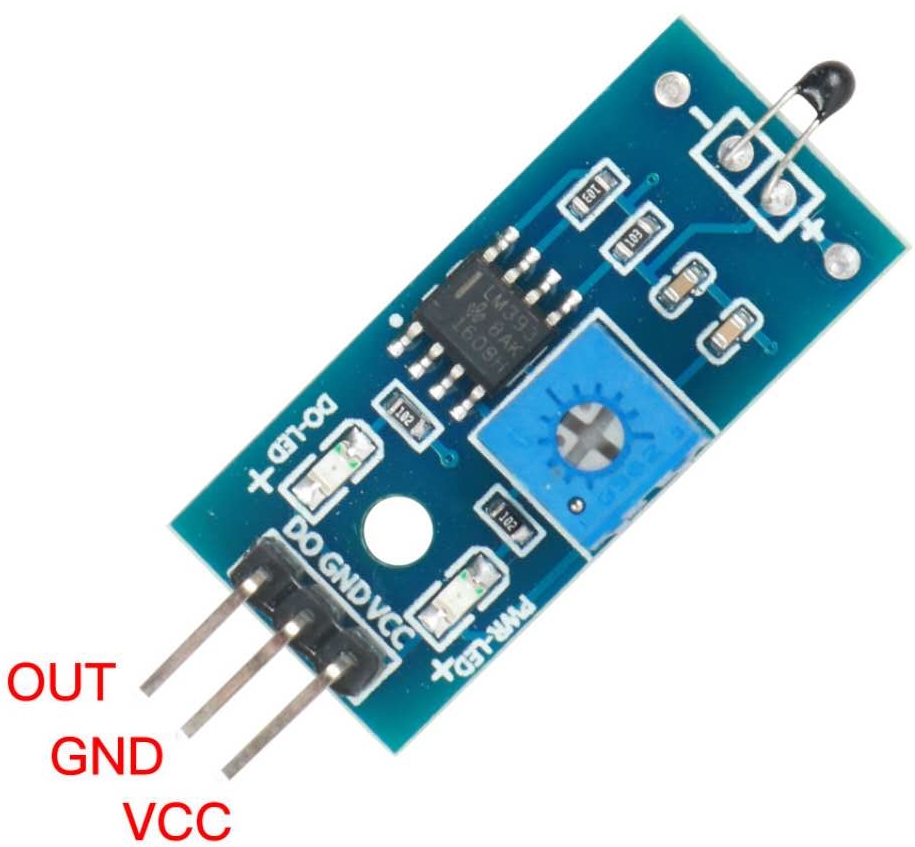
# Temperature Alarm Experiment

## Introduction

Thermistor is a type of the sensitive component which is divided into the positive temperature coefficient thermistor (PTC) and the negative temperature coefficient thermistor (NTC) according to the temperature coefficient. The typical characteristic of thermistor is that it is sensitive to temperature and exhibits different resistance values at different temperatures.The positive temperature coefficient thermistor (PTC) has a higher resistance value while the temperature is higher and the negative temperature coefficient thermistor (NTC) has a lower resistance value while the temperature is higher. They are both semiconductor devices.



Thermistor

## Experimental Purpose

* Read the resistance value of the thermistor at different temperatures
* Use the thermistor to control the on and off of the LED.

