**PT100 là gì?** PT100 là loại cảm biến điện trở, sự thay đổi về nhiệt độ dẫn đến sự thay đổi về điện trở (khác với LM35). Do vậy, cách kết nối cũng hoàn toàn khác nhau. Bảng tham số nhiệt độ và điện trở như hình bên dưới.

PT-100 có 3 loại: 2 dây, 3 dây và 4 dây. Cụ thể:

* Loại 2 dây: chỉ có 1 đầu nguồn + (đỏ) và 1 đầu đất (trắng).
* Loại 3 dây: 1 đầu nguồn + (đỏ) và 2 dây đất (trắng). Nếu muốn tận dụng điện trở tổng trên 2 dây trắng (chập 2 đây làm 1).
* Loại 4 dây tương tự như loại 3 dây.

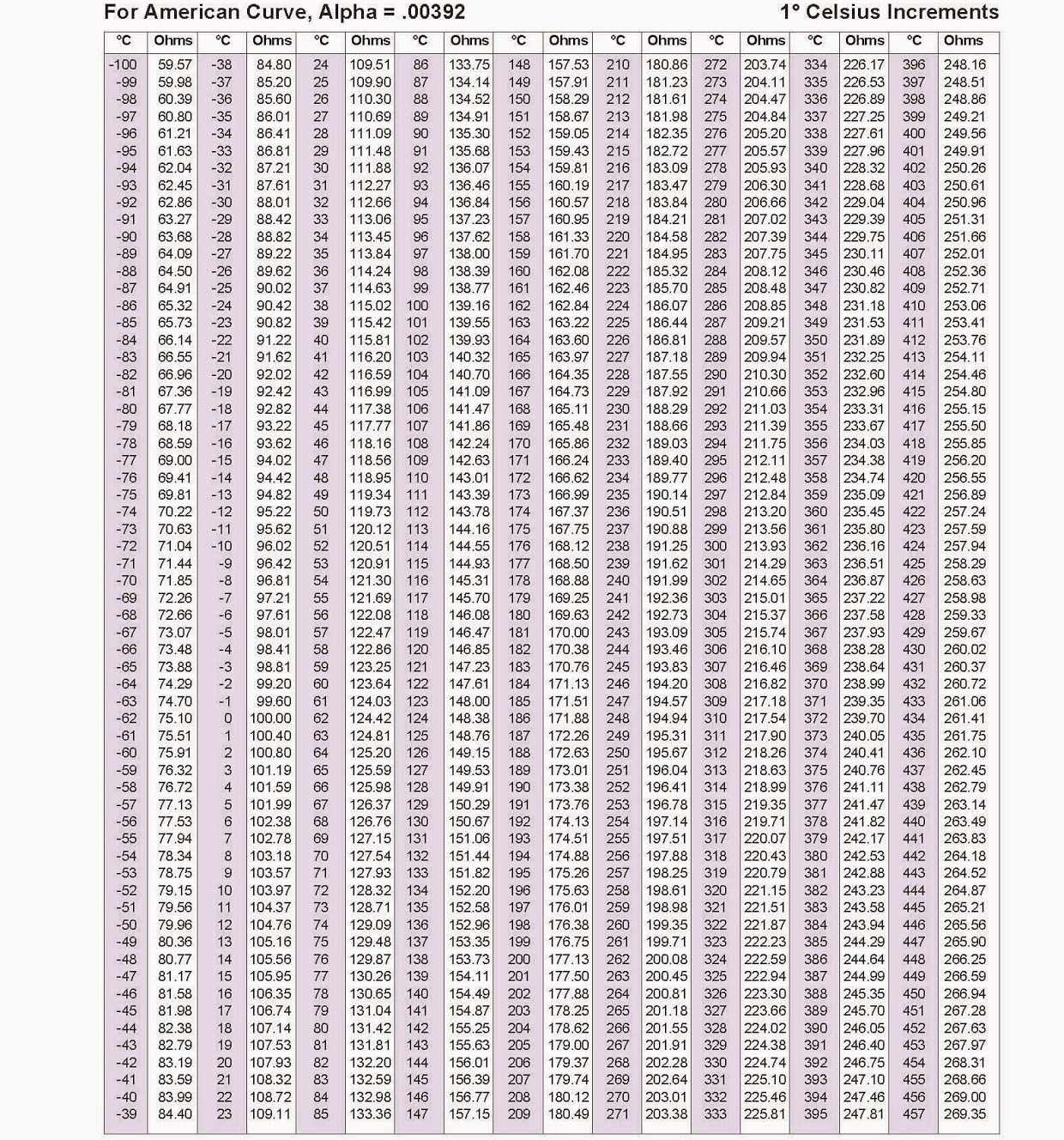


Figure 1-Bảng tham số điện trở và nhiệt độ

Cách ghép nối PT-100 như sơ đồ dưới đây. Khi nhiệt độ thay đổi – điện trở thay đổi. Trong khi đó, ADC chỉ có thể nhận biết sự thay đổi về mặt điện áp, do vậy không thể đưa tín hiệu đầu vào của PT-100 vào chân ADC của Arduino được mà cần nối đầu dò PT-100 với tải phụ để sinh ra điện áp tương ứng.

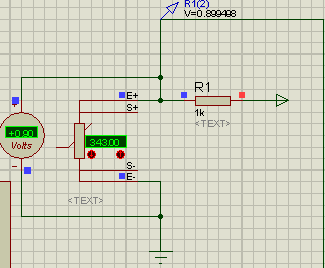


Figure 2-Sơ đồ mạch chia áp

Sơ đồ mạch điện như trên được gọi là mạch chia điện áp, công thức chia áp được tính như sau:

V_{out} = \dfrac{V_{cc} \times R_{TD}}{R_{TD} + R}

Với:

RTD là điện trở của PT-100 và R là điện trở được sử dụng để chia áp.

Vcc: nguồn cấp (+5V), Vout là nguồn ra của PT-100

2 chú ý khi lựa chọn giá trị cho R:

1. R đủ lớn để chịu nhiệt cho chính nó, vì khi PT-100 có R tối thiểu toàn bộ điện áp Vcc đều rơi trên R.
2. R đủ bé để không gây ra nhiễu cho cảm biến.

**Tính toán giá trị điện áp đầu ra**

Giả sử cảm biến với giới hạn nhiệt độ là: 00C đến 1000C. Giá trị R là 1KΩ, điện trở nội của PT-100 khi nhiệt độ 00C là RTD=100Ω và khi ở nhiệt độ 1000C là RTD=138.5Ω.

