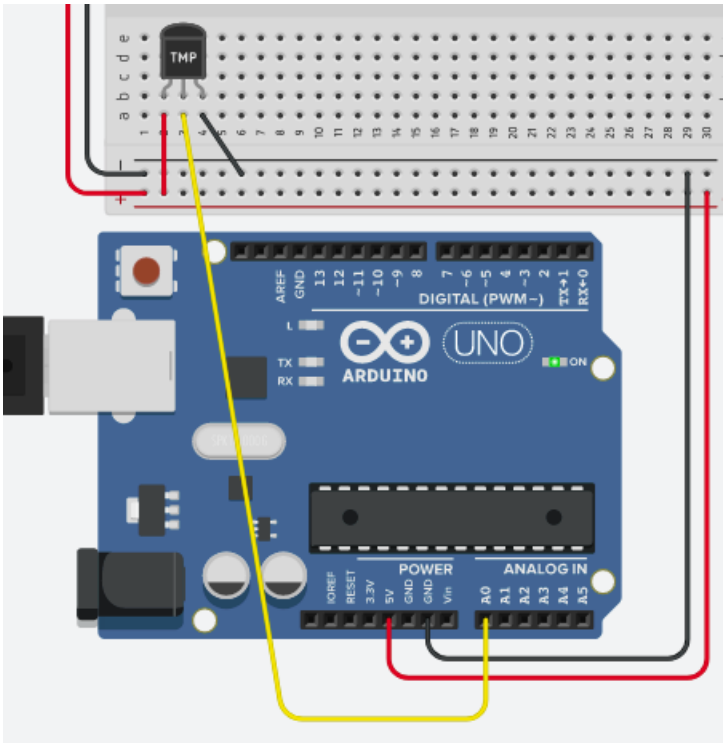
Arduino Summative – Advanced Thermometer

We can use our digital world to make something useful and different – an advanced analogue and digital thermometer.

There is a temperature sensor (TMP36) in Tinkercad that has three pins (POWER, GND, and Vout). The Vout pin should be connected to any one of the six analog input pins (A0-A5). This reading, through the equations below, will allow you to see what the current temperature is in fahrenheit or celsius. We will be using celsius for this assignment. Use the equations to determine the temperature in Degrees Celsius. Both variables below are of type float.

```
voltage = analogRead(A0) * 0.004882813;   //convert the analog reading (from 0 - 1023) back to a voltage value from 0-5 volts
degreesC = (voltage - 0.5) * 100.0;       //convert the voltage to a temperature in degrees Celsius
```



To the left, you will see how to wire the temperature sensor. You will notice that once you run the program and click on the sensor and adjust the sliding bar that you get an unrealistic range of temperatures for any indoor environment. Your job is to make it so that the minimum temperature is 0 degrees celsius and the maximum temperature is 40 degrees celsius.
Hint: What function have we learned in the labs that can do this for us?

Now that we have an understanding of how the sensor works, you will be given two different scenarios and you will be responsible to make modifications to the circuit. Each scenario will have a range of marks that are obtainable based off of the level of difficulty of implementing the advanced features. The modifications will make the thermometer more useful to us.

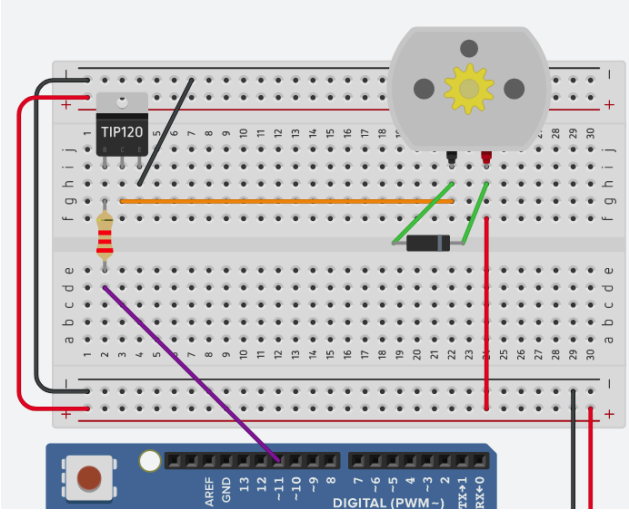
Programming Requirements:

Scenario (These requirements must be done in order to increase your mark):

- [Base Requirement] analogue temperature be displayed over a 180-degree scale by using a servo motor. When the servo motor is all the way over to the left (0 degrees) it will represent 0 degrees celsius. When the servo motor is at 180 degrees (all the way to the right), it will represent the max temperature which is 40 degrees celsius. The range will vary depending on how to change the slider (left and right) when clicking on the temperature sensor.

