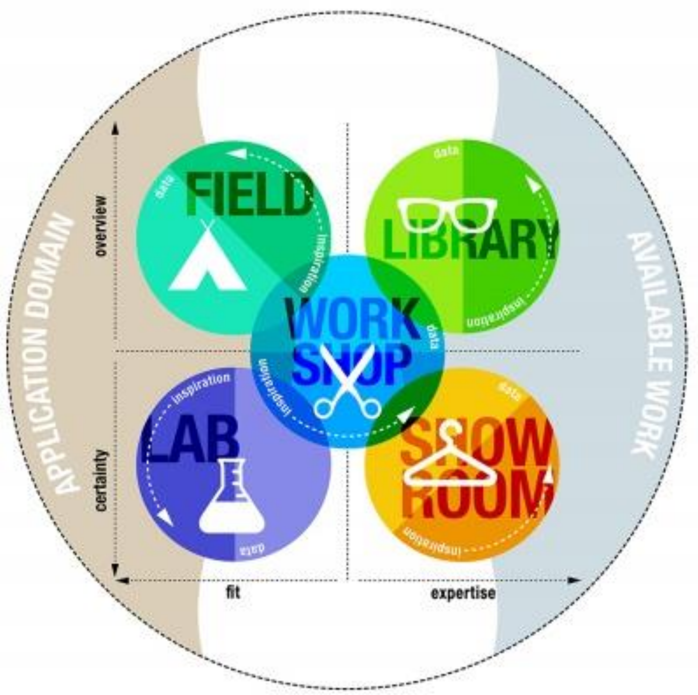
Applied Research Report



Topic: *THERMISTORS*

Author(s): *Johnson Domacasse.*

Team name: *group 7.*

Document date / version: *2023-03-13,* ***V1.0***

Document status: *concept.*

1 Contents

2 Abstract

A 3D printer makes use of high temperatures to shape an object into existence. These temperature levels can be dangerous and need to be monitored and controlled. That is why this research report is written. This document is written in order to get a better understanding of how we can utilise thermistors in order to receive relative, accurate and valuable data and then use this data for other purposes. This is done with research methods from the DOT framework. The thermistor has both pros and cons but it is one of those cases where the good outweighs the bad. As long as the sensor is connected properly and the software written for it is indeed maintainable. This article contains valuable information about the thermistor sensor and how it can be used in your own projects.

3 Introduction

Anyone can sit and wonder how exactly a thermometer can display the temperature in an area. But what they may not wonder is how this temperature is being measured and **how**. In this research report we are going to assess the measuring of temperature using a thermistor and how this can be done effectively.
A 3D-printer makes use of high temperature to "mold" an object of choice. The 3D-printer makes use of high temperatures in order to do this. These temperature can be dangerous and therefore must be monitored in order to keep a safe and controlled environment. Just like the thermometer, the temperature is displayed on a nice display. What we want to know is how we can measure this temperature using a thermistor. Which sensors are used within the printer to acquire this temperature? Is there a cheaper way to do this?

4 Research question

What is an efficient way to receive data from a thermistor and utilize this data?

How can we determine the temperature using the thermistors capabilities to measure resistance and transform this into data we can use.

1. How does a thermistor work?
2. What are the limitations of using a thermistor.
3. What is the interaction between the program and the thermistor.
4. How can we determine clear and relatively accurate temperature from a thermistor.

The acceptance criteria for this research question is if we can efficiently measure the temperature using a thermistor.

5 Research strategy

The (sub)questions from this research will be answered according to a few methods and strategies using the DOT framework.

The definition of done for this research is when all the questions are answered and we can efficiently measure the temperature using a thermistor.

For the main question, I plan to:

- Use “literature study (from library)” by finding and studying the datasheet of the thermistor and relevant application notes.
- Use “prototyping (from workshop)” to experiment with the technology and get some hands-on experience.

For (sub)question 1, I plan to:

- Use “component test (from lab)” to better understand how the thermistor functions and how we can manipulate its behaviour.
- Use “available product analysis (from library)” by scanning the internet to see if there are any available information on the thermistor.

For (sub)question 2, I plan to:

- Use “literature study (from library)” by finding and studying the datasheet of the thermistor and relevant application notes.
- Use “component test (from lab)” to better understand how the thermistor functions and how we can manipulate its behaviour.

For (sub)question 3, I plan to:

- Use “prototyping (from workshop)” to experiment with the technology and get some hands-on experience with both components and software.
- Use “component test (from lab)” to better understand how the thermistor functions and how we can manipulate its behaviour.
- Use “data analytics (from lab)” to analyse the acquired data and make changes according to this data.

For (sub)question 4, I plan to:

- Use “component test (from lab)” to better understand how the thermistor and other components function together.
- Use “literature study (from library)” by finding and studying the datasheet of the thermistor and the other components.
- Use “system test (from lab)” in order to test the combined sensors and document the result.

6 Research results

6.1 Main question.

For this question we need to dive a little deeper in the understanding of components. Components on their own work fine given the right circumstances. But when connected with each other it becomes rather different. You have to pay attention to both the hardware side of things as the software side. For now we assume that all of the hardware specification are met. In our case, the thermistor is connected using the provided datasheet. See datasheet reference. Both the fan and the thermistor can work fine on their own. How they interact with one another happens within the software. Within our code we must specify for example that If the temperature reaches a certain threshold, the fan will automatically turn on. While the fan doesn't reach the lower temperature threshold it will work. Meaning once it reaches the low point, the fan will turn off. Of course we don't want to spam the program so we implement the "Millis" function in order to get the measurements at a certain rate. Once you combine the code of the thermistor and the code of the fan, it will be a matter of reading outputs.

6.2 Sub question 1:

To put it simply, a thermistor is a type of semiconductor that react like a resistor that is sensitive to temperature in its environment. The term "thermistor" derives from "Thermally sensitive resistor". The relationship between the thermistor's temperature and its resistance is highly dependent on which materials it is made out of. The manufacturers of thermistors will typically a high degree of accuracy. Meaning you will have an accurate measurement. There are two types of thermistors. The "negative temperature coefficient" (NTC) thermistors and the "positive temperature coefficient". The way an NTC would work is if the temperature increases, the thermistors resistance would decrease. It is the opposite for the PTC. If a PTC's resistance increases, it means that the temperature has also increased. For this research, we will be using an NTC thermistor. We also connected the resistor into a circuit with an Arduino. We then add a simple program on the Arduino to read and print the results from the thermistor. This way we can see what happens when we interact with the thermistor. For example, if you add heat the value increases and the opposite when you apply cool air. This is based on information found on the internet and hands-on experimentation.

So a thermistor uses a built in resistor to determine the temperature of its surroundings.

6.3 sub question 2:

Thermistors, like any other component, have their advantages and disadvantages. Some advantages of the thermistor includes:

- The fact that they are highly accurate with an impressive range from± 0.05°C to ± 1.5°C.
- The thermistor is not prone to malfunction in harsh environments or where electrical noise is present.
- They are well suited for both controlled environments and harsh ones.

Some disadvantages of the thermistor include:

- That the thermistor is indeed highly accurate however it only over a limited temperature range that is within about 50°C of a base temperature.
- That the thermistor is a fragile piece of hardware and can be easily damaged if not properly handled.
- Not easily interchangeable. The process of replacing a thermistor is very difficult, although possible.

- Being non-linear. Meaning the temperature to resistance values plot o a graph as a curve rather than a straight line.

Based off of these pros and cons we can draw some conclusions about the limitations of the thermistor. It is indeed accurate however within a certain range. It is also fragile. Finally it’s not prone for repairs. So if it breaks, best course of action would be to replace it.

6.4 sub question 3

We must first write a program to interact with the thermistor. We will then document our findings. For this research we will be using the “KY-028” NTC thermistor. See figure 1.

