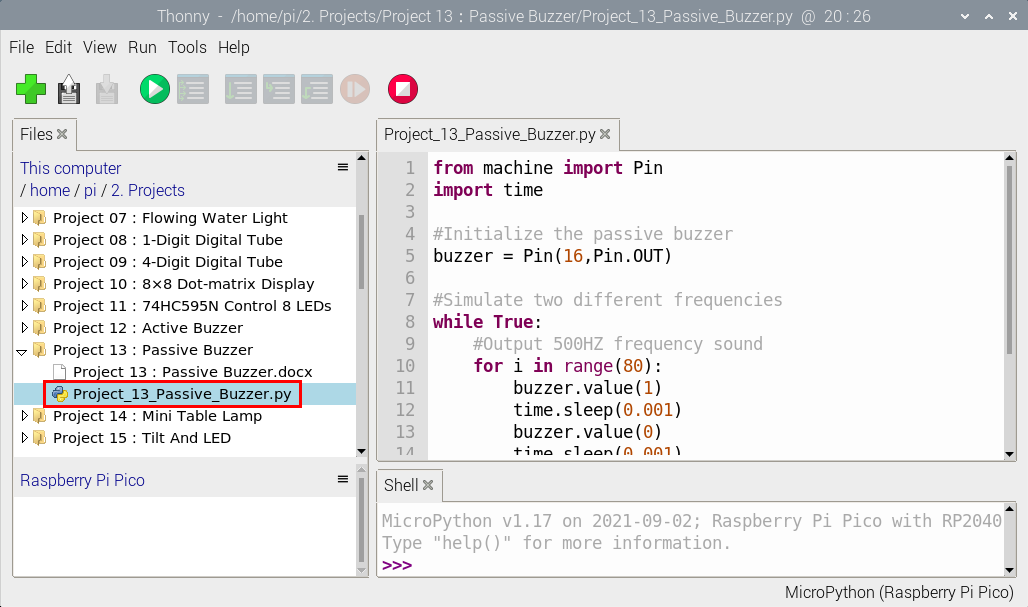


1. **Test Code**

The code used in this project is saved in the file KS3020 Keyestudio Raspberry Pi Pico Learning Kit Ultimate Edition\3. Raspberry Pi System\Python\_Tutorial\2. Projects\Project 13：Passive Buzzer. You can move the code anywhere. We save the code to the pi folder of the Raspberry Pi system. The path:home/pi/2. Projects



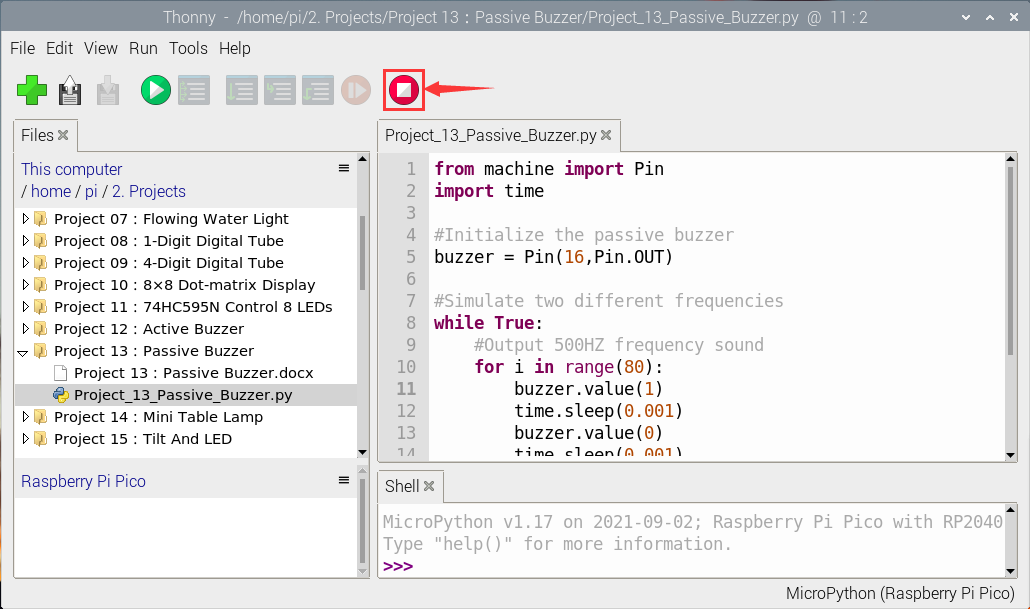
Open“Thonny”, click“This computer”→“home”→“pi”→“2. Projects”→“Project 13：Passive Buzzer”. And double left-click the“Project\_13\_Passive\_Buzzer.py”.



|  |
| --- |
| from machine import Pin  import time  #Initialize the passive buzzer  buzzer = Pin(16,Pin.OUT)  #Simulate two different frequencies  while True:  #Output 500HZ frequency sound  for i in range(80):  buzzer.value(1)  time.sleep(0.001)  buzzer.value(0)  time.sleep(0.001)  #Output 250HZ frequency sound  for i in range(100):  buzzer.value(1)  time.sleep(0.002)  buzzer.value(0)  time.sleep(0.002) |

1. **Test Result**

Ensure that the Raspberry Pi Pico is connected to the computer，click“Stop/Restart backend”.



Click “Run current script”, the code starts executing, we will see that the the passive buzzer sounds the alarm. Click “Stop/Restart backend”to exit the program.

