**Lab Exercise 11/21/2022**

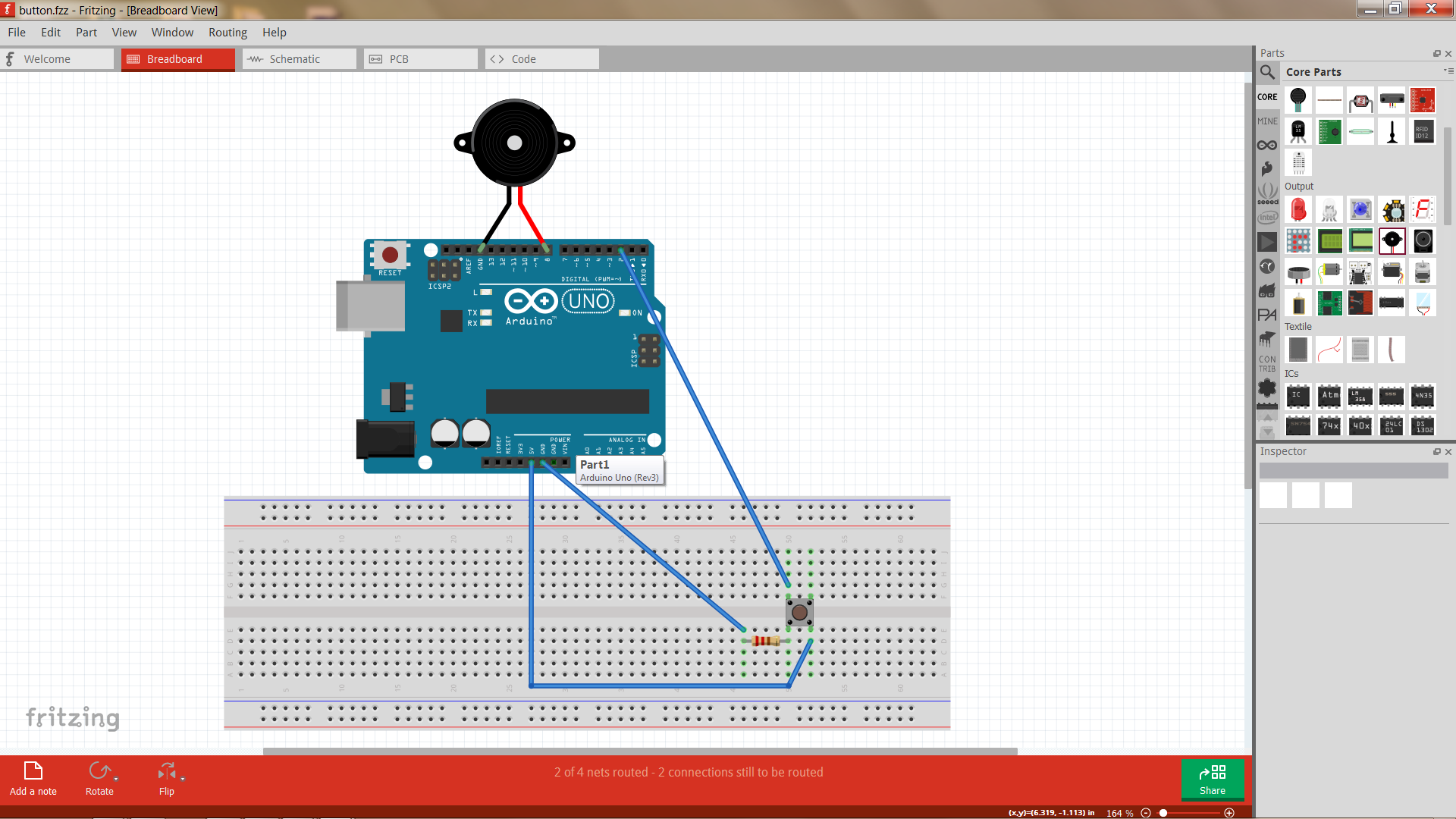
**Advanced Programming in Java**

**Creating a Music Box (“Ode to Joy”)**

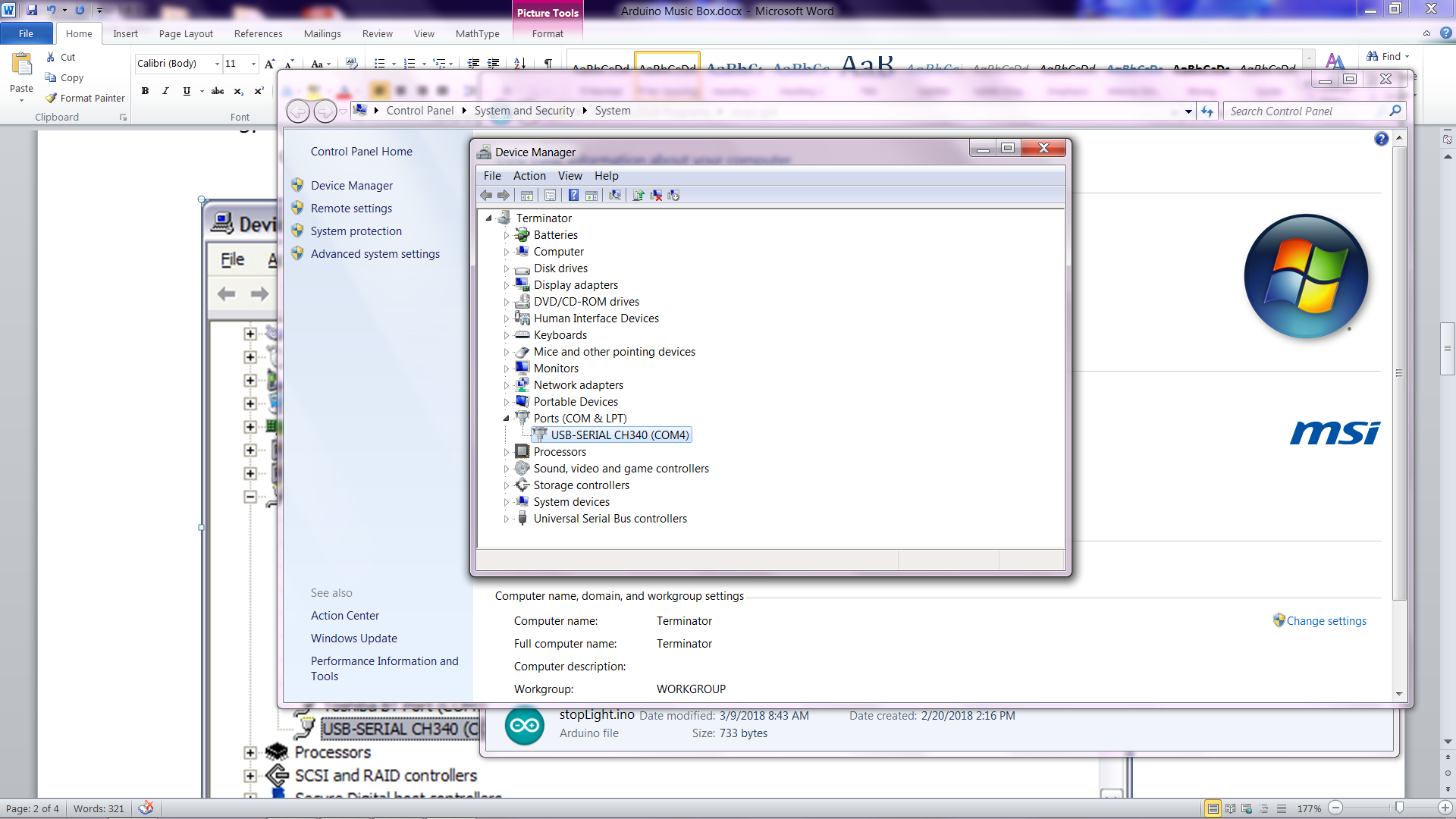
In this activity, you will require the following:

* Arduino
* Breadboard
* Pushbutton Switch
* One 10000 Ohm Resistor
* Hookup wires
* USB A-B cable

1. Connect our pushbutton to the Arduino by mounting it on the breadboard.
2. Place the 10000 Ohm resistor in the breadboard
3. Hook up the Arduino to the breadboard as in the diagram attached.
4. Connect your Arduino to your computer with the USB A-B cable.

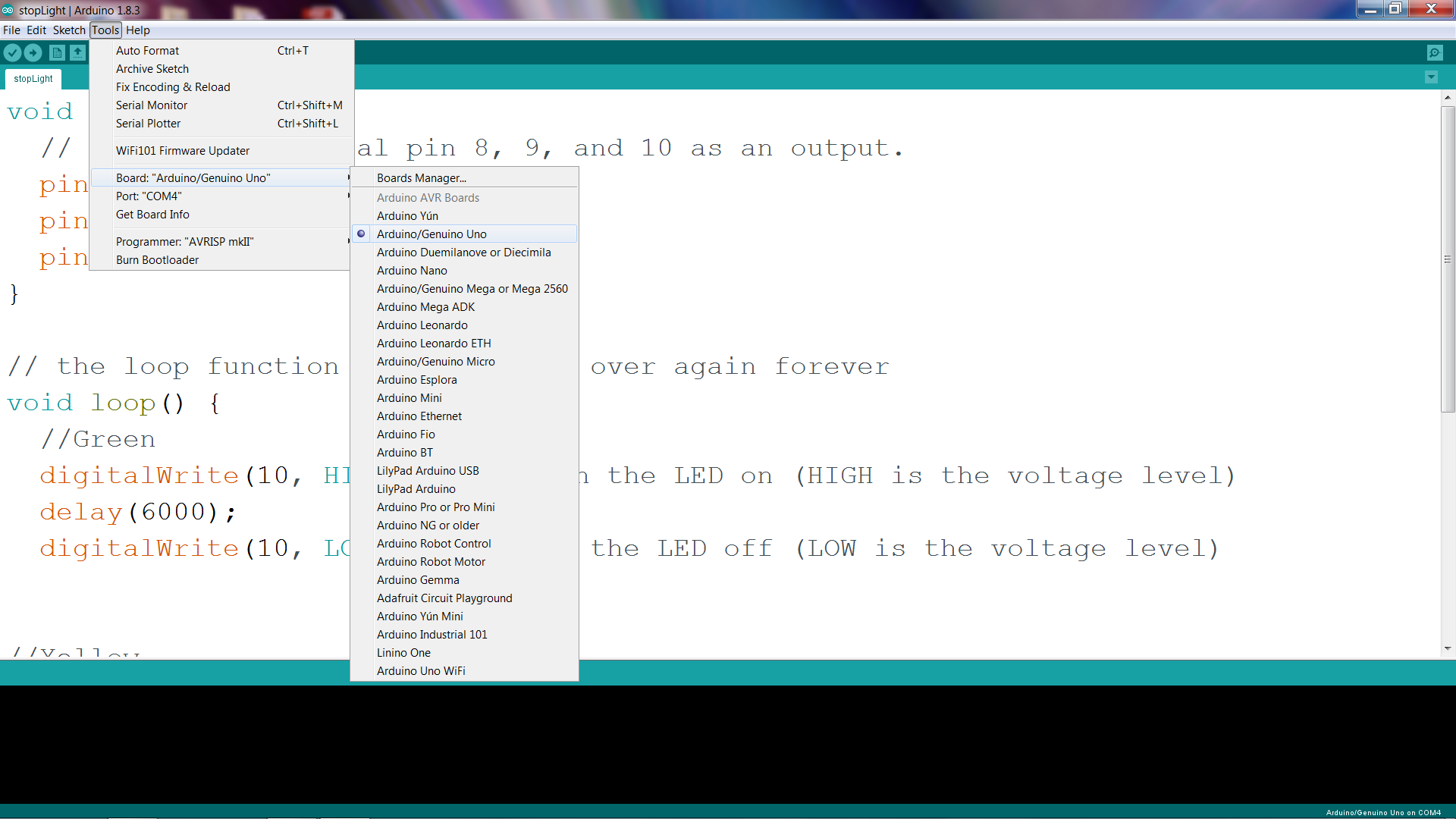


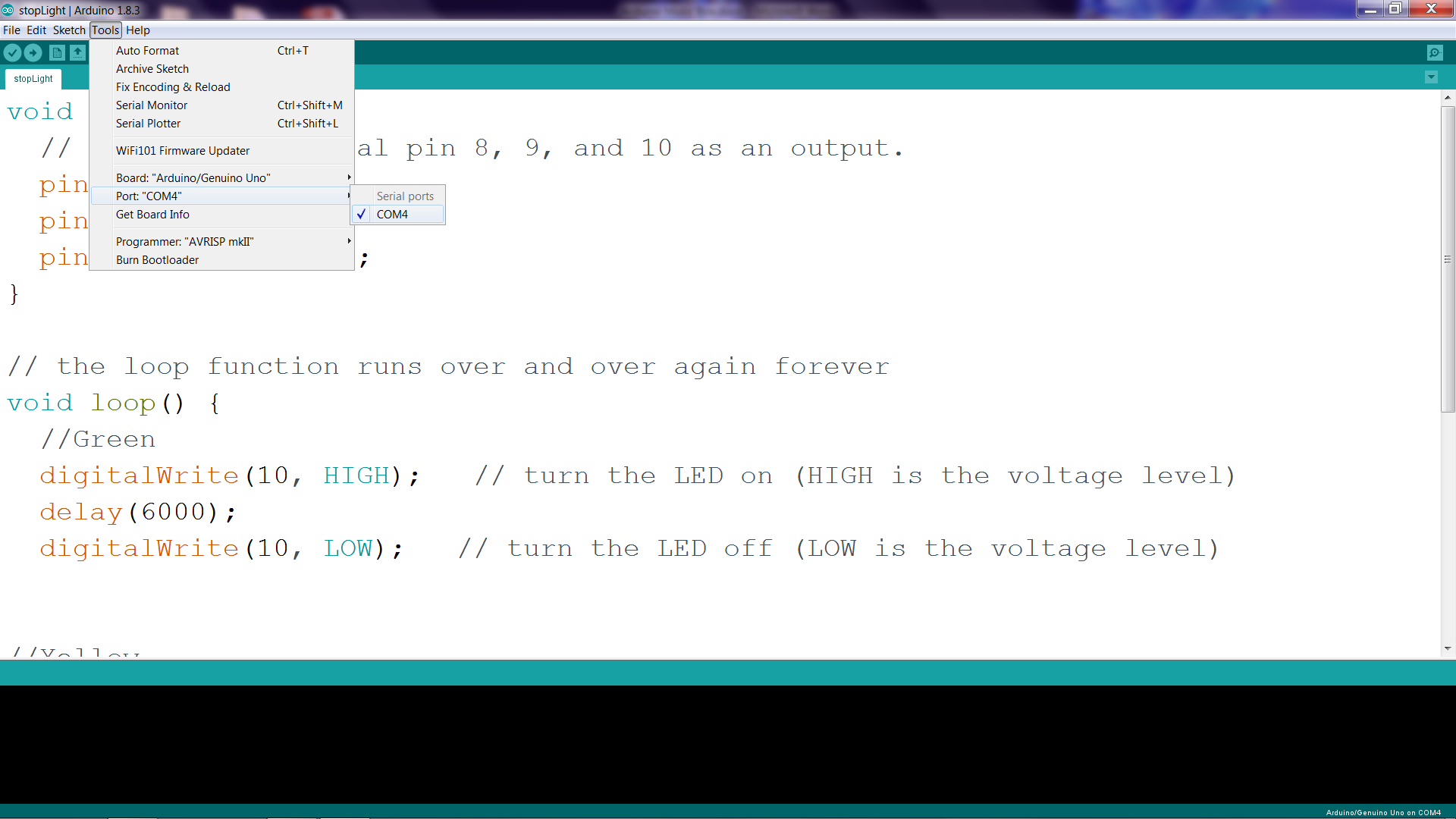
1. (Optional) Make sure Arduino is you can check out device manager to find out what Port we are connected to. You can see this in the Device Manager.