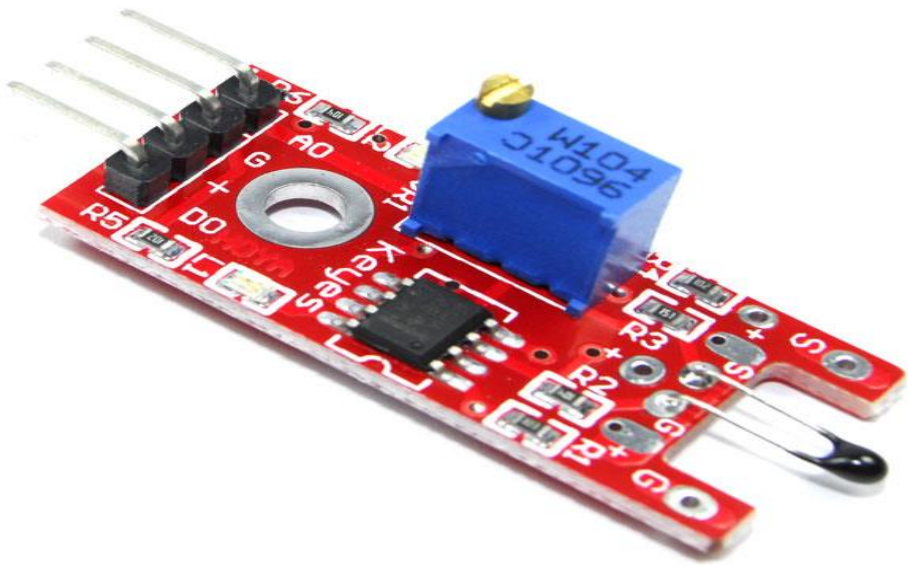


Figure 1. ky-028; NTC Thermistor module.

According to the datasheet of this thermistor. Before we can make a circuit to connect it to an *Arduino*, we need see if the circuit we are building is safe. The thermistor does require another resistor to function. In this case we will use a specially built module for it that does not require one . We can hook this up to our Arduino using the circuit you can see in figure 2.

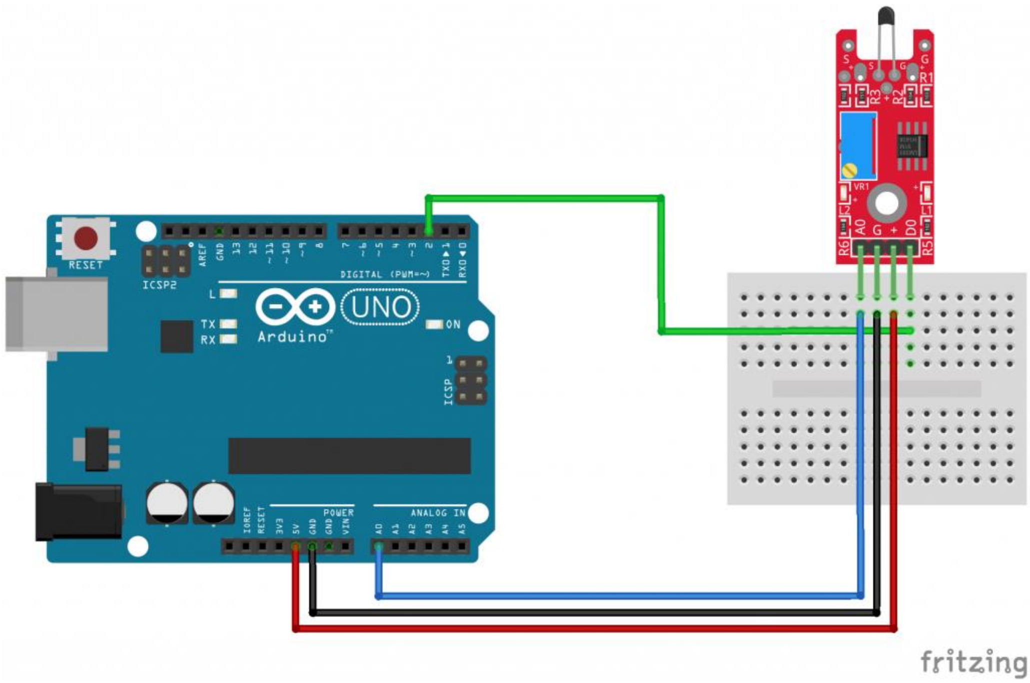


Figure 2. circuit that connects the Arduino and the thermistor.