## Experimental Principle

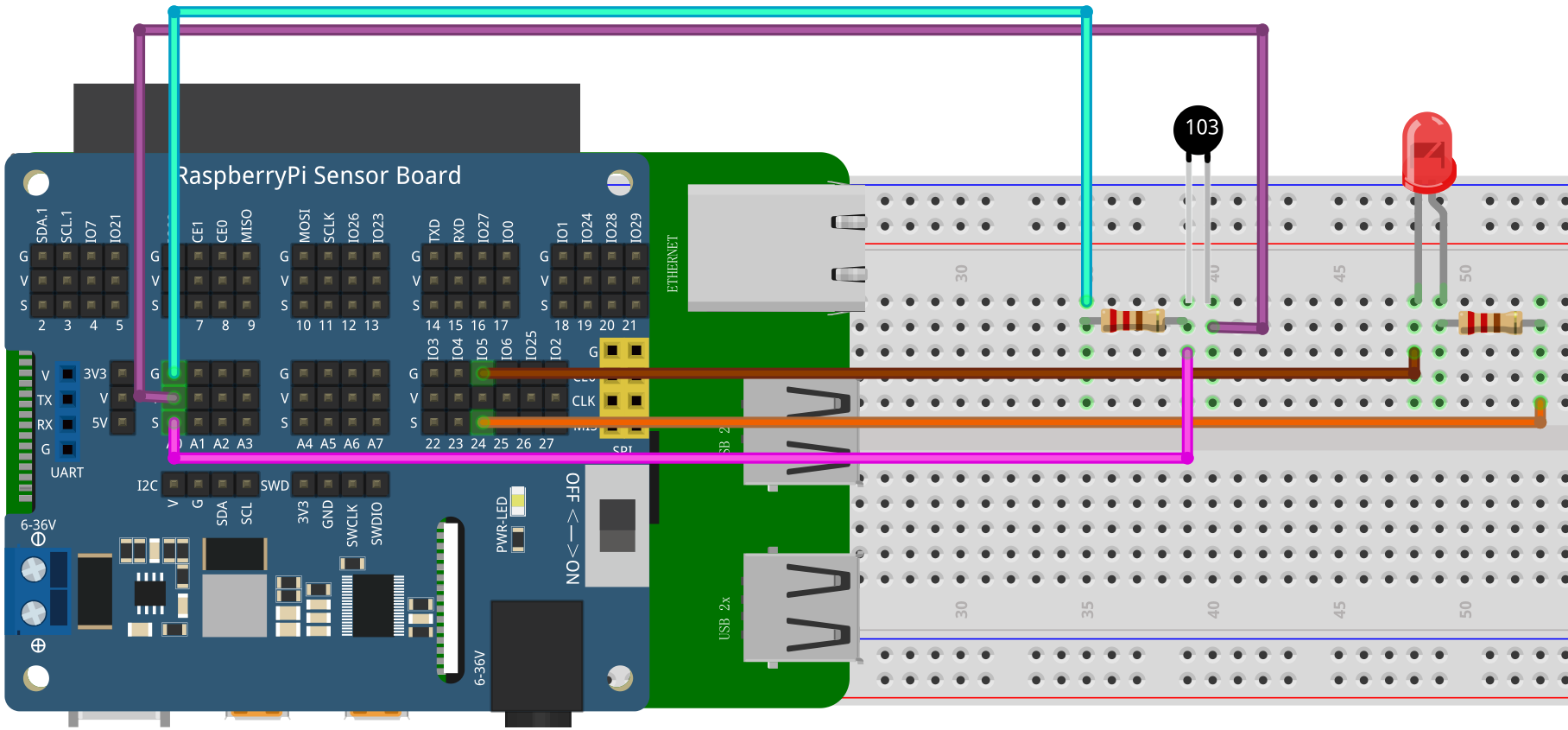
In this experiment, the LED and thermistor are connected in series and a 10K pull-up resistor is connected to one end of the thermistor. Read the thermistor voltage that changes with temperature through the analog port and print these parameters through the serial monitor. Observe the changes of the LED brightness.

