Analog temperature sensor

Overview

This course will use the Raspberry Pi to acquire the signal of the analog temperature sensor, which control the on and off of the LED lamp.

Experimental Materials

RaspberryPi \*1

Breadboard \*1

analogTempSensor \*1

ADC0832 ` \*1

Led \*1

Resistor(330Ω) x1

Dupont Line

Ready to work

1. Install python interpreter in your Raspberry Pi system

2. Install the RPi.GPIO library in your Raspberry Pi system

3. Install the wiringPi library in your Raspberry Pi system

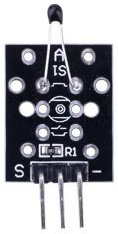
Refer to the attached "Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System" for details.

product description

1. Introduction:

● Function: thermistor, is a non-linear resistor that is sensitive to temperature response, and the resistance change with temperature. It is usually made of single crystal or polycrystalline semiconductor materials. This resistor is with a series of special electrical properties, which basic feature is that the resistance varies greatly with temperature, the resistance and volt-ampere curve are nonlinear relationship.

● Application: Thermistor is high temperature sensitive, small thermal inertia, long life, small size, simple structure, and various shapes and structures. Therefore, with the development of industrial, agricultural production, science and technology, this element is widely used, such as temperature measurement, temperature control, temperature compensation, liquid level measurement, pressure measurement, fire alarm, meteorological detection, switch circuit, overload protection, suppression voltage Fluctuating, time delay, amplitude stability, automatic gain adjustment, laser and microwave power measurements, and more.



1. Characteristic parameters

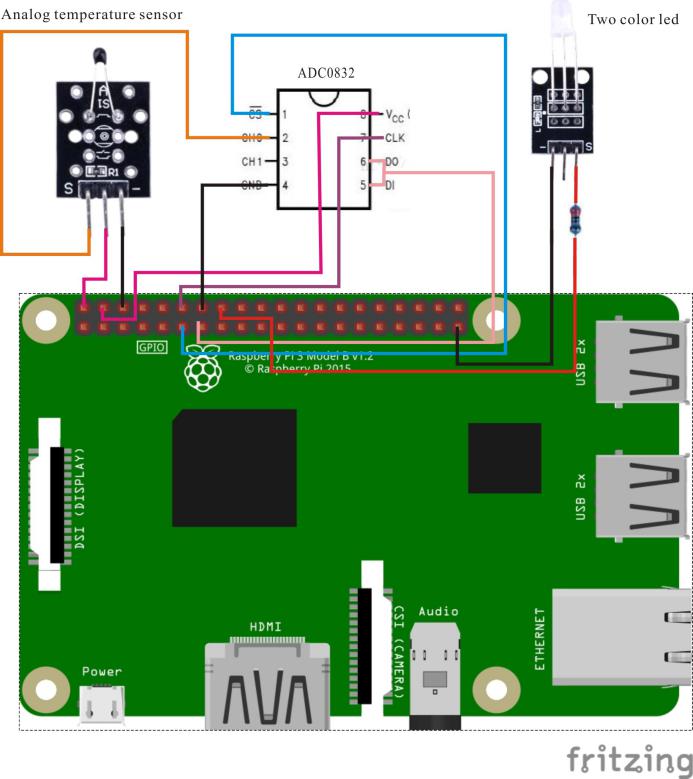
◆Analog signal output

◆Temperature range: - 55 ~ +125°

◆Temperature measurement accuracy: ±0.5°C

◆Working voltage: DC 5V

1. Wiring diagram



Sample code

1. python code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

import math

LedPin = 16

threshold = 200

def init():

ADC0832.setup()

GPIO.setmode(GPIO.BOARD)

GPIO.setup(LedPin, GPIO.OUT)

GPIO.output(LedPin, GPIO.LOW)

def loop():

while True:

analogVal = ADC0832.getResult(0)

print 'analogVal = %d' % analogVal

if(analogVal < threshold):

GPIO.output(LedPin, True)

else:

GPIO.output(LedPin, False)

time.sleep(0.2)

if \_\_name\_\_ == '\_\_main\_\_':

init()

try:

loop()

except KeyboardInterrupt:

ADC0832.destroy()

print 'The end !'

1. C code

/\*

\* compile with -lm for math library

\* gcc analogTempSensor.c -lwiringPi -lm

\*/

#include <wiringPi.h>

#include <stdio.h>

#include <math.h>

typedef unsigned char uchar;

typedef unsigned int uint;

#define ADC\_CS 0

#define ADC\_CLK 1

#define ADC\_DIO 2

#define LedPin 4

#define threshold 200

uchar get\_ADC\_Result(uint channel)

{

uchar i;

uchar dat1=0, dat2=0;

digitalWrite(ADC\_CS, 0);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,channel); delayMicroseconds(2);

digitalWrite(ADC\_CLK,1);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0);

digitalWrite(ADC\_DIO,1); delayMicroseconds(2);

for(i=0;i<8;i++)

{

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

pinMode(ADC\_DIO, INPUT);

dat1=dat1<<1 | digitalRead(ADC\_DIO);

}

for(i=0;i<8;i++)

{

dat2 = dat2 | ((uchar)(digitalRead(ADC\_DIO))<<i);

digitalWrite(ADC\_CLK,1); delayMicroseconds(2);

digitalWrite(ADC\_CLK,0); delayMicroseconds(2);

}

digitalWrite(ADC\_CS,1);

return(dat1==dat2) ? dat1 : 0;

}

int main(void)

{

uchar analogVal;

double Vr, Rt, temp;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !\n");

return -1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(LedPin, OUTPUT);

while(1)

{

pinMode(ADC\_DIO, OUTPUT);

analogVal = get\_ADC\_Result(0);

printf("analogVal = %d.\n", analogVal);

if(analogVal < threshold)

{

digitalWrite(LedPin, HIGH);

}

else

{

digitalWrite(LedPin, LOW);

}

delay(1000);

}

return 0;

}

Experimental phenomena

When the analog temperature sensor's analog output value meets the threshold condition, the LED will light on; when the threshold condition is not satisfied, the LED light goes out.