Note: Mine is connected to Port 4 but yours might be different. You should make a note of this.

1. Start your Arduino development environment.
2. On your Tools menu, check that the Board is set to Uno and the Port is set to the value you noted in step 5.





In this program, we will be using two special functions; tone and delay.

The tone function has 3 parameters; pin to send signal to, frequency in Hertz, and duration in milliseconds

The delay function has 1 parameter; time in milliseconds

Now let’s make some music by typing in the following code and running it. Press the button and see what happens. Do you recognize the song?

**Now let’s add the code**

//Plays a melody

const int buttonPin = 2;

int pbState = 0;

void setup(){

pinMode(buttonPin, INPUT);

}

void loop(){

pbState = digitalRead(buttonPin);

if (pbState == HIGH){

//First group of notes

tone(8, 247, 300);

delay(500);

tone(8, 247, 300);

delay(500);

tone(8, 262, 300);

delay(500);

tone(8, 294, 300);

delay(500);

tone(8, 294, 300);

delay(500);

//second group of notes

tone(8, 262, 300);

delay(500);

tone(8, 247, 300);

delay(500);

tone(8, 220, 300);

delay(500);

tone(8, 196, 300);

delay(500);

tone(8, 196, 300);

delay(500);

//third group of notes

tone(8, 220, 300);

delay(500);

tone(8, 247, 300);

delay(500);

tone(8, 247, 500);

delay(650);

tone(8, 220, 200);

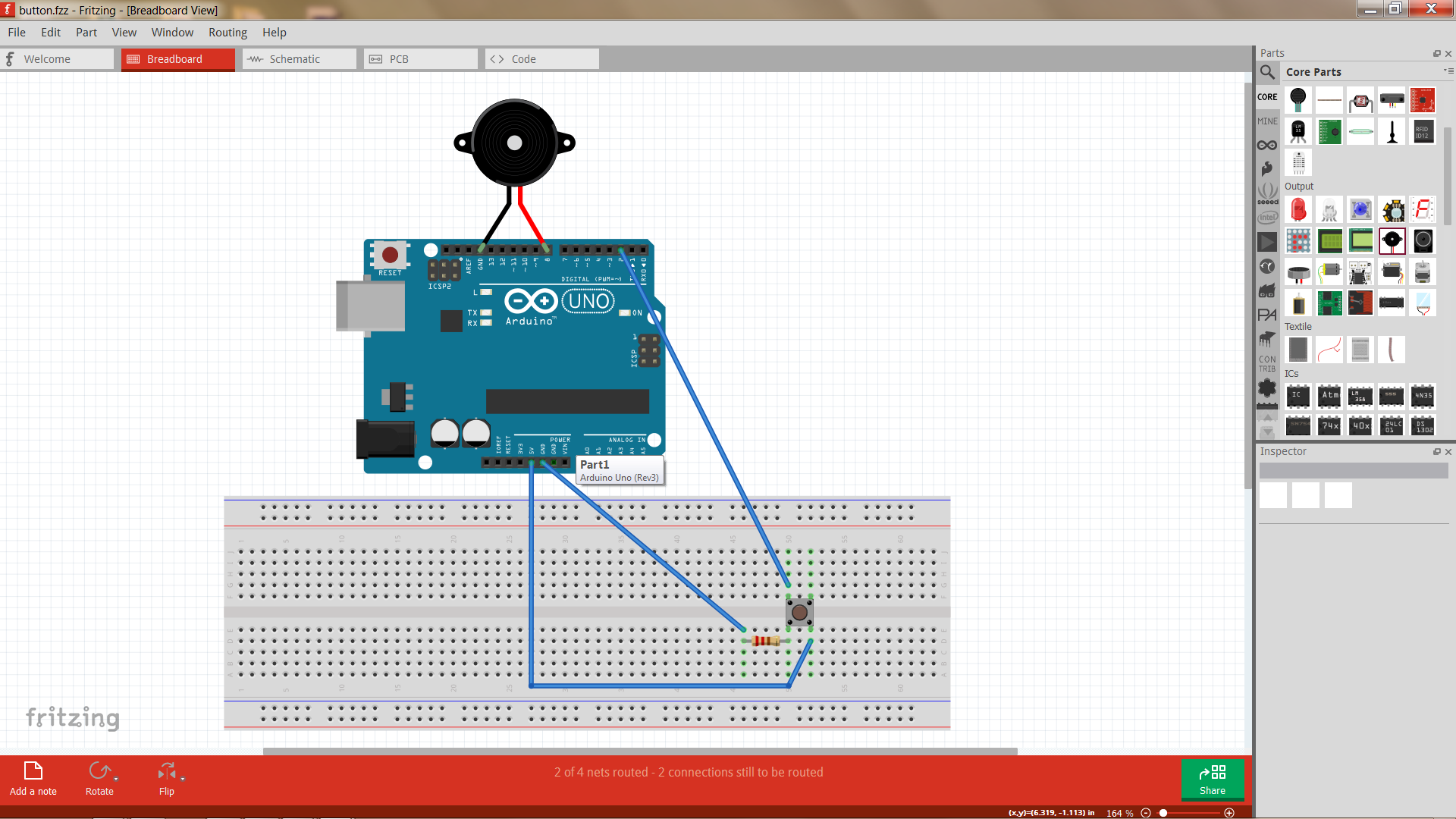
delay(250);

