**Project 26：Sound Control Fan**

1. **Introduction**

The sound sensor has a built-in capacitive electret microphone and power amplifier. It can be used to detect the sound intensity of the environment. In this project, we use a Raspberry Pi Pico to control a sound sensor and a motor module to make a voice-activated fan.

1. **Components Required**

|  |  |  |  |
| --- | --- | --- | --- |
| 17a6d7f241a04d4e932cb06b758197c4 | _DSC2552 | 6 |  |
| Raspberry Pi Pico\*1 | Raspberry Pi Pico Expansion Board\*1 | Sound Sensor\*1 | USB Cable\*1 |
| KS6038 130电机驱动模块 | 杜邦线-1 | 杜邦线-3 |  |
| 130 Motor Module\*1 | F-F Dupont Wires | M-F Dupont Wires |  |

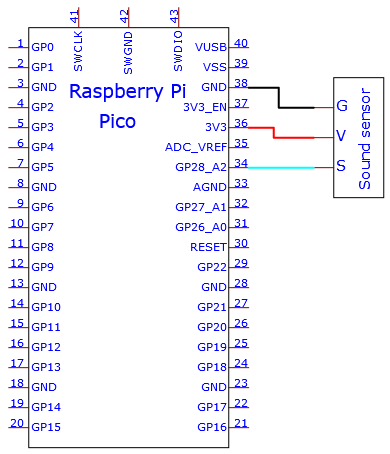
1. **Component Knowledge**

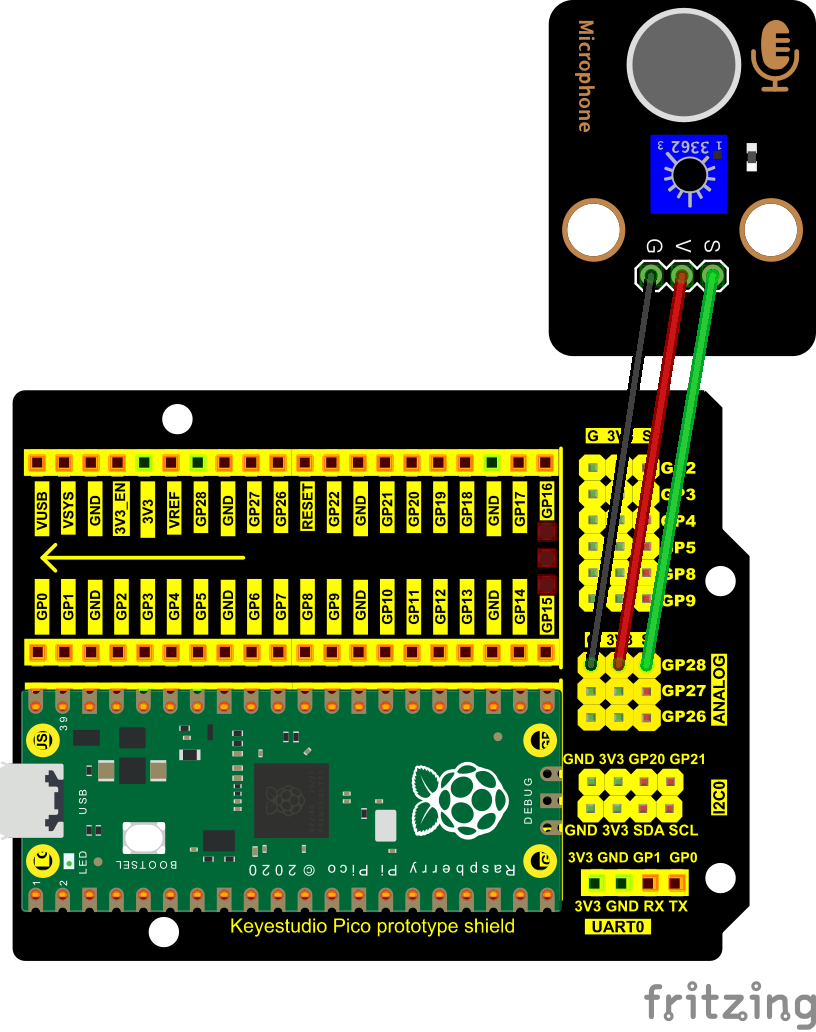
# 19-2_画板 1_画板 1

**Sound sensor** is usually used to detect the loudness of the sound in the surroundings. Arduino can collect its output signal through the analog input