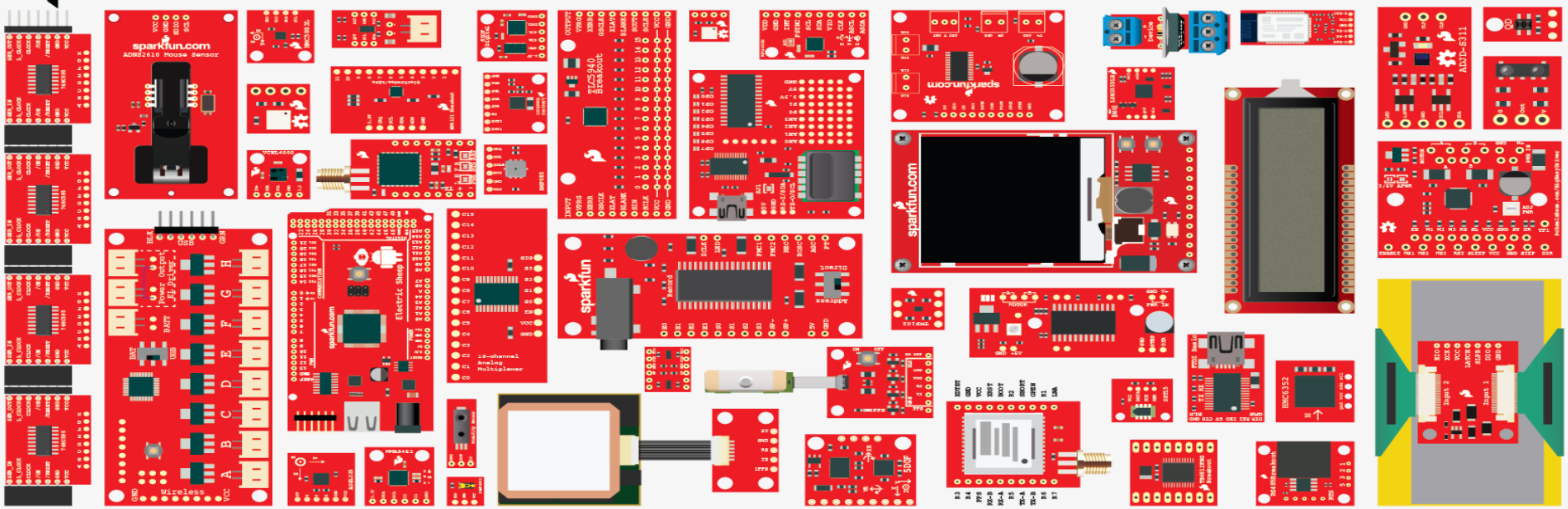
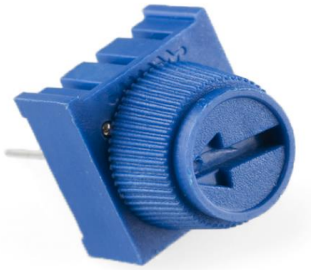
sparkfun.com



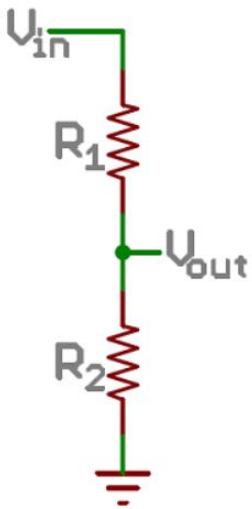
# Sparkfun Advanced

## Sparkfun Inventor's Kits [SIK]

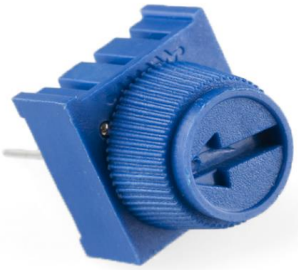
# What is a potentiometer?



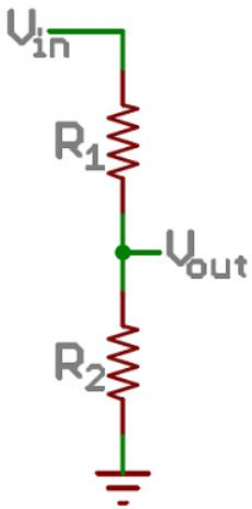
- a variable resistor!
- Pots connect two resistors internally, in series, and adjust a center tap between them creating an adjustable voltage divider.




