LAB2-MC

Sensors and DC Motors

CMP(N)211

Spring 2021

By: Salma Abdel Monem

Lab Objectives

- ✔ Control DC motors using PWM and H-bridge
- ✔ Read Analog Signals.
- ✔ Pull up resistance.
- ✔ Temperature sensors types
- ✔ Map Thermistor readings to Celsius temperature
- ✔ Use the SimuLIDE.

Recap

Till Now we have learnt the following:

- ✔ Read Digital Signals . Example ?
- ✔ Write Digital Signals. Example ?

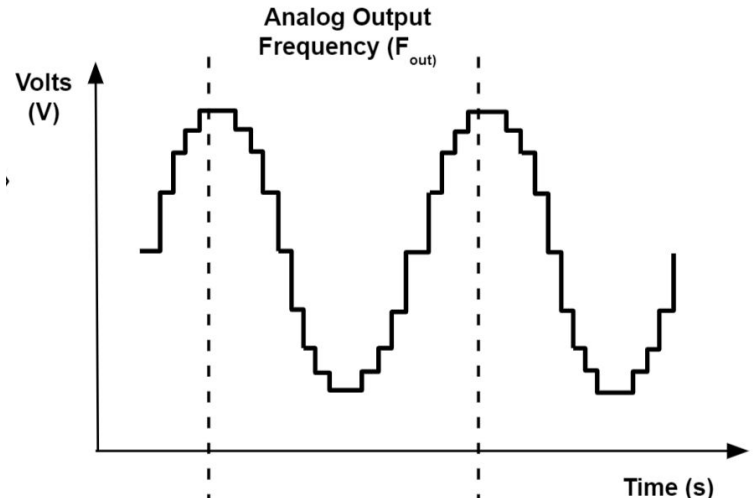
Now it’s time to learn how to Read and Write Analog Signals

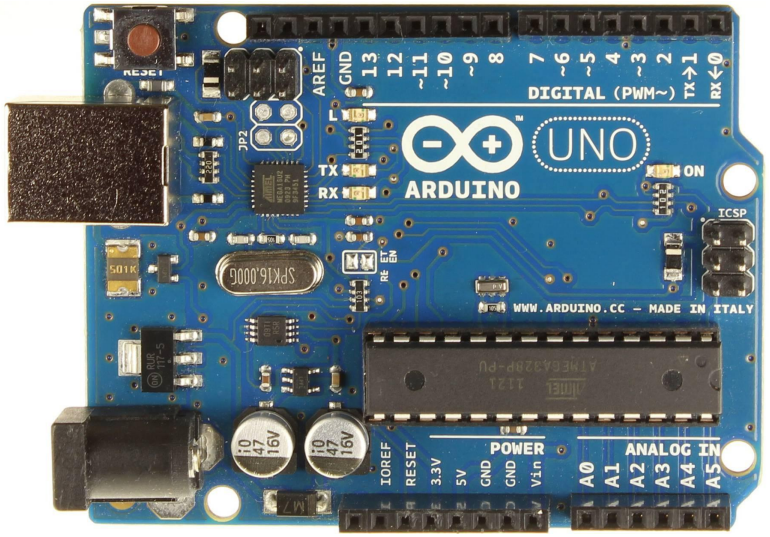
PWM

We will get into it’s details in the next lab.

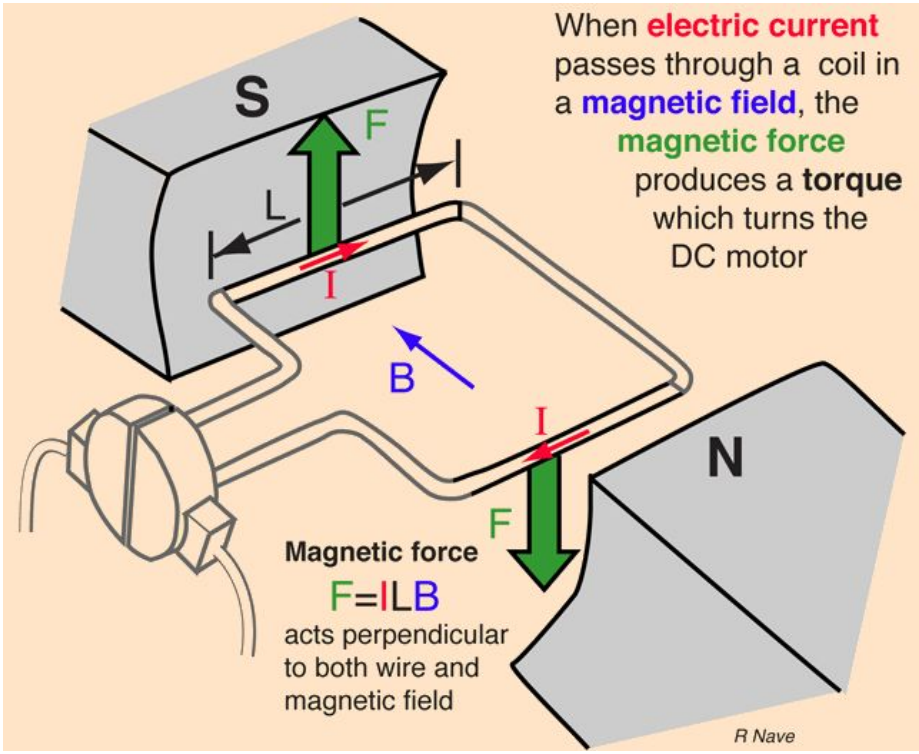
✔ Till now, what you need to know is that PWM pins(~) produce Analog like / discretized signal from (0 to 254).

✔ Example of writing to PWM pin analogWrite(9, 254);



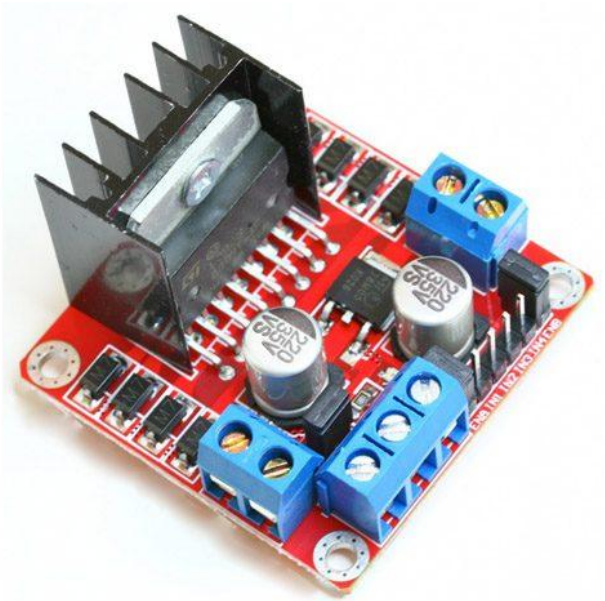


DC motors


When electric current passes through a coil in a magnetic field, the magnetic force produces a torque which turns the DC motor
S
N
F
L
I
B
I
F
Magnetic force
F=ILB
acts perpendicular to both wire and magnetic field
R Nave

