Exercise #B1(**AnalogWrite, Pulse Width Modulation, PWM**):

Reading an analog sensor (TMP 36)

The TMP36 is a low voltage, precision centigrade temperature sensor. It provides a voltage output that is linearly proportional to the Celsius temperature. It also doesn't require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range.

Table 4. TMP3x Output Characteristics

Sensor	Offset Voltage (V)	Output Voltage Scaling (mV/°C)	Output Voltage @ 25°C (mV)
TMP35	0	10	250
TMP36	0.5	10	750
TMP37	0	20	500

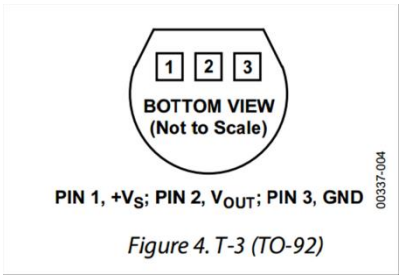


Figure 4. T-3 (TO-92)

For the complete datasheet of the TMP36, please check the 'links' page at the workshop webpages.

Controlling LED illumination

Light emitting diodes (LEDs) are two-terminals devices that emit light at an intensity that is proportional to the current the flows through the device.

LEDs are polar devices: the long leg should be connected to the positive voltage and the short leg should be connected to the GND pin.

In order not to kill an LED, its current should be limited. Here we use 270 Ohms resistor (which is connected in series to the LED) to limit the LED current to below 20 mA (For a 5V source).

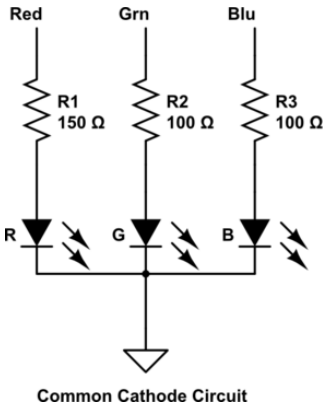
<u>The circuit:</u>

* thermometer TMP 36 (pin 2 – center) attached to analog input 0;

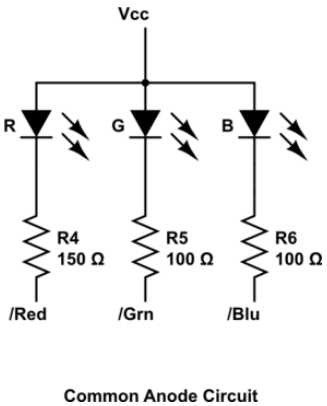
* thermometer TMP 36 (pin 3) to ground; * thermometer TMP 36 (pin 1) to +5V

* 3 color LED: short leg(s) to the GND; long leg(s) to the 5V via 150-270 ohm resistor.

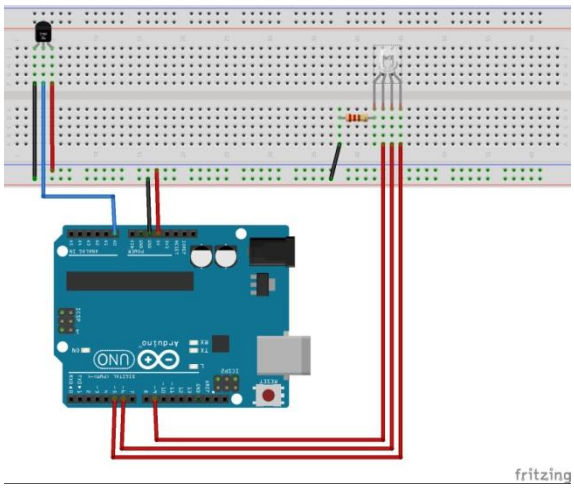
!!! Connecting the TMP in an opposite polarity causes an excessive heating that may kill the device and/or cause burns !!!



Common Cathode Circuit



Common Anode Circuit



#B1: Take a temperature reading every exactly 3 second. Between readings, the color of the LED should turn from blue to red with increase in the reading temperature – e.g. from red at 18°C to blue at 28°C.

Let's break the task to intermediate steps:

1. Build a simple circuit with *only* the TMP36 temperature sensor.
2. Write a code that reads the ADC value of the sensor (number between 0 and 1023) every 2 seconds.
   ***** Since you would like to 'do' things between readings (blinking the LEDs), use the 'blinkWithoutDelay example to execute a reading every 2 seconds *without* using the 'delay' function.**
   For the 'AnalogRead' function use the 'AnalogInput' Example
3. Convert the reading value (number between 0 and 1023) to temperature in Celsius degrees.
   Use the code we wrote last week with using the 'map' function.
4. Add to your circuit the 3-colours LED (remember to disconnect the Arduino from your computer when messing up with the circuit)
5. Test the circuit with simple 'if' condition to see that you can alternate the colour between blue and red of the LED by touching the temperature sensor.
6. Map the temperature range of interest (22 to 28) to the range of 0-255.
   int colour=map(int(sensorTemp),20,24,0,255);
   Use the function 'constrain' to ensure that the colour value is always between 0 and 255:
   https://www.arduino.cc/en/Reference/constrain
7. Use **analogWrite(LED_PIN, Value_between_0_and_255);** to control the colour of the red/blue LEDs
   analogWrite(LEDPinBlue, colour);
   analogWrite(LEDPinRed, 255-colour);

Use the following example from the Arduino examples library: Analog/AnalogInOutSerial

When you're done, explore the other two Sketches in the course library for examples of **plotting** and **sending data to your Arduino using the Serial port**

Exercise #B2 (reading sensor, and **plotting results**).

Download Megunolink LITE from the following link: http**://www.megunolink.com/megunolink-lite/**

Download and install the Arduino Library: **http://www.megunolink.com/download/GraphSeries.zip**

- Plot the measured temperature
- Save the data to a text file

The circuit:

* thermometer TMP 36 (pin 2 – center) attached to analog input 0

* thermometer TMP 36 (pin 3) to ground

* thermometer TMP 36 (pin 1) to +5V