# What is a potentiometer?



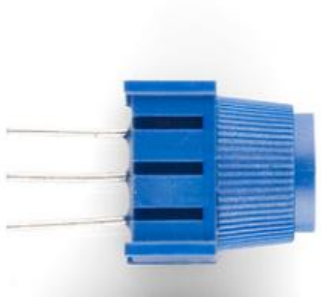
- a variable resistor!
- Pots connect two resistors internally, in series, and adjust a center tap between them creating an adjustable voltage divider.
- Vin →power
- Vout→ connets to analog pins on the board (A0)
- When powered with 5V, the middle pin outputs a voltage between 0V and 5V, depending on the position of the knob on the potentiometer.
- Used for inputs, control the blinking rate of an LED, the contrast of LCD, …




# Reading a Potentiometer

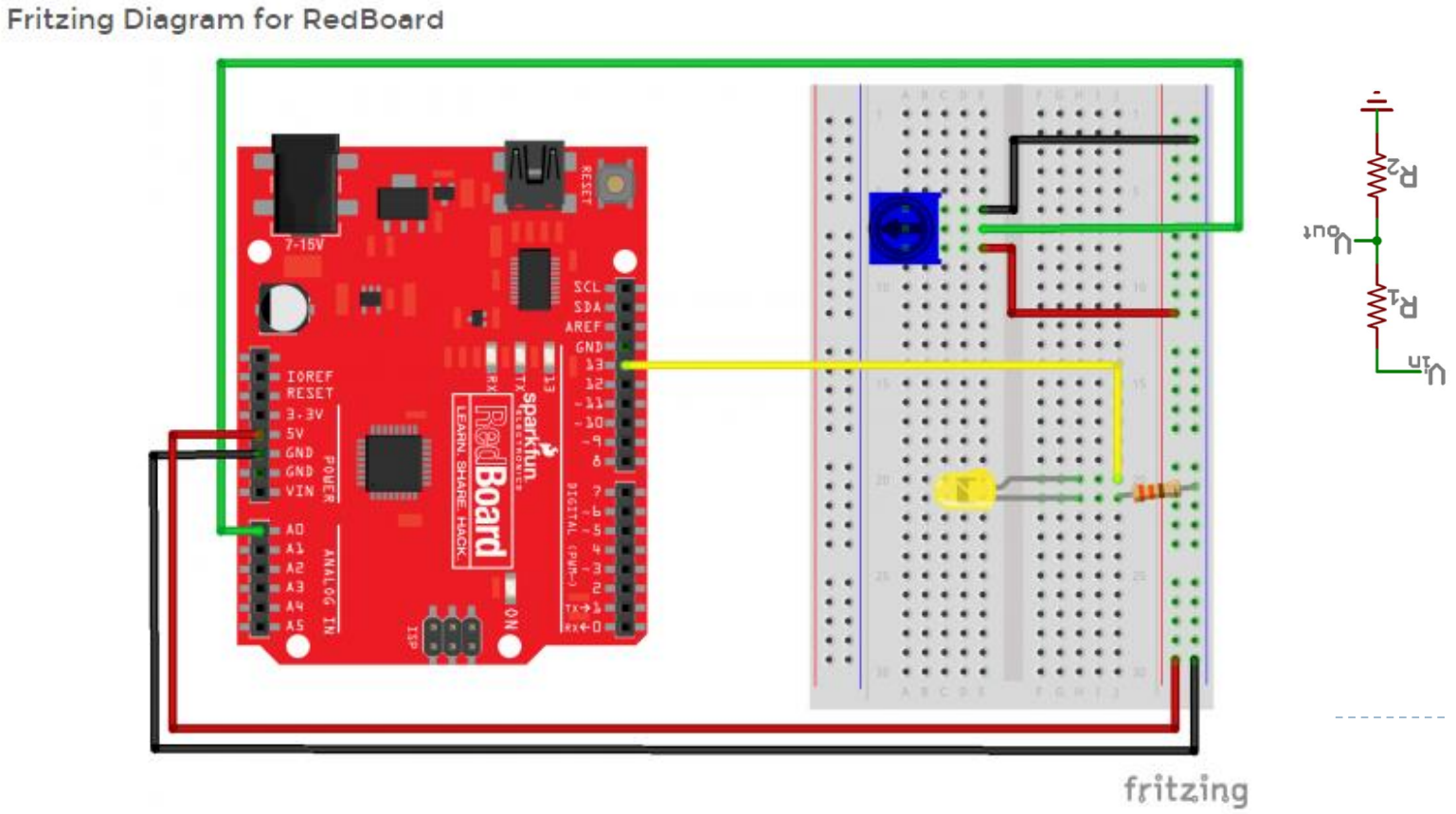
You will need the following parts:

- **1x** Red Board
- **1x** Breadboard
- **1x** LED (polarized)
- **1x** 330Ω Resistor
- **6x** Jumper Wires
- **1x** Potentiometer



# Circuit #2: Potentiometer

- Build the following circuit.
- Open circuit #2 in your Arduino software
- Compile and upload to your board

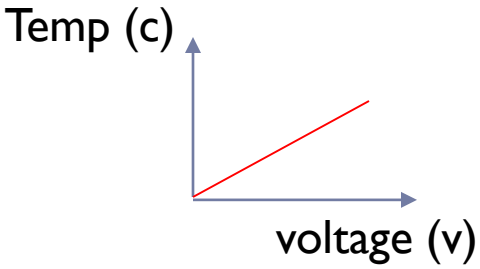


# SF-4 Challenge #1: dim LED using pot

- Control the brightness of LED with the potentiometer
- Turning the potentiometer all the way to the left turns LED off
- Turning Pot all the way to the right, turns LED ON
- Since you are dimming an LED use a PWM pin

# A temperature sensor


- Polarized component
- Measures ambient temperature
- Three pins – a positive, a ground, and a signal
- The signal pin outputs the change in voltage due to change in temperature to the analog pin on the Redboard.
- Linear temperature sensor:




- TMP36: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/TMP35_36_37.pdf

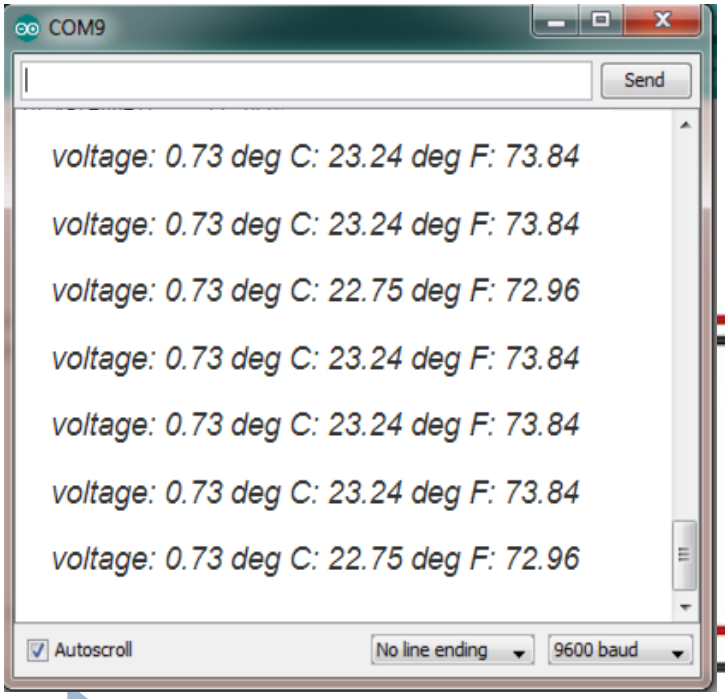
# Reading a temperature sensor

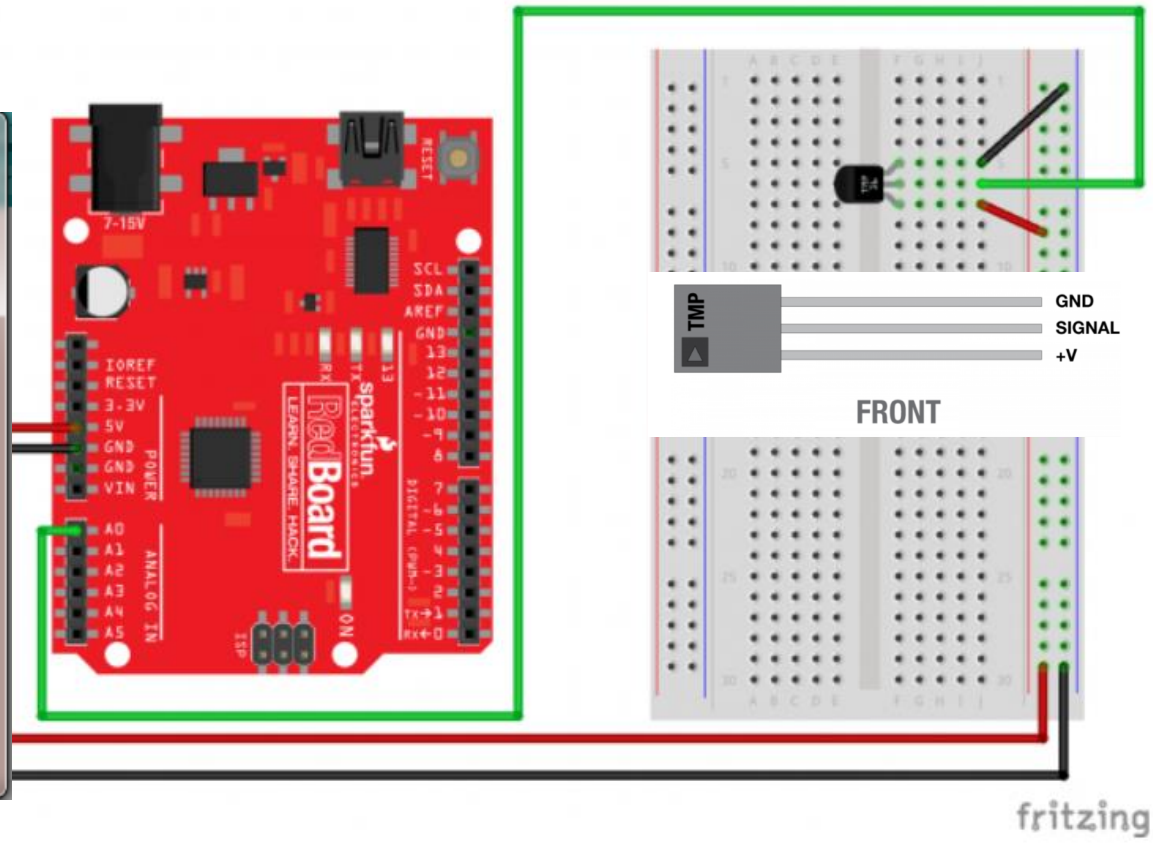