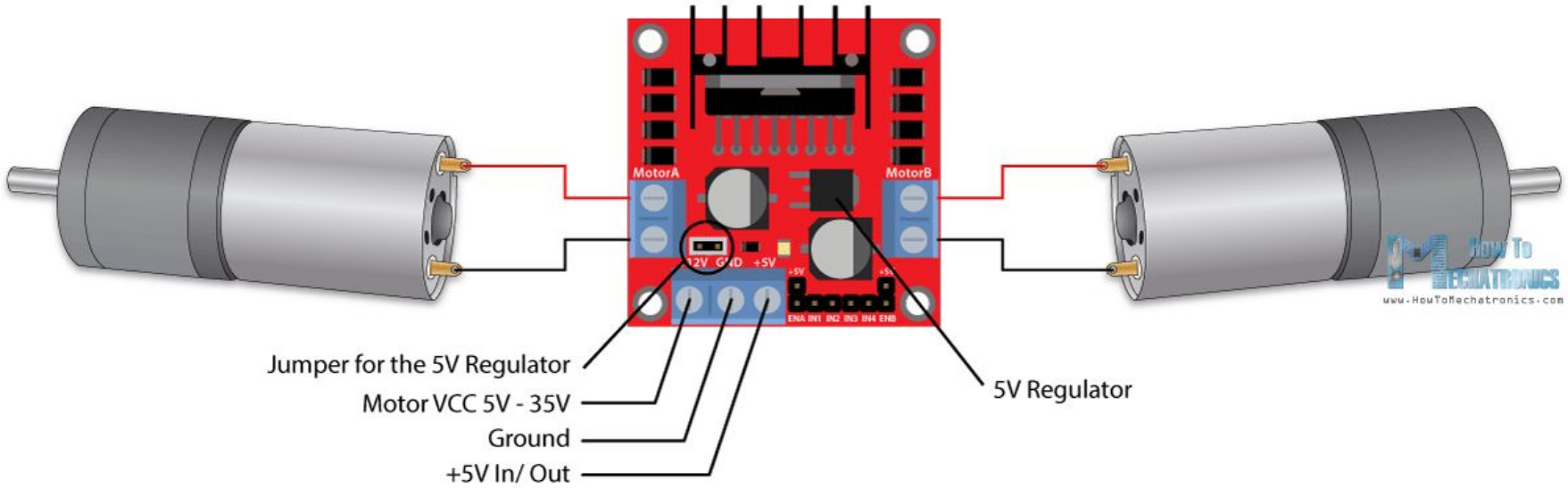
DC motors and H-Bridge

✔ The H-Bridge (L298p here) can be used to control the direction of the spinning of the motor and it’s speed as well.