Từ công thức:

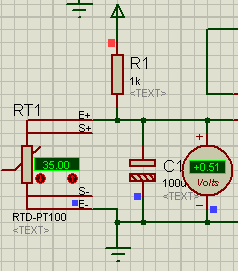
V_{out} = \dfrac{V_{cc} \times R_{TD}}{R_{TD} + R}

Figure 3-Bộ lọc nhiễu cho PT-100

Như vậy, ngưỡng điện áp của đầu ra: 0.4545V - 0.6083V

**Bộ lọc thông thấp:**

Do PT-100 có đầu dò bằng kim loại dài và dây dẫn khá dài (1m), vô tình tạo ra một ăng ten thu sóng (nhiễu) gây rối loạn cho việc biến đổi. Do vậy, một giải pháp được đưa ra là sử dụng tụ 1 chiều có dung lượng 100μF, mắc song song với RTD để lọc nhiễu (Figure 3-Bộ lọc nhiễu cho PT-100)

**Mạch khuếch đại điện áp**

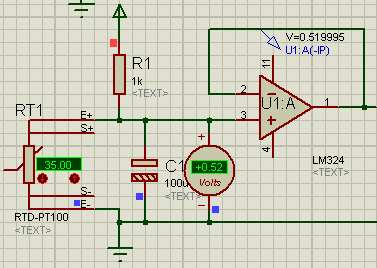
Có 2 lý do để sử dụng mạch khuếch đại điện áp:

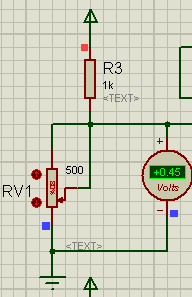
Figure 4-Khuếch đại thuật toán

1. Tránh hiệu ứng tải (To avoid loading effect)
2. Đầu vào trở kháng cao (Higher input resistance)

Sử dụng bộ khuếch đại không đảo (tín hiệu đầu ra cùng pha với tín hiệu đầu vào) trước khi đưa vào đầu Analog của ADC trên Arduino(Figure 4-Khuếch đại thuật toán).

Bộ khuếch đại hoạt động với đầu vào đảo được nối với đầu ra của LM324, điều đó cho phép với mọi điện áp đầu ra của PT-100 đều được phản hồi ở đầu ra (2 điện kế ở 2 đầu vào và ra của LM324)

**Điện áp hằng:**

Nguồn điện áp không đổi được thiết kế bằng cách sử dụng mạng phân áp làm từ điện trở (R2) và biến trở (VR1), điều chỉnh biến trở sao cho điện áp đầu ra phải bằng điện áp ở nhiệt độ tối thiểu bạn sẽ đo (giả sử 00C, ở điện áp 00C từ PT100 = 0,4545V).

