**Lab Exercise 11/18/2024**

**Advanced Programming in Java**

**Binary Dice**

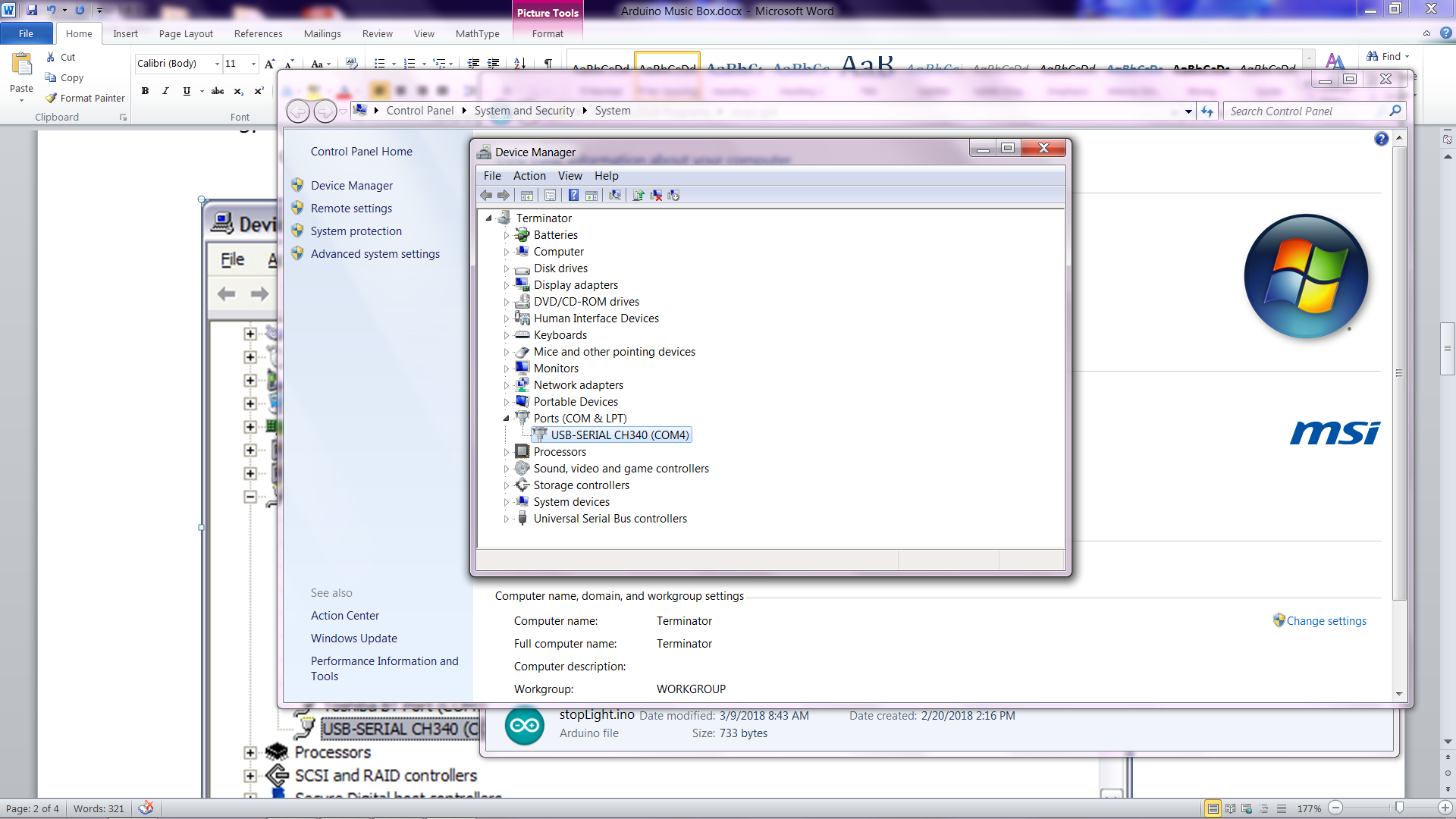
In this activity, you will require the following:

* Arduino
* Breadboard
* 3 LED’s (polarity sensitive)
* 200 Ohm resistor
* Hookup wire
* USB A-B cable