You will need the following parts:

- 1x RedBoard
- 1x Breadboard
- 5x Jumper Wires
- 1x Temperature Sensor → sends analog signal to the board so we use analog pins on the board (A0)


# Circuit #7: Reading a temperature sensor

- Build the following circuit.
- Download circuit #7 from BB
- Open in Arduino, compile and upload to your board



# Circuit #7 Reading a temperature sensor script

- Declare and initialize: const int temperaturePin=A0
  float voltage, degreesC, defreesF

- Reads voltage on analog pin A0:

voltage = analogRead(temperaturePin)

How do we translate voltage to temperature?

- convert voltage to degreeC

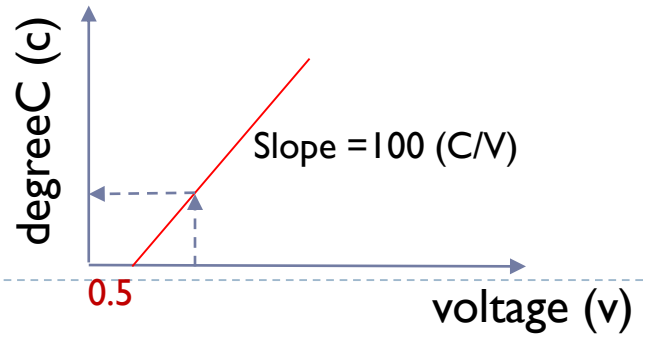
Table 4. TMP3x Output Characteristics

| 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 |

1 v = 100 C

(voltage - 0.5)

degreesC = (voltage - 0.5) * 100



# Circuit #7 Reading a temperature sensor script

- Declare and initialize: const int temperaturePin=A0
  float voltage, degreesC, defreesF

