**Cảm biến nhiệt độ LM35 và cách sử dụng nó trong môi trường Arduino**

1. **Giới thiệu, nội dung chính**

Cảm biến nhiệt độ LM35 là một loại *cảm biến tương tự* rất hay được ứng dụng trong các ứng dụng đo nhiệt độ thời gian thực. Vì nó hoạt động khá chính xác với sai số nhỏ, đồng thời với kích thước nhỏ và giá thành rẻ là một trong những ưu điểm của nó. Vì đây là cảm biến tương tự (analog sensor) nên ta có thể dễ dàng đọc được giá trị của nó bằng hàm [analogRead()](http://arduino.vn/reference/analogread). Nào, cùng nhau tìm hiểu thôi!

Qua bài viết này, mình hi vọng nó sẽ đem lại cho bạn một vài mẹo nhỏ để ứng dụng hàm [analogRead()](http://arduino.vn/reference/analogread) một cách nhuần nhuyển, và bật mí cho bạn cách sử dụng cảm biến LM35 - một cảm biến nhiệt độ rất hay và dễ sử dụng!

1. **Phần cứng**

* Arduino UNO
* Breadboard
* Dây cắm breadboard
* 1 cảm biến LM35

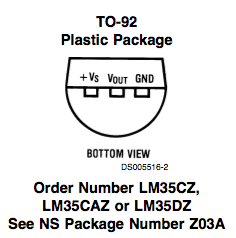
1. **Giới thiệu về cảm biến LM35**

**LM35 là một cảm biến nhiệt độ analog**

Nhiệt độ được xác định bằng cách đo hiệu điện thế ngõ ra của LM35.

→ Đơn vị nhiệt độ: °C.

→ Nhiệt độ thay đổi tuyến tính: 10mV/°C

[](http://machtudong.vn/sanpham/lm35-cam-bien-nhiet-do.html)  
***Sơ đồ chân của LM35***

**LM35 không cần phải canh chỉnh nhiệt độ khi sử dụng.**

Độ chính xác thực tế: 1/4°C ở nhiệt độ phòng và 3/4°C ngoài khoảng 2°C tới 150°C

LM35 có hiệu năng cao, công suất tiêu thụ là 60uA

**Cảm biến LM35 hoạt động bằng cách cho ra một giá trị hiệu điện thế nhất định tại chân Vout (chân giữa) ứng với mỗi mức nhiệt độ.**

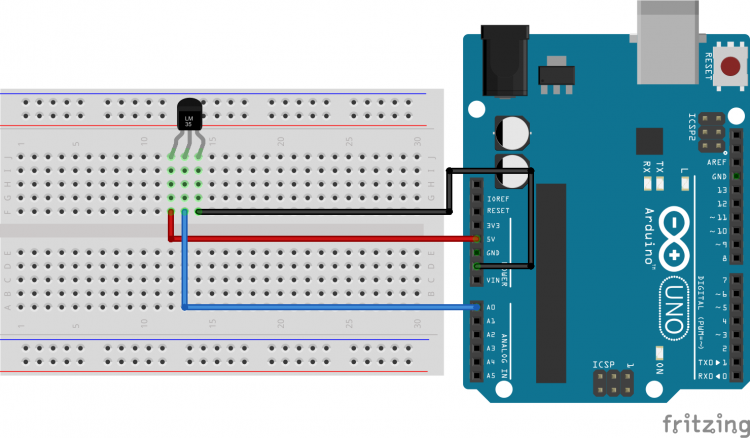
Như vậy, bằng cách đưa vào chân bên trái của cảm biến LM35 hiệu điện thế 5V, chân phải nối đất, đo hiệu điện thế ở chân giữa bằng các pin A0 trên arduino (giống y hệt cách đọc giá trị biến trở), bạn sẽ có được nhiệt độ (0-100ºC) bằng công thức:

1. float temperature = (5.0\*analogRead(A0)\*100.0/1024.0);

Với LM35, bạn có thể tự tạo cho mình mạch cảm biến nhiệt độ sử dụng LM35 và tự động ngắt điện khi nhiệt độ vượt ngưỡng tối đa, đóng điện khi nhiệt độ thấp hơn ngưỡng tối thiểu thông qua module rơ le...

**LM35 thay đổi nhiệt độ nhanh và chính xác.**

1. **Lắp mạch**

[](http://k3.arduino.vn/img/2014/08/23/0/731_8121-1408764231-0-lm35-bb.png)

1. **Lập trình**
2. int sensorPin = A0;// chân analog kết nối tới cảm biến LM35
4. void setup() {
5. Serial.begin(9600); //Khởi động Serial ở mức baudrate 9600
6. // Bạn không cần phải pinMode cho các chân analog trước khi dùng nó
7. }
9. void loop() {
10. //đọc giá trị từ cảm biến LM35
11. int reading = analogRead(sensorPin);
13. //tính ra giá trị hiệu điện thế (đơn vị Volt) từ giá trị cảm biến
14. float voltage = reading \* 5.0 / 1024.0;
16. // ở trên mình đã giới thiệu, cứ mỗi 10mV = 1 độ C.
17. // Vì vậy nếu biến voltage là biến lưu hiệu điện thế (đơn vị Volt)
18. // thì ta chỉ việc nhân voltage cho 100 là ra được nhiệt độ!
20. float temp = voltage \* 100.0;
22. Serial.println(temp);
23. /\*Mẹo:
24. Các bạn phaỉ khai báo phần thực cho toàn bộ các số thực nhé!
25. \*/
26. delay(1000);//đợi 1 giây cho lần đọc tiếp theo
27. }
28. **Đóng góp từ cộng đồng**

**Đọc nhiệt độ âm với cảm biến DS18D20**

Về đóng góp IC thì nó không khác gì con LM35 nhưng nó có khả năng đọc được nhiệt độ âm và độ sai số ít hơn.

1. #include < onewire.h >
2. #include < dallastemperature.h >
3. #include < liquidcrystal.h >
4. #define ONE\_WIRE\_BUS 3 //Chân cảm biến DS20B18 được nối với cổng digital D3 (sơ đồ đấu nối chân này thông qua trở 4,7K với chân Vcc 5V
6. OneWire oneWire(ONE\_WIRE\_BUS);
7. DallasTemperature sensors( & oneWire);
8. DeviceAddress cambienDS20B18;
10. // Khai báo chân xuất ra màn hình tinh thể lỏng lưu ý chân đấu nối
12. LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
13. void setup(void) {
14. sensors.begin();
15. lcd.begin(16, 2);
16. }
18. void loop() {
19. sensors.requestTemperatures(); // Đọc giá trị từ cảm biến
20. float nhietdoDS18B20 = sensors.getTempC(cambienDS20B18);
22. // Xuất thông tin cảm biến ra màn hình LCD
23. lcd.clear();
24. lcd.setCursor(0, 0);
25. lcd.print("Nhiet do: ");
26. lcd.setCursor(10, 0);
27. lcd.print(nhietdoDS18B20, 1);
28. lcd.write(223);
29. lcd.print("C");
31. }