interface. The S pin is an analog output, which is the real-time output of the microphone voltage signal. The sensor comes with a potentiometer so you can adjust the signal strength. It also has two fixing holes so that the sensor can be installed on any other equipment. You can use it to make some interactive works, such as voice-operated switches.

1. **Read the Analog Value of the Sound Sensor**

We first use a simple code to read the analog value of the sound sensor and print it to the serial monitor, please refer to the following wiring diagram for the wiring.

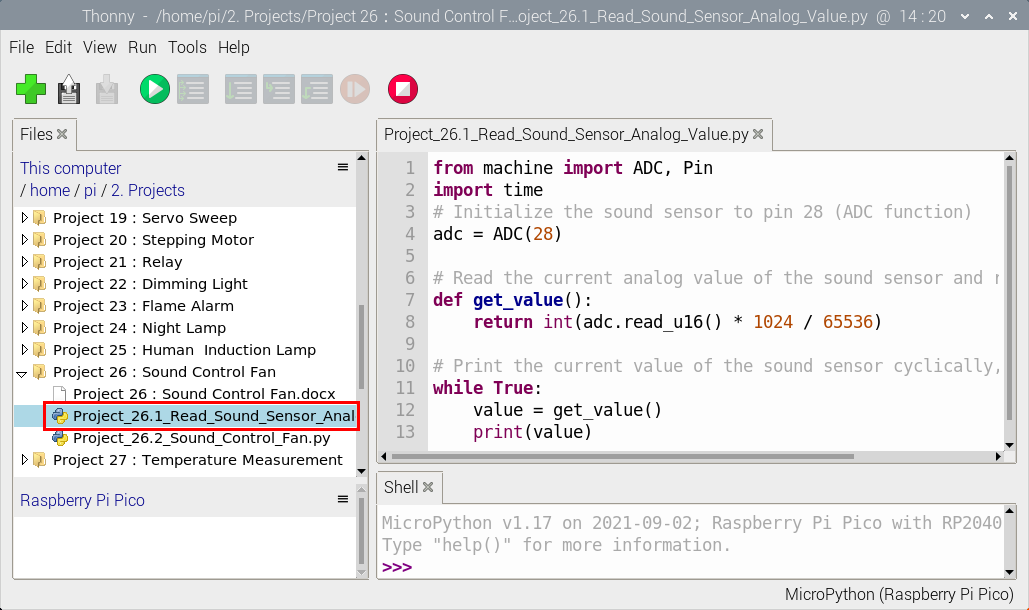




The code used in this project is saved in the file KS3020 Keyestudio Raspberry Pi Pico Learning Kit Ultimate Edition\3. Raspberry Pi System\Python\_Tutorial\2. Projects\Project 26：Sound Control Fan. You can move the code anywhere. We save the code to the pi folder of the Raspberry Pi system. The path:home/pi/2. Projects