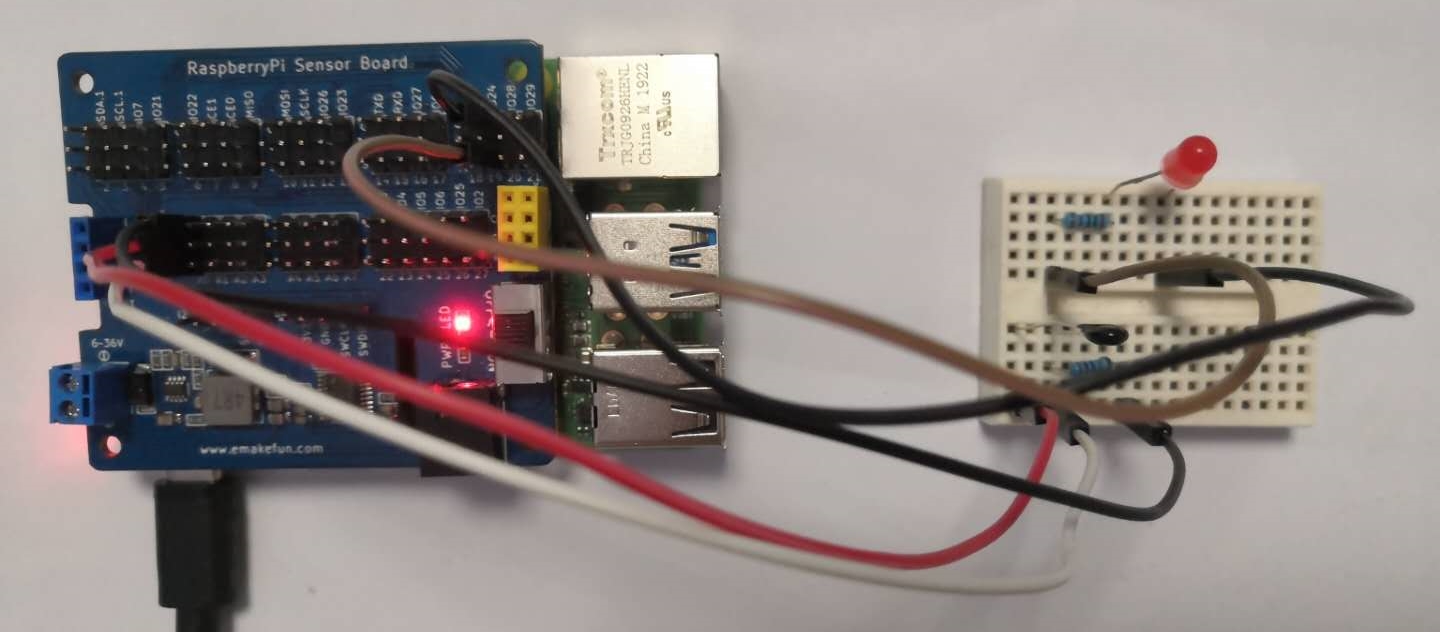
## Component List

* Raspberry Pi main board
* Raspberry Pi expansion board
* Breadboard
* Cable
* Thermistor \*1
* Red LED \*1
* 10kΩ Resistor \* 1
* 220Ω Resistor \* 1
* Several jumper wires

## Wiring

|  |  |
| --- | --- |
| Raspberry Pi | Thermistor |
| 5V | 1 |
| GND | 3 |
| A0 | 2 |
| Raspberry Pi | 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 is connected 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**);** //delay 1s

**}**

**}**

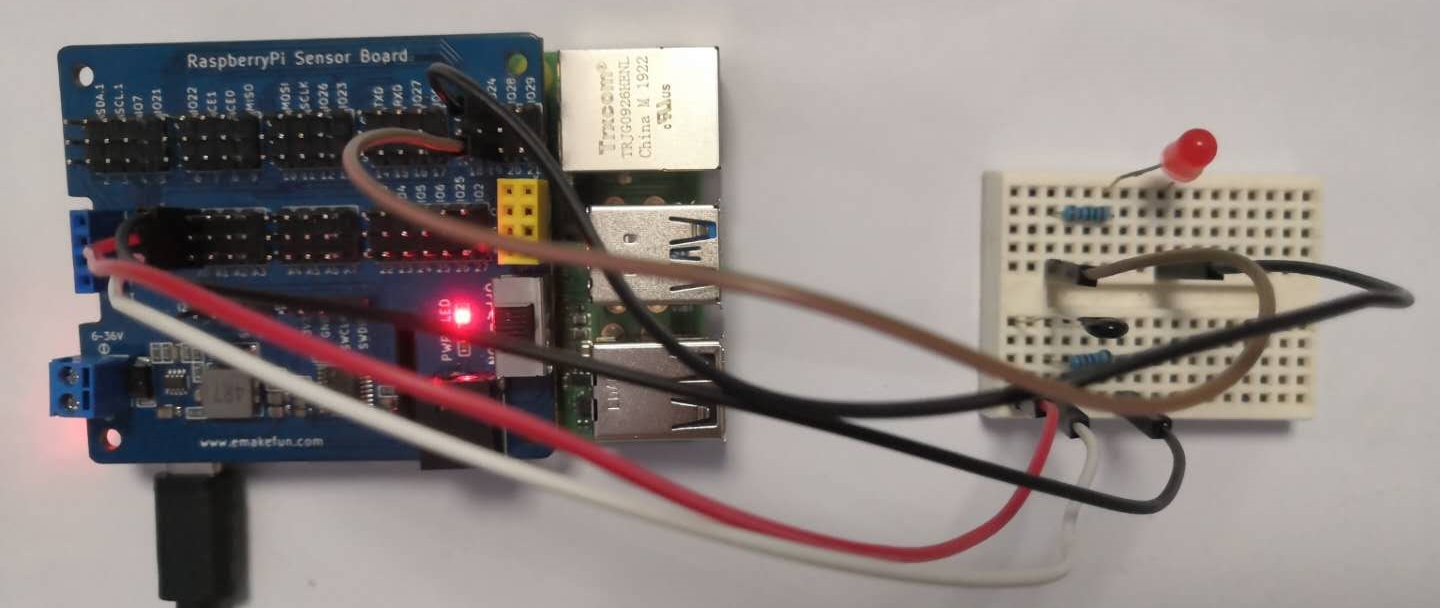
## Python program

|  |
| --- |
| 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**);**  **}**  **}**  **}**  **}** |

## Experimental Effect



After the program runs, we observe the voltage value printed on the serial port, then use a flame or a higher temperature medium to approach the thermistor and finally observe the voltage change and the brightness of the LED.