There is another catch to this thermistor!

We will discuss this after we have discovered it within the written program. See figure 3.

```
#define NTC_ANALOGPIN A0

//Delay for reading values at a second interval.
unsigned long current_time = 0;
unsigned long previous_time = 0;
const int TEMP_INTERVAL = 1000;

/// @brief this function converts the raw values from the sensor into celcius.
/// @param RawADC
/// @return Temp
double ThermistorConverter(int RawADC){
  double Temp;
  Temp = log(10000.0*((1024.0 / RawADC - 1)));
  Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp )) * Temp );
  Temp -= 273.15;            // Convert Kelvin to Celcius
  return Temp;
}

/// @brief This functions reads and display the temperature from the NTC sensor.
void DisplayTemperature(){
  if(current_time - previous_time > TEMP_INTERVAL){
    int rawValue = analogRead(NTC_ANALOGPIN);
    double temperature = ThermistorConverter(rawValue);
    Serial.print("|Temperature: ");
    Serial.print(temperature);
    Serial.println(" °C|");
    previous_time = current_time;
  }
}

void setup() {
  pinMode(NTC_ANALOGPIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  current_time = millis();
  DisplayTemperature();
}
```

Figure 3. program to communicate with the thermistor.

The thermistor reads analog values. So from there we know that we need to somehow convert this raw analog signal into a more readable value. In this case we use a function specially made that can take the raw signal and convert it into temperature.

We then use this value in another function to display our information. Once you compile the code, you will notice that everything seems fine except for **1** crucial detail.

When you increase the temperature the values go down and the opposite for when you decrease the temperature. By taking another glance at the datasheets of this thermistor you may notice that this thermistor's values are inverse. This means that at a high voltage it will show a low voltage value at the analog output. This can be solved using voltage divider. Aside from this, there seems to be no abnormalities when observing the interaction between the thermistor and the program. We then calibrate the thermistor using the built in screw on top of the blue case to get more accurate data.

6.5 sub question 4

In order to answer this question we need to take a step back and review how we can determine the value of temperature using the code from the previous sub question. On its own the code can already generate accurate enough analog values. It would be a matter of transforming this received signal into readable data. An additional feature of this thermistor is the fact that you can adjust its sensitivity using the screw on top of the blue box. once adjusted to the desired value, the thermistor will read accurate data on its own and transmit this to your serial monitor. We learned about the screw in the datasheet and did some minor experimentations with it. Mainly what would happen if you turned it some degree clockwise and counterclockwise.

7 Conclusions

7.1 Main question

The conclusion here is that once you connect both the thermistor and in this case a fan into a safe circuit, it is a matter of implementing the right code and reading the output from the thermistor. The fan will work according to this output value then.

7.2 Sub question 1

No conclusion was determined for this sub question.

7.3 Sub question 2

The conclusion here is that the thermistor is indeed a prominent sensor for reading temperature. Despite it having its minor drawbacks. One thing is for certain is that these drawbacks do not keep you from efficiently getting data within a certain range of temperature.

7.4 Sub question 3

The conclusion here is that it's a matter of writing the proper software, receiving this data and processing it into information. Assuming your thermistor has been wired correctly within a circuit.

7.5 Sub question 4

The conclusion here is that you first set the desired sensitivity using the screw on top. Then you measure the "Analog" signal and then transform this into readable information. Multiple tests must be done in order to get the desired sensitivity.

