LECTURE 5

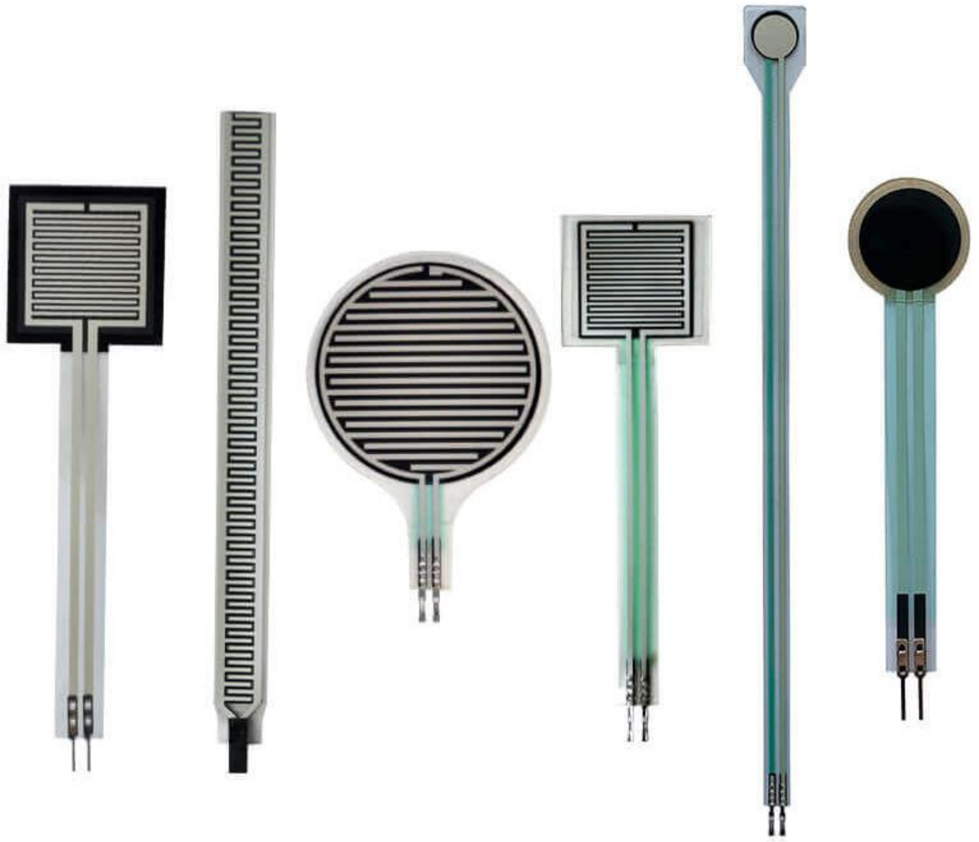
Force Sensor and Application: Control Multiple LEDs

Instructor: Osman Gul, Ph.D. Candidate
Department of Mechanical Engineering
Korea Advanced Institute of Science and Technology (KAIST)

Syllabus:

Week	Date& Time*	Content
1		No Class (Course Add/Drop Period)
2	March 6 (Mon) 20-22	Orientation and Course Overview
3	March 13 (Mon) 20-22	Sensor Fundamentals
4	March 20 (Mon) 20-22	Arduino Fundamentals
5	March 27 (Mon) 20-22	Arduino Programming and Applications
6	April 3 (Mon) 20-22	Force Sensor and Application: Control Multiple LEDs
7	April 10 (Mon) 20-22	Light Sensor and Application: Solar Tracker
8	April 17	No Class (Midterm)
9	April 24 (Mon) 20-22	Humidity and Temperature Sensor and Application: Temperature Controlled Fan
10	May 1 (Mon) 20-22	Gas Sensor and Application: Smoke Detector
11	May 8 (Mon) 20-22	Sound Sensor and Application: Control LED by Clapping
12	May 15 (Mon) 20-22	Accelerometer Sensor and Application: Ping Pong Game
13	May 22 (Mon) 20-22	Ultrasonic Sensor and Application: Flappy Bird Game
14	May 29 (Mon) 20-22	Course Wrap-up
15	June 5	No Class (Final)
16	June 12	No Class (Final)

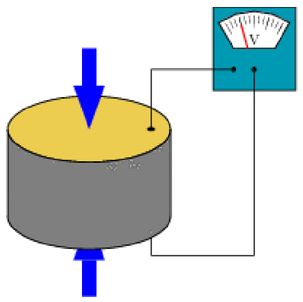
Force Sensing Resistor (FSR)

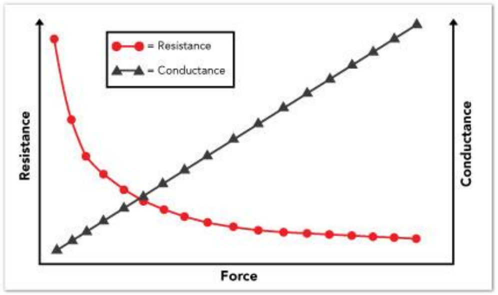


- ✓ Force sensing resistors are a piezoresistive sensing technology. This means they are passive elements that function as a variable resistor in an electrical circuit.

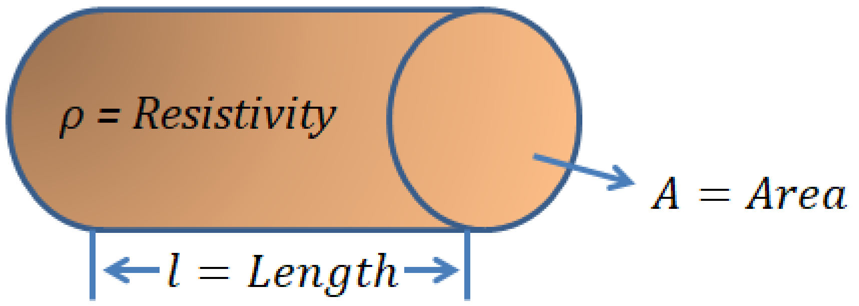
- ✓ Piezo is derived from the Greek πιέζω, which means to **squeeze** or **press**

- ✓ The **piezoresistive effect** is a change in the electrical resistivity of a semiconductor or metal when mechanical strain is applied. In contrast to the piezoelectric effect, the piezoresistive effect causes a change only in **electrical resistance**, not in electric potential.

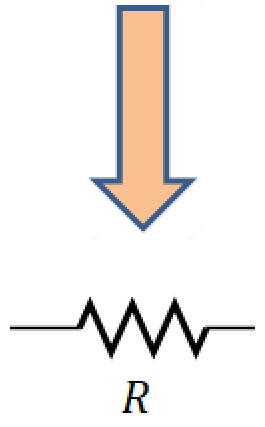




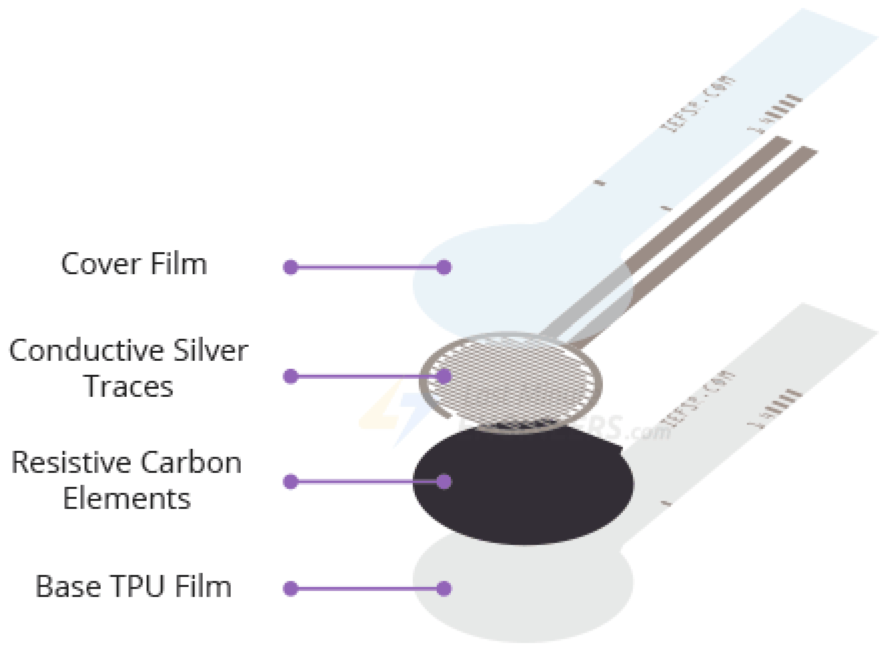
- ✓ When unloaded, the sensor has a high resistance (on the order of Megaohms (MΩ)) that drops as force is applied (usually on the order of Kiloohms (KΩ)).
- ✓ When you consider the inverse of resistance (conductance), the conductance response as a function of force is linear within the sensor's designated force range.

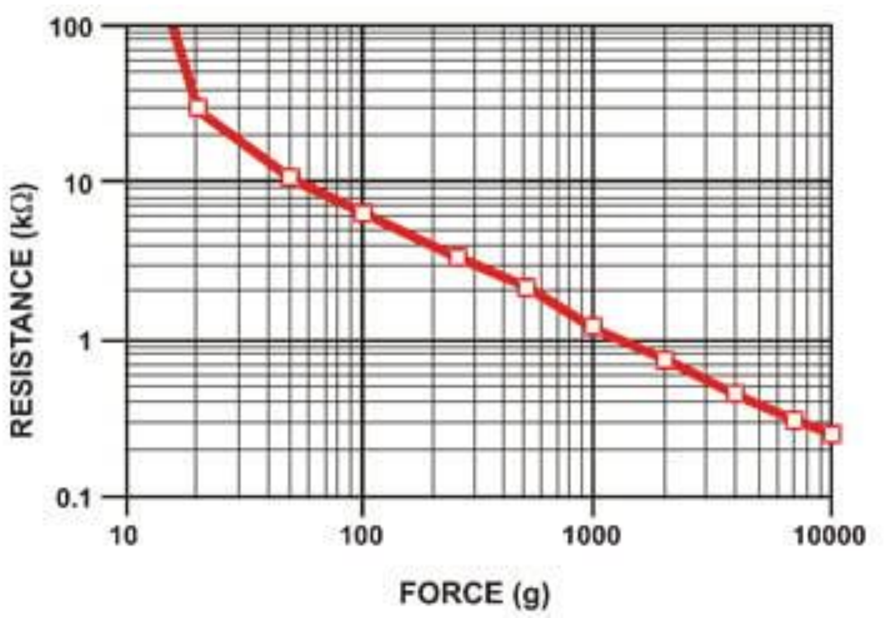


$$R = \frac{\rho l}{A}$$



✓ It is made up of several thin, flexible layers. When squeezed, more of the carbon elements that normally offer resistance are brought into contact with the conductive traces, thereby lowering the resistance.

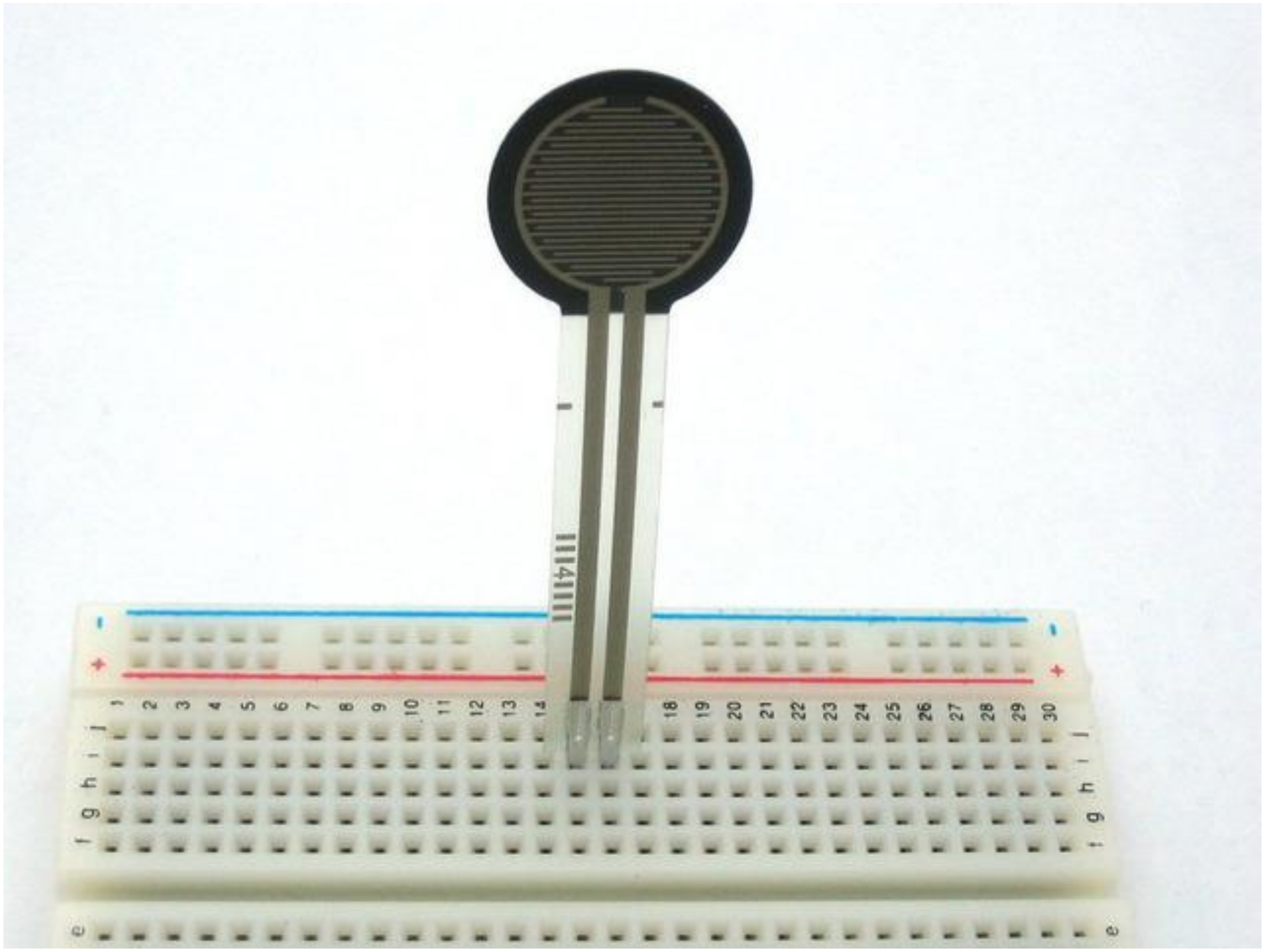




How FSR works?

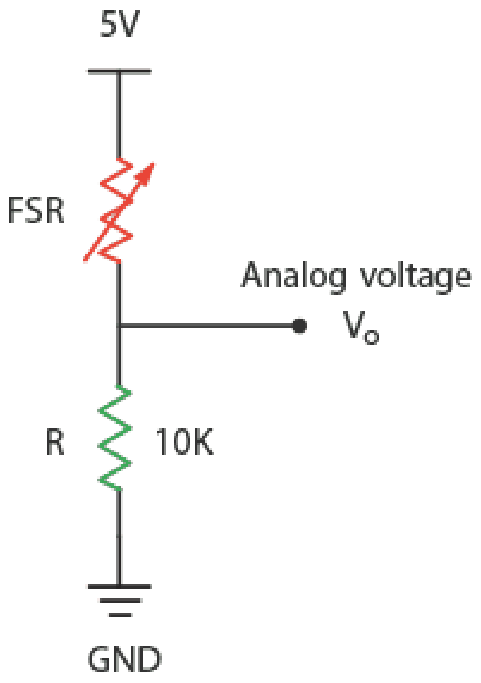
Max
Min
Resistance
Last Minute ENGINEERS.com

✓ In the absence of pressure, the sensor will read infinite resistance (larger than 1MΩ). Applying more pressure to the sensor's head reduces the resistance between its terminals, while releasing the pressure restores the resistance to its original value.



Reading an FSR

- ✓ The simplest way to read the FSR is to combine it with a static resistor to form a voltage divider, which produces a variable voltage that can be read by the analog-to-digital converter of a microcontroller.

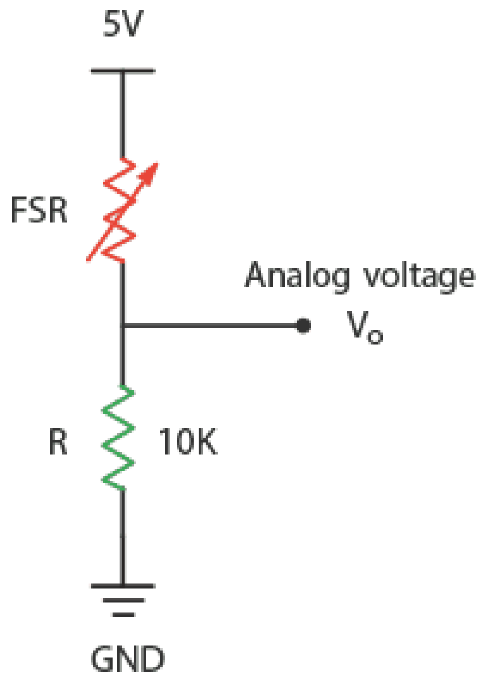


- ✓ It is important to note that the output voltage you measure is the voltage drop across the pull-down resistor, not the voltage drop across the FSR.

- ✓ We can use this equation to calculate the output voltage (Vo).

$$V_O = V_{CC}\frac{R}{R+FSR}$$

https://lastminuteengineers.com/

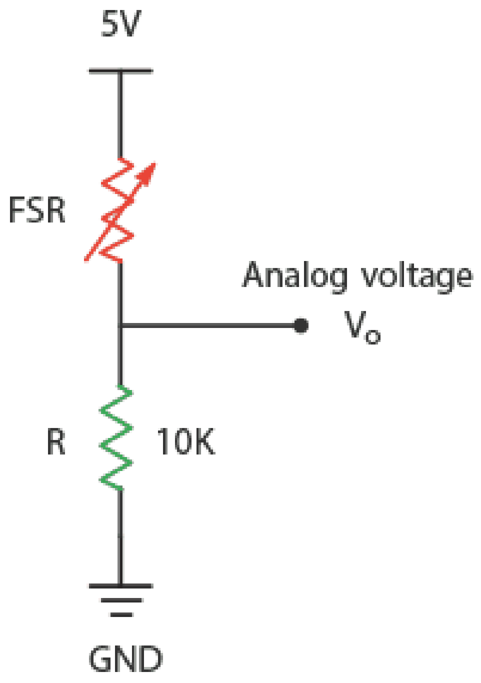


- ✓ In this configuration, the output voltage increases as the applied force increases.
- ✓ For example, with a 5V supply and a 10K pull-down resistor, when there is no pressure, the FSR resistance is extremely high (around 10M). This produces the following output voltage:

$$V_O = 5V\frac{10k\Omega}{10k\Omega + 10M\Omega}$$

$$= 0.005V$$

$$\approx 0V$$

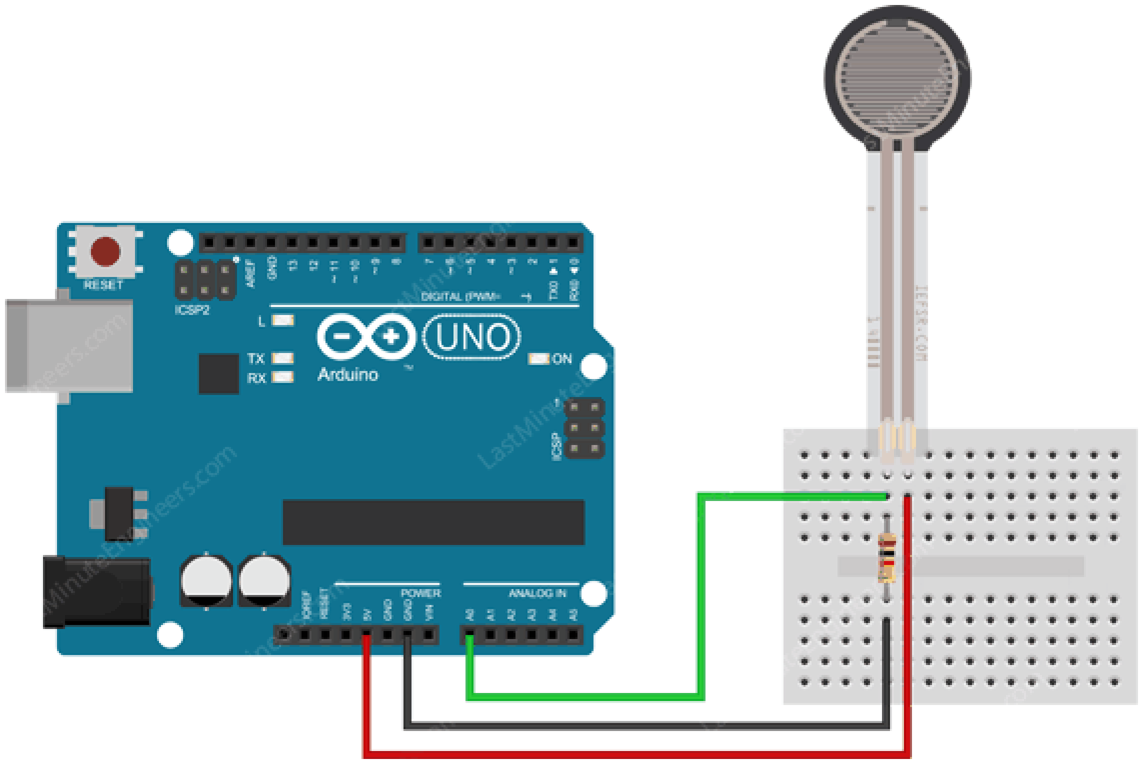


- ✓ If you apply significant force to the FSR, the resistance will drop to approximately 250 Ω. As a result, the output voltage becomes:

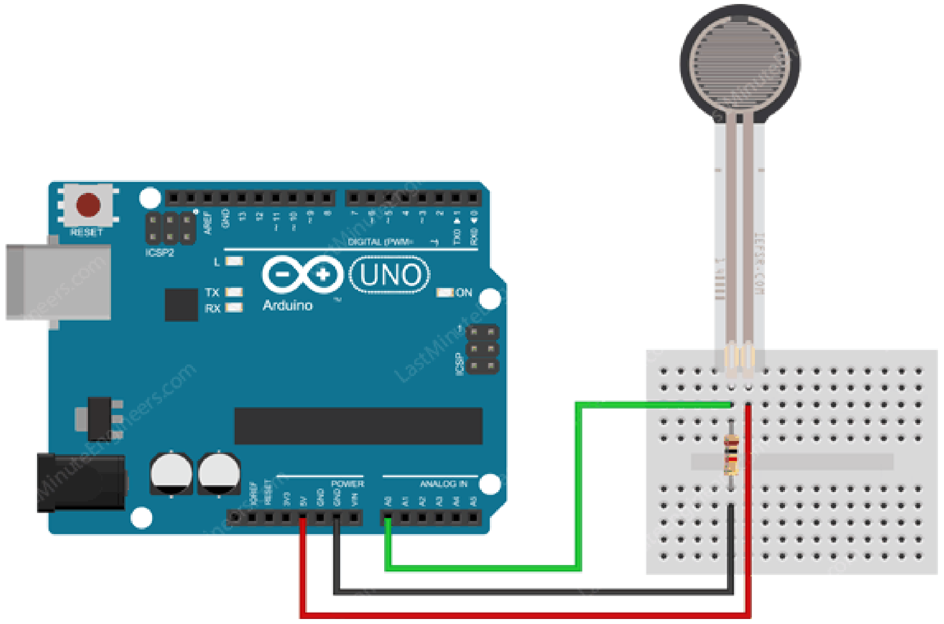
$$V_O = 5V \frac{10k\Omega}{10k\Omega + 250\Omega}$$
$$= 4.9V$$
$$\approx 5V$$

- ✓ As you can see, the output voltage varies from 0 to 5V depending on the amount of force applied to the sensor.

Wiring an FSR to an Arduino



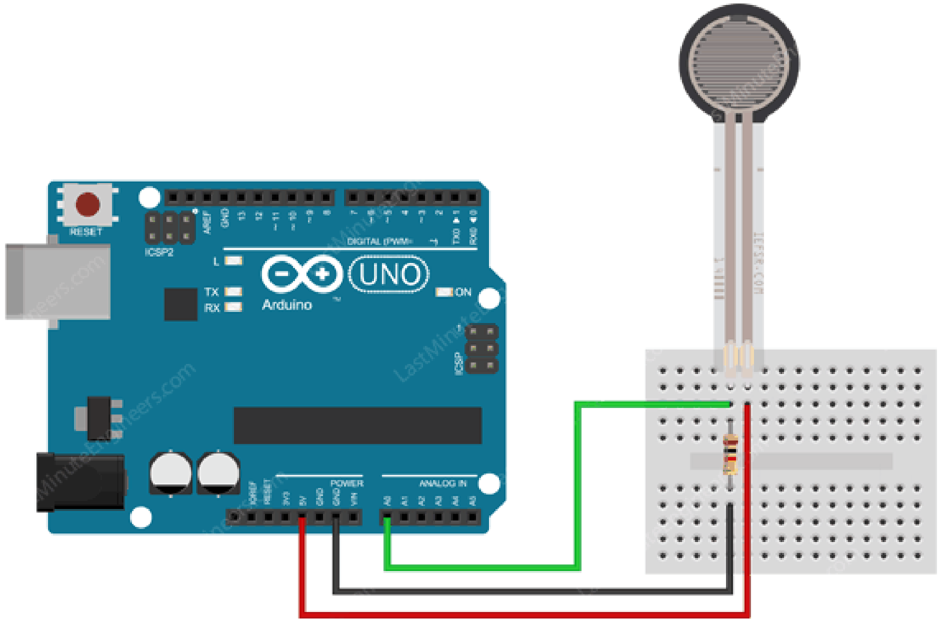
- ✓ You need to connect a **10kΩ** pull-down resistor in series with the FSR to create a voltage divider circuit. Next, the A0 ADC input of an Arduino is wired to the junction of the pull-down resistor and the FSR.



- ✓ Keep in mind that FSRs are really just resistors, which means you can connect them either way and they will still work.

Simple Analog FSR Measurements

```
int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrReading;     // the analog reading from the FSR resistor divider

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  fsrReading = analogRead(fsrPin);

  Serial.print("Analog reading = ");
  Serial.println(fsrReading);     // print the raw analog reading

  delay(1000);
}
```



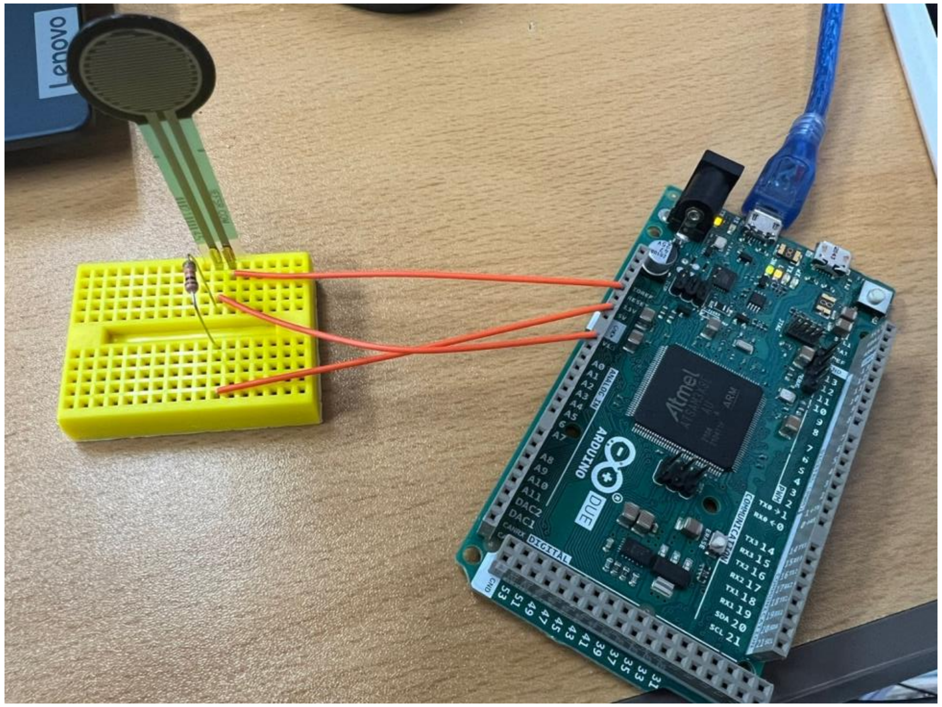
- ✓ The sketch begins with the declaration of the Arduino pin to which FSR is connected. Next, we define the fsrReading variable to store the FSR's raw analog reading.

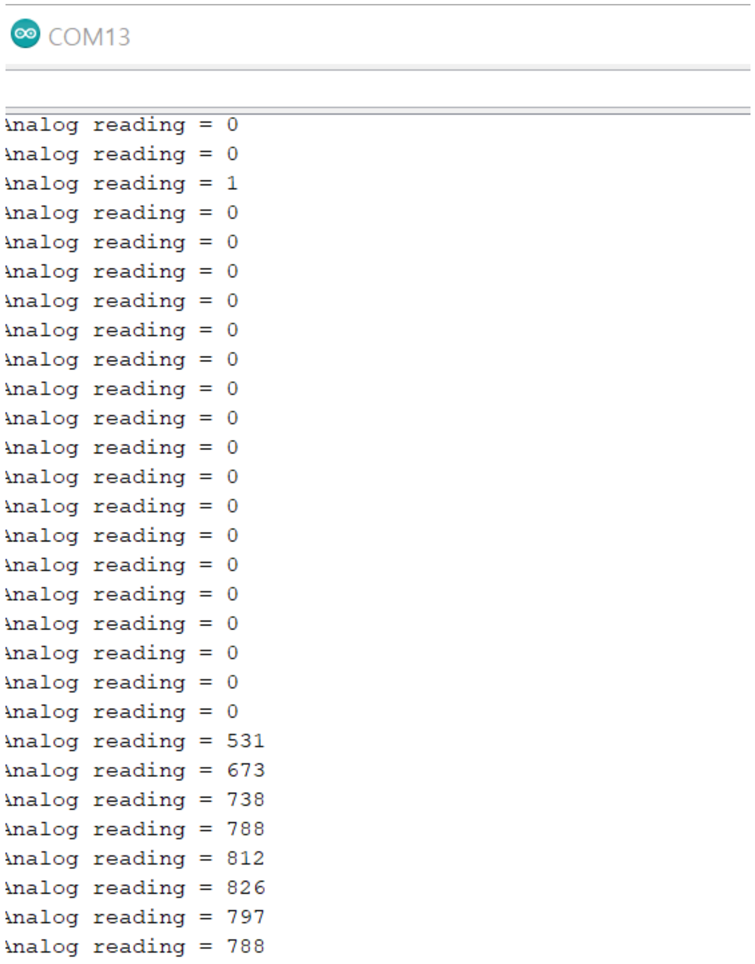
```
int fsrPin = 0;
int fsrReading;
```

- ✓ In the setup function, serial communication is established.

```
void setup(void) {
 Serial.begin(9600);
}
```