A computer screen shot of a computer

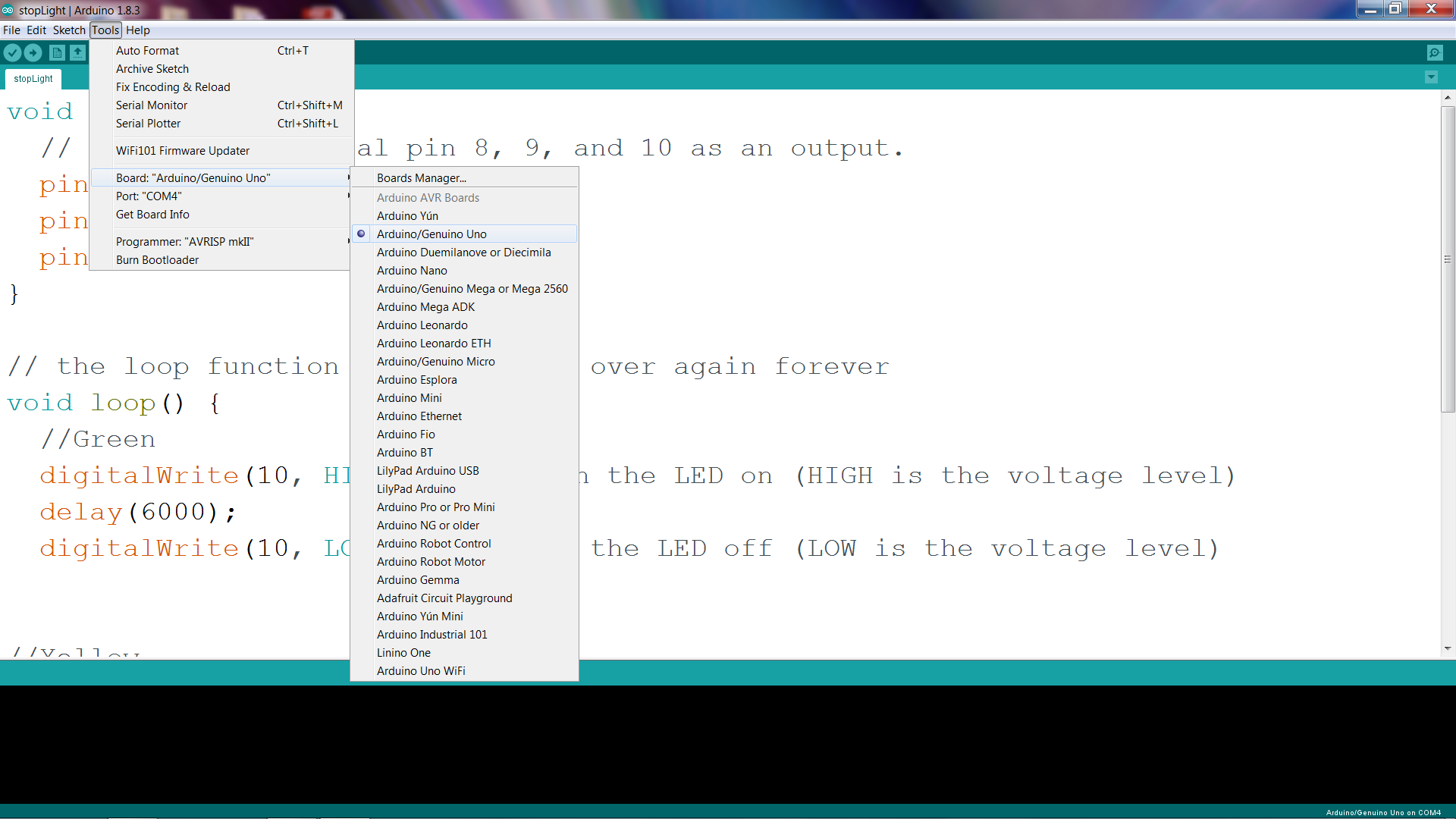
Description automatically generated

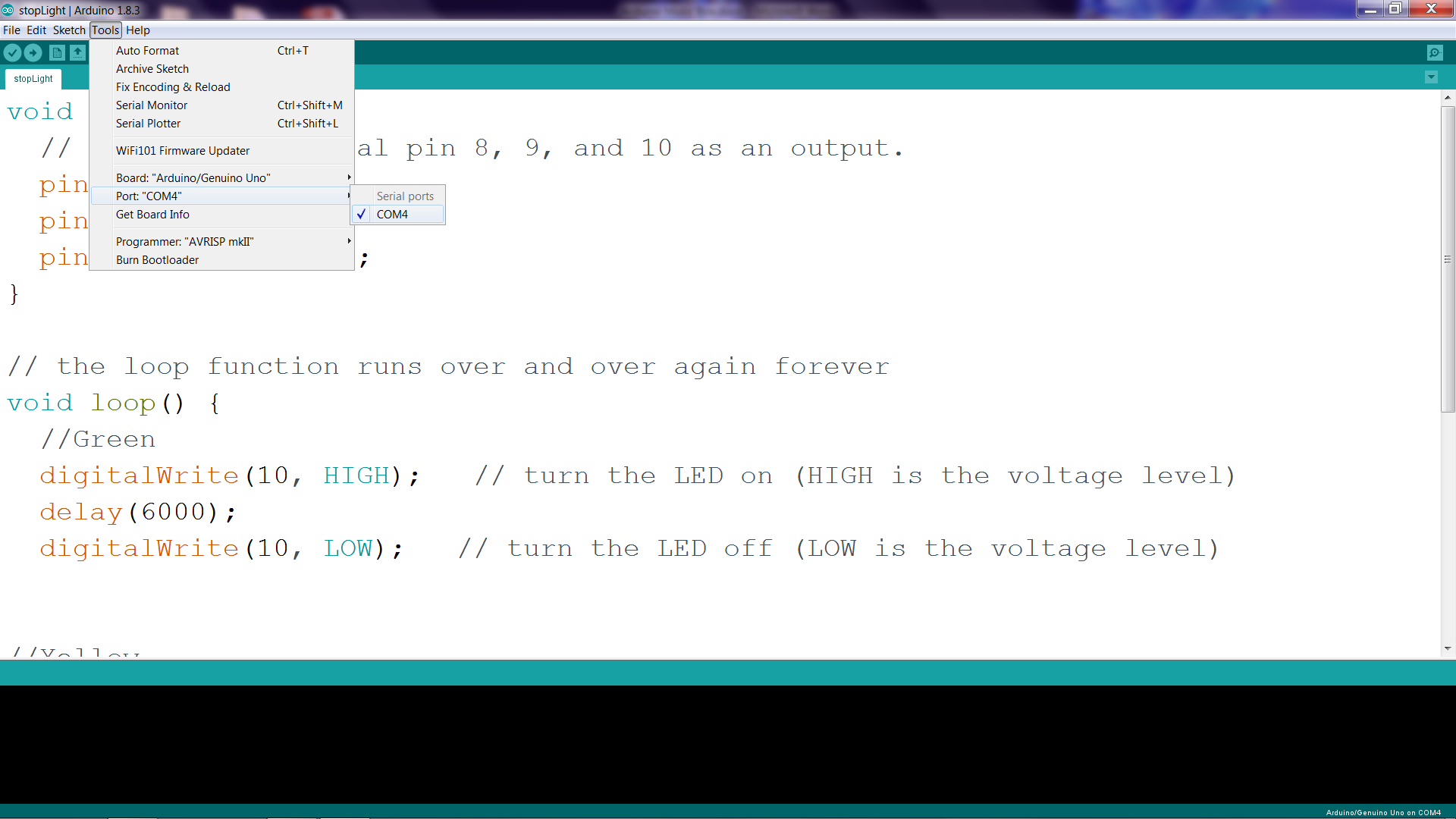
1. (Optional) Make sure Arduino is you can check out device manager to find out what Port we are connected to. You can see this in the Device Manager.



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.





**Now let’s add the code**

const unsigned int LED\_BIT0 = 12;

const unsigned int LED\_BIT1 = 11;

const unsigned int LED\_BIT2 = 10;

void setup() {

// put your setup code here, to run once:

pinMode(LED\_BIT0, OUTPUT);

pinMode(LED\_BIT1, OUTPUT);

pinMode(LED\_BIT2, OUTPUT);

randomSeed(analogRead(A0));

long result = random(1, 7);

output\_result(result);

}

void loop() {

//To roll dice again, press the reset button on the Arduino

}

void output\_result(const long result){

digitalWrite(LED\_BIT0, result & B001);

digitalWrite(LED\_BIT1, result & B010);

digitalWrite(LED\_BIT2, result & B100);

}

**Now that you have this working, modify it to be a counter from 0 (000) to 7 (111).**

In case you are not aware, a 3 bit binary counter has 8 states (23 conditions). Specifically, 000, 001, 010, 011, 100, 101, 110, 111 where 0 represents an off LED and 1 represents an on LED. Hint: instead of sending a random number from 1 to 6 to the output result function, send it the numbers from 0 to 7 in that order.

FYI: result & B001 is an example of binary masking where only the numbers 1 (001), 3 (011), and 5 (101) will result in LED\_BIT0 being on. The same can be applied to result & B010 and result & B100.

**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.

A computer screen shot of a computer

Description automatically generated

**The program**

int switchPin = 2;

int ledPin = 13 ;

booleanlastButton = LOW;

booleancurrentButton = 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);

}

booleandebounce(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