- Reads voltage on analog pin A0:

voltage = analogRead(temperaturePin)

returns a value 0-1023

- converts voltage to degreeC:

degreesC = (voltage - 0.5) * 100.0

Needs a voltage 0-5 v

# Circuit #7 Reading a temperature sensor script

- Declare and initialize: const int temperaturePin=A0
  float voltage, degreesC, defreesF

- Reads voltage on analog pin A0:

voltage = analogRead(temperaturePin) * 0.004882814

returns a value 0-1023

=5(v)/1023
Converts 0-1023 to 0-5 to find the true voltage

- converts voltage to degreeC:

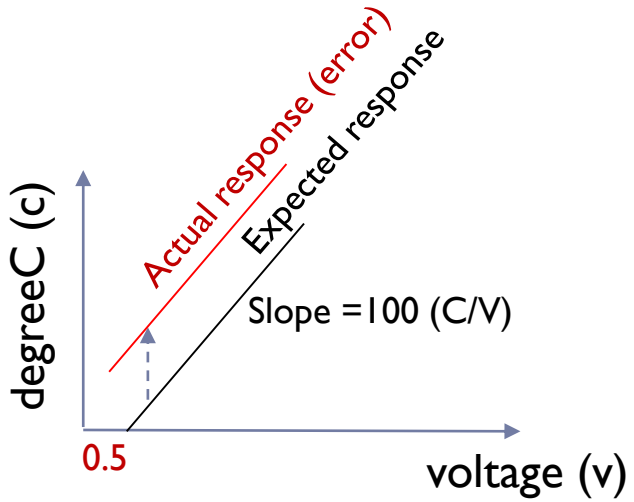
degreesC = (voltage - 0.5) * 100.0

Needs a voltage 0-5 v

For precise calculations (accurate to 3 decimal places), avoid map() and implement the calculations manually in your code yourself.

# What is calibration

- A method of improving **sensor** performance by removing errors in the **sensor** outputs. Errors are differences between a **sensors** expected output and its measured output, which show up consistently every time a new measurement is taken.




One-point calibration

# Why calibrate?

- **No Sensor is perfect!**
  - Two sensors from the same manufacturer production run may yield slightly different readings.
  - Sensors subject to heat, cold, shock, humidity etc. during storage, shipment and/or assembly may show a change in response.
  - Sensors age! And require re-calibration
- **The Sensor is only one component in the measurement system**
  - Example: for analog sensor (temp, photo resistor, potentiometer,..), ADC is also part of the measurement

# How to calibrate?

- **Standard References:**
  - **Compare the results against a calibrated sensor**
  - **Use sensor data sheet:** outputs 750 mV at 25°C (TMP36 data sheet)

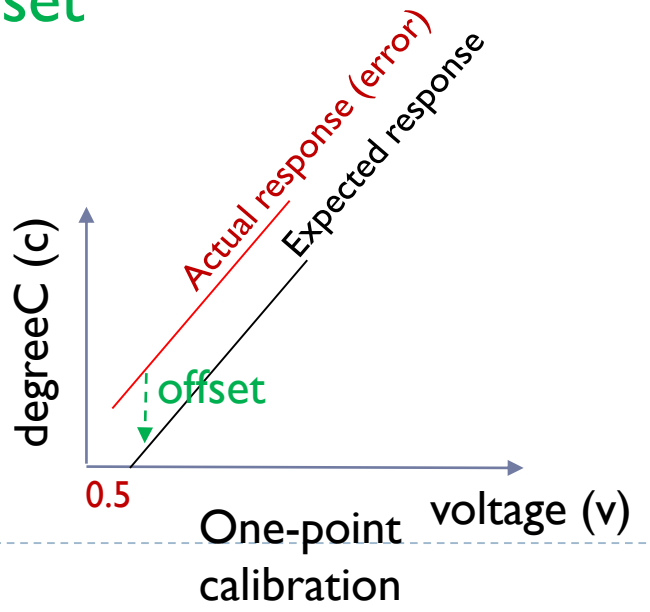
- **Standard physical reference:**
  - **Range finders:** Rulers and Meter sticks
  - **Temperature:** Ice-water Bath - The "Triple Point" of water is 0.01°C at sea-level
  - **Accelerometers:** Gravity is a constant 1G on the surface of the earth.

# Let's calibrate temperature sensor

1. Take a measurement with your sensor (degreeC actual)
2. Compare that measurement with your standard reference (degreeC expected)
3. offset= (degreeC actual – degreeC expected)
4. Update the equation by subtracting offset value:

degreeC=(voltage - 0.5) * 100.0 – offset

Compile, run and verify!



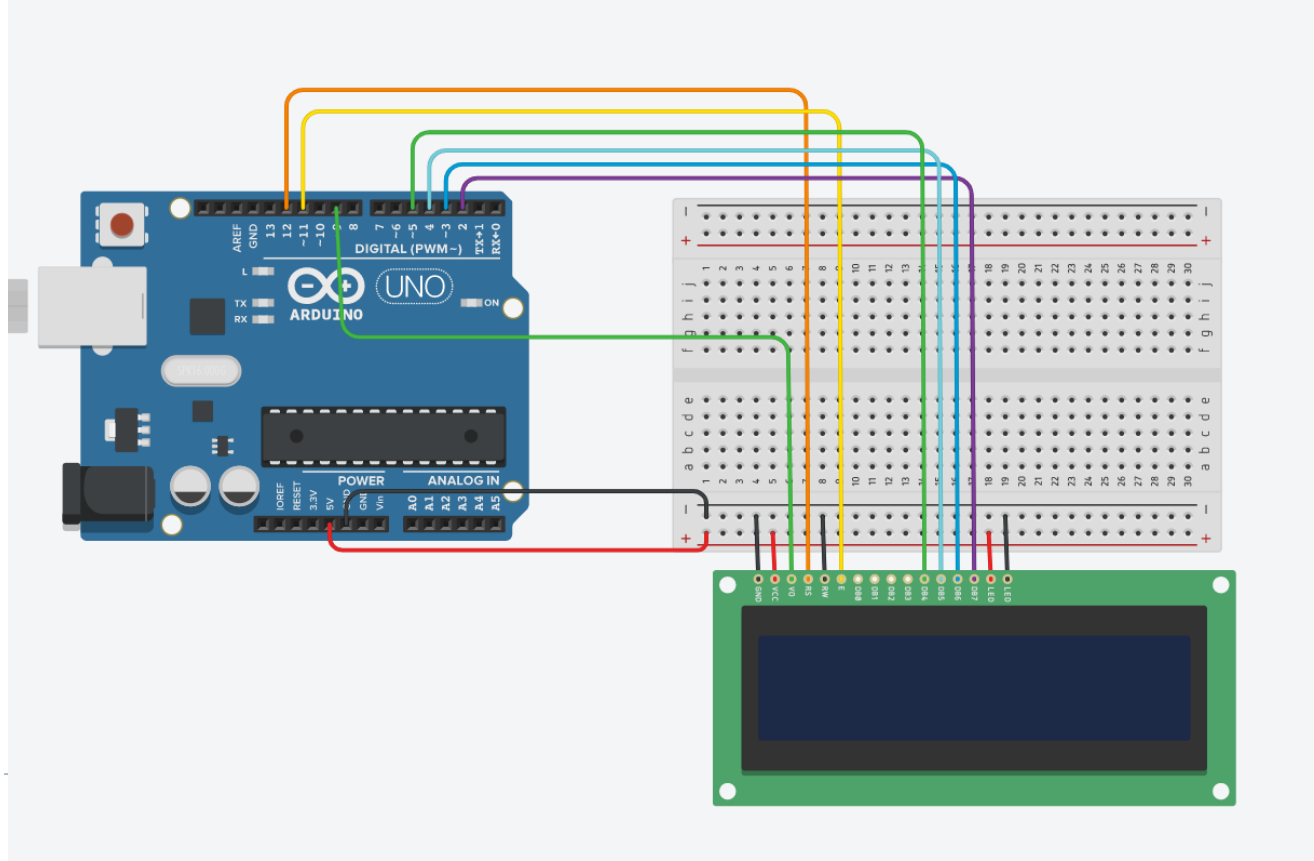

# Using an LCD

You will need the following parts:

- 1x RedBoard
- 1x Breadboard
- 13x Jumper Wires
- 1x LCD : a basic 16 character by 2 line display, display commands, bits of information, or readings from your sensor


# Circuit #15: LCD

- Build the following circuit.
- Open circuit #15LCDscreenv2_NoPot in Arduino
- Compile and upload to your board



# Circuit #15: LCD script

- LiquidCrystal Library:
  - #include <LiquidCrystal.h>

- Declare an object from class LiquidCrystal and use functions:
  - LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

Class Object properties

```
void setup(){
```

  - lcd.begin(16,2); //initializes16 columns, 2 rows

Object methods

  - lcd.clear();
  - lcd.print("hello, world!");

```
}
```

Function
- LiquidCrystal()
- begin()
- clear()
- home()
- setCursor()
- write()
- print()
- cursor()
- noCursor()
- blink()

https://www.arduino.cc/en/Reference/LiquidCrystal

# Circuit #15: LCD script

```
lcd.setCursor(0, 0); // top left
lcd.setCursor(15, 0); // top right
lcd.setCursor(0, 1); // bottom left
lcd.setCursor(15, 1); // bottom right
```

Function
- LiquidCrystal()
- begin()
- clear()
- home()
- setCursor()
- write()
- print()
- cursor()
- noCursor()
- blink()

- Void loop(){
  - lcd.setCursor(0, 1); //Set the (invisible) cursor to column 0, row 1.

Object

methods

  - lcd.print(millis() / 1000); //Print the number of seconds since reset.

}

- Built-in function: millis() →Returns the number of milliseconds since the Arduino board began running the current program

https://www.arduino.cc/en/Reference/LiquidCrystal

# SF-4 :
# Creating a home thermostat!

- Read the room temperature using the temperature sensor and print to serial monitor (you might have to calibrate it again)
- Set your desired room temperature using the pot. Print the set temperature in the serial monitor
  - SetTemp=analogRead(potPin); → returns a value 0-1023 from potentiometer
  - SetTemp=map(SetTemp,0,1023,20,100), squezzes the 0-1023 range to 20deg F - 100 degF. (or any other range)
  - Note: map is not converting voltage to temperature. It is only adjusting the pot range to something meaningful.
- if the room temperature is below the set temperature (the room is cold), LED turns on and display "heater on!" on the serial monitor.

# SF-4:
# Creating a home thermostat!

- Read the room temperature using the temperature sensor and print to serial monitor (you might have to calibrate it again)
- Set your desired room temperature using the pot. Print the set temperature in the serial monitor
  - SetTemp=analogRead(potPin); → returns a value 0-1023 from potentiometer
  - SetTemp=map(SetTemp,0,1023,20,100), squezzes the 0-1023 range to 20deg F - 100 degF. (or any other range)
  - Note: map is not converting voltage to temperature. It is only adjusting the pot range to something meaningful.
- if the room temperature is below the set temperature (the room is cold), LED turns on and display "heater on!" on the serial monitor.

# SF-4:
# Creating a home thermostat!

- Read the room temperature using the temperature sensor and print to serial monitor (you might have to calibrate it again)
- Set your desired room temperature using the pot. Print the set temperature in the serial monitor
  - SetTemp=analogRead(potPin); → returns a value 0-1023 from potentiometer
  - SetTemp=map(SetTemp,0,1023,20,100), squezzes the 0-1023 range to 20deg F - 100 degF. (or any other range)
  - Note: map is not converting voltage to temperature. It is only adjusting the pot range to something meaningful.
- if the room temperature is below the set temperature (the room is cold), LED turns on and display "heater on!" on the serial monitor.

# SF-4:
# Creating a home thermostat!

- Read the room temperature using the temperature sensor and print to serial monitor (you might have to calibrate it again)
- Set your desired room temperature using the pot. Print the set temperature in the serial monitor
  - SetTemp=analogRead(potPin); → returns a value 0-1023 from potentiometer
  - SetTemp=map(SetTemp,0,1023,20,100), squezzes the 0-1023 range to 20deg F - 100 degF. (or any other range)
  - Note: map is not converting voltage to temperature. It is only adjusting the pot range to something meaningful.
- if the room temperature is below the set temperature (the room is cold), LED turns on and display "heater on!" on the serial monitor.
- If the room is warmer then another LED lights up and display "AC on" on the monitor