**Project 31：Temperature Instrument**

1. **Introduction**

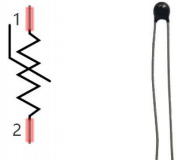
Thermistor is a kind of resistor whose resistance depends on temperature changes, which is widely used in gardening, home alarm system and other devices. Therefore, we can use this feature to make a temperature instrument.

1. **Components Required**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 17a6d7f241a04d4e932cb06b758197c4 | _DSC2552 | |  | 10k欧 |
| Raspberry Pi Pico\*1 | Raspberry Pi Pico Expansion Board\*1 | | Thermistor\*1 | 10KΩResistor\*1 |
| 杜邦线-3 |  | 5 |  |  |
| 10CM M-F Dupont Wires | Breadboard\*1 | LCD 128X32 DOT\*1 | Jumper Wires | USB Cable\*1 |

1. **Component Knowledge**

Thermistor: A thermistor is a temperature sensitive resistor. When it senses a change in temperature, the thermistor's resistance changes. We can use this feature to detect temperature intensity with thermistor. Thermistors and its electronic symbols are shown below:



The relation between thermistor resistance and temperature is:



**[in](javascript:;) [the](javascript:;) [formula](javascript:;):**

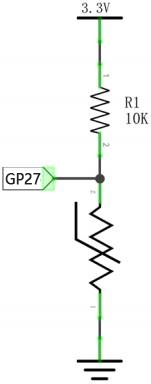
Rt is the resistance of the thermistor at T2 temperature.

R is the nominal resistance value of the thermistor at T1 room temperature.

EXP[n] is the nth power of e.

B is the temperature index

T1 and T2 refer to K degrees, that is, Kelvin temperature. Kelvin temperature =273.15 + Celsius temperature. For thermistor parameters, we use : B=3950, R=10KΩ，T1=25℃.The circuit connection method of thermistor is similar to that the photoresistor, as shown below :

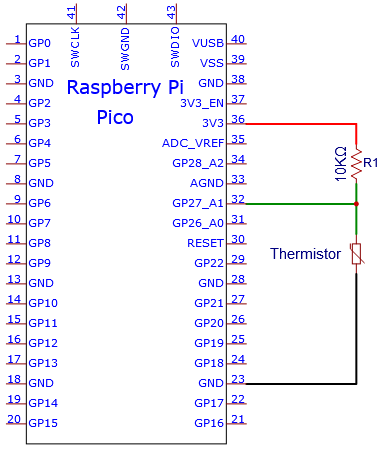


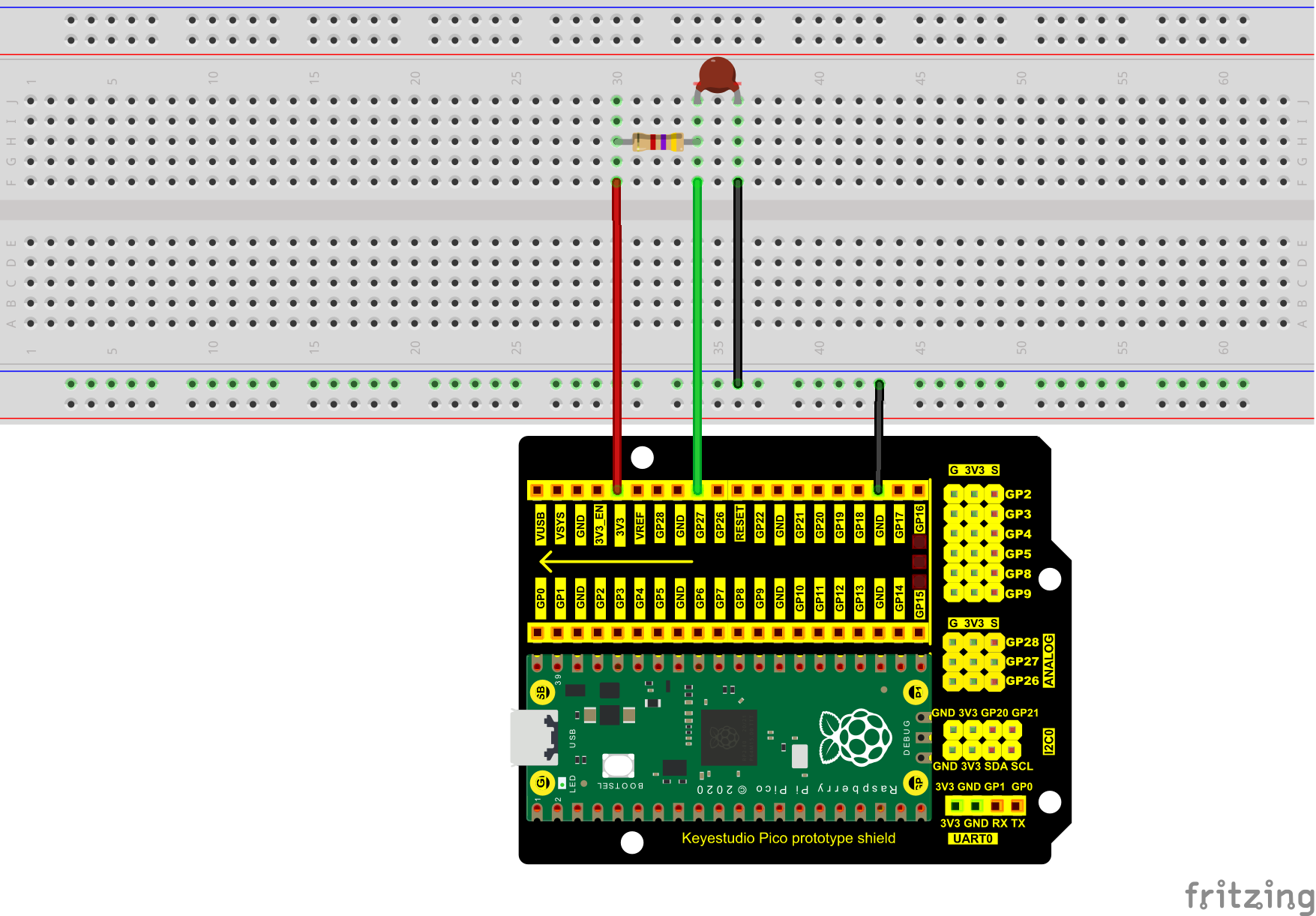
We can use the value measured by the ADC converter to get the resistance value of the thermistor, and then use the formula to get the temperature value. Therefore, the temperature formula can be deduced as:



1. **Read the Values**

First we will learn the thermistor to read the current ADC value, voltage value and temperature value and print them out . Please connect the wires according to the following wiring diagram.



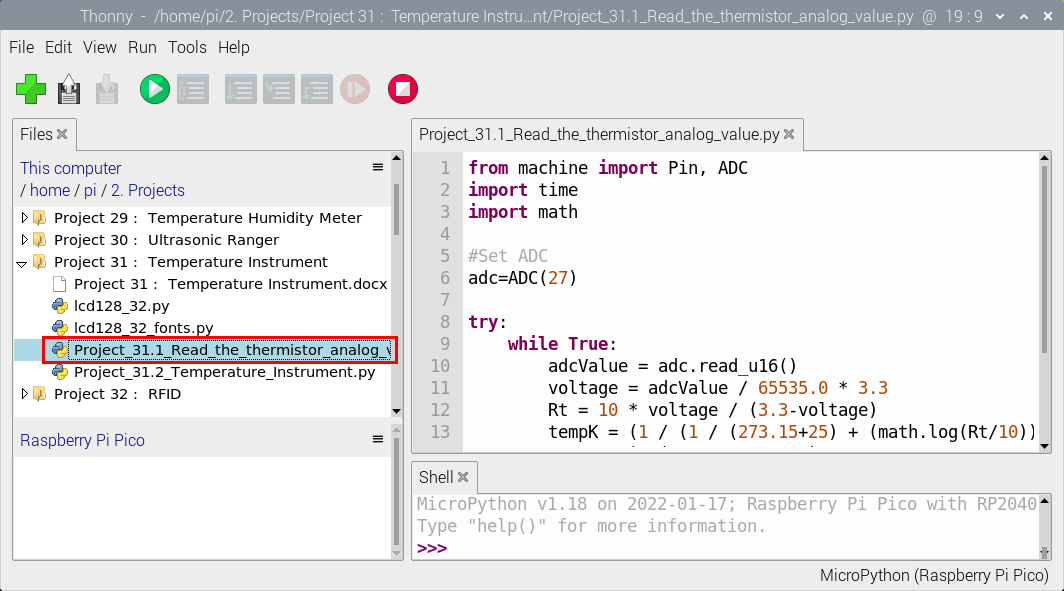


The code used in this project is saved in the file KS3020 Keyestudio Raspberry Pi Pico Learning Kit Ultimate Edition\3. Raspberry Pi System\Python\_Tutorial\2. Projects\Project 31：Temperature Instrument. You can move the code anywhere. We save the code to the pi folder of the Raspberry Pi system. The path:home/pi/2. Projects.

Open“Thonny”, click“This computer”→“home”→“pi”→“2. Projects”→“Project 31: Temperature Instrument”. And double left-click the“Project\_31.1\_Read\_the\_thermistor\_analog\_value.py”.

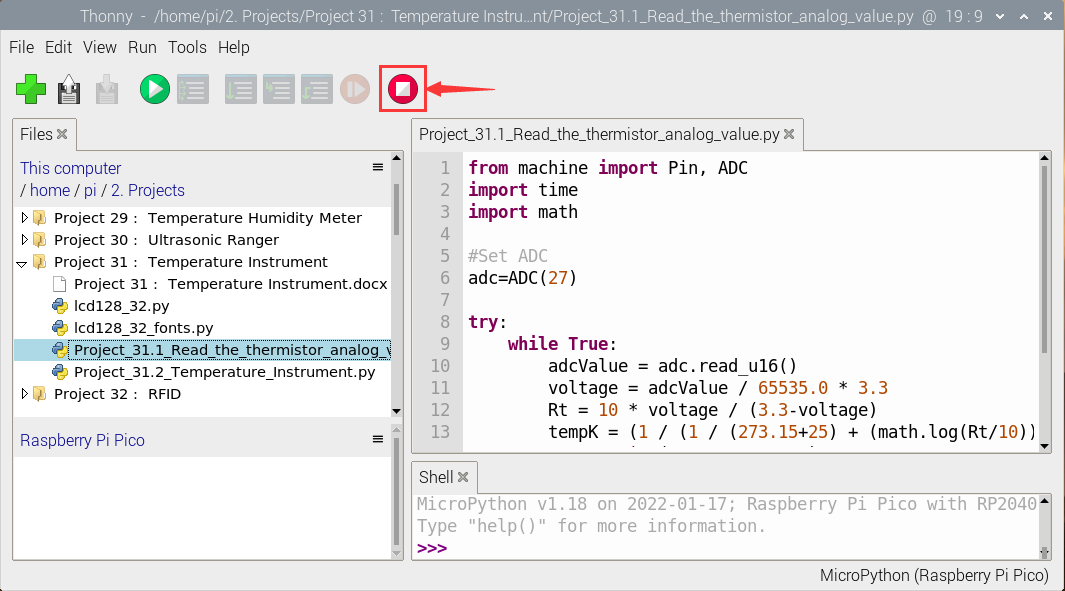


Open“Thonny”, click“This computer”→“home”→“pi”→“2. Projects”→“Project 31: Temperature Instrument”. And double left-click the“Project\_31.1\_Read\_the\_thermistor\_analog\_value.py”.

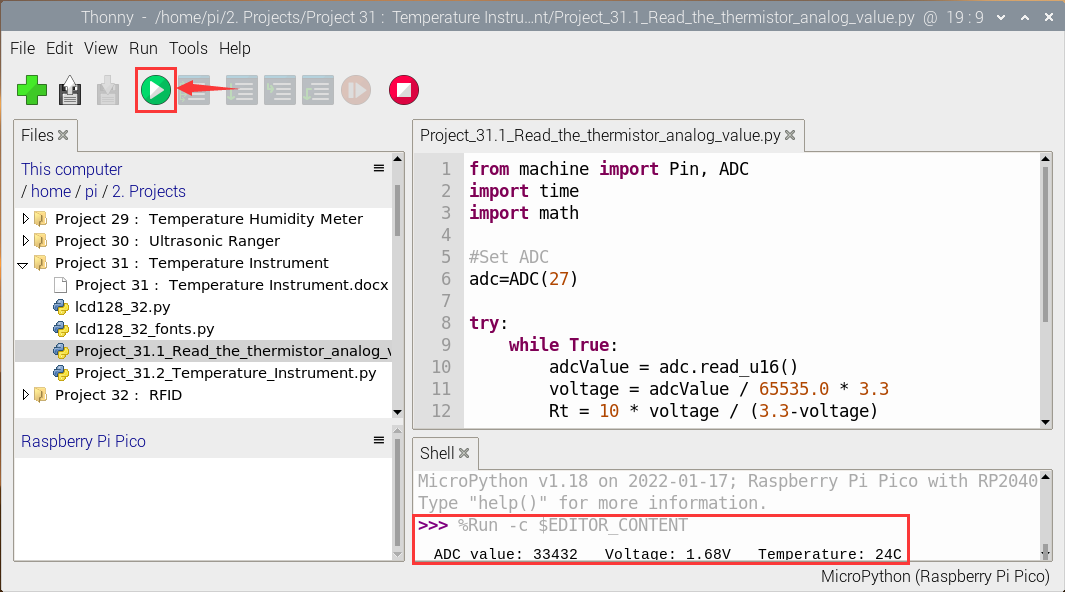


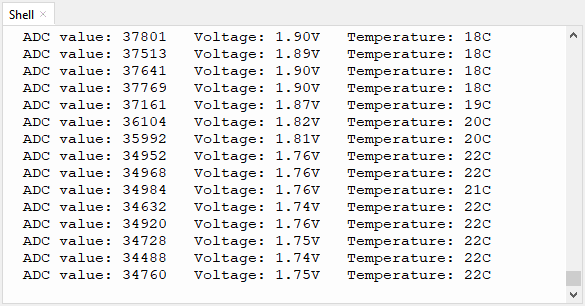
|  |
| --- |
| from machine import Pin, ADC  import time  import math  #Set ADC  adc=ADC(27)  try:  while True:  adcValue = adc.read\_u16()  voltage = adcValue / 65535.0 \* 3.3  Rt = 10 \* voltage / (3.3-voltage)  tempK = (1 / (1 / (273.15+25) + (math.log(Rt/10)) / 3950))  tempC = int(tempK - 273.15)  print("ADC value:", adcValue, " Voltage: %0.2f"%voltage + "V",  " Temperature: " + str(tempC) + "C")  time.sleep(1)  except:  pass |

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



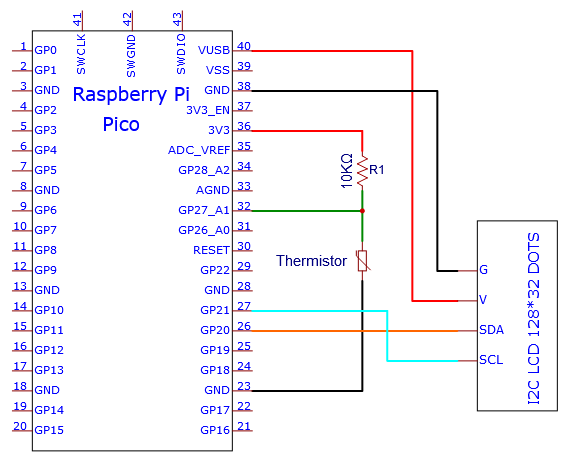
Click “Run current script”, the code starts executing, we will see that the "Shell" window of Thonny IDE will continuously display the thermistor's current ADC value, voltage value, and temperature value.  Try pinching the thermistor with your index finger and thumb (don't touch the wire) for a while, and you will see the temperature increase. Press“Ctrl+C”or click“Stop/Restart backend”to exit the program.

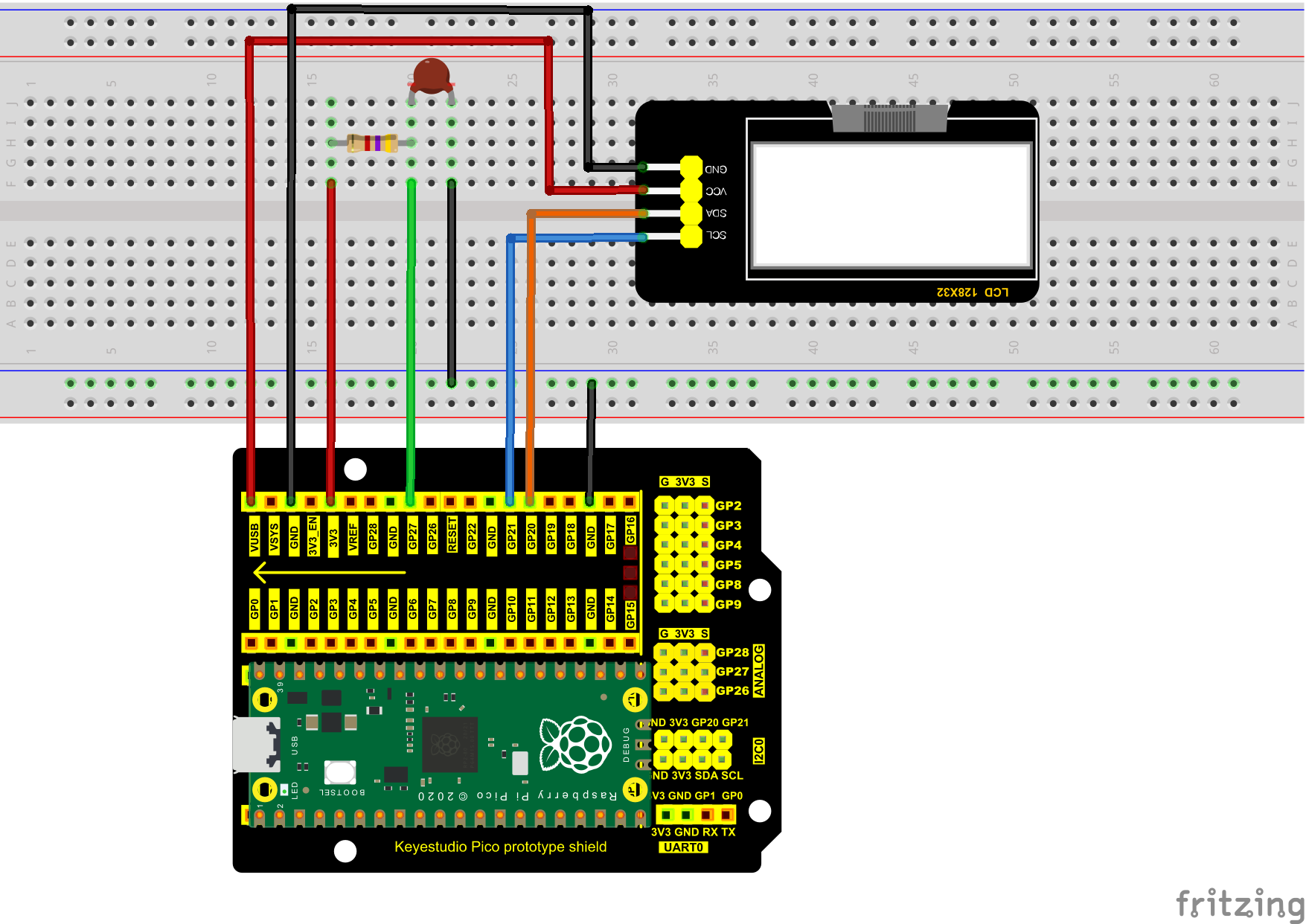




1. **Circuit Diagram and Wiring Diagram**

Note : LCD\_128X32\_DOT must be connected with a 10CM M-F Dupont wire, the LCD\_128X32\_DOT will display normally. Otherwise, using a 20CM M-F Dupont wire may cause the LCD\_128X32\_DOT display abnormally.



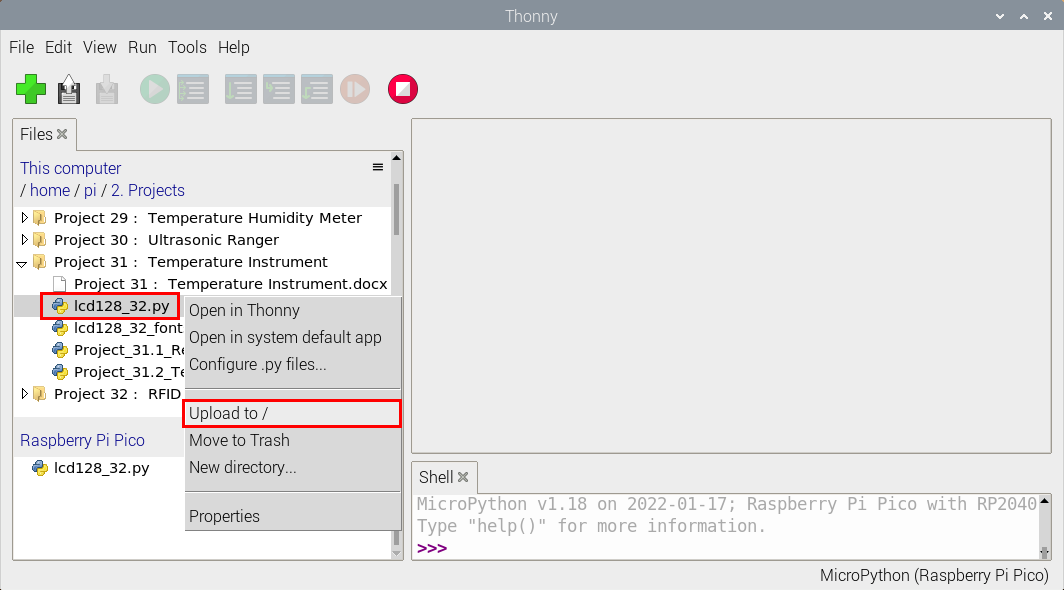


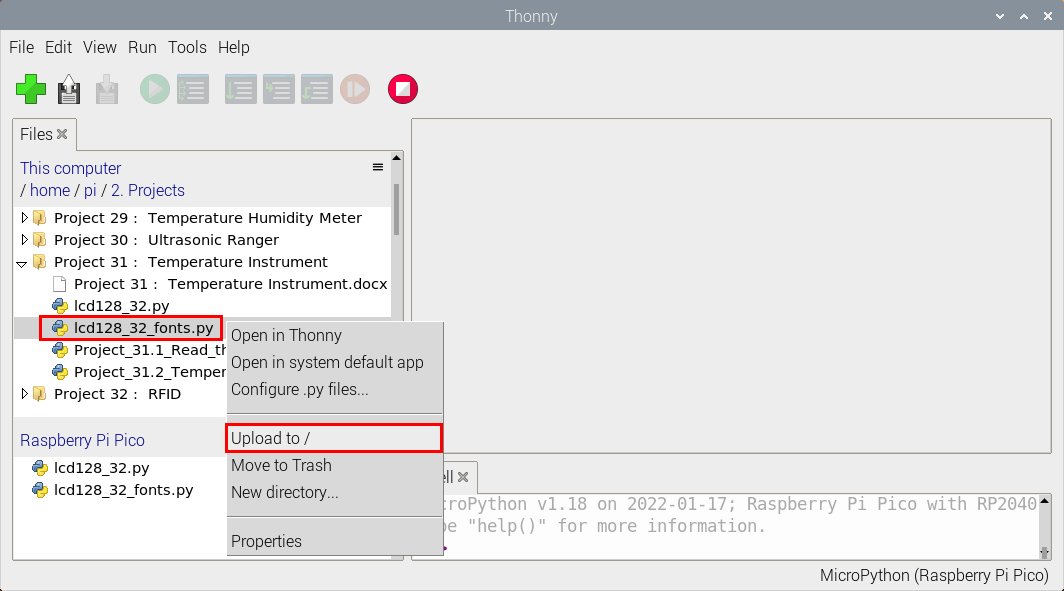
1. **Test Code**

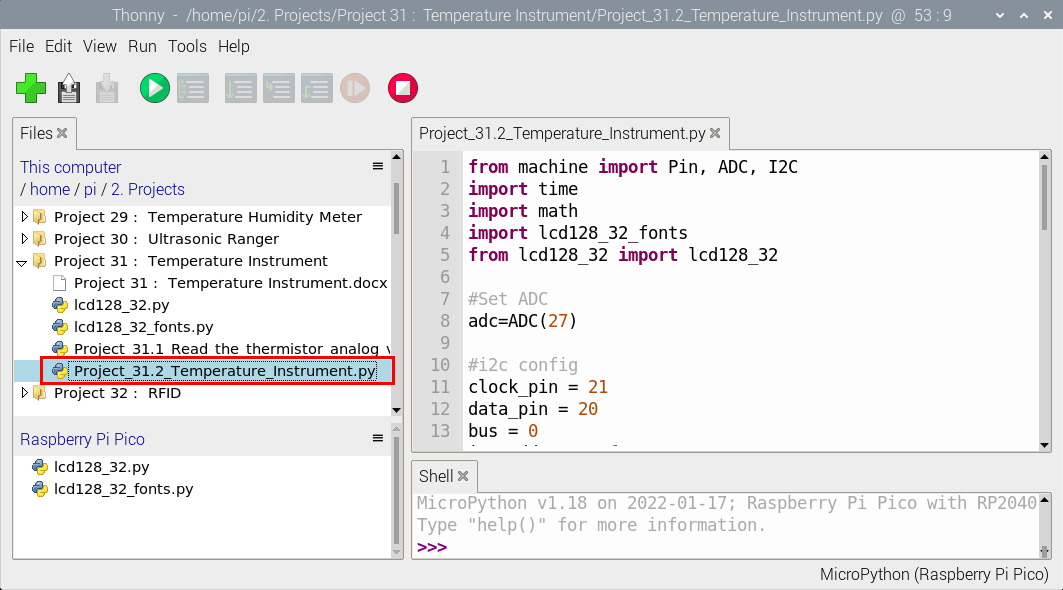
The code used in this project is saved in the file KS3020 Keyestudio Raspberry Pi Pico Learning Kit Ultimate Edition\3. Raspberry Pi System\Python\_Tutorial\2. Projects\Project 31：Temperature Instrument. You can move the code anywhere. We save the code to the pi folder of the Raspberry Pi system. The path:home/pi/2. Projects.



Open“Thonny”, click “This computer”→“home”→“pi”→“2. Projects”→“Project 31：Temperature Instrument”. Select“lcd128\_32.py”and“lcd128\_32\_fonts.py”， right-click and select“Upload to /”，waiting for the “lcd128\_32.py”and“lcd128\_32\_fonts.py”to be uploaded to the Raspberry Pi Pico. And double left-click the“Project\_31.2\_Temperature\_Instrument.py”.



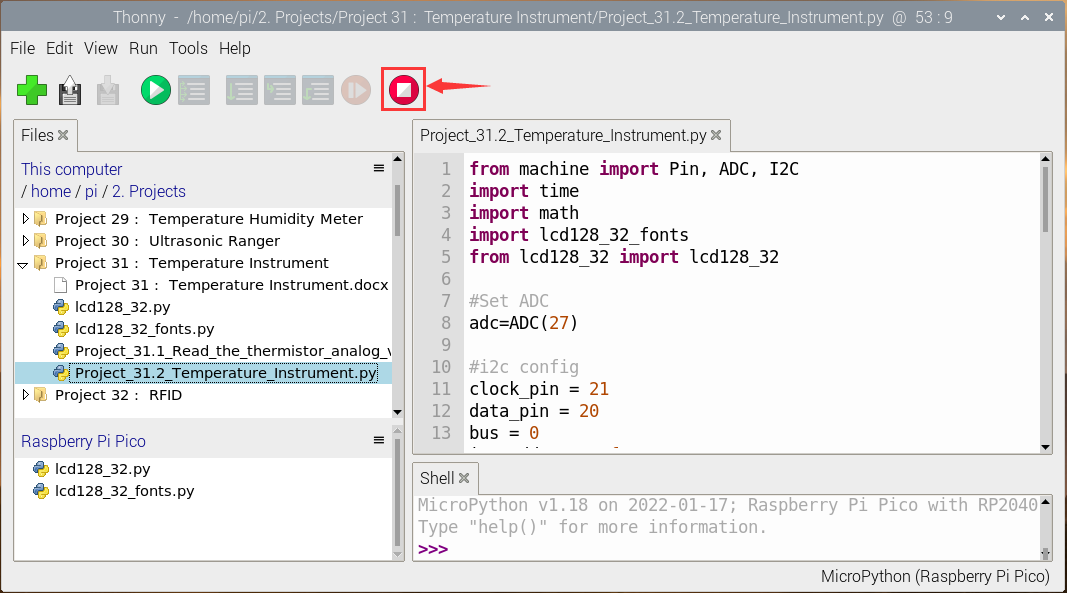




|  |
| --- |
| from machine import Pin, ADC, I2C  import time  import math  import lcd128\_32\_fonts  from lcd128\_32 import lcd128\_32  #Set ADC  adc=ADC(27)  #i2c config  clock\_pin = 21  data\_pin = 20  bus = 0  i2c\_addr = 0x3f  use\_i2c = True  def scan\_for\_devices():  i2c = machine.I2C(bus,sda=machine.Pin(data\_pin),scl=machine.Pin(clock\_pin))  devices = i2c.scan()  if devices:  for d in devices:  print(hex(d))  else:  print('no i2c devices')  try:  while True:  adcValue = adc.read\_u16()  voltage = adcValue / 65535.0 \* 3.3  Rt = 10 \* voltage / (3.3-voltage)  tempK = (1 / (1 / (273.15+25) + (math.log(Rt/10)) / 3950))  tempC = int(tempK - 273.15)    if use\_i2c:  scan\_for\_devices()  lcd = lcd128\_32(data\_pin, clock\_pin, bus, i2c\_addr)    lcd.Clear()  lcd.Cursor(0, 0)  lcd.Display("Voltage:")  lcd.Cursor(0, 8)  lcd.Display(str(voltage))  lcd.Cursor(0, 20)  lcd.Display("V")  lcd.Cursor(2, 0)  lcd.Display("Temperature:")  lcd.Cursor(2, 12)  lcd.Display(str(tempC))  lcd.Cursor(2, 15)  lcd.Display("C")  time.sleep(0.5)  except:  pass |

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 LCD 128X32 DOT displays the voltage value of the thermistor and the temperature value in the current environment. Press“Ctrl+C”or click“Stop/Restart backend”to exit the program.