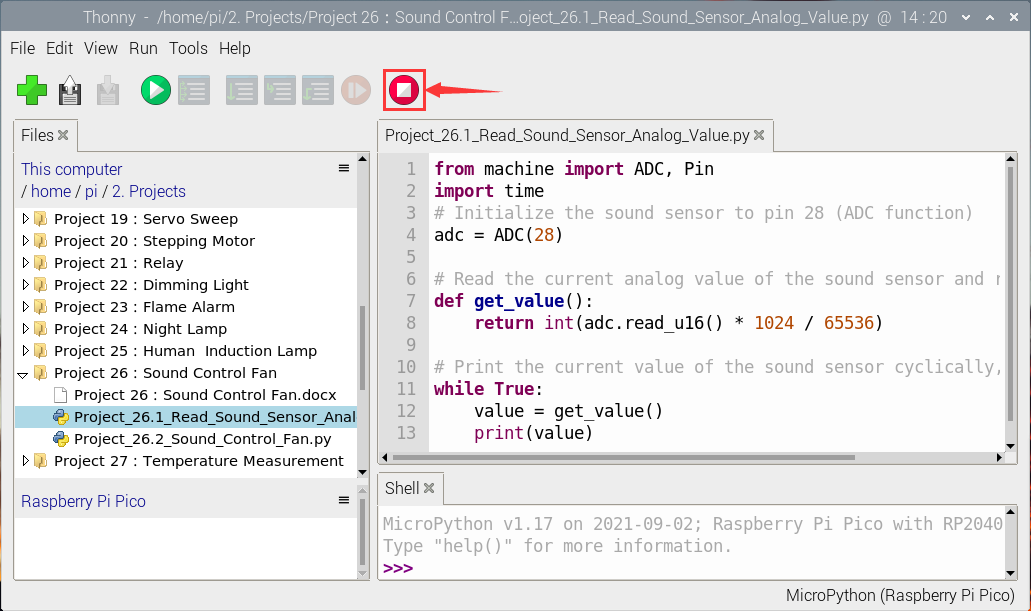


Open“Thonny”, click“This computer”→“home”→“pi”→“2. Projects”→“Project 26：Sound Control Fan”. And double left-click the“Project\_26.1\_Read\_Sound\_Sensor\_Analog\_Value.py”.

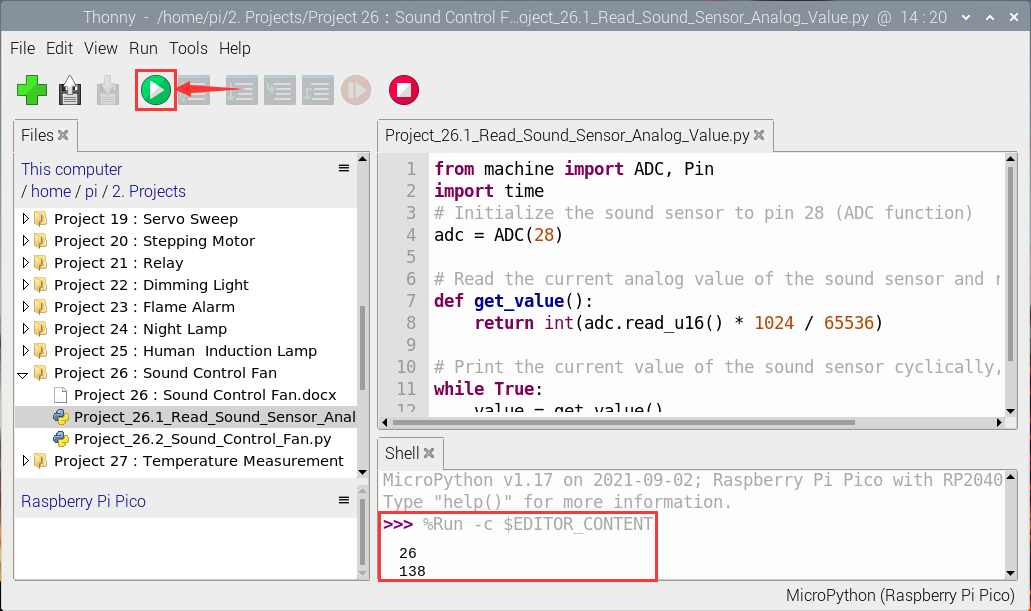


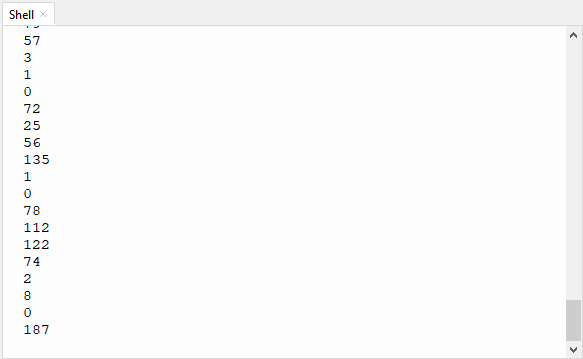
|  |
| --- |
| from machine import ADC, Pin  import time  # Initialize the sound sensor to pin 28 (ADC function)  adc = ADC(28)  # Read the current analog value of the sound sensor and return [0, 1023]  def get\_value():  return int(adc.read\_u16() \* 1024 / 65536)    # Print the current value of the sound sensor cyclically, value=[0, 1023]  while True:  value = get\_value()  print(value)  time.sleep(0.1) |

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



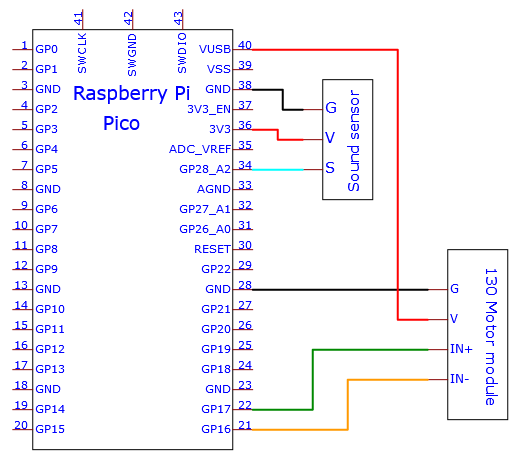
Click “Run current script”, the code starts executing, we will see that the "Shell" window of Thonny IDE will print the analog values read by the sound sensor. When you clap your hands to the sensor, the analog value of the sound sensor will change significantly. Click“Stop/Restart backend”to exit the program.

