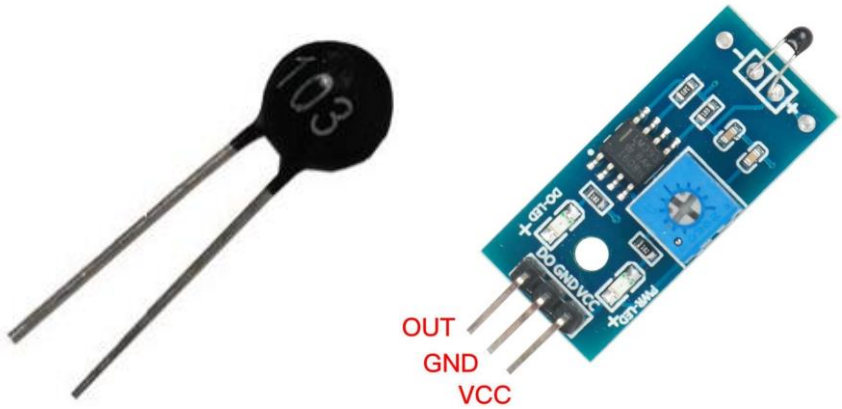
Introduction of thermistors

Thermistors are a class of sensing elements, which can be divided into positive temperature coefficient thermistors (PTC) and negative temperature coefficient thermistors (NTC) according to different temperature coefficients. The typical characteristic of thermistors is that they are sensitive to temperature, showing different resistance values at different temperatures. The positive temperature coefficient thermistor (PTC) has higher resistance value when the temperature is higher, and the negative temperature coefficient thermistor (NTC) has lower resistance value when the temperature is higher, they both belong to semiconductor devices.



Thermistor

Experiment purpose

- Read the resistance values of thermistors at different temperatures
- Use thermistor to control LED on and off

The experimental principle

The LED is connected in series with the thermistor, and the end of the thermistor is connected with a 10K pull-up resistance. The voltage change of the thermistor varies with the temperature is read through the analog port, and these parameters are printed out through the serial port monitor. Meanwhile, the change of LED light brightness is observed.