- [60% max] Wire two LED's with each LED being a different colour. Use any colour except for blue and red. Make sure you put an annotation in close proximity to the LED's stating which one is the minimum temperature and which one represents the maximum temperature. When the temperature is 0 degrees, turn on the Min. LED. When the temperature is 40 degrees, turn on the maximum temperature LED. Make sure that both LED's are never on at the same time. Also, make sure that both LED's turn off when you are not at the min and max temperatures.
- [65% max] Use an RGB LED to indicate when the heater or the air conditioner should be on. Use RED for heat and BLUE for A/C. *You choose* what values these are for when the A/C should turn on, and when the heater is on. Use a different colour to show when neither one is on. This means that when it is not cold enough for the heater and when it is not hot enough for the air conditioner, a DIFFERENT colour should be on. Example: pink.
- [70% max] Wire a push button that when pressed and released, causes the circuit to RESET. This means that all LED's (MIN/MAX/RGB) should turn off. The Servo motor should rotate all the way to the left slowly and all the way to the right. Then the circuit should be recalibrated and it should function the way it is supposed to. If you have a motor or other functions, they should resume normally as well but should stop during the reset phase.
- [75% max]
  Initial Setup: You will be using a DC motor to represent an Air Conditioner. Normally, an Arduino is not capable of powering a DC motor simply by using a digital I/O pin due to its low current (in mA). You need to amplify the current in order to be able to use the motor. An NPN transistor (TIP120) will be required here to accomplish this. You must send a current from a PWM (~) pin on the Arduino to send a current to the base of the transistor in order to activate the transistor. Once the transistor is activated, the +5v from the Arduino board can flow from the Collector to the emitter. You will be using a DC Motor, TIP120 (NPN transistor), a 2.2 kilo-Ohm resistor and a diode.

  Use this wiring diagram:



TASK:

- You need to automatically turn on the fan (DC motor) when it is hot enough in your environment. Example: As soon as it is 20 degrees the fan will turn on at its full speed. You choose the threshold temperature yourself. What temperature is unbearable for YOU? When the temperature drops below the threshold, you must turn the motor OFF. Be sure that whatever Arduino pin you use for the motor that it has a PWM (~).
- [80 max] Make it so that when the temperature is between your threshold and 40 degrees celsius, the motor will spin faster as the environment becomes warmer. Hint: Use the map() function.

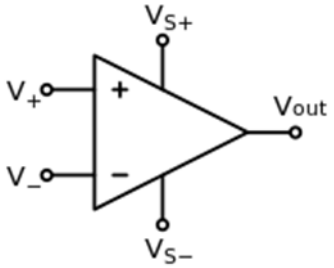
- [90 max] Your circuit will not change in terms of its overall operation but the fan (DC Motor) must turn ON immediately once the threshold is met using an INTERRUPT and it will turn OFF immediately once the temperature is below the threshold using an interrupt.

  The Vout pin on the temperature sensor sends its voltage to the analog input pin. The value ranges from 0 to 5 volts. Based on the voltage read, your equations tell you what the actual temperature is in degrees celsius. The Interrupt pins (2 and 3) only work with digital inputs (0 volts and 5 volts). You cannot send a range of voltages to them. You must use the **LM358 chip** (called 741 Operational Amplifier in Tinkercad) to send +5v or 0 volts to the Interrupt pin. Once the temperature has reached a threshold (your choice) the fan should automatically start up). Hint: You need to create your OWN voltage divider circuit to compare with the value from the temperature sensor. Show your work in a report attached to this assignment post.
- Create a second interrupt that turns the fan OFF immediately when the temperature is below the threshold. There is more than one way to do this. Be creative! Hint: Do you really need to use two interrupt pins to accomplish this task?

Read this to understand how to create the required interrupts described above.

An Operational Amplifier (LM358) is a device capable of amplifying DC and AC input signals. It has two inputs and a single output. It gets its name OP Amp (Operational Amplifier) because it was initially designed for analog computers to allow them to perform mathematical operations such as addition, subtraction, multiplication and division etc.

Overview:



- Vs+ = Positive power supply
- Vs- = Negative power supply
- V+ = non-inverting input voltage
- V- = inverting input voltage
- Vout = Output voltage

If the voltage sent to V+ is higher V-, then Vout will send out a +5 volt signal. Otherwise, it will send out 0 volts. On Tinkercad, V- is IN- and V+ is IN+.

- [MAX 100% - with all other requirements met] Wire a slideswitch that is capable of putting the Arduino is AUTO or MANUAL mode for the A/C. When the switch is to the LEFT the mode will be MANUAL. This means that you can turn on the fan whenever you want using another input component in Tinkercad. This is UP TO YOU. When in manual mode, the fan can spin at its maximum speed until you wish for it to stop. Make sure there is an easy way to control this using your Arduino. When the fan is in AUTO mode, it will spin according to your threshold temperature. This is controlled by your interrupts. You do not need an interrupt for when AUTO or MANUAL is selected.

  To understand how a slideswitch works, try wiring and testing out this circuit: