

Arduino force

Due Monday 4/11/22 (7th period) Tuesday 4/12/22 (6th period)

Each partner needs to write the code for today's lesson. You will only make one circuit.

Today we'll be using a force sensing resistor. It looks like this:



This thing is a resistor. When you press it, its resistance changes, which can be sensed by an Arduino. It's a lot like the LDR (light dependent resistor) except that it's the force you apply to it that changes its value instead of light.

Create the circuit from page 63 of the Getting Started with Arduino book except use the force sensor instead of the LDR. Start by putting the force sensor into holes e1 and e2 on the breadboard.

Write a program that reads the value of pin A0 and prints it to the serial monitor (or open your LDR program and do a Save as for today's program). Make sure it has a `Serial.begin(9600)` inside Setup and that you are printing the value of the variable used in the `analogRead(0)` command. Try pressing on the sensor with your thumb and forefinger, and make sure the values listed in the monitor change. If you are not pressing or touching the sensor, you should have a reading of zero. The harder you press the closer the value gets to 1023, the maximum possible reading. As tempting as it is to try to achieve perfection and get a 1023, please don't break this thing with your efforts! Debug this circuit and your program until you have it working.

I don't need to sign you off on this part, just move on once you're ready.

We're going to make a reaction time program. It should work like this:

A light turns on and off (1 second each), then stays on for a random amount of time (between 3 and 8 seconds.) When it turns off the user squeezes the pressure sensor and is told his/her reaction time. The program pauses 3 seconds then starts over.

To do this, you will need to do a few things:

Insert an LED into pins 13 and GND (remember, long leg into 13).

Use the `random()` function (see the Arduino reference) to choose the random time between 3 and 8 seconds. It's actually pretty easy, you can add a command like this:

```
long r = random(3000,8000);  
delay(r);
```

(Continued on next page)

To time how long it took for someone to press the sensor you need to read the sensor over and over again in a loop waiting for the value to reach a certain trigger point. You can use a while loop to do this with the force variable. For example, assuming that the force sensor returns 200 or less when no one is pressing it you could do this:

```
while(val<200)
{
    val = analogRead(0);
}
```

This will stay in the loop until the user presses the sensor hard enough to make it read 200. You also need to use the millis() function which gives you the number of milliseconds since the Arduino was turned on which allows us to measure the passage of time like this:

```
long before = millis();

<loop goes here>

long after = millis();

long elapsedTime = after - before;
```

Notice that I'm using "long" instead of "int" for the before and after times. That is because milliseconds is a very large number; too big for an int (which rolls over near 32767). The value stored in "elapsedTime" at this point is the number of milliseconds it took the user to press the sensor.

Print the results out in the serial monitor. Here is sample output:

```
Ready...
When the light goes out, press the sensor.
Begin!
Reaction time: 1060 milliseconds or 1.06 seconds

Ready...
When the light goes out, press the sensor.
Begin!
You can't press the sensor until the light goes out!

Ready...
When the light goes out, press the sensor.
Begin!
Reaction time: 359 milliseconds or 0.36 seconds
Congratulations, you hold the new record.
```

For 9/10, do everything above except tracking the best score.

For 10/10 also add code to make sure the user is not pressing the sensor before the light goes out (see middle sample above).

For 11/10 (extra credit) do all of the above and also keep track of the best performance and report when the user beats the previous record.