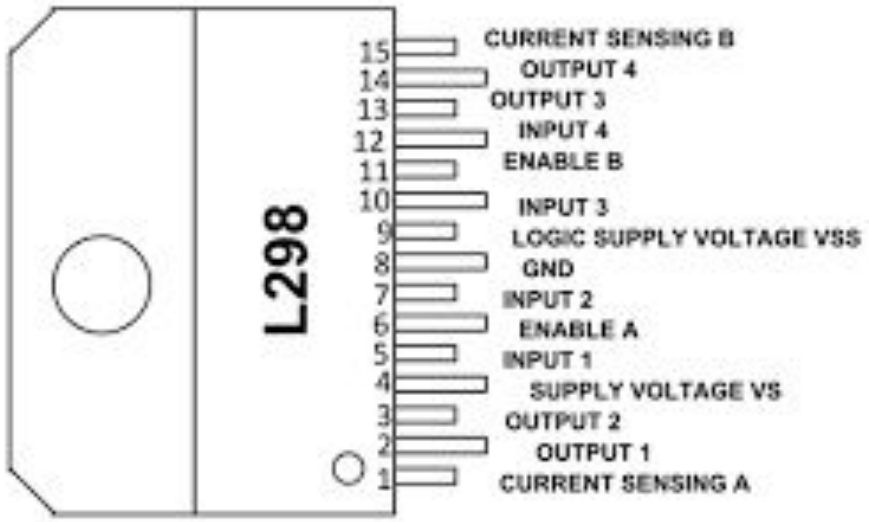
DC motors and H-Bridge

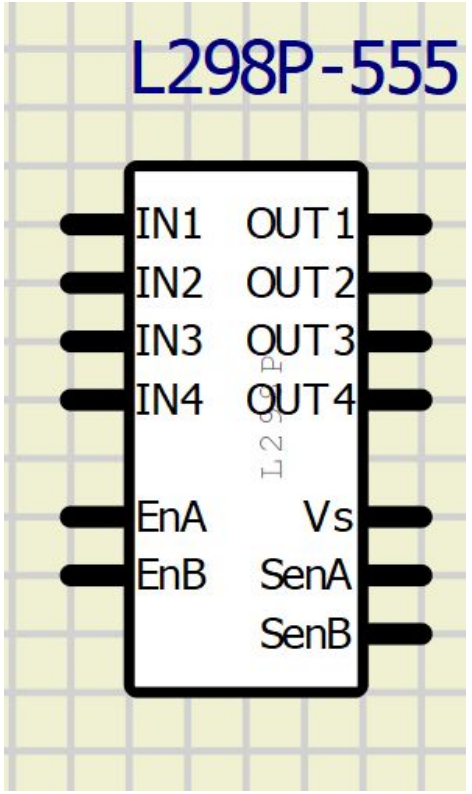
✔ L298p can connect two motors



DC motors and H-Bridge

✔ L298p pinout

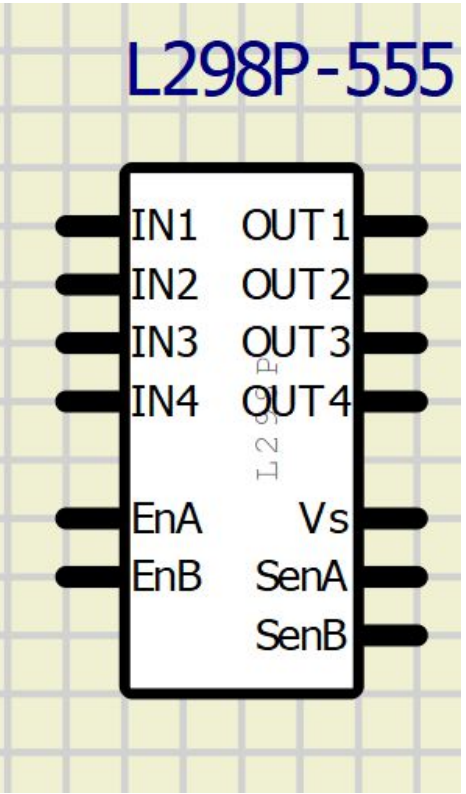




DC motors and H-Bridge

L298p Pinout

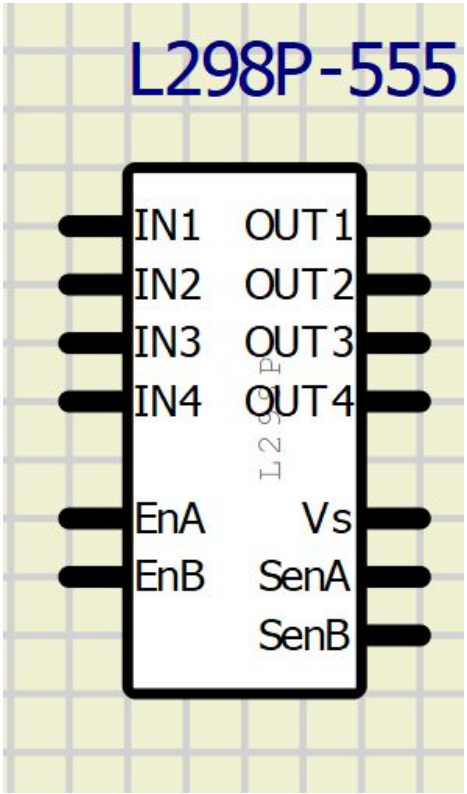
- ✔ IN1 and IN2 control direction of motor 1 (arduino)
- ✔ IN3 and IN3 control direction of motor 2 (arduino)
- ✔ EnA controls the speed of the motor 1 (PWM arduino)
- ✔ EnB controls the speed of the motor 2 (PWM arduino)
- ✔ SenA and senB should be grounded
- ✔ OUT1 and OUT2 are connected to the both edges of the motor 1
- ✔ OUT3 and OUT4 are connected to the both edges of the motor 2
- ✔ Vs is connected to 3.3 v from arduino



DC motors and H-Bridge

L298p Pinout

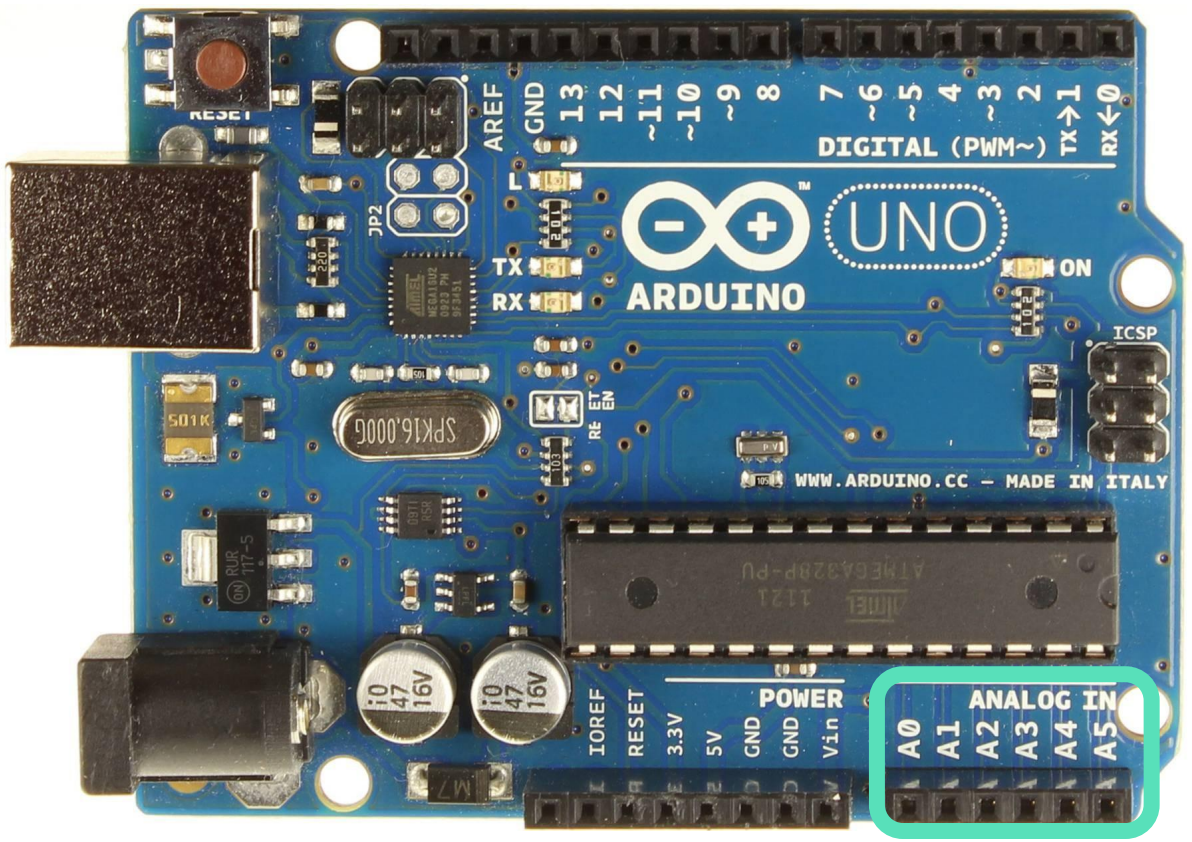
IN1	IN2	MOTOR
0	0	BRAKE
1	0	FORWARD
0	1	BACKWARD
1	1	BRAKE



✔Given that OUT1 is connected to the +ve edge of the motor

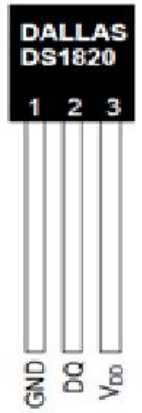
Read Analog Signal

- Use the analog pins {A0 ….A5} to read analog signals.
- The analog input can vary from { 0 to 1023} .
- You might want to map 0 to LOW , and 1023 to HIGH {5volt} in Arduino.
- Pin numbers are A(0 : 5) and no need to set them as inputs in the setups.
- For example: analogRead(A3)



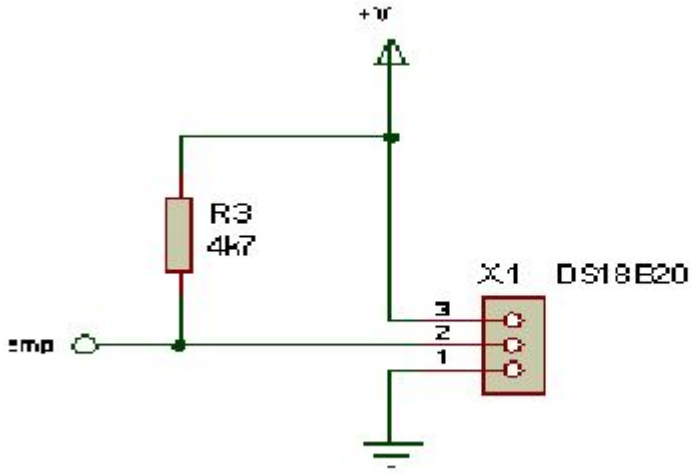
Temperature sensor types

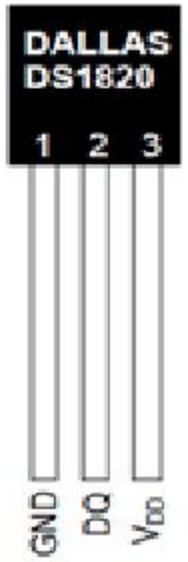
- Thermistor (cheap, wide range, less accurate, non linear)
- Thermocouple
- RTDs (resistive temperature detectors)
- Digital thermometer ICs (more expensive, more accurate, linear)
- Analog thermometer ICs



Digital thermometer IC sensor (DS1820)

- It has three legs like the following:

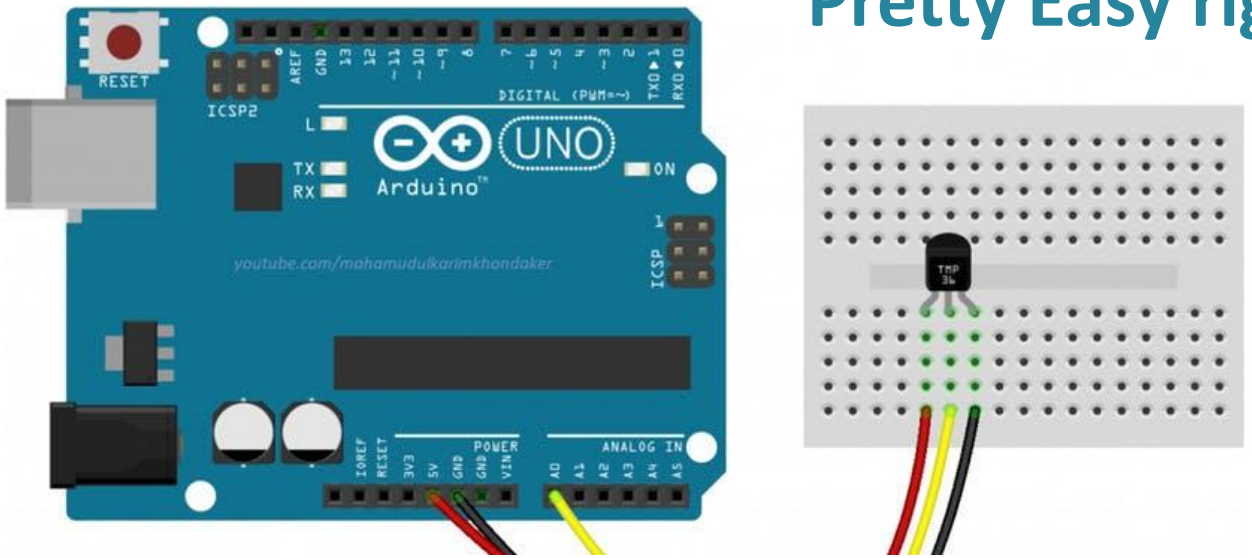




So let’s connect it with an Arduino ??

Yet this is not recommended !

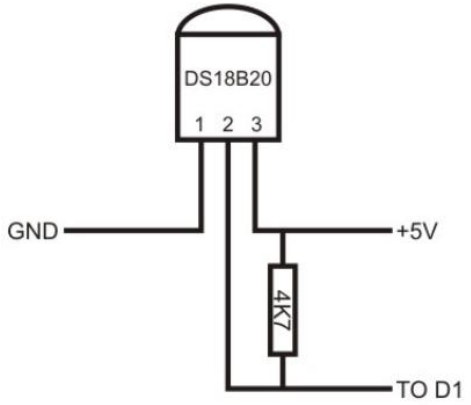
Pretty Easy right?

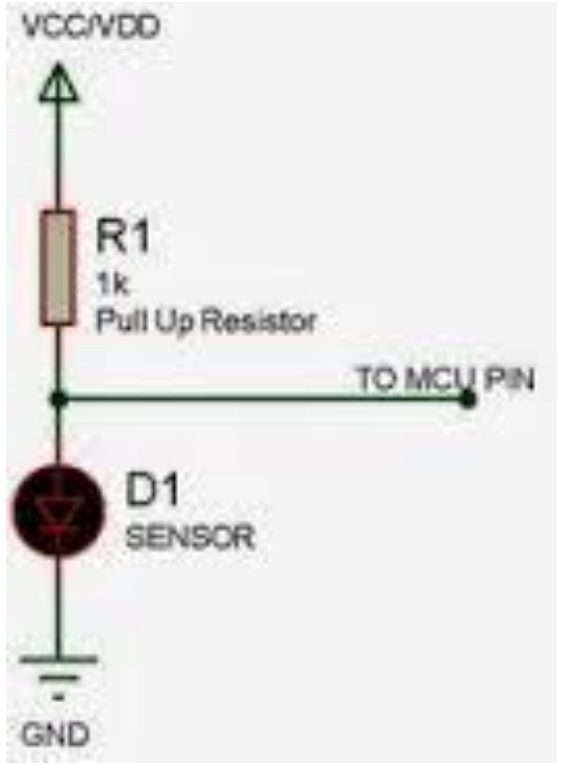


This might cause unstable readings if the sensor is not sending a reading signal

Solution? Pull Up resistors

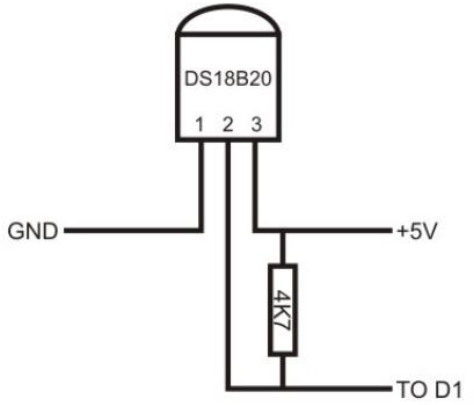
- What happens when the sensor is open circuit?
- Answer is : floating input !!

