Digital temperature sensor

Overview

We will use the Raspberry Pi to capture the value of the digital temperature sensor and print it to the command line interface of the Raspberry Pi system after ADC conversion.

Material Needed

RaspberryPi \*1

Breadboard \*1

Digital temperature sensor \*1

ADC0832 \*1

Dupont Line

Preparatory 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

See the attached “Installing a Python Interpreter and Corresponding Libraries in a Raspberry Pi System” for details.

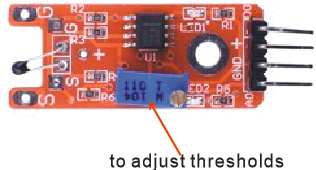
Product Description

Brief Introduction

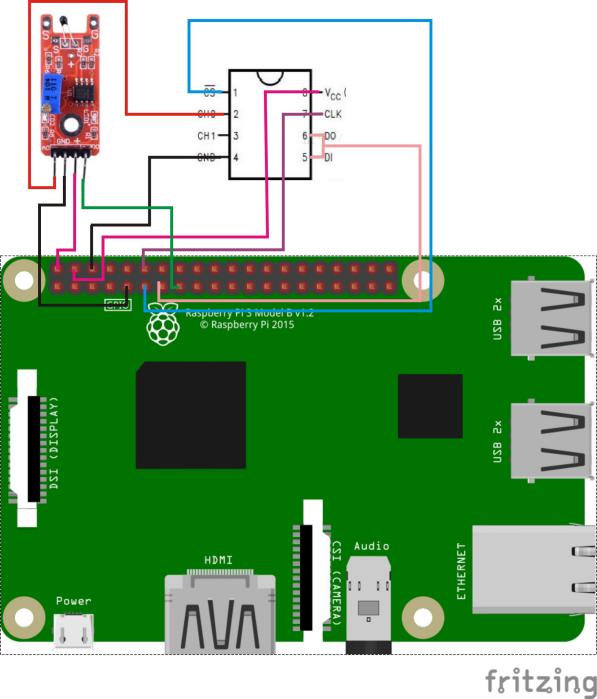
● Function: thermistor thermistor is a non-linear resistor that is sensitive to temperature response and whose resistance will change with temperature. It is usually made of single crystal, polycrystalline semiconductor materials. This resistor has a series of special electrical properties. The basic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

● Application: Thermistor is sensitive to high temperature, small thermal inertia, long life, small size, simple structure, and various shapes and structures. Therefore, with the development of industrial and 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.

In this experiment, the analog voltage value of the pin AO is converted into a digital quantity by the ADC and then output to the Raspberry Pi. The digital level of the pin DO is input to the Raspberry Pi when the level of the DO pin is detected. When level flipping occurs, the Raspberry Pi will read the value of the pin A0 via the ADC input Raspberry Pi and print the readout value to the Raspberry Pi command line interface.



Wiring diagram



Sample Code

1. Python Code

#!/usr/bin/env python

import RPi.GPIO as GPIO

import ADC0832

import time

TempSensor\_DO\_PIN = 15

def init():

GPIO.setmode(GPIO.BOARD)

GPIO.setup(TempSensor\_DO\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

ADC0832.setup()

def loop():

while True:

global digitalVal

digitalVal = GPIO.input(TempSensor\_DO\_PIN)

if(digitalVal == 1):

print 'DO is %d' % digitalVal

print "Temprature alarm ..."

print 'Current analog value is %d'% ADC0832.getResult(0)

time.sleep(0.2)

else:

pass

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

init()

try:

loop()

except KeyboardInterrupt:

ADC0832.destroy()

print 'The end !'

2. C Code

#include <wiringPi.h>

#include <stdio.h>

#include <string.h>

#include <errno.h>

#include <stdlib.h>

#define ADC\_CS 0

#define ADC\_CLK 1

#define ADC\_DIO 2

#define TempSensor\_DO\_Pin 3

typedef unsigned char uchar;

typedef unsigned int uint;

uchar get\_ADC\_Result(void)

{

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,0); 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);

pinMode(ADC\_DIO, OUTPUT);

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

}

int main(void)

{

uchar digitalVal = 1;

uchar analogVal = 0;

if(wiringPiSetup() == -1)

{

printf("setup wiringPi failed !");

return -1;

}

pinMode(ADC\_CS, OUTPUT);

pinMode(ADC\_CLK, OUTPUT);

pinMode(TempSensor\_DO\_Pin, INPUT);

pullUpDnControl(TempSensor\_DO\_Pin, PUD\_UP);

while(1)

{

if((digitalVal = digitalRead(TempSensor\_DO\_Pin)))

{

printf("Do is %d.\n", digitalVal);

printf("Temprature alarm...");

printf("Current analog value is %d.\n", get\_ADC\_Result());

delay(200);

}

else

{

;

}

}

return 0;

}

Experimental phenomena

When the digital temperature sensor detects a temperature change, if the sensor's analog voltage satisfies the level-reversal threshold condition, the Raspberry Pi will read the value of the pin A0 through the ADC input Raspberry Pi and will read it. The value is printed to the command line interface of the Raspberry Pi.