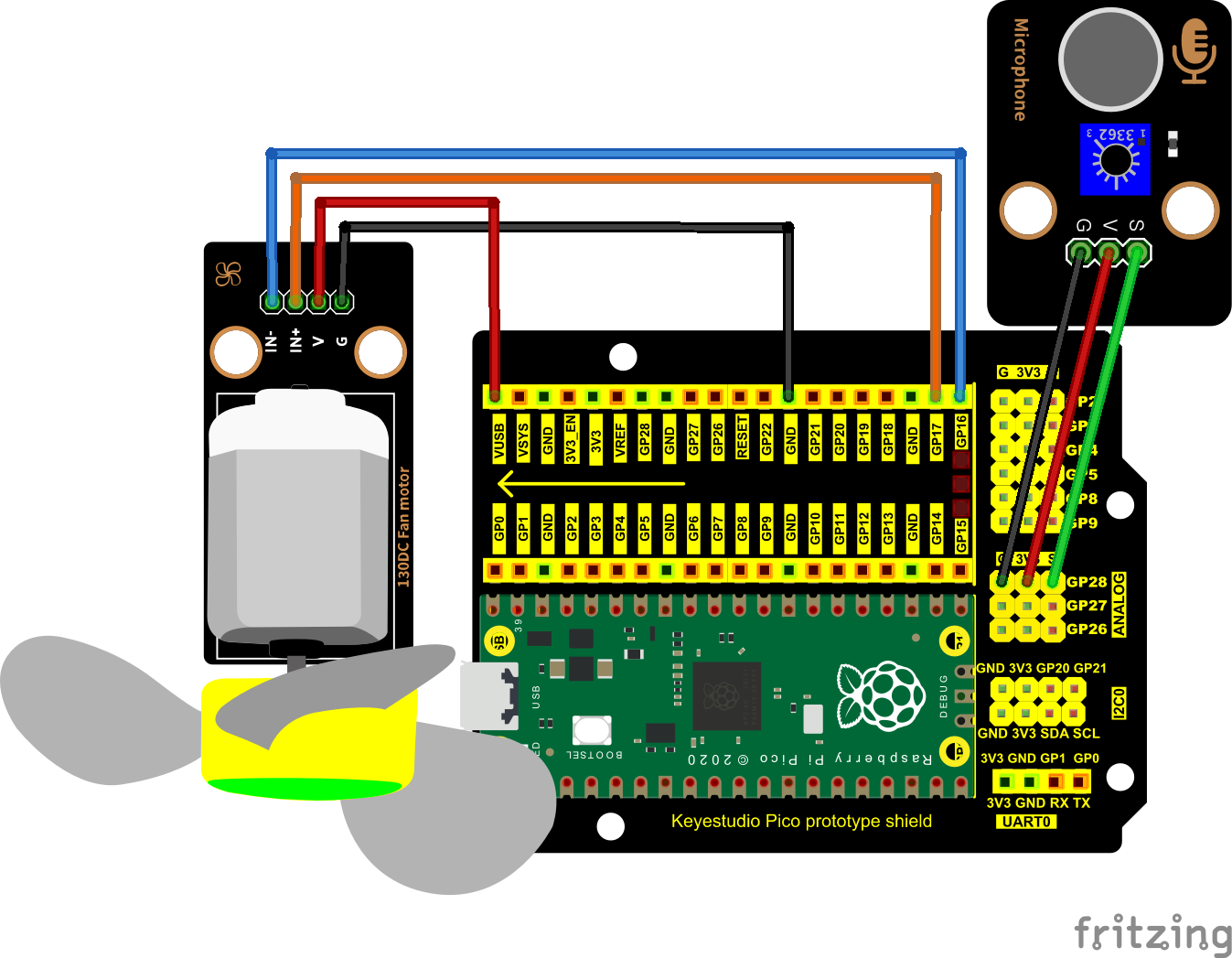




1. **Wiring Diagram**

Next, we use a sound sensor, a 130 motor module and a fan leaf to make a voice-activated fan. The wiring diagram is as follows.



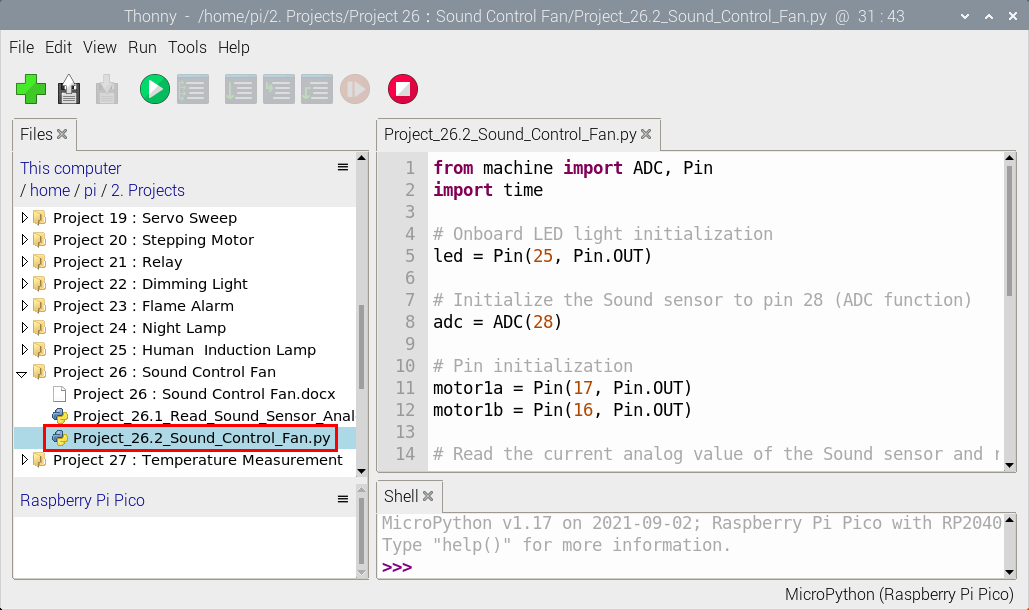


1. **Test Code**（Note：The threshold 600 in the code can be reset itself as needed）