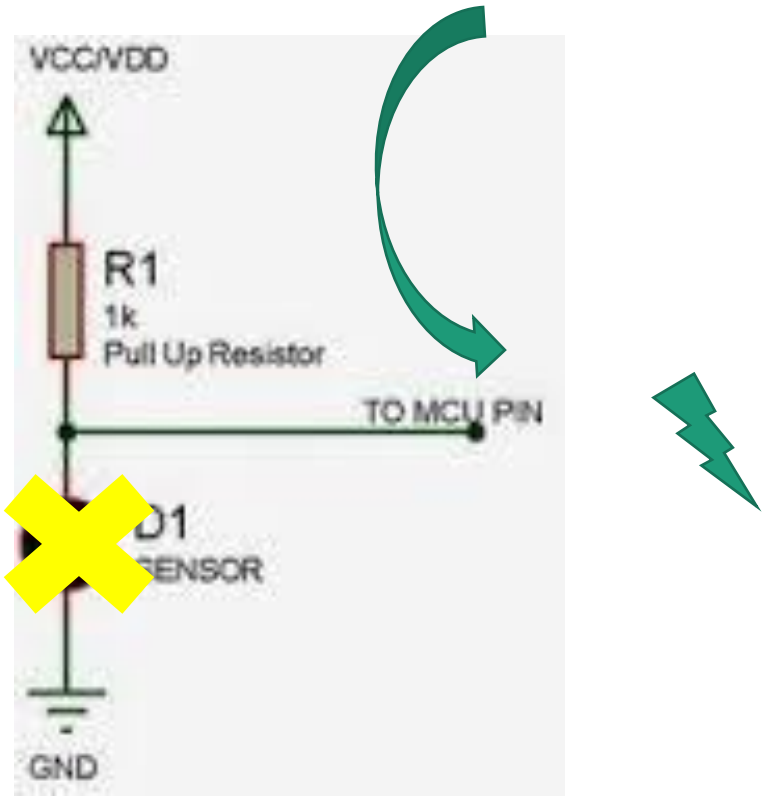




Solution? Pull Up resistors

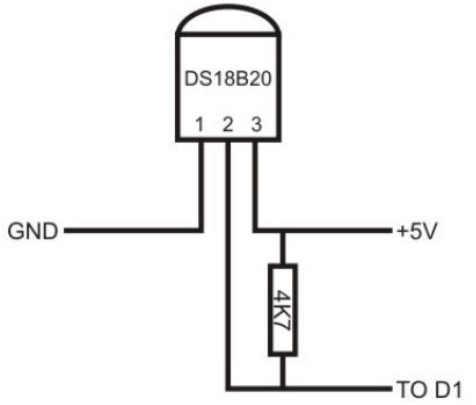
- The sensor is open circuit

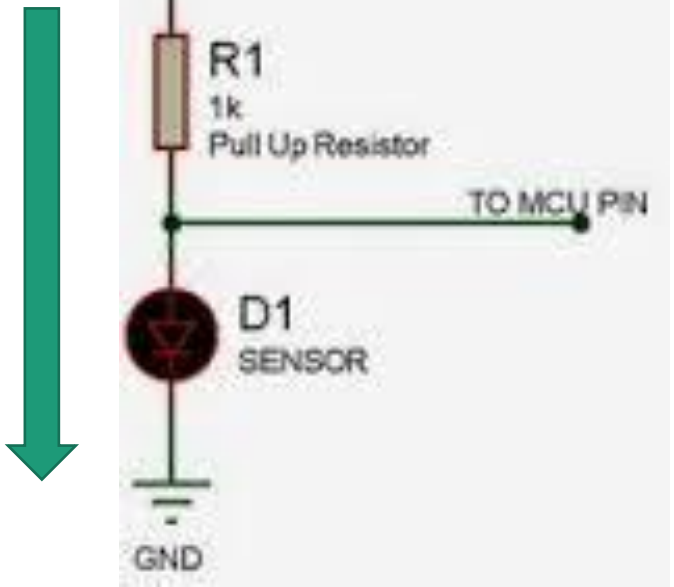




Solution? Pull Up resistors

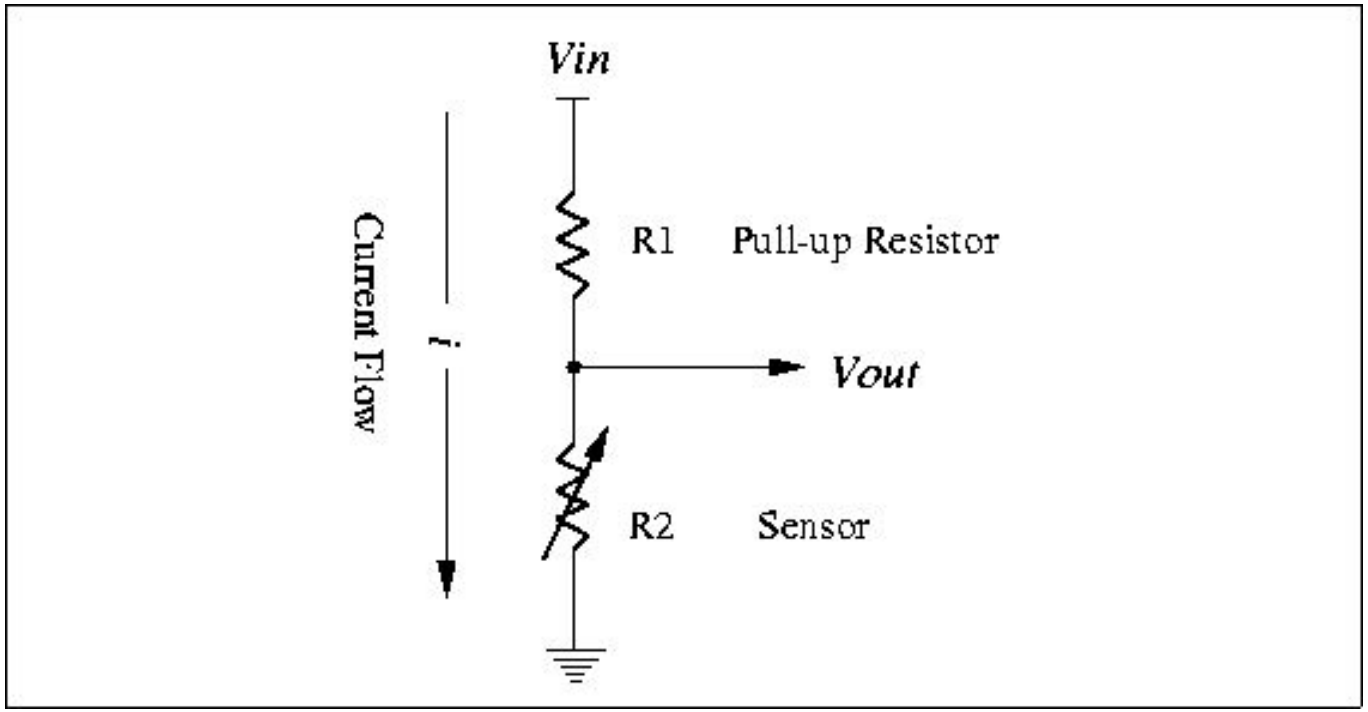
- The sensor is closed circuit





What will be the reading then?

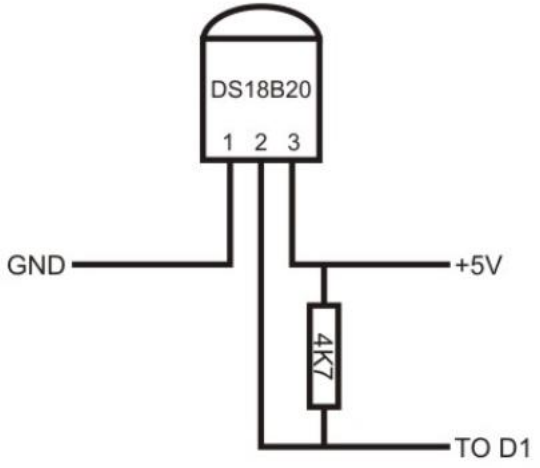
- Remember the voltage divider rule.
- When the sensor is closed--circuit; it's resistance changes with the temperature.
- The reading voltage *Vout* is then what the Arduino senses.

