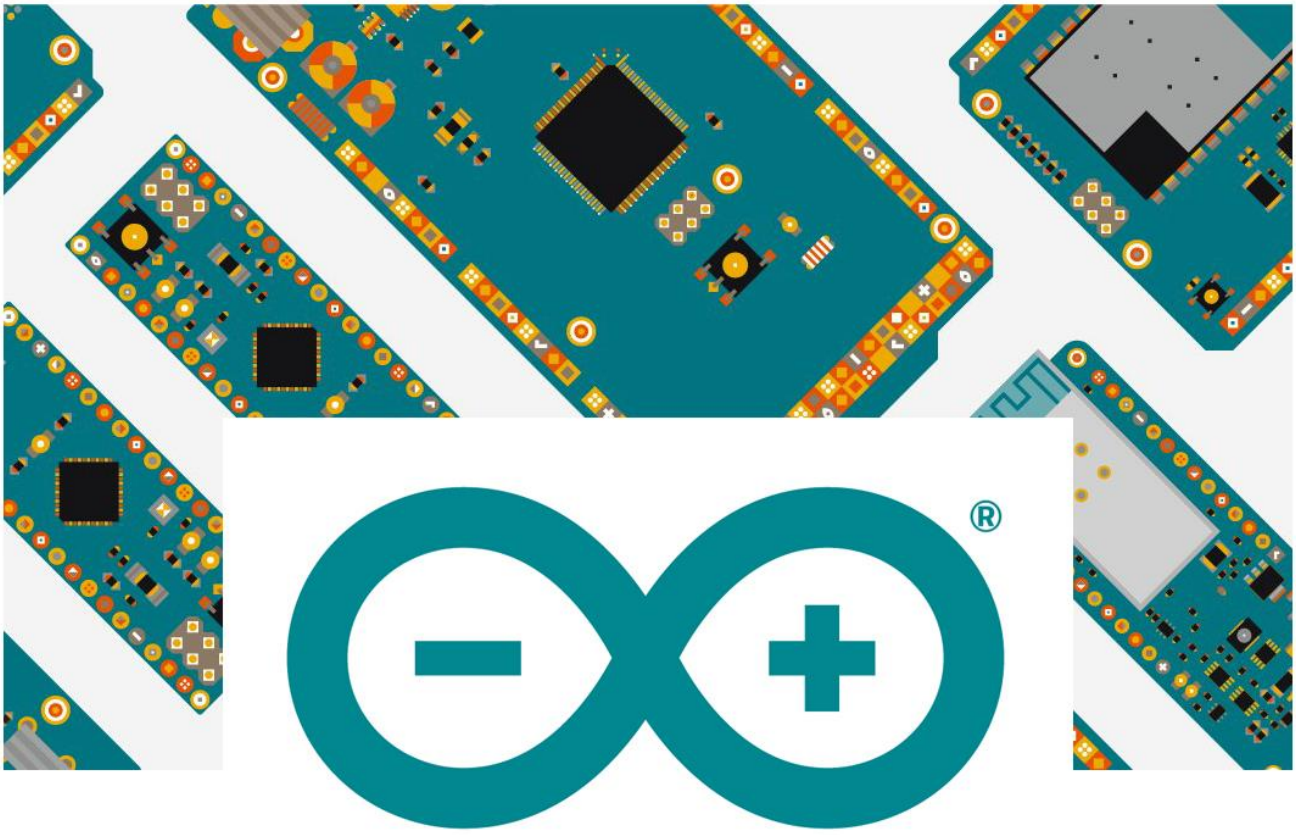


ARDUINO WORKSHOP

IEEE and ASME UCF

ieee@knights.ucf.edu

ieee.cecs.ucf.edu

ucfasme@gmail.com

www.asmeatucf.com

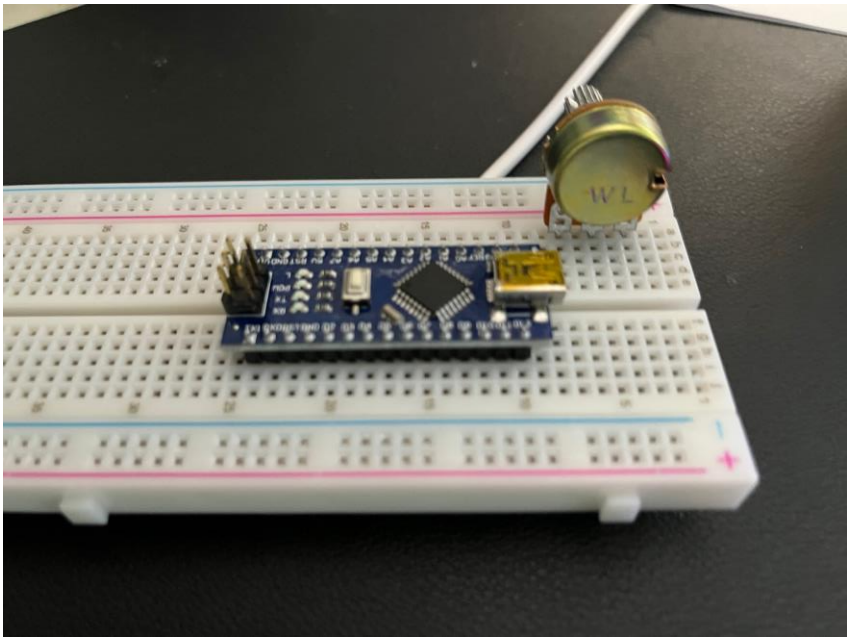
Components

- 1x Arduino Nano
- 2x DC Brushless Motors
- 1x L293D Motor Driver IC
- 2x SG90 Micro-servo Motor
- 2x Joystick
- 1x 10 kΩ Rotary Potentiometer
- 1x HC-SR04 Ultrasonic Sensor
- 10x Male-to-Male Wires
- 1x USB 2.0 Micro-B Cable

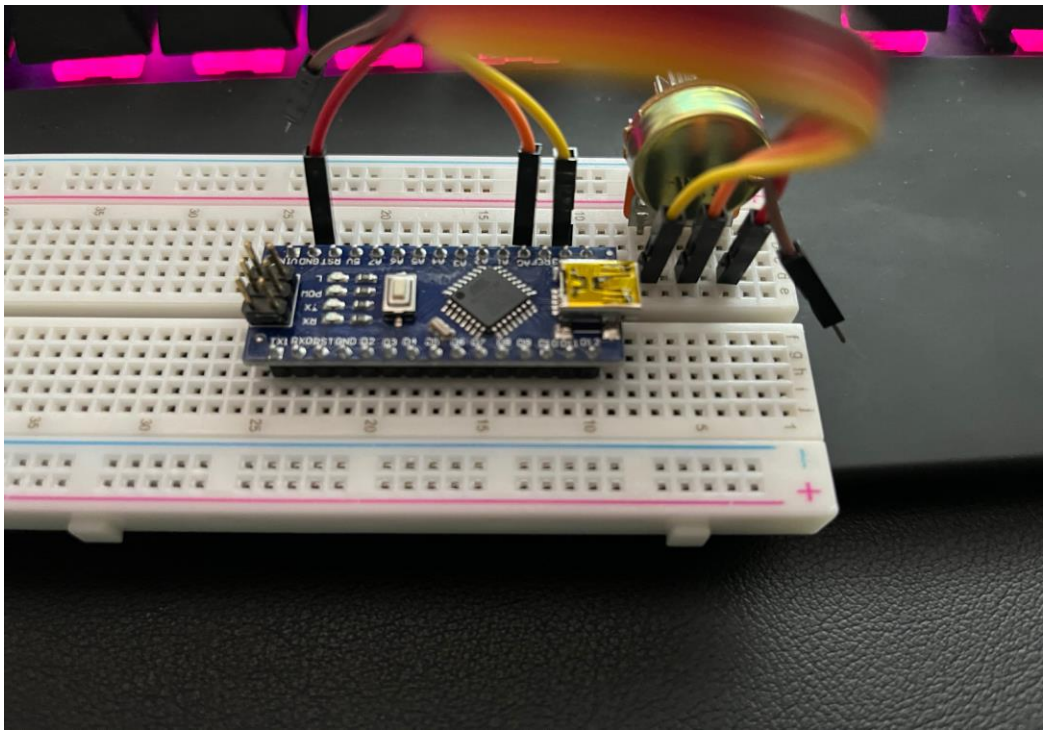
Servo Control with Potentiometer

1. Plug in the 3-pin potentiometer into column A rows 3, 5, and 7 (**A3**, **A5**, **A7**) of the breadboard.

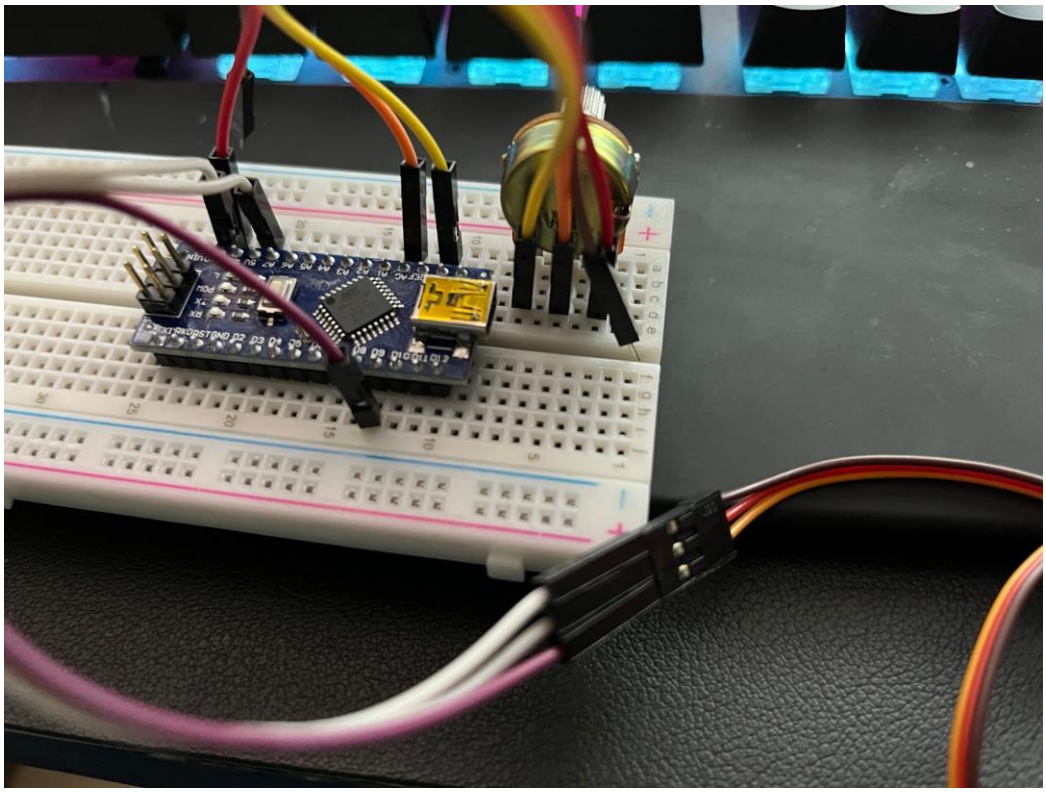
2. Plug in the Arduino NANO to columns D and H rows 10-24 (**D10-D24** and **H10-H24**) with the USB connector on the NANO facing the potentiometer.



3. Connect wires from NANO to potentiometer
   - ***a) B11 → D7***
   - ***b) B13 → D5***
   - ***c) B23 → D3***



4. Connect 3 wires from the NANO to the Servo
   a) **C23 → Brown Connection**
   b) **B21 → Red Connection**
   c) **J13 → Orange connection**



Plug in USB from NANO to computer and run code