8 References

[1] *KY-028 datasheet*. (n.d.). https://datasheetspdf.com/pdf-file/1402039/Joy-IT/KY-028/1

[2] Engineering, O. (2022, 17 November). *What Is A Thermistor And How Does It Work?* https://www.omega.com/en-us/. https://www.omega.com/en-us/resources/thermistor

[3] *THERMISTOR BASICS – Wavelength Electronics*. (n.d.). https://www.teamwavelength.com/thermistor-basics/#:~:text=Since%20thermistors%20are%20nonlinear%2C%20meaning,do%20not%20get%20recorded%20correctly.

[4] *Dr Hilderink, G. H.*. (2022.March). Writing technical reports for practicums.

9 Main Research Question.

how can we determine the temperature using the thermistors capabilities to measure resistance and transform this into data we can use. With this research we are going to evaluate how we can efficiently work with the thermistor alongside with other components.

9.1.1 Research strategy

In this research I applied the follow methods from the DOT framework (HBOi, 2021):

Strategy	Method	What
Library	Literature study	Find and study datasheets of thermistor and other components.
Workshop	prototyping	To experiment and get hands-on experience.
Library	Available product analysis	Scanning to internet to see if there is any information on projects around a thermistor.
Lab	System test	To test the final system and document findings.

9.1.2 Research results.

For this question we need to dive a little deeper in the understanding of components. Components on their own work fine given the right circumstances. But when connected with each other it becomes rather different. You have to pay attention to both the hardware side of things as the software side. For now we assume that all of the hardware specification are met. In our case, the thermistor is connected using the provided datasheet. See datasheet reference. Both the fan and the thermistor can work fine on their own. How they interact with one another happens within the software. Within our code we must specify for example that If the temperature reaches a certain threshold, the fan will automatically turn on. While the fan doesn't reach the lower temperature threshold it will work. Meaning once it reaches the low point, the fan will turn off. Of course we don't want to spam the program so we implement the "Millis" function in order to get the measurements at a certain rate. Once you combine the code of the thermistor and the code of the fan, it will be a matter of reading outputs.

The ky-028 NTC thermistor sensor.

To put it simply, a thermistor is a type of semiconductor that react like a resistor that is sensitive to temperature in its environment. The term "thermistor" derives from "Thermally sensitive resistor". The relationship between the thermistor's temperature and its resistance is highly dependent on which materials it is made out of. The manufacturers of thermistors will typically a high degree of accuracy. Meaning you will have an accurate measurement. There are two types of thermistors. The "negative temperature coefficient" (NTC) thermistors and the "positive temperature coefficient". The way an NTC would work is if the temperature increases, the thermistors resistance would decrease. It is the opposite for the PTC. If a PTC's resistance increases, it means that the temperature has also increased. For this research, we will be using an NTC thermistor. An additional feature of this thermistor is the fact that you can adjust its sensitivity using the screw on top of the blue box. once adjusted to the desired value, the thermistor will read accurate data on its own and transmit this to your serial monitor.

The program for the thermistor.

The thermistor reads analog values. So from there we know that we need to somehow convert this raw analog signal into a more readable value. In this case we use a function in our program, specially made that can take the raw signal and convert it into temperature.

We then use this value in another function to display our information. Once you compile the code, you will notice that everything seems fine except for **1** crucial detail.

When you increase the temperature the values go down and the opposite for when you decrease the temperature. By taking another glance at the datasheets of this thermistor you may notice that this thermistor's values are inverse. This means that at a high voltage it will show a low voltage value at the analog output. This can be solved using voltage divider. Aside from this, there seems to be no abnormalities when observing the interaction between the thermistor and the program.

9.1.3 Conclusions

The conclusion here is that a thermistor is a solid choice for reading temperature values. it's a matter of writing proper software and continuous testing in order to get the correct sensitivity on the thermistor. Naturally you would need to connect both the thermistor and, in this case, the fan into a properly made and safe circuit for better functionality. Once you have this you can make the component react on the values that you read from the thermistor. For example the fan will turn on when the temperature being read reaches a certain threshold.