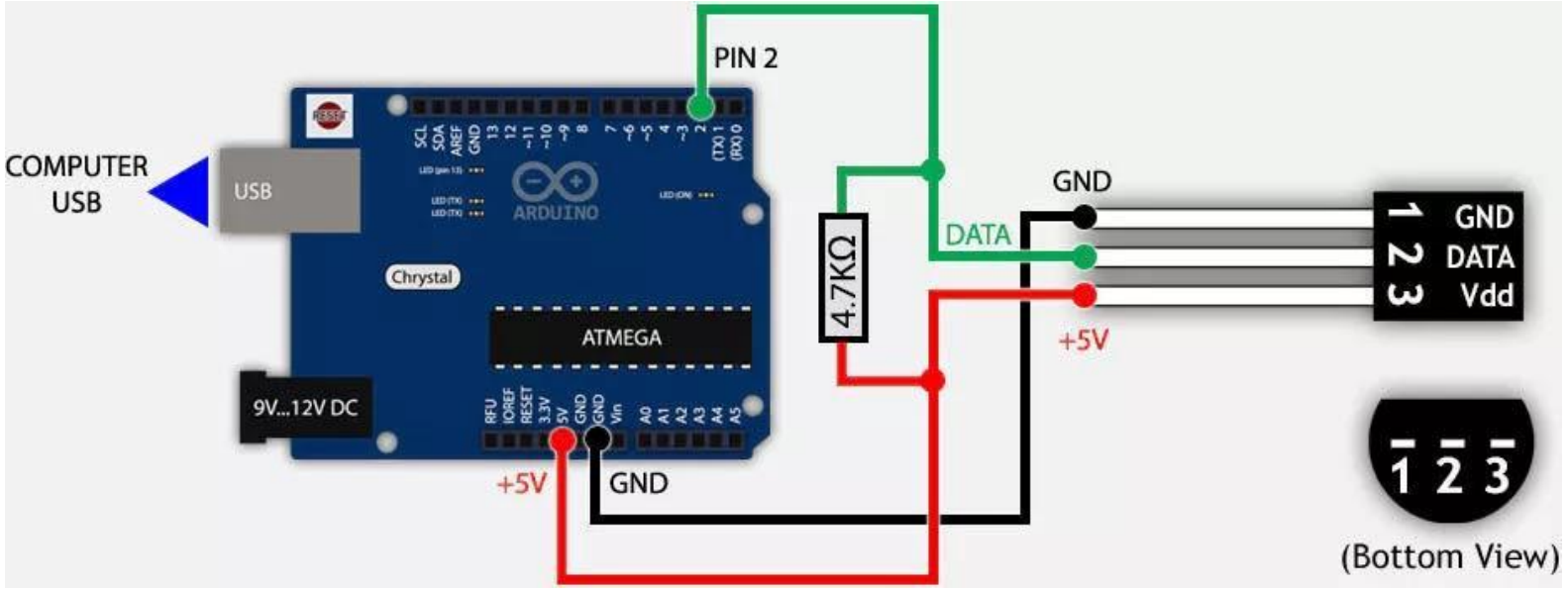


$$\mathbf{Vout} = \frac{\boldsymbol{R2}}{\boldsymbol{R2 + R1}} * \boldsymbol{Vin}$$

Then how should we connect it ?

- Using a pull-up resistance



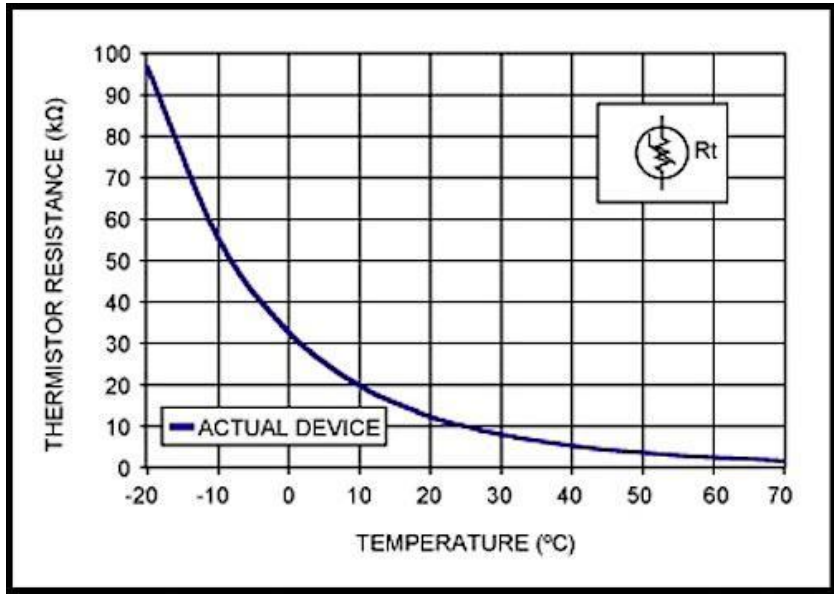


Additional notes to connect the DS1820

- You have first to use the 1-wire library with arduino.
- Following the steps in this site.
- You can also see the details of the ADC in the datasheet.
- It has linear mapping between the reading and the temperature.

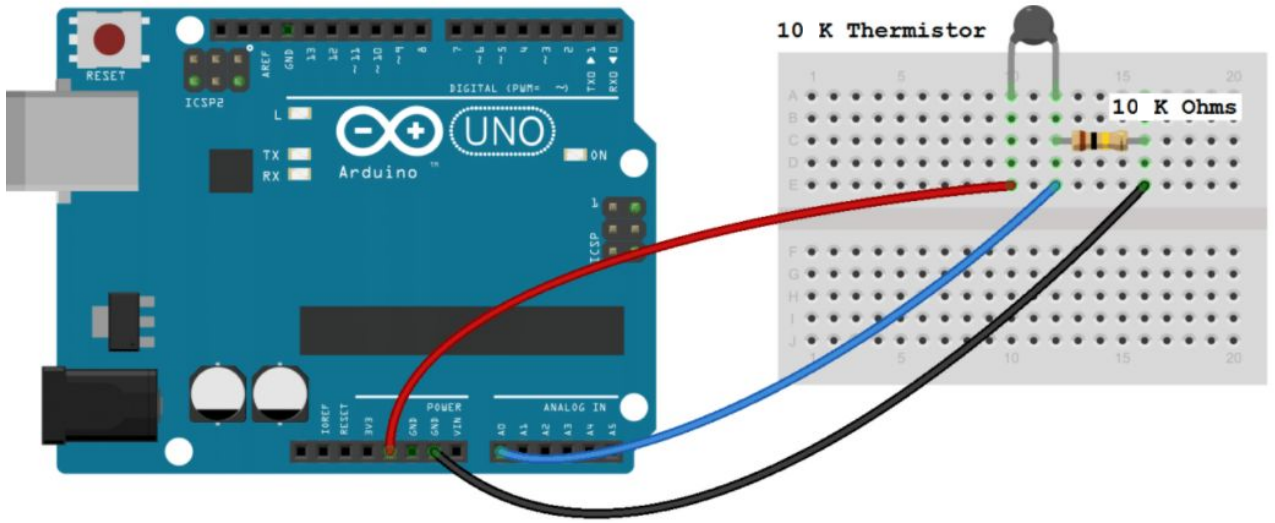
Thermistor(NTC)

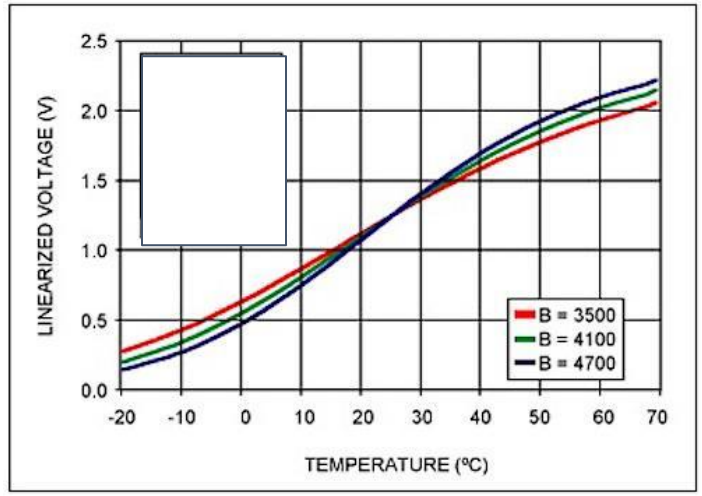
- It has two legs like the following
- Simple and cheap but it doesn't have linear mapping for temperature vs. it's analog readings.



Thermistor(NTC)

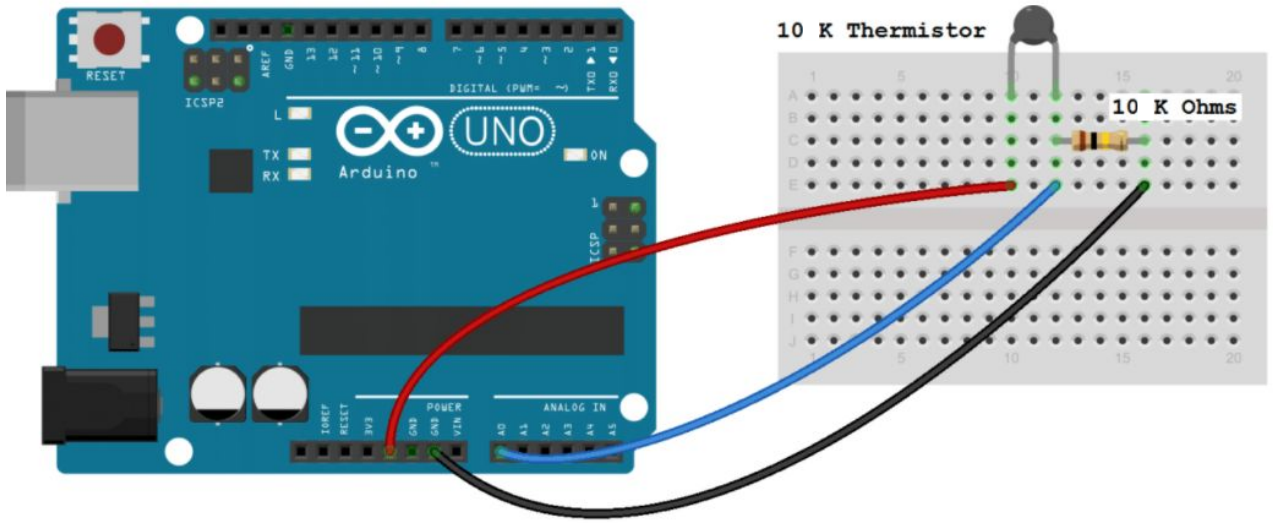
- Placing the Thermistor in series with a normal resistor forms a voltage divider circuit.
- Suppose we will connect a pull down resistance.

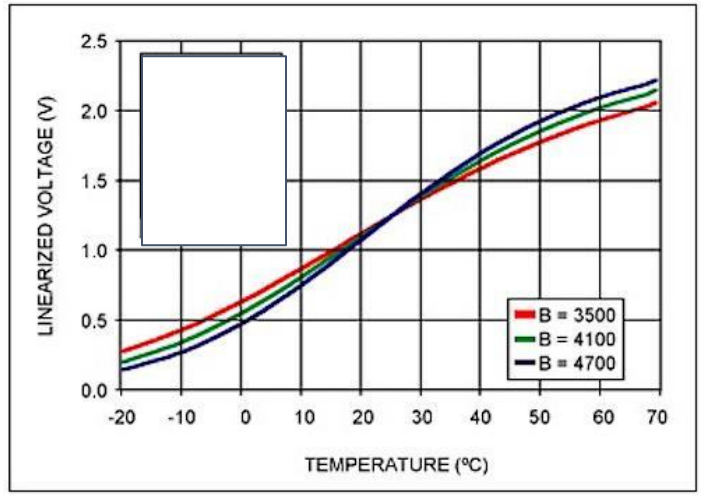




Thermistor(NTC)

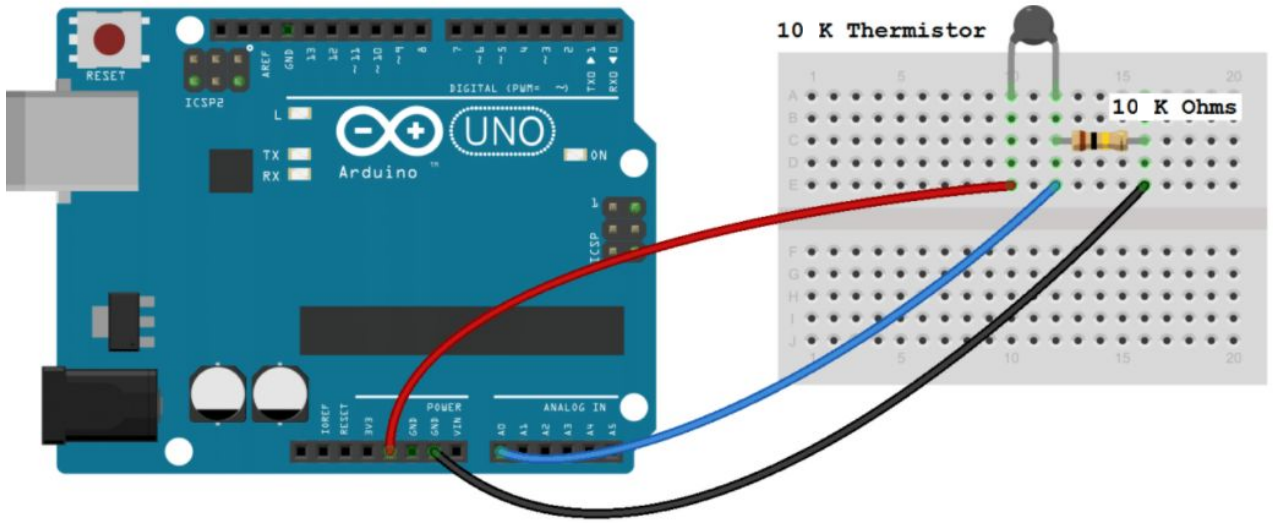
- Linear mapping technique (Voltage Mode Linearization)