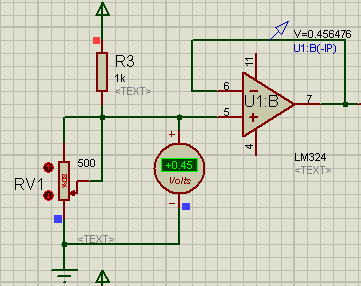
Điều chỉnh RV1 sao cho điện áp đầu ra = 0.454545

Giá trị RV1 = 80%\*500Ω = 400 ohm.

Điện áp hằng này được hiểu là điện áp tối thiểu của cảm biến PT-100 dự định sử dụng để đo nhiệt độ. Vì trong ứng dụng này, tác giả chỉ mong muốn sử dụng PT-100 đo nhiệt độ trong ngưỡng (0oC – 100oC) và điện áp tối thiểu được quy định từ đầu của mạch chia áp (Figure 2-Sơ đồ mạch chia áp) nên 2 giá trị điện áp này cần phải tương ứng với nhau.

Sau đó, điện áp hằng này lại được khuếch đại không đảo một lần nữa (để tránh trường hợp trở kháng cao và tải kéo theo sự thay đổi đột ngột điện áp trên mạch).

Hình 5

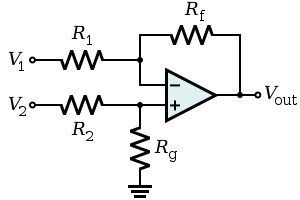


**Khuếch đại vi sai:**

Vì ADC trên Arduino có ngưỡng điện áp biến đổi là 0V – 5V (0-1023 mức), nhưng giá trị của điện áp ở cả 2 mạch chia điện áp và điện áp hằng có ngưỡng 0.4545V – 0.6083V. Không có bất kỳ mối liên hệ nào giữa 2 mức điện áp này. Ý tưởng như sau:

1. Kéo điện áp 0.4545V xuống bằng 0V (khi nhiệt độ là 00C)
2. Nâng mức điện áp 0.6083V lên bằng 5.00V (khi nhiệt độ là 1000C)

Bộ khuếch đại vi sai được ứng dụng để thực hiện ý tưởng này như sau:



**Vout** = Rf / R1 \* ( V2 - V1 )

Rf = Điện trở hồi tiếp (Feedback Resistor)

R1 = Điện trở vào (Input Resistor)

V1 = Điện áp đầu vào đảo được lấy từ bộ khuếch đại của PT-100

V2 = Điện áp đầu vào không đảo được lấy từ bộ điện áp hằng số

Trường hợp 1: Khi nhiệt độ là 00C, đầu ra của PT100 điện trở là 100Ω tương đương với điện áp là 0.4545V. Vì thế hệ số khuếch đại vi sai là 0

**Vout** = Rf / R1 \* (0.4545 – 0.4545) = 0

Trường họp 2: Khi nhiệt độ là 1000C, điện áp tương đương là 0.6083V, hệ số khuếch đại là:

**Vout** = Rf / R1 \* (0.6083-0.4545) = Rf / R1 \* 0.1538

Câu hỏi: Rf / R1 là bao nhiêu?

**Lựa chọn hệ số khuếch đại:**

Như đã đề cập về ADC của Arduino, điện áp tham chiếu nội của ADC 10 bit là 5V, có nghĩa là với giá trị 0 – 1023 thì giá trị điện áp tương xứng 0 – 5V. Hay độ phân giải là: 5V = 5/1024 = 4.88mV.

Trong khi đó PT100 có giá trị điện áp biến thiên là: 0.4545 V – 0.6083V, độ lệch là: 0.6083 – 0.4545 = 0.1538V.

Do vậy độ chia của PT100 được tính như sau: 0.1538V/4.88mV = 31.516 (~32) cho khoảng nhiệt độ 00C -1000C.

