

ESPRUINO

About

Video

Blog

Get it

Tutorials & Projects

Code Examples

Reference

Search

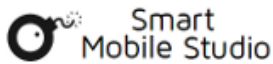
FAQ

Forum

Contact Us



Press

Sponsored by.



Like Share 146

Share:   

Or [follow us on](#)  or  for news and updates!

THERMISTORS

This Tutorial assumes that you are able to enter code in Espruino. If you can't do that yet, please follow the steps on the [Quick Start](#) page.

INTRODUCTION

One of the most useful parts of modern Microcontrollers is the analog input. On the STM32 chips that Espruino uses you have 12 bit analog inputs on some pins - and I'll show you how to use these to work out the temperature using a [Thermistor](#).

Thermistors are relatively cheap electronic components. They have 2 wires, and a resistance that changes depending on their Temperature. These are available as small components, but they are also used in virtually all cars in order to determine air pressure and water temperature - and when used in cars they are packaged up in to strong, weatherproof modules. I'll be using one of these that I got from a scrap heap.

There are several different types of thermistor, and their resistance varies in different ways. Most manufacturers provide tables of values (linking temperature to resistance) but you can also work out temperature from resistance using a formula called the [Steinhart-hart](#) equation.

To work out which type of thermistor you have, the first step is to measure the resistance of it, and guess what temperature it is currently at. Then simply search online for 'thermistor table' and figure out which thermistor type most closely matches (thermistor types are generally named by their resistance at 25 degrees C). It should be really obvious - mine measured 6k Ohms at around 20 degrees C - so it was rated as a 5k Ohm thermistor at 25 degrees C.

Once you know that, search online for the equation, for instance "5k thermistor equation". For mine, I got the steinhart equation and coefficients:

```
W = log(R)
T = 1 / (A + B*W + C*W3)
A: 1.2874E-03
B: 2.3573E-04
C: 9.5052E-08
```

Note that T is in Kelvin, so we have to subtract 273.15 from it to get degrees C!

WIRING UP

Disclaimer: This is not the most accurate way to wire a thermistor up - however it works, and is very simple!

Try and find a resistor that is roughly the same resistance as your thermistor's rating. I chose 5.6 kOhms as this is a standard value (and my thermistor is 5 kOhms).

Choose 3 pins next to each other, the middle one of the pins must be able to read analog values. Just type analogRead() in to Espruino to get a list of pins that support analog inputs. I'll be using an Olimexino, so am using D0,D1 and D2 - if you use different pins, change the names in the software below.

Then, connect the pins up as follows:

Pin Name	Connection
C0	One of the 2 thermistor wires
C1	The other thermistor wire, AND one of the resistor's wires
C2	The other resistor wire

We'll be making the first pin 0v, the third pin 3.3v (the chip's voltage), and reading the voltage off of the middle pin.

SOFTWARE

Now our wiring is done, to the software! First off, we'll make the two pins either side of the middle 0 and 3.3v volts respectively:

```
digitalWrite(C0, 0);
digitalWrite(C2, 0);
```

And we can read off the voltage:

```
analogRead(C1)
```

This should display something a bit like 0.543 (it is a value between 0 and 1)

We can use this to work out the resistance in Ohms. Because we have made a potential divider, the value we expect is $V = \frac{R_t}{R_t + R_r}$ (Where R_t is our thermistor's resistance, and R_r is our resistor's resistance). Fiddling around with the algebra gives $R_t = \frac{V R_r}{1 - V}$

```
= Rr*V / (1-V);
```

So now , let's put this all into a function:

```
function getTemp() {  
  digitalWrite(C0,0); // set voltage either side  
  digitalWrite(C2,1);  
  var val = analogRead(C1); // read voltage  
  var ohms = 5600*val/(1-val); // work out ohms  
  var A = 0.0012874; // Steinhart equation  
  var B = 0.00023573;  
  var C = 0.000000095052;  
  var W = Math.log(ohms);  
  var temp = 1 / (A + W * (B+C * W*W)) - 273.15;  
  return temp; // and return the temperature  
}
```

All you have to do now is type:

```
getTemp()
```

And the temperature in degrees C will be displayed!

So now , maybe you want to make a fan that comes on when it gets too hot. I won't cover wiring up a fan here, so let's just light an LED up.

All we'll do, is every second we will check the temperature, and if it is above 25 degrees C (it is England here...) we'll light the LED!

```
function checkTemp() {  
  var temp = getTemp();  
  digitalWrite(LED1, temp > 25);  
}  
setInterval(checkTemp, 1000);
```

Sorted!

EXTRA SOFTWARE

Maybe that's not enough... You might want to display the temperature on a display. For this, we'll use the LCD Driver code from this website.

Just go to the LCD Driver page, and copy in the first section of code. See the documentation on the page for how to wire up the LCD.

After that, initialise the LCD, and print some text to make sure it's working!

```
var lcd = new LCD(A4,A5,A0,A1,A2,A3);  
lcd.print("Hello World");
```

Now , if that works, we can print the temperature every second. Note that we're just overwriting the function that we made before:

```
function checkTemp() {  
  var temp = getTemp();  
  digitalWrite(LED1, temp > 25);  
  lcd.clear();  
  lcd.print("Temp is "+temp);  
}
```

But that's a bit small - so let's add big numbers. Go back to the LCD Driver page, and copy and paste the code immediately under the title 'Big Number Library'.

We need to initialise it:

```
lcd.bigInit();
```

And now we can modify our function to draw big characters. This time, we can only draw the numbers 0-9, so we need to use a bit of maths to extract these and then draw them one at a time. Finally, we just print a decimal point and a 'degrees C' mark.

```
function checkTemp() {  
  var temp = getTemp();  
  digitalWrite(LED1, temp > 25);
```

```
lcd.clear();  
lcd.bigNum(0, Math.floor(temp/10)%10);  
lcd.bigNum(3, Math.floor(temp)%10);  
lcd.bigNum(9, Math.floor(temp*10)%10);  
lcd.bigNum(12, Math.floor(temp*100)%10);  
lcd.setCursor(7,3);lcd.print("2");  
lcd.setCursor(16,0);lcd.print("'C");  
}
```

That's it!

Note, if you want to save this and have it start automatically, you need to initialise the LCD and set up the special characters needed for the large numbers. To do this, add:

```
function onInit() {  
  var lcd = new LCD(A4,A5,A0,A1,A2,A3);  
  lcd.bigInit();  
}
```

You can now type `save()`, and when the power goes off and back on, onInit will be called and the LCD will be initialised.

This page is auto-generated from [GitHub](#). If you see any mistakes or have suggestions, please [let us know](#).