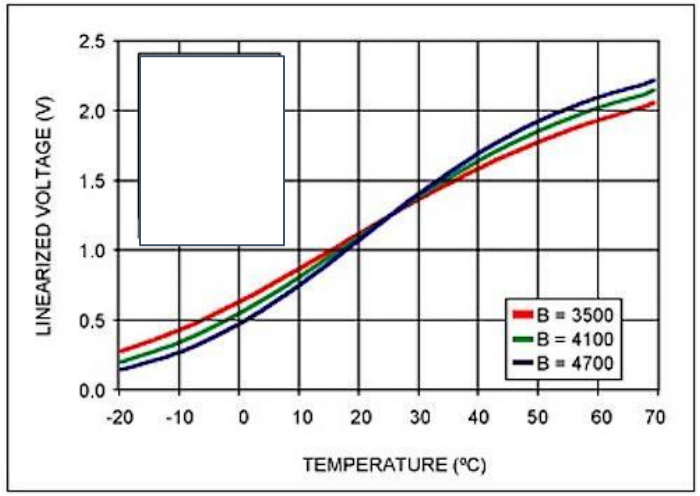




Thermistor(NTC)

- If the resistor's value is equal to the Thermistor's resistance at room temperature, then the region of linearization will be symmetrical around room temperature





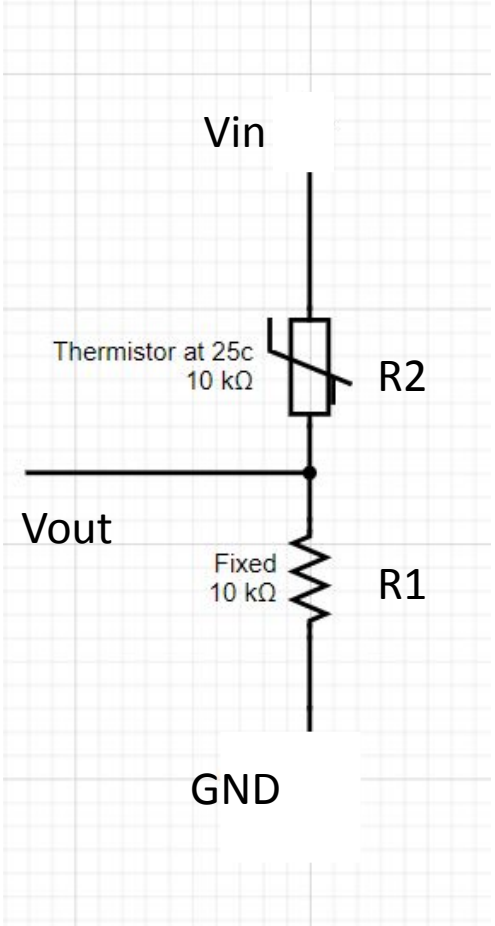
So what is the actual temperature ?

- Remember that the Arduino reads the voltage *Vout*.
- Using the voltage Vin=1023 and *Vout* (reading) from {0:1023} we can get the sensor's resistance right ?
- We connect a pull down resistance.

$$\mathbf{Vout} = \frac{\boldsymbol{R1}}{\boldsymbol{R1} + \boldsymbol{R2}} * \boldsymbol{Vin}$$

$$\mathbf{R2} = \boldsymbol{R1} * \left(\frac{\boldsymbol{Vin}}{\boldsymbol{Vout}} - \mathbf{1}\right)$$

=10000.0*((1023.0/Vout)-1)



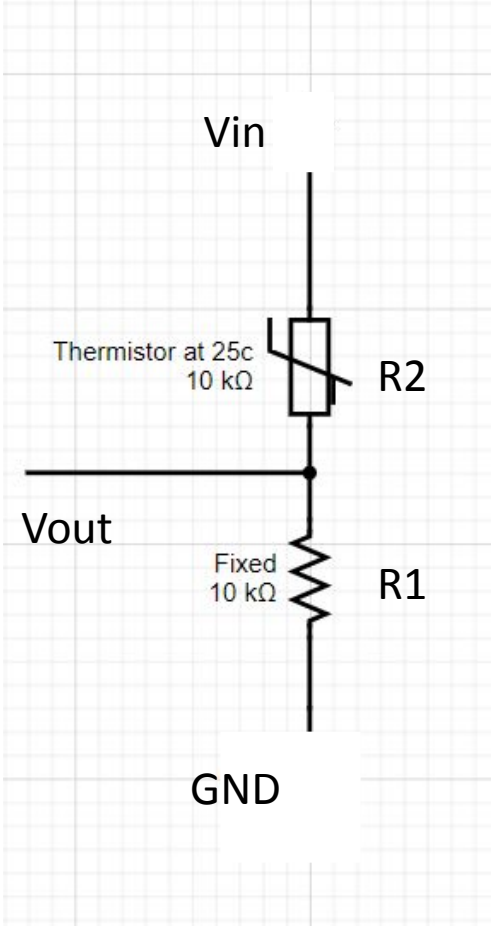
So what is the actual temperature ?

$$\mathbf{R2} = \boldsymbol{R1} * \left(\frac{\boldsymbol{Vin}}{\boldsymbol{Vout}} - \mathbf{1} \right)$$

- Using the “Stein–Hart equation”
- A,B and C are constants related with the sensor.
- T is temperature and R is the sensor’s resistance.

$$\frac{1}{T} = A + B \ln R + C(\ln R)^3,$$

- T is the current temperature + 273.15
- So subtract 273.15 from T and you got the temperature.



Exercise

- Use the SimuLIDE to Perform the following requirement.
- Read the temperature from a Thermistor and control a dc motor using it.
- If the reading is between (0 and 50 c) the motor stops.
- If the reading is between (51 and 100c) the motor moves forward with half the speed.
- If the reading is higher than 100c the motor moves backwards with maximum speed.
- You might need a small delay between readings.
- Use the Serial.println for printing the “Vout, R2, and T”.

Exercise Credit hours

- Use the SimuLIDE to Perform the following requirement.
- EX1 :
  - Connect two input switches to the arduino.
  - If the input switches = 0,0 or 1,1 -> stop the dc motor
  - If the input switches = 1,0 -> move the motor forward with half the speed.
  - If the input switches = 0,1 -> move the motor backwards with maximum speed.

Exercise Credit hours

- Use the SimuLIDE to Perform the following requirement.
- EX2 :
- Connect a Thermistor to the arduino and print the temperature to the serial monitor.
- Use the Serial.println() for printing.
- You might need a small delay between readings.

Exercise Credit hours

- EX3 bonus :
- Merge the two exercises as follows:
  - Connect a Thermistor to the arduino and print the temperature to the serial monitor.
  - If the reading is between (0 and 50 c) the motor stops.
  - If the reading is between (51 and 100c) the motor moves forward with half the speed.
  - If the reading is higher than 100c the motor moves backwards with maximum speed.

Appendix

- Thermistor constants for NTC 10k ohm are :
- float A = 1.009249522e-03, B = 2.378405444e-04, C = 2.019202697e-07;
- You can use ‘log’ function in arduino instead of ‘ln’.