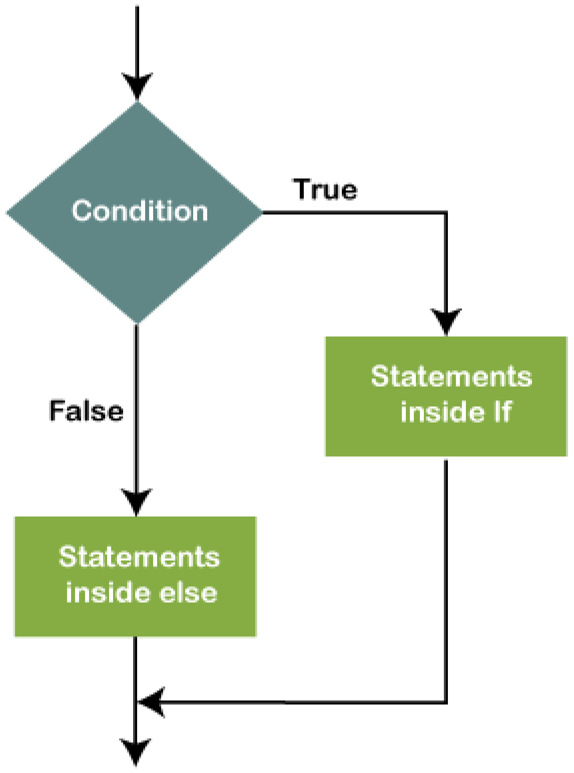
```
void loop(void) {
 fsrReading = analogRead(fsrPin);

 Serial.print("Analog reading = ");
 Serial.println(fsrReading);     // print the raw analog reading

 delay(1000);
}
```

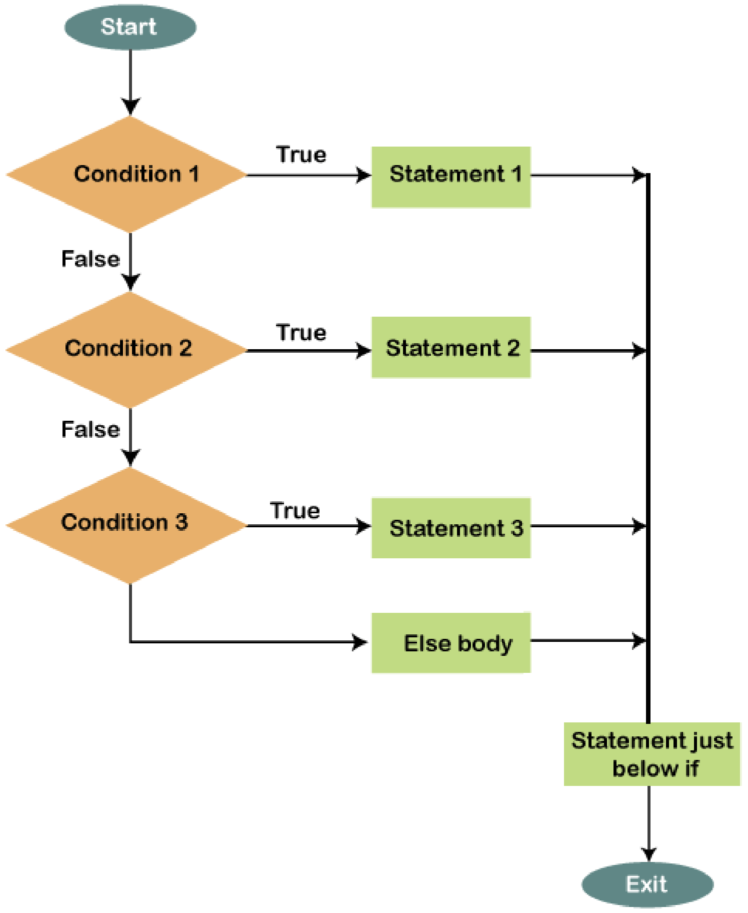




```
Analog reading = 0
Analog reading = 0
Analog reading = 1
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 0
Analog reading = 531
Analog reading = 673
Analog reading = 738
Analog reading = 788
Analog reading = 812
Analog reading = 826
Analog reading = 797
Analog reading = 788
```

If loop:


Condition
True
False
Statements inside If
Statements inside else


Start
Condition 1
True
Statement 1
False
Condition 2
True
Statement 2
False
Condition 3
True
Statement 3
Else body
Statement just below if
Exit

If loop:

```
if (condition) {
    body;
}
```

```
if (condition) {

body;

 }
```

- x > y: x is greater than y
- x < y: x is less than y
- x >= y: x is greater than or equal to y
- x <= y: x is less than or equal to y
- x == y: x equals y
- x != y: x is not equal to y

```
if (condition) {

body;

 }
```

```
if (x < y) {

Serial.println("x is less than y");

}
```

```
if (condition) {

body; // executed if the condition is
true

}

else{
body; // executed if the condition is
false

}
```

```
if (condition 1) {

// code executed if condition 1 is true

}
else if (condition 2) {

// execute code only if condition 1 is false and
condition 2 is true

}
```

- ✓ In the loop, we read the FSR's analog output and show it on the serial monitor.

- ✓ As mentioned previously, the sensor's output voltage varies from 0V (no pressure applied) to approximately 5V (maximum pressure applied). When the Arduino converts this analog voltage to a digital value, it converts it to a 10-bit number between 0 and 1023. Therefore, the value displayed on the serial monitor will range from 0 to 1023 based on how hard you squeeze the sensor.

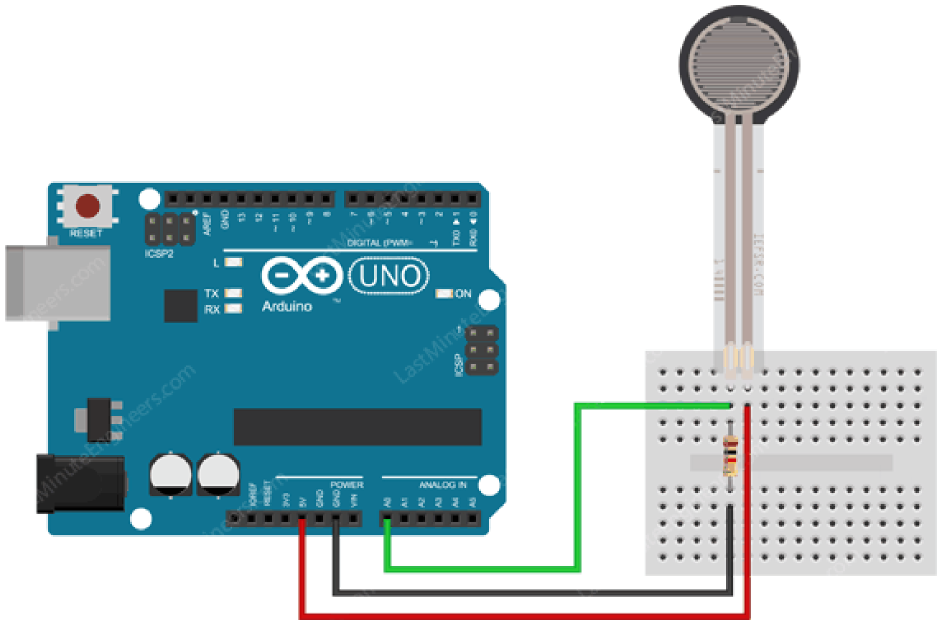
```
fsrReading = analogRead(fsrPin);

Serial.print("Analog reading = ");
Serial.print(fsrReading);
```

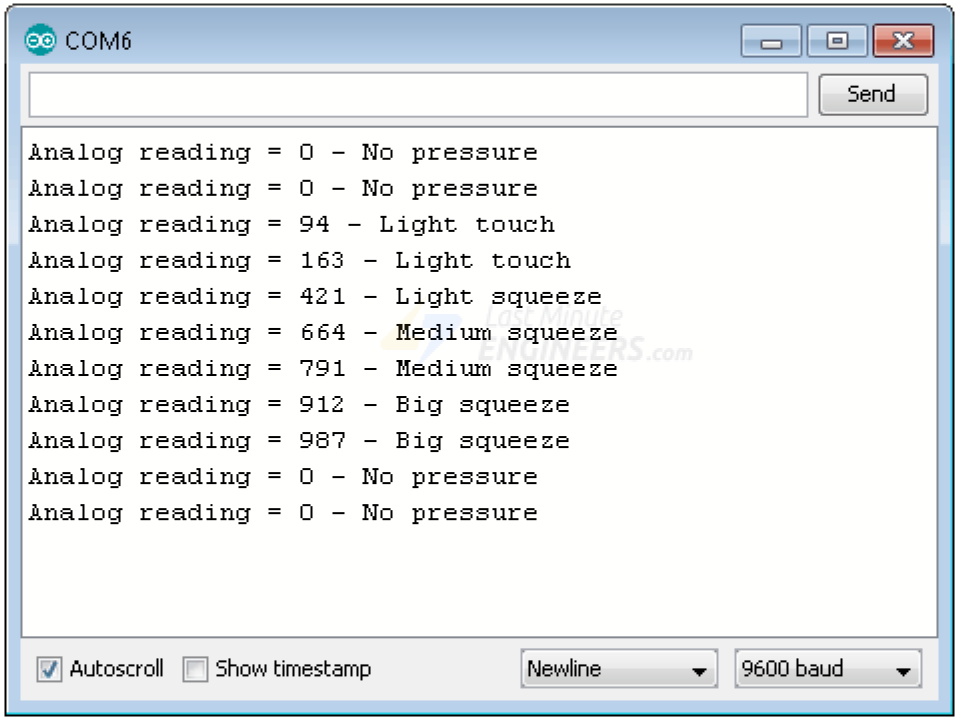
- ✓ Finally, we print the amount of pressure measured qualitatively.

```
if (fsrReading < 10) {
        Serial.println(" - No pressure");
} else if (fsrReading < 200) {
        Serial.println(" - Light touch");
} else if (fsrReading < 500) {
        Serial.println(" - Light squeeze");
} else if (fsrReading < 800) {
        Serial.println(" - Medium squeeze");
} else {
        Serial.println(" - Big squeeze");
}
```

Simple Analog FSR Measurements

```
int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrReading;     // the analog reading from the FSR resistor divider

void setup(void) {
  Serial.begin(9600);
}

void loop(void) {
  fsrReading = analogRead(fsrPin);

  Serial.print("Analog reading = ");
  Serial.print(fsrReading);     // print the raw analog reading

  if (fsrReading < 10) {
    Serial.println(" - No pressure");
  } else if (fsrReading < 200) {
    Serial.println(" - Light touch");
  } else if (fsrReading < 500) {
    Serial.println(" - Light squeeze");
  } else if (fsrReading < 800) {
    Serial.println(" - Medium squeeze");
  } else {
    Serial.println(" - Big squeeze");
  }
  delay(100);
}
```





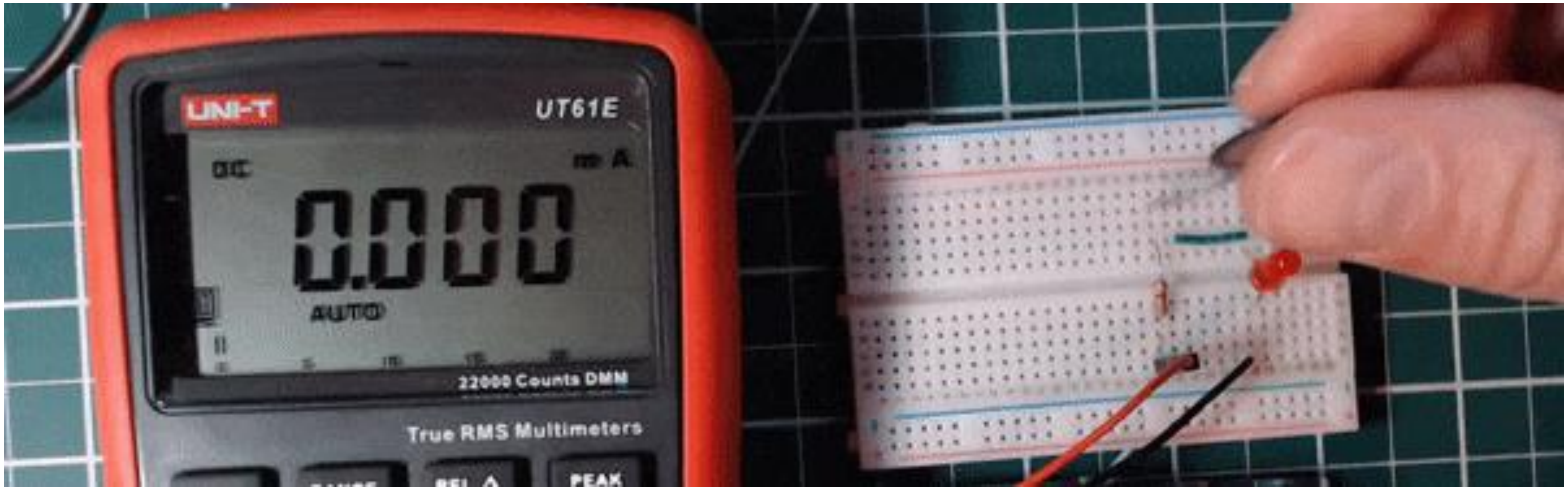
COM6

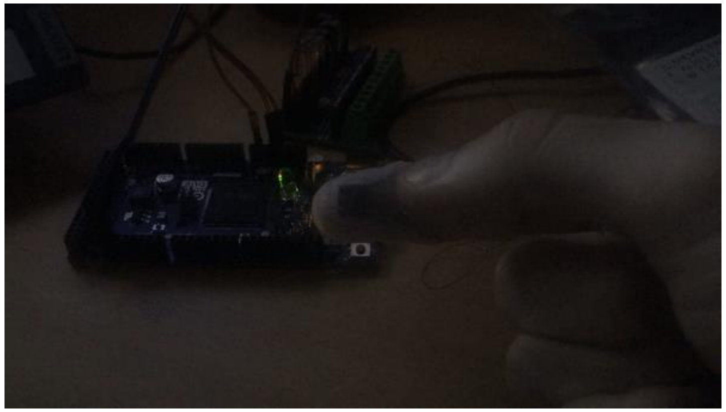
Send

```
Analog reading = 0 - No pressure
Analog reading = 0 - No pressure
Analog reading = 94 - Light touch
Analog reading = 163 - Light touch
Analog reading = 421 - Light squeeze
Analog reading = 664 - Medium squeeze
Analog reading = 791 - Medium squeeze
Analog reading = 912 - Big squeeze
Analog reading = 987 - Big squeeze
Analog reading = 0 - No pressure
Analog reading = 0 - No pressure
```

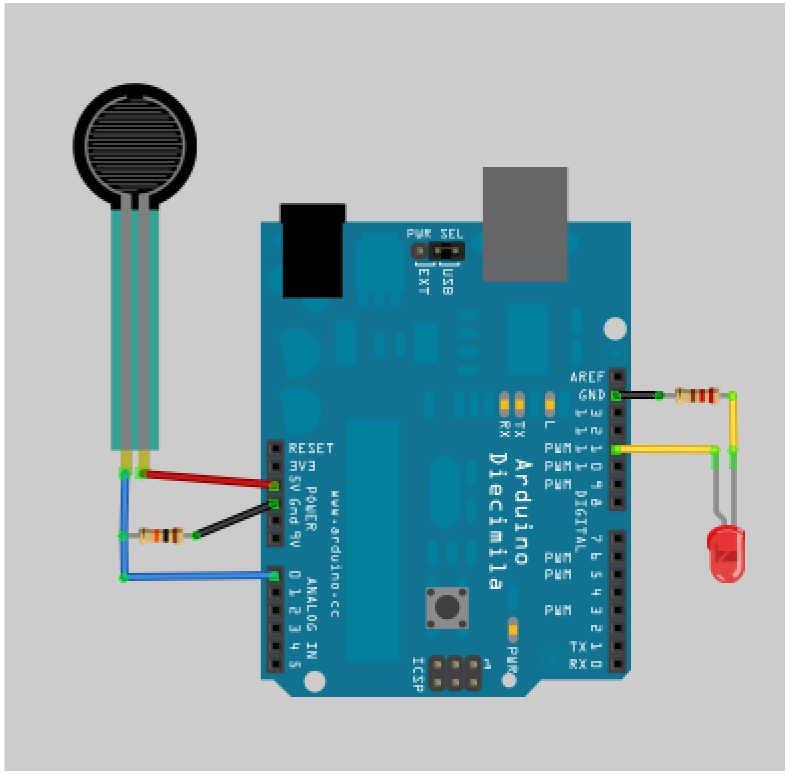
Autoscroll [✓] Show timestamp [ ] Newline 9600 baud

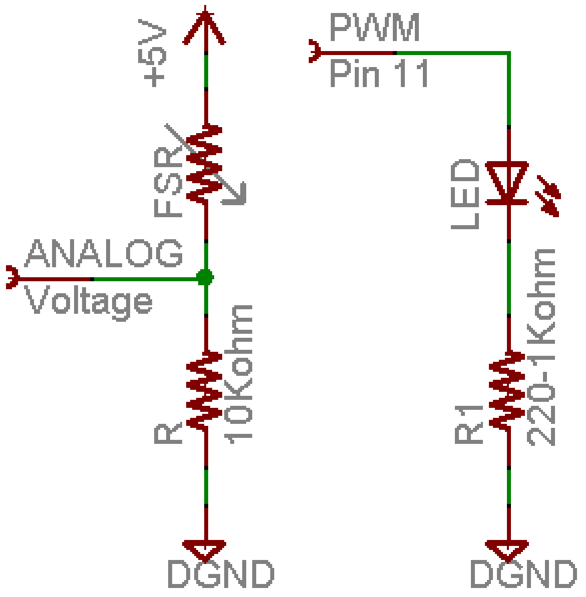
Controlling LED brightness with FSR:





Controlling LED brightness with FSR:





FSR-based LED fader

- ✓ For our FSR-based LED fade code, we′re going to read in the FSR value from the voltage divider using analogRead and then use this to proportionally set our LED brightness using analogWrite.

CONVERTING ANALOGREAD RANGE TO ANALOGWRITE RANGE USING MAP()

- ✓ analogRead uses a 10-bit ADC. Thus, the analogRead value ranges from 0-1023; however, the analogWrite value is 8 bits, which means that it ranges from 0-255. So, we have to do a simple conversion between the two ranges: 0-1023 to 0-255.

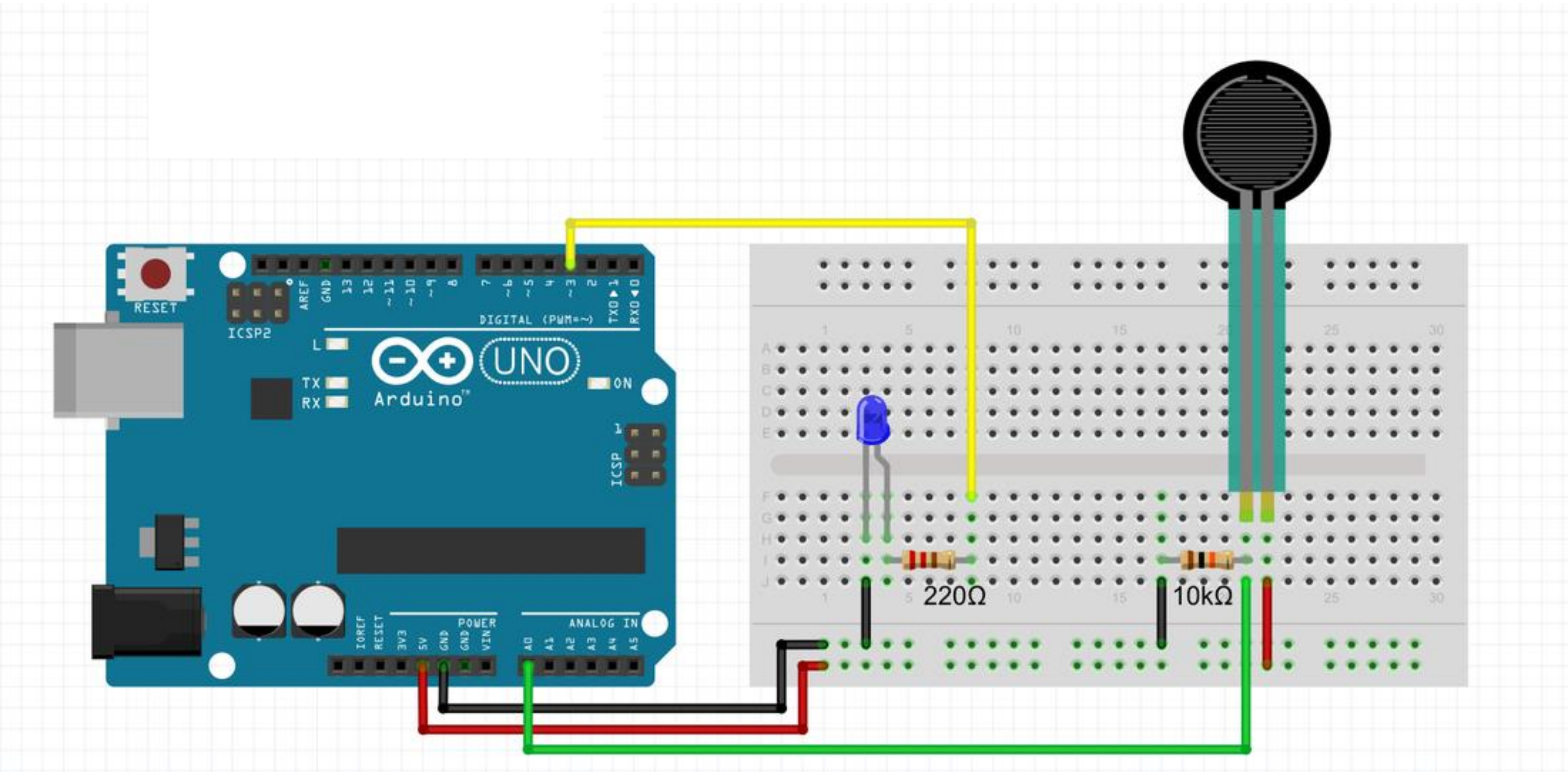
- ✓ If we assume both ranges start at zero (as they do in this case), our conversion is simply: int outputVal = (int)(inputVal/1023.0 * 255);.

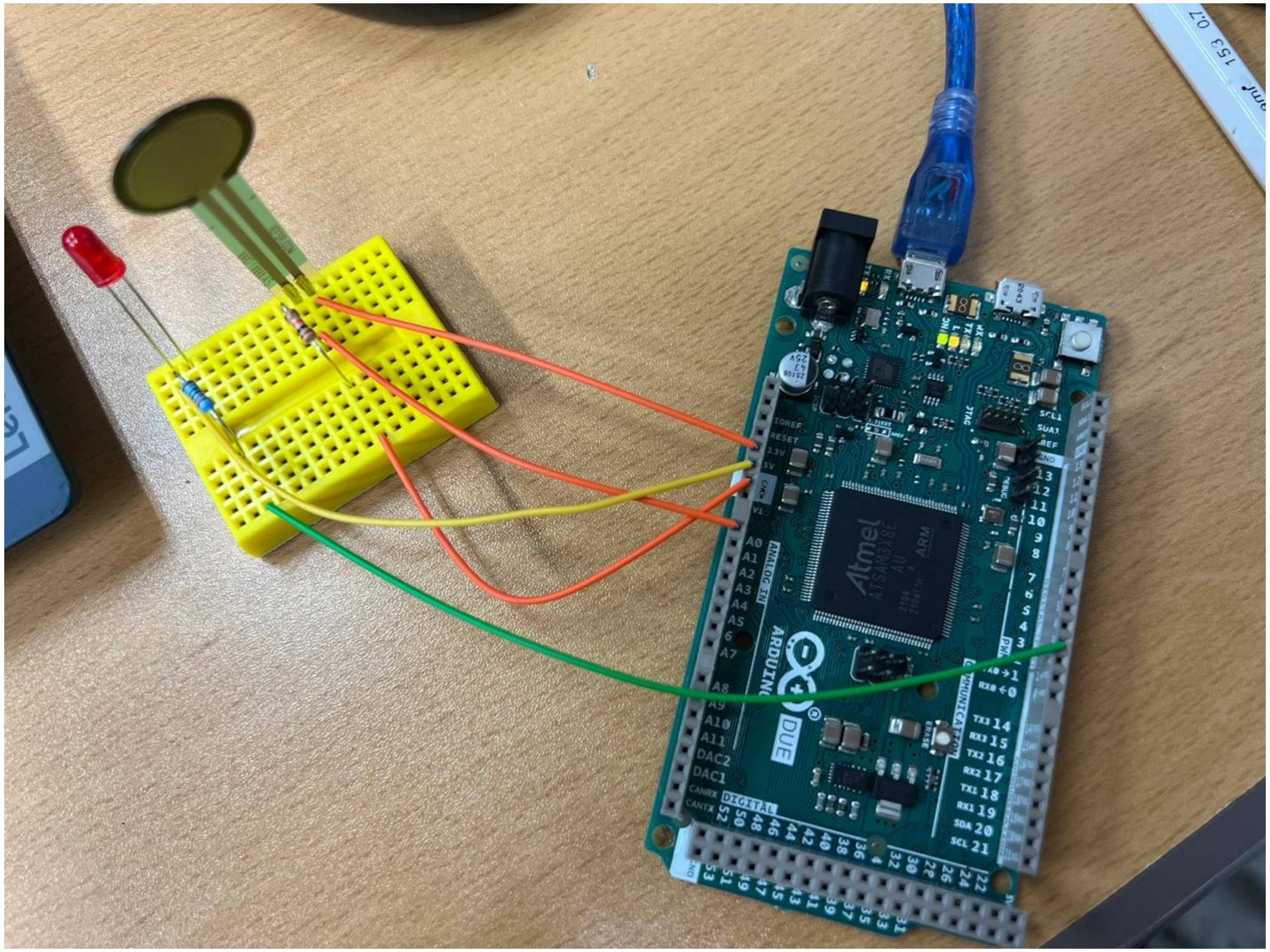
- ✓ If, instead, we can't assume that both ranges start at zero, the more general conversion algorithm is: int outputVal = OUTPUT_MIN + (inputVal - INPUT_MIN)/(INPUT_MAX - INPUT_MIN) * (OUTPUT_MAX - OUTPUT_MIN);

- ✓ This type of range conversion is so common that Arduino (and Processing and many other programming libraries) have a built-in function for it called map. It's called "map" because we want to re-map a number from one range to another. Indeed, here's the entire map function from the Arduino source code:

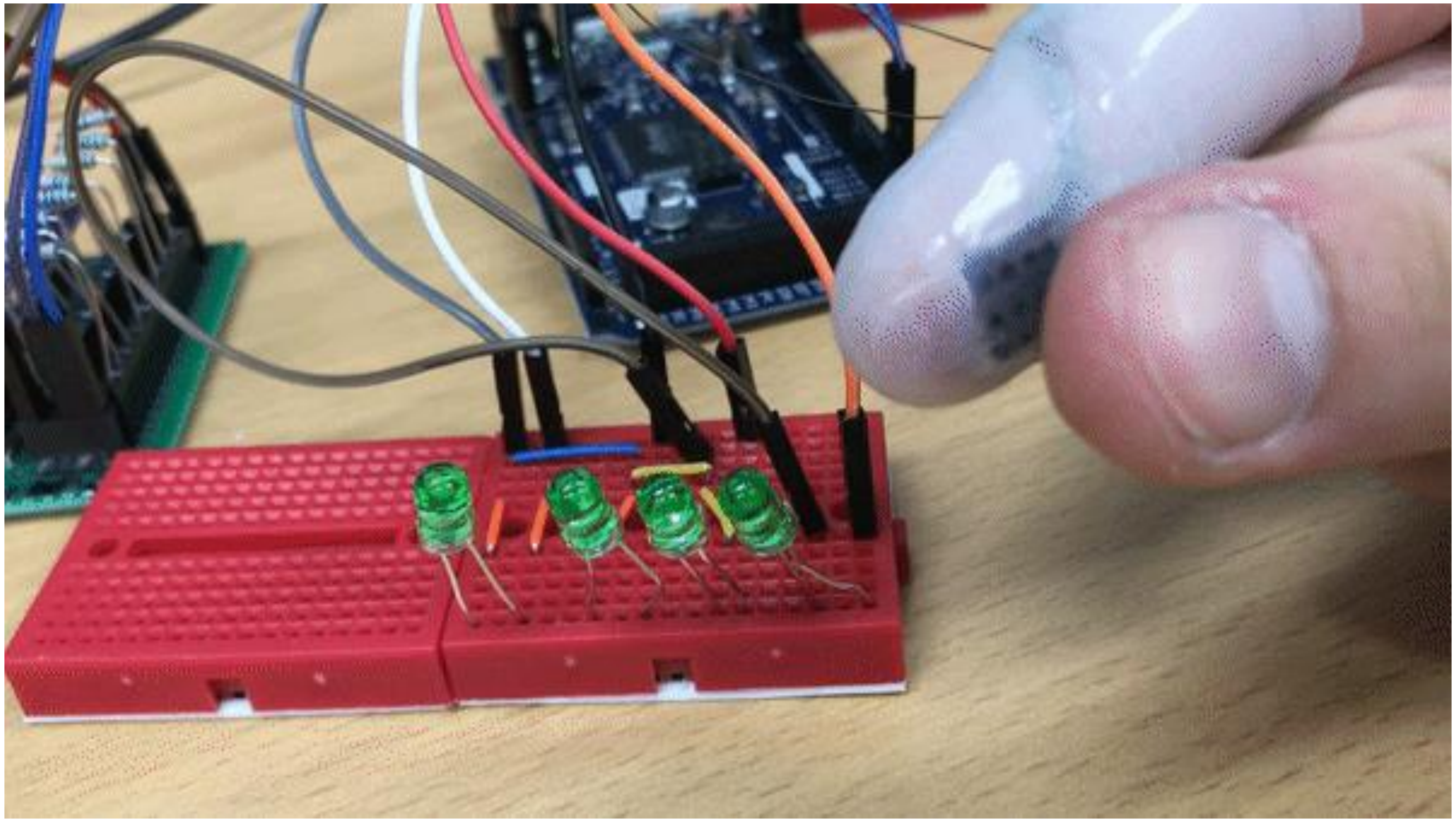
```
long map(long x, long in_min, long in_max, long out_min, long out_max) {
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

```
const int OUTPUT_LED_PIN = 3;
const int INPUT_FSR_PIN = A0;
const int DELAY = 50; // how often to read from the sensor input

void setup() {
  pinMode(OUTPUT_LED_PIN, OUTPUT);
  pinMode(INPUT_FSR_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {

  // Read the force-sensitive resistor value
  int fsrVal = analogRead(INPUT_FSR_PIN);

  // Remap the value for output.
  int ledVal = map(fsrVal, 0, 1023, 0, 255);

  // Print the raw sensor value and the converted led value (e,g., for Serial Plotter)
  Serial.print(fsrVal);
  Serial.print(",");
  Serial.println(ledVal);

  // Write out the LED value.
  analogWrite(OUTPUT_LED_PIN, ledVal);

  delay(DELAY);
}
```

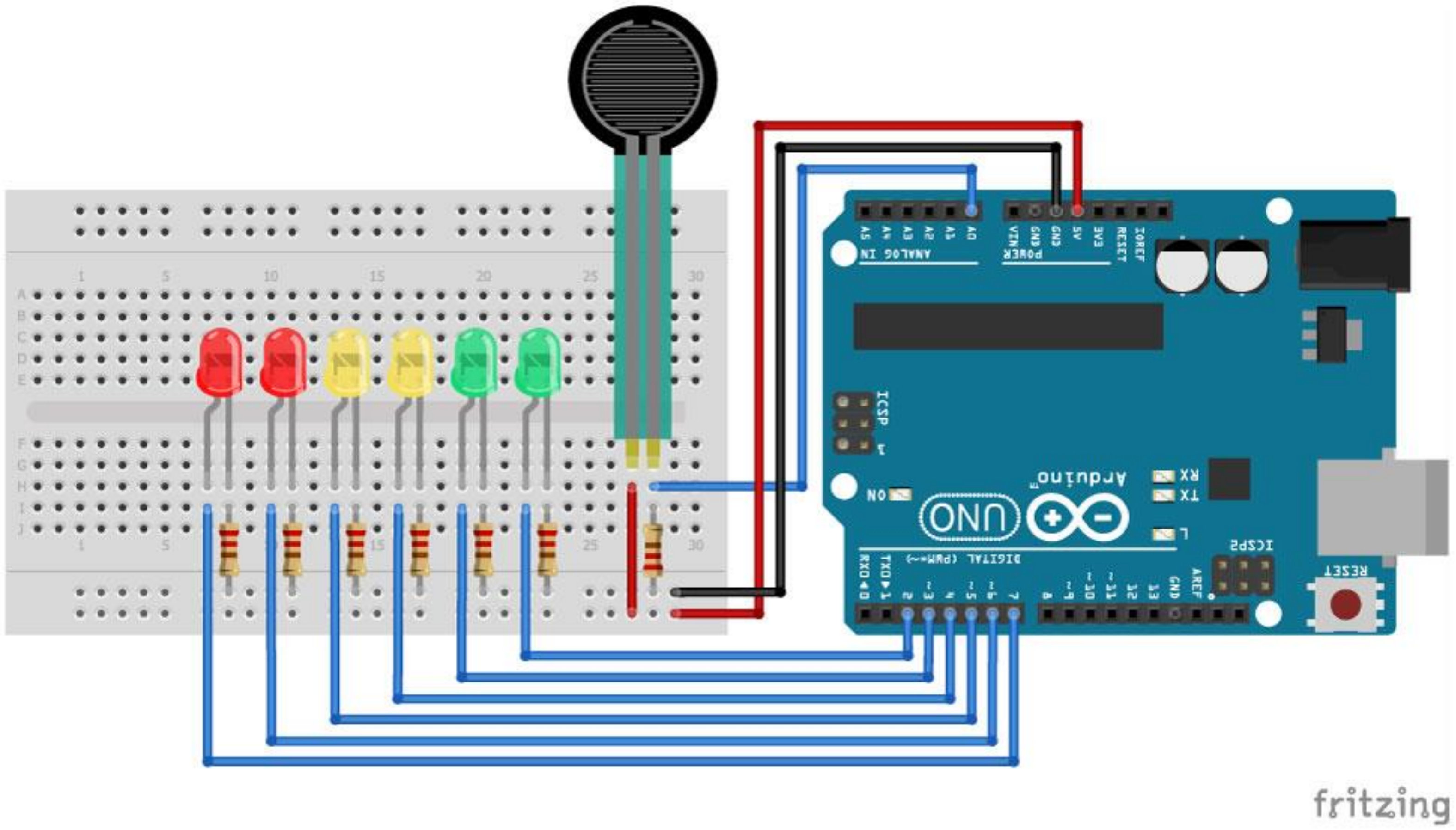

220Ω
10kΩ
UNO
Arduino
RESET
ICSP2
DIGITAL (PWM=~)
POWER
ANALOG IN

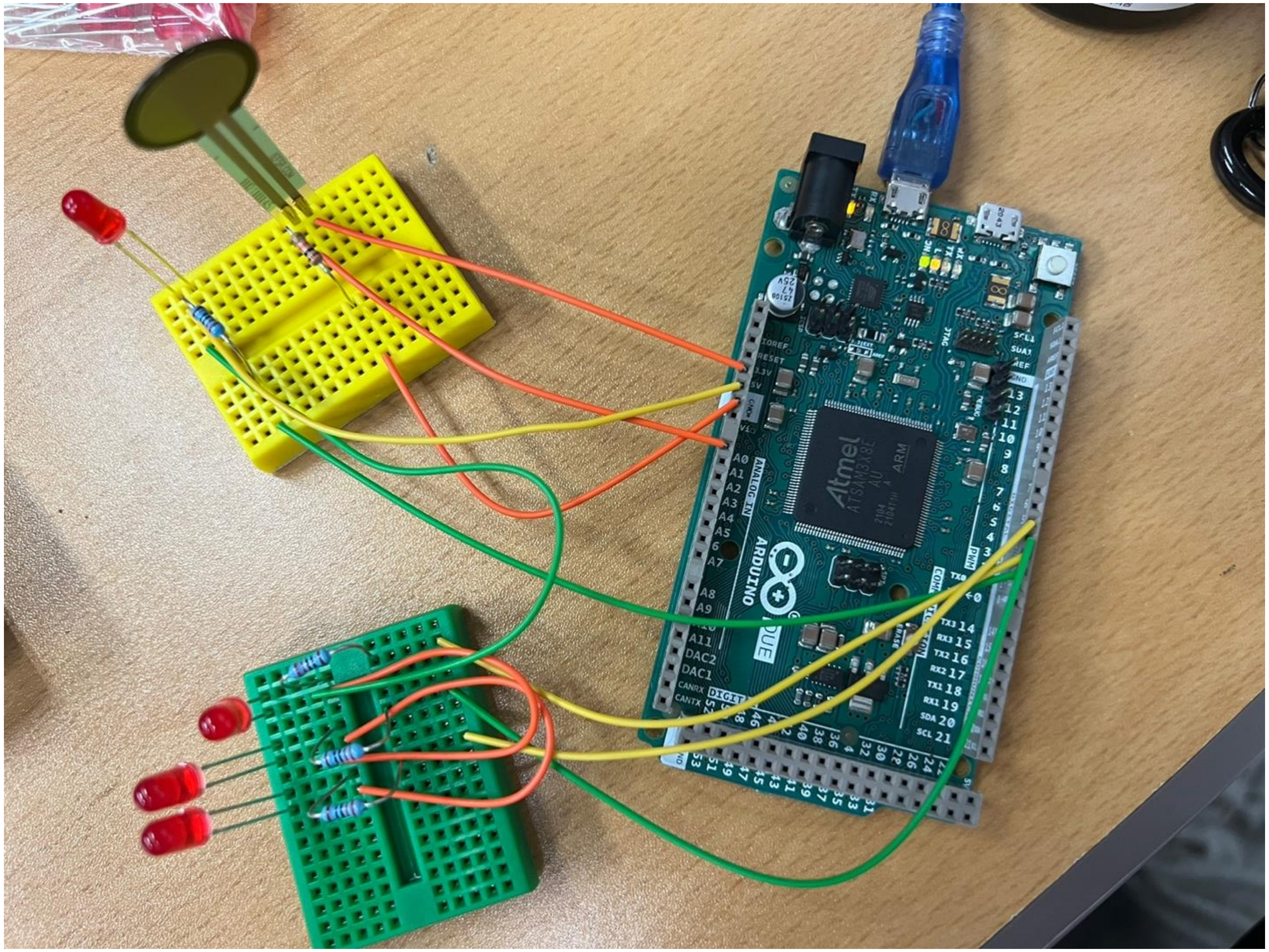


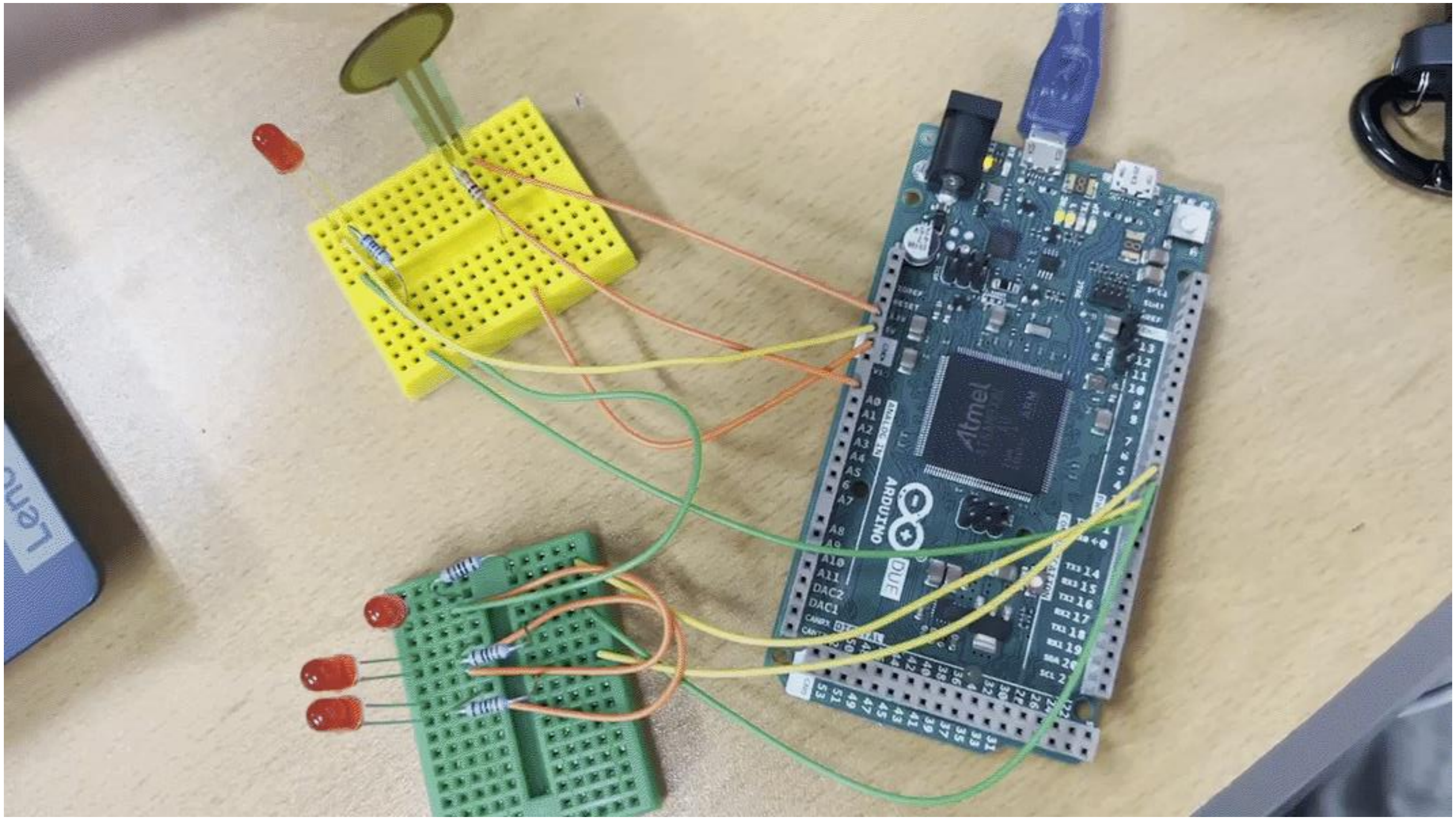
Controlling multi-LED with FSR



Controlling multi-LEDs with FSR







Controlling multi-LEDs with FSR