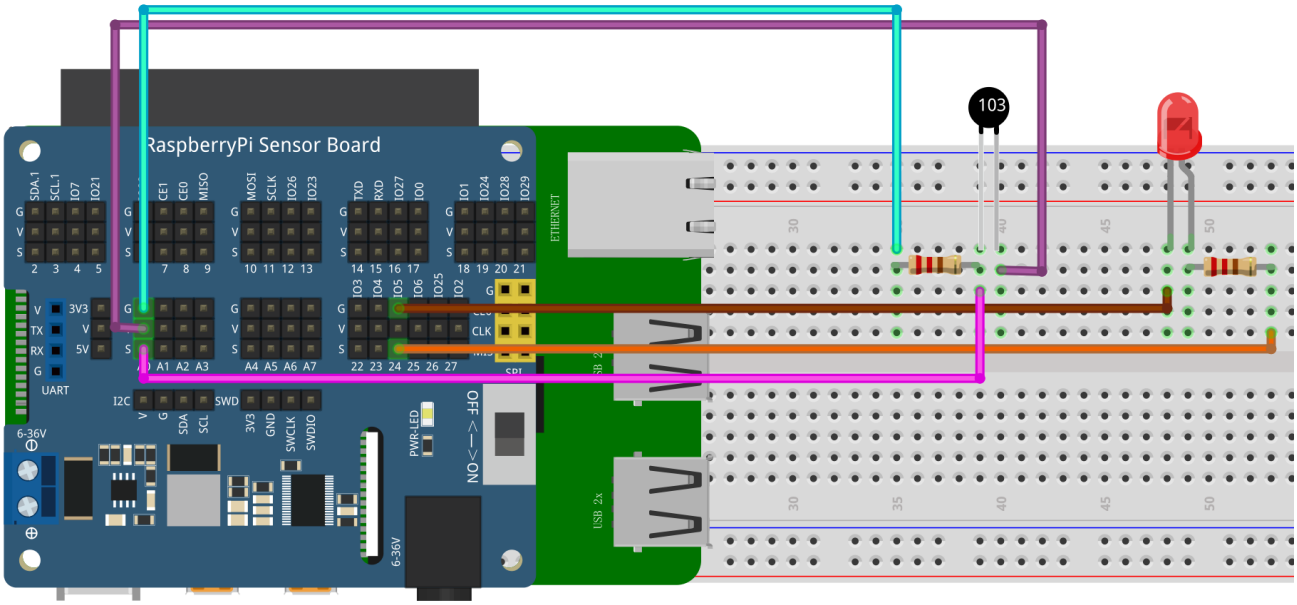
The component list

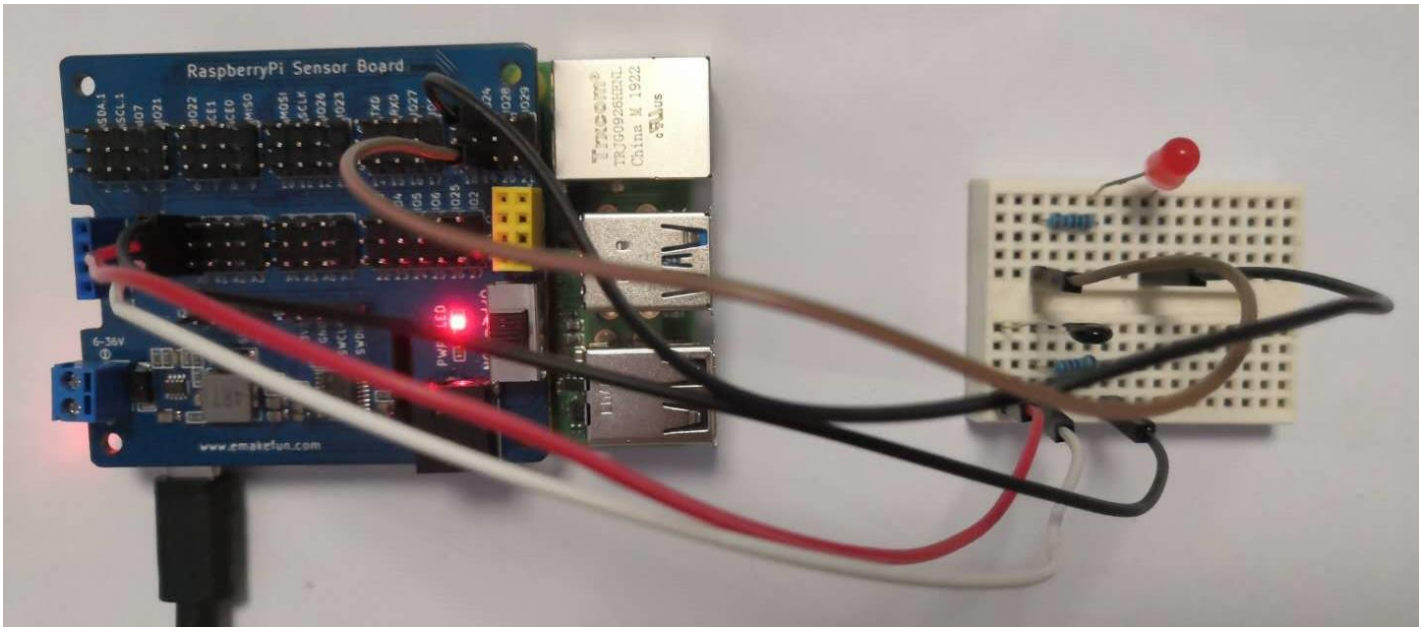
- RaspberryPi mainboard
- Raspberry Pi expansion board
- Breadboard
- Power cord
- Thermistor *1
- Red LED*1

- 10kΩ resistance * 1
- 220Ω resistance * 1
- Several jumpers

Wiring

RaspberryPi	Thermistor
5V	1
GND	3
A0	2
RaspberryPi	LED module
IO24(wiringPi)/19(BCM)	+
GND	—





C++ program

```
#include <stdio.h>//Import the basic library
#include <wiringPi.h>//Import the Raspberry Pi WiringPi encoding IO control library
#include <wiringPiI2C.h>//Import the Raspberry Pi WiringPi coding I2C control library

int  LEDPIN = 24; // LED light connects to IO0 port
int  value = 0 ;
float voltage = 0.0 ;
int main()
{
    wiringPiSetup();
    wiringPiI2CSetup(0x04);
    pinMode(LEDPIN, OUTPUT);
    while(1)
    {
        value =  wiringPiI2CReadReg16(0x04, 0x10);
        printf("value %d\n",value); //print value
        if(value >= 800)
        {
            digitalWrite(LEDPIN, HIGH);
            delay(500);
        }else{
            digitalWrite(LEDPIN, LOW);
        }
        delay(1000); // Print Value
    }
}
```