tone(8, 220, 200);

delay(250);

}

}



**Building a Reaction Tester**

Start by connecting one leg of the button to the positive rail on the bread board. Next, add the 10K resistor between the other leg and ground. Then, add a led between pin 13 on the Arduino and ground.   After that, connect the negative rail on the bread board to the ground on the Arduino and the positive rail to 5V.  Finally, connect the leg of the push button with the 10K resistor to pin two on the Arduino.

Graphical user interface, application

Description automatically generated

**The program**

int switchPin = 2;

int ledPin = 13 ;

boolean lastButton = LOW;

boolean currentButton = LOW;

boolean started = false;

boolean timer = false;

long startTime;

long endTime;

long randomTime;

float elapsedTime;

void setup()

{

pinMode(switchPin, INPUT);

pinMode(ledPin, OUTPUT);

Serial.begin(9600);

}

boolean debounce(boolean last)

{

boolean current = digitalRead(switchPin);

if(last != current)

{

delay(5);

current = digitalRead(switchPin);

}

return current;

}

void loop()

{

currentButton = debounce(lastButton);

if(lastButton == LOW && currentButton == HIGH)

{

started = !started;

lastButton = HIGH;

}

lastButton = currentButton;

if(started == true && timer == false)

{

Random();

timer = true;

}

if(started == false && timer == true)

{

Stop();

timer = false;

}

}

void Random()

{

randomTime = random(4,10);

randomTime = randomTime\*1000;

digitalWrite(ledPin, HIGH);

delay(100);

digitalWrite(ledPin, LOW);

delay(randomTime);

Start();

}

void Start()

{

startTime = millis();

digitalWrite(ledPin, HIGH);

}

void Stop()

{

endTime = millis();

elapsedTime = (endTime - startTime)+5;

elapsedTime = elapsedTime/1000;

Serial.print("Time Seconds: ");

Serial.println(elapsedTime);

digitalWrite(ledPin, LOW);

}

**Running the test**

1. Upload your program to the Arduino
2. Open the Serial Monitor at 9600 baud
3. Push the pushbutton and the LED will flash momentarily.
4. The LED will come on at some random time from 4 to 10 seconds.
5. When the light comes on, push the button and the serial monitor will display your reaction time.

Graphical user interface, application

Description automatically generated