```
/* Arduino example code to control multiple LEDs with a Force Sensitive Resistor (FSR) as pressure sensor.

// Define pins:
#define fsrpin A0
#define led1 2
#define led2 3
#define led3 4
#define led4 5
#define led5 6
#define led6 7

// Define variables:
int fsrreading;

void setup() {
 // Begin serial communication at a baud rate of 9600:
 Serial.begin(9600);
 // Set LED pins as output:
 pinMode(led1, OUTPUT);
 pinMode(led2, OUTPUT);
 pinMode(led3, OUTPUT);
 pinMode(led4, OUTPUT);
 pinMode(led5, OUTPUT);
 pinMode(led6, OUTPUT);
}

void loop() {
 // Read the FSR pin and store the output as fsrreading:
 fsrreading = analogRead(fsrpin);

 // Print the fsrreading in the serial monitor:
 Serial.println(fsrreading);

 // Control the LEDs:
 if (fsrreading > 200) {
  digitalWrite(led1, HIGH);
 }
 else digitalWrite(led1, LOW);
 if (fsrreading > 450) {
  digitalWrite(led2, HIGH);
 }
 else digitalWrite(led2, LOW);
 if (fsrreading > 550) {
  digitalWrite(led3, HIGH);
 }
 else digitalWrite(led3, LOW);
 if (fsrreading > 650) {
  digitalWrite(led4, HIGH);
 }
 else digitalWrite(led4, LOW);
 if (fsrreading > 800) {
  digitalWrite(led5, HIGH);
 }
 else digitalWrite(led5, LOW);
 if (fsrreading > 900) {
  digitalWrite(led6, HIGH);
 }
 else digitalWrite(led6, LOW);
}
```

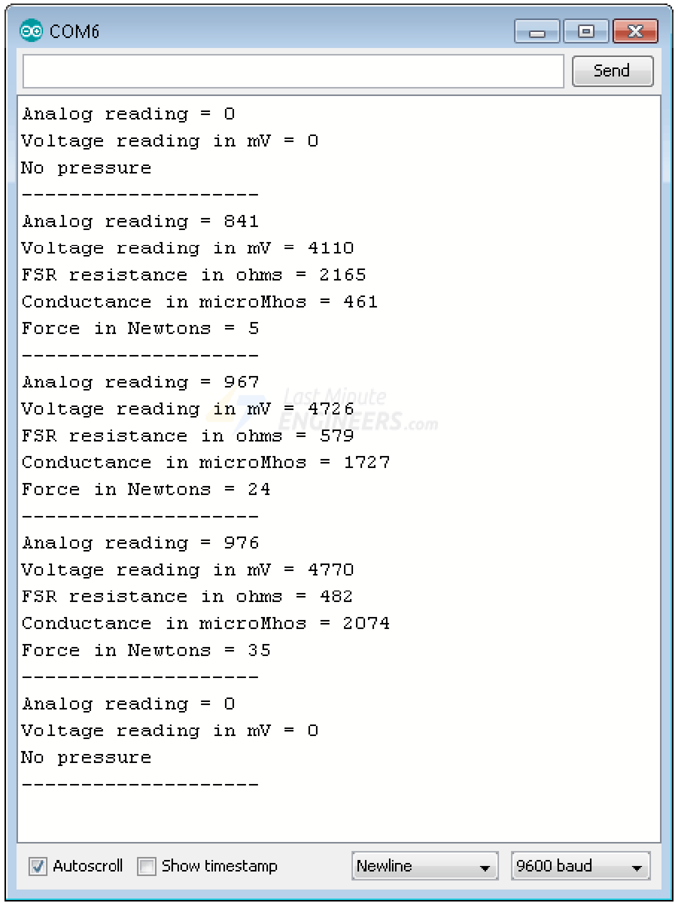
```
// Define pins:
#define fsrpin A0
#define led1 2
#define led2 3
#define led3 4
#define led4 5
#define led5 6
#define led6 7


// Define variables:
int fsrreading;
```

```
void setup() {
  // Begin serial communication at a baud rate
of 9600:
  Serial.begin(9600);
  // Set LED pins as output:
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  pinMode(led6, OUTPUT);
}
```

```
void loop() {
 // Read the FSR pin and store the output as fsrreading:
 fsrreading = analogRead(fsrpin);
 // Print the fsrreading in the serial monitor:
 Serial.println(fsrreading);
 // Control the LEDs:
 if (fsrreading > 200) {
  digitalWrite(led1, HIGH);
 }
 else digitalWrite(led1, LOW);
 if (fsrreading > 450) {
  digitalWrite(led2, HIGH);
 }
 else digitalWrite(led2, LOW);
 if (fsrreading > 550) {
  digitalWrite(led3, HIGH);
 }
 else digitalWrite(led3, LOW);
 if (fsrreading > 650) {
  digitalWrite(led4, HIGH);
 }
 else digitalWrite(led4, LOW);
 if (fsrreading > 800) {
  digitalWrite(led5, HIGH);
 }
 else digitalWrite(led5, LOW);
 if (fsrreading > 900) {
  digitalWrite(led6, HIGH);
 }
 else digitalWrite(led6, LOW);
}
```

Advanced Analog FSR Measurements

```
int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrReading;     // the analog reading from the FSR resistor divider
int fsrVoltage;     // the analog reading converted to voltage
unsigned long fsrResistance;  // The voltage converted to resistance
unsigned long fsrConductance;
long fsrForce;       // Finally, the resistance converted to force

void setup(void) {
  Serial.begin(9600);   // We'll send debugging information via the Serial monitor
}

void loop(void) {
  fsrReading = analogRead(fsrPin);
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

  // analog voltage reading ranges from about 0 to 1023 which maps to 0V to 5V (= 5000mV)
  fsrVoltage = map(fsrReading, 0, 1023, 0, 5000);
  Serial.print("Voltage reading in mV = ");
  Serial.println(fsrVoltage);

  if (fsrVoltage == 0) {
    Serial.println("No pressure");
  } else {
    // The voltage = Vcc * R / (R + FSR) where R = 10K and Vcc = 5V
    // so FSR = ((Vcc - V) * R) / V        yay math!
    fsrResistance = 5000 - fsrVoltage;     // fsrVoltage is in millivolts so 5V = 5000mV
    fsrResistance *= 10000;                // 10K resistor
    fsrResistance /= fsrVoltage;
    Serial.print("FSR resistance in ohms = ");
    Serial.println(fsrResistance);

    fsrConductance = 1000000;           // we measure in micromhos so
    fsrConductance /= fsrResistance;
    Serial.print("Conductance in microMhos: ");
    Serial.println(fsrConductance);

    // Use the two FSR guide graphs to approximate the force
    if (fsrConductance <= 1000) {
      fsrForce = fsrConductance / 80;
      Serial.print("Force in Newtons: ");
      Serial.println(fsrForce);
    } else {
      fsrForce = fsrConductance - 1000;
      fsrForce /= 30;
      Serial.print("Force in Newtons: ");
      Serial.println(fsrForce);
    }
  }
  Serial.println("--------------------");
  delay(1000);
}
```



```
Analog reading = 0
Voltage reading in mV = 0
No pressure
--------------------
Analog reading = 841
Voltage reading in mV = 4110
FSR resistance in ohms = 2165
Conductance in microMhos = 461
Force in Newtons = 5
--------------------
Analog reading = 967
Voltage reading in mV = 4726
FSR resistance in ohms = 579
Conductance in microMhos = 1727
Force in Newtons = 24
--------------------
Analog reading = 976
Voltage reading in mV = 4770
FSR resistance in ohms = 482
Conductance in microMhos = 2074
Force in Newtons = 35
--------------------
Analog reading = 0
Voltage reading in mV = 0
No pressure
--------------------
```

Autoscroll [✓] Show timestamp [ ] Newline 9600 baud