Python program

```
import time
import smbus as smbus
import RPi.GPIO as GPIO
import time

LEDPIN = 19
cyc =0.5
ADC=smbus.SMBus(1)#Declare to use I2C 1
GPIO.setmode(GPIO.BCM)
GPIO.setup(LEDPIN, GPIO.OUT)
```

```
while True:
    ADC.write_byte(0x04, 0x10)#Write a byte to the slave
    val = ADC.read_word_data(0x04, 0x10)
    print(val)#Raspberry Pi reads the data returned by the expansion board and prints it
out
    if val>=800:
        GPIO.output(LEDPIN, True)
        time.sleep(0.5)
    else:
        GPIO.output(LEDPIN, False)
    time.sleep(1)#Delay 1 second
```

Java program

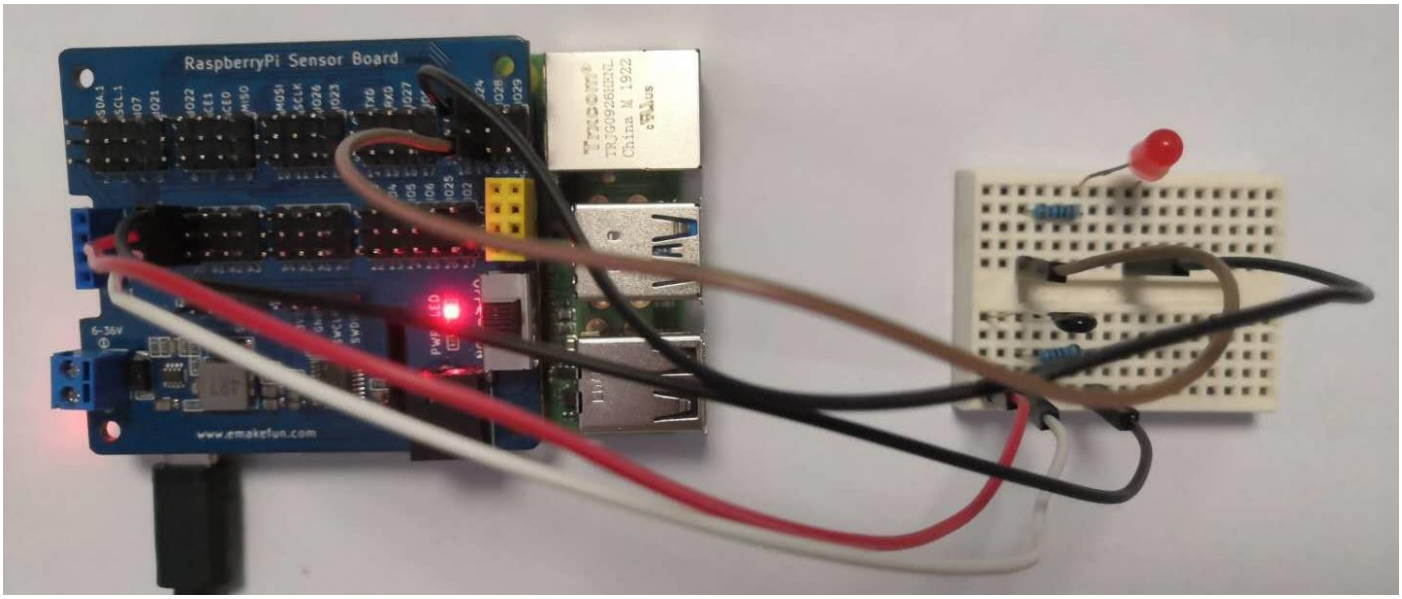
```
import com.pi4j.wiringpi.I2C;
import com.pi4j.wiringpi.Gpio;

public class Thermistor {
    static int Buzzer = 4, led_pin = 5, value = 0, fd;

    static {
        Gpio.wiringPiSetup();
        fd = I2C.wiringPiI2CSetup(0x04);
        Gpio.pinMode(led_pin, Gpio.OUTPUT);
    }

    public static void main(String[] args){
        for ( ; ;){
            value =  I2C.wiringPiI2CReadReg16(fd, 0x10);
            if(value > 200) {
                Gpio.digitalWrite(led_pin, Gpio.HIGH);
                Gpio.digitalWrite(Buzzer, Gpio.HIGH);
                Gpio.delay(1000);
            } else {
                Gpio.digitalWrite(led_pin, Gpio.LOW);
                Gpio.digitalWrite(Buzzer, Gpio.LOW);
            }
        }
    }
}
```

Experiment Result



After the program runs, first observe the voltage value printed by the serial port, and then use a flame or a medium with a higher temperature to approach the thermistor to observe the voltage change printed by the serial port and the brightness of the LED