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 26：Sound Control Fan. 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



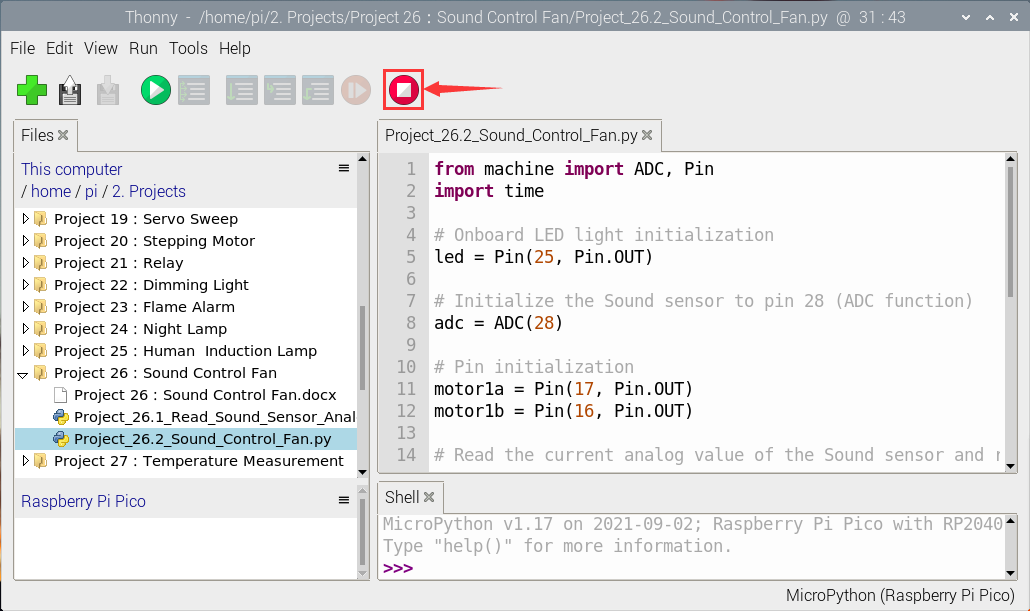
Open“Thonny”, click““This computer”→“home”→“pi”→“2. Projects”→“Project 26：Sound Control Fan”. And double left-click the“Project\_26.2\_Sound\_Control\_Fan.py”.



|  |
| --- |
| from machine import ADC, Pin  import time    # Onboard LED light initialization  led = Pin(25, Pin.OUT)    # Initialize the Sound sensor to pin 28 (ADC function)  adc = ADC(28)    # Pin initialization  motor1a = Pin(17, Pin.OUT)  motor1b = Pin(16, Pin.OUT)  # Read the current analog value of the Sound sensor and return [0, 1023]  def get\_value():  return int(adc.read\_u16() \* 1024 / 65536)    # If the Sound sensor detects Sounds, the built-in LED on the Pico board will blink  # and the motor will rotate when the analog value is greater than 600  # Otherwise, the motor does not rotate and the LED goes off  while True:  value = get\_value()  if value >600:  led.value(1) # Set led turn on  motor1a.high() # Set motor1a high  motor1b.low() # Set motor1b low  time.sleep(5) # delay time  else:  motor1a.low() # Set motor1a low  motor1b.low() # Set motor1b low  led.value(0) # Set led turn off |

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 clap your hands to the sound sensor, and when the sound intensity exceeds a threshold, the small fan spins while the Raspberry Pi Pico's built-in LED lights up.  Instead, the small fan doesn't rotate, and the Raspberry Pi Pico's built-in LEDS don't light up. Press“Ctrl+C”or click“Stop/Restart backend”to exit the program.