**Cách tính khác để tìm hệ số khuếch đại vi sai:**

Tại nhiệt độ là 1000C, ta mong muốn điện áp đạt giá trị là 5V, Vout = 5V

Trong đó:

V1 (điện áp sau khuếch đại của PT-100): 0.6083V

V2 (điện áp sau khuếch đại của mạch điện áp hằng): 0.4545V

Từ công thức:

**Vout** = Rf / R1 \* (0.6083-0.4545)

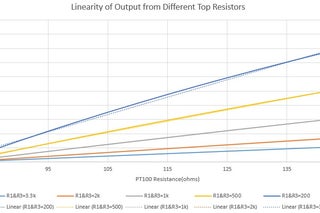
5 = Rf / R1 \* (0.1538)

Hay Rf / R1 = 32.5 (~32)

Để có hệ số 32 thì điện trở Rf lớn hơn R1 32 lần. Tuy nhiên, chú ý 2 điểm cơ bản sau:

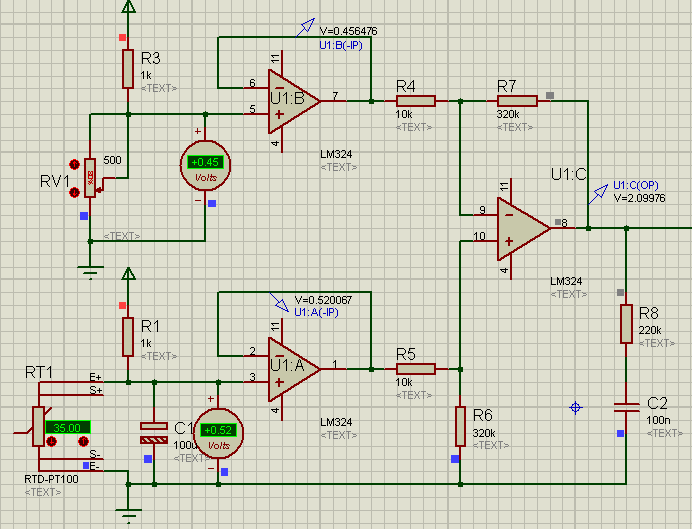
1. Hệ số phải xấp xỉ 32
2. Điện trở càng tăng thì công suất càng giảm

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Rf | Noise | Power |
| 1K | 32K | Low | High |
| 10K | 320K | Medium | Medium |
| 100K | 3200K | High | Low |



Mối liên hệ giữa các hệ số điện trở với công suất tiêu thị và hệ số nhiễu

**Thiết kế phần cứng:**



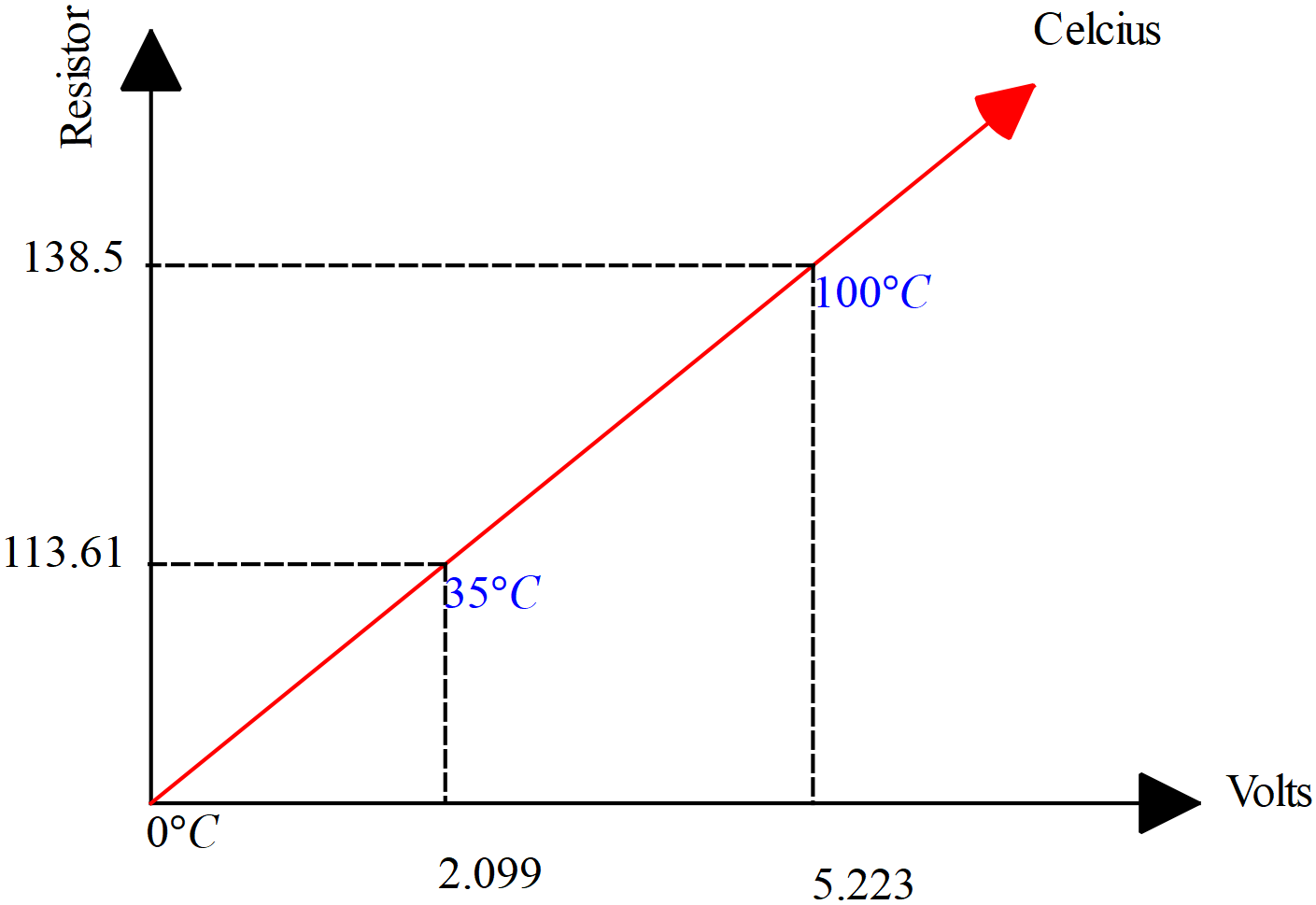
Hệ số khuếch đại khi V1 = 0.45476 (hơi khác so với 0.4545 được yêu cầu) và V2 = 0.520067V. Khi đó điện áp Vout = 2.09976 V [32\* (0.520067 - 0.45476)]

Một số trường hợp đã kiểm thử:

|  |  |  |  |
| --- | --- | --- | --- |
| T0 | V1 | V2 | Vout |
| 0 | 0.4545 | 0.4546 | ~0 |
| 29 | 0.5107 | 0.4546 | ~1.800 |
| 35 | 0.5200 | 0.4546 | ~2.099 |
| 50 | 0.5431 | 0.4546 | ~2.834 |
| 80 | 0.5843 | 0.4546 | ~4.154 |
| 100 | 0.6178 | 0.4546 | ~5.223 |

**Lập trình cho Arduino**

Tìm ra mối liên quan giữa hệ số biến đổi điện áp của ADC với điện áp đầu ra của PT-100



Ta có phương trình đường thẳng đi qua 2 điểm 350C và 1000C là:

Trong đó:

Và hằng sô C được xác định:

Tại nhiệt độ 1000C, hằng số C:

Như đã đề cập, ADC chỉ nhận giá trị đầu vào là điện áp trong ngưỡng 0 – 5V và quy đổi ở đầu ra 0 – 1023 trạng thái khác nhau. Nhưng, màn hình LCD cần hiển thị giá trị nhiệt độ chứ không phải giá trị là các số nguyên từ 0 – 1023.

Từ công thức quy đổi điện trở theo tài liệu hướng dẫn của PT-100:

Rt = Ro\*(1+alpha\*T)

Trong đó:

R0 là điện trở tại nhiệt độ 00C

Rt là điện trở tại thời điểm t

T là nhiệt độ tại thời điểm t

Alpha: hệ số chuyển đổi [0.00375 - 0,003928]

Từ công thức trên, giá trị T là giá trị cần được hiển thị lên màn hình LCD.

Quy trình thực hiện như sau:

Analog in = A0-A5

Volts = (analogRead(Vin)/1023.0)\*5.0

Cấu hình chân vào là A0

Chuyển giá trị đọc được sang điện áp

Rx = Volts\*m+C

Chuyển điện áp thành điện trở

temp1= (Rx/R0-1.0)/alpha

Chuyển điện trở thành nhiệt độ

calibration=0.3+(0.005\*temp1)

Sai số khi chuyển đổi

tempCelcius=Calibration+temp1

Nhiệt độ Celcius cuối cùng

tempFara = tempCelcius\*1.8+32

Nhiệt độ Fara cuối cùng

Ví dụ minh họa:

Tính toán giá trị của T (Celcius) khi nhiệt độ là 300C.

Theo lý thuyết Rt = Ro\*(1+alpha\*T)

Ta có: Rt = 111.88; R0 = 100; alpha = 0.00392

Xác định T (lý thuyết là 30)?

**Bước 1. Quy đổi giá trị điện áp**

Từ mô phỏng ta có: Vout = 1.85052 (cái này có thể tính được nếu lấy giá trị đọc được của ADC và áp dụng theo công thức: Volts = (analogRead(Vin)/1023.0)\*5.0

Suy luận: 5V= 1023 nên 1.85052V ~ 378

**Bước 2. Quy đổi giá trị điện áp sang điện trở**

Từ phương trình đường thẳng:

(tương ứng với lý thuyết tại 300C)

**Bước 3. Quy đổi giá trị điện trở sang nhiệt độ**

Từ phương trình:

**Bước 4. Tính sai số**

Từ phương trình:

**Bước 5. Nhiệt độ cuối cùng:**

T = 29.816 + 0.45 = 30.309

**Code:**

//LCD Pin name: 4=RS

//LCD Pin name: 5=RW

//LCD Pin name: 6=E

//String stringOne = String(13); // using a constant integer

//String stringOne = String(analogRead(0), DEC); // using an int and a base

//String stringOne = String(45, HEX); // using an int and a base (hexadecimal)

//String stringOne = String(255, BIN); // using an int and a base (binary)

//String stringOne = String(millis(), DEC); // using a long and a base

//String stringOne = String(5.698, 3); // using a float and the decimal places

int RS = 12;

int RW = 11;

int E = 13;

//khai bao bien

int i;

float val;

int analogPin = A0; //Doc du lieu cho ADC

float Volts; //voltage read by arduino

float tempC; //final temperature in degree celsius after calibration

float tempF; //final temperature in degree fahrenhiet after calibration

float temp1; //temperatuere before calibration

float calibration; //calibration

float Rx; //Resistance of PT100

// variables that required to convert voltage into resistance

float C = 95; //Constant of straight line (Y = mx + C)

float slope = 8.7; // Slope of straight line (Y = mx + C)

// variables that required to convert resistance into temperatures

float R0 = 100.0; //Resistance of minimum temperature to be measured (at 0 degree)

float alpha = 0.00375; // value of alpha from datasheet

//khai bao thu vien chan LCD

void setup()

{

DDRD=0xFF;

pinMode(RS,OUTPUT);

pinMode(RW,OUTPUT);

pinMode(E,OUTPUT);

}

void LCD\_Command(unsigned char lenh)

{

PORTD = lenh;

digitalWrite(RS,0);//RS=0, ghi lệnh

digitalWrite(RW,0);

digitalWrite(E,1);

delay(100);

digitalWrite(E,0);

}

void LCD\_Data(unsigned char dulieu) //Function to send display data to LCD

{

PORTD = dulieu;

digitalWrite(RS,1);//RS=1, ghi du lieu

digitalWrite(RW,0);

digitalWrite(E,1);

delay(100);

digitalWrite(E,0);

}

void LCD\_Init() //Function to prepare the LCD and get it ready

{

LCD\_Command(0x38); // for using 2 lines and 5X7 matrix of LCD

delay(10);

LCD\_Command(0x0C); // turn display ON, cursor blinking

delay(10);

LCD\_Command(0x01); //clear screen

delay(10);

LCD\_Command(0x80); // bring cursor to position 0 of line 1

delay(10);

}

void LCD\_ShiftLeft()

{

LCD\_Command(0x18);// Shift display entire to left

delay(10);

}

void LCD\_SecondRow()

{

LCD\_Command(0xC0);// Sencond Row begining

//delay(10);

}

void Display()

{

String S1="Ndo: ";

S1 = S1 + String(tempC, 2); //lam tron 2 chu so thap phan

int j=0;

while(S1[j] != '\0') // searching the null terminator in the sentence

{

LCD\_Data(S1[j]);

j++;

}

}

void loop()

{

val = analogRead(analogPin); // read the input pin

//Hien thi gia tri dien ap

// Bits to Voltage

Volts = (val/1024.0)\*5;

// Converting voltage to resistance

Rx = Volts\*slope+C; //y=mx+c

// Converting resistnace to temperature

temp1= (Rx/R0-1.0)/alpha; // from Rx = R0(1+alpha\*X)

calibration=0.3+(0.005\*temp1); //tolerance for class B PT100

tempC=temp1-calibration; // Final temperature in celsius

// conversion of celsius to fehrenheit

tempF = tempC\*1.8+32; //Final temperature in fehrenheit

LCD\_Init();

Display();

}

