

20.Thermometer

Introduction

The resistance of thermistor will be changed with temperature change. in this lesson, we can make a thermometer according to this feature.

Hardware Required

- ✓ 1 * Raspberry Pi
- ✓ 1 * T-Extension Board
- ✓ 1 * PCF8591
- ✓ 1 * 40-pin Cable
- ✓ Several Jumper Wires
- ✓ 1 * Breadboard

Principle

Thermistor

Thermistor is a temperature sensitive resistor. When the temperature changes, resistance of thermistor will change. With this feature, we can use thermistor to detect temperature intensity. Thermistor and symbol are as follows.



The relationship between resistance value and temperature of thermistor is:

$$R_t = R \cdot \text{EXP} [B \cdot (1/T_2 - 1/T_1)]$$

Where:

R_t is the thermistor resistance under T₂ temperature;

R is in the nominal resistance of thermistor under T₁ temperature;

EXP[n] is nth power of E;

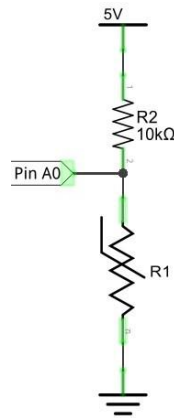
B is for thermal index;

T₁, T₂ is Kelvin temperature (absolute temperature). Kelvin temperature=273.15+celsius

temperature.

Parameters of the thermistor we use is: $B=3950$, $R=10k$, $T_1=25$.

The circuit connection method of the thermistor is similar to photoresistor, like the following method:



We can use the value measured by ADC converter to obtain resistance value of thermistor, and then can use the formula to obtain the temperature value.

Consequently, the temperature formula can be concluded:

$$T_2 = 1 / (1/T_1 + \ln(R_t/R)/B)$$

The PCF8591 four modules red short-circuit cap instructions

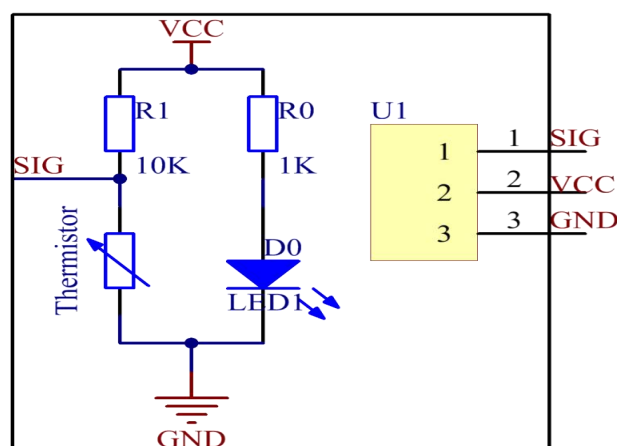
Module a total of three red short-circuit cap, respectively, as follows:

P4 connected to P4 short-circuit cap, select the thermistor access circuit

P5 connected to P5 short-circuit cap, select the photosensitive resistor access circuit

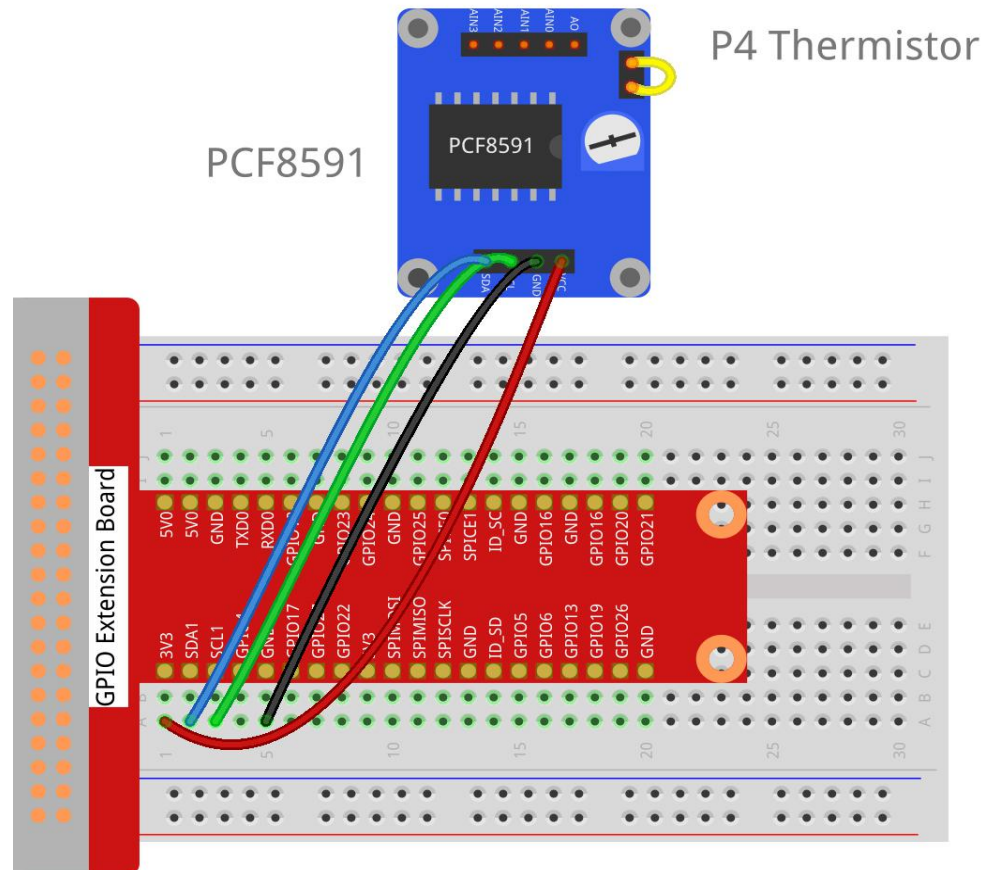
P6 connected to P6 short-circuit cap, select 0-5V adjustable voltage access circuit

Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.



For C Language Users

Step 2: Get into the folder of the code.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/C/20.Thermometer
```

Step 3: Compile the code.

```
gcc 20.Thermometer.c -o Thermometer.out -lwiringPi
```

Step 4: Run the executable file .

```
sudo ./Thermometer.out
```

Code

```
#include <stdio.h>
#include <wiringPi.h>
```

```

#include <pcf8591.h>

#include <math.h>

#define PCF 120

#define DOpin 0

void Print(int x)
{
    switch(x)
    {
        case 0:
            printf("\n*****\n" );
            printf( " * Too Hot! *\n" );
            printf( "*****\n\n");

            break;

        case 1:
            printf("\n*****\n" );
            printf( " * Better~ *\n" );
            printf( "*****\n\n");

            break;

        default:
            printf("\n*****\n" );
            printf( " * Print value error. *\n" );
            printf( "*****\n\n");

            break;
    }
}

int main()
{
    unsigned char analogVal;

    double Vr, Rt, temp;

    int tmp, status;

```

```

if(wiringPiSetup() == -1){
    printf("setup wiringPi failed !");
    return 1;
}

// Setup pcf8591 on base pin 120, and address 0x48

pcf8591Setup(PCF, 0x48);
pinMode(DOPin, INPUT);
status = 0;
while(1) // loop forever
{
    printf("loop");

    analogVal = analogRead(PCF + 0);
    Vr = 5 * (double)(analogVal) / 255;
    Rt = 10000 * (double)(Vr) / (5 - (double)(Vr));
    temp = 1 / (((log(Rt/10000)) / 3950)+(1 / (273.15 + 25)));
    temp = temp - 273.15;
    printf("Current temperature : %lf\n", temp);

    // For a threshold, uncomment one of the code for
    // which module you use. DONOT UNCOMMENT BOTH!
    //-----

    // 1. For Analog Temperature module(with DO)
    tmp = digitalRead(DOPin);

    // 2. For Thermister module(with sig pin)
    // if (temp > 33) tmp = 0;
    // else if (temp < 31) tmp = 1;
    //-----

    if (tmp != status)
    {
        Print(tmp);
        status = tmp;
    }
}

```

```
    }

    delay (200);

}

return 0;

}
```

For Python Language Users

Step 2: Get into the folder of the code.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/Python
```

Step 3: Run the executable file.

```
sudo python3 20.Thermometer.py
```

As the code runs, the stepper motor first turns clockwise for a round then stop for a second, after that, it rotates anticlockwise for a round; subsequently, the motor stops for a second. This series of actions will be executed repeatedly.

Code

```
#!/usr/bin/env python3

import PCF8591 as ADC
import RPi.GPIO as GPIO
import time
import math

DO = 17

GPIO.setmode(GPIO.BCM)

def setup():
    ADC.setup(0x48)
    GPIO.setup(DO, GPIO.IN)

def Print(x):
    if x == 1:
        print ("
```

```

    print ('*****')

    print (* Better~ *)

    print ('*****')

    print ("

if x == 0:

    print ("

    print ('*****')

    print (* Too Hot! *)

    print ('*****')

    print ("

def loop():

    status = 1

    tmp = 1

    while True:

        analogVal = ADC.read(0)

        Vr = 5 * float(analogVal) / 255

        Rt = 10000 * Vr / (5 - Vr)

        temp = 1/(((math.log(Rt / 10000)) / 3950) + (1 / (273.15+25)))

        temp = temp - 273.15

        print ('temperature = ', temp, 'C')

        # For a threshold, uncomment one of the code for

        # which module you use. DONOT UNCOMMENT BOTH!

        #####

        # 1. For Analog Temperature module(with DO)

        tmp = GPIO.input(DO)

        #

        # 2. For Thermister module(with sig pin)

        #if temp > 33:

        #    tmp = 0

        #elif temp < 31:

```

```
# tmp = 1

#####

if tmp != status:

    Print(tmp)

    status = tmp

    time.sleep(0.2)
if __name__ == '__main__':

    try:

        setup()

        loop()

    except KeyboardInterrupt:

        pass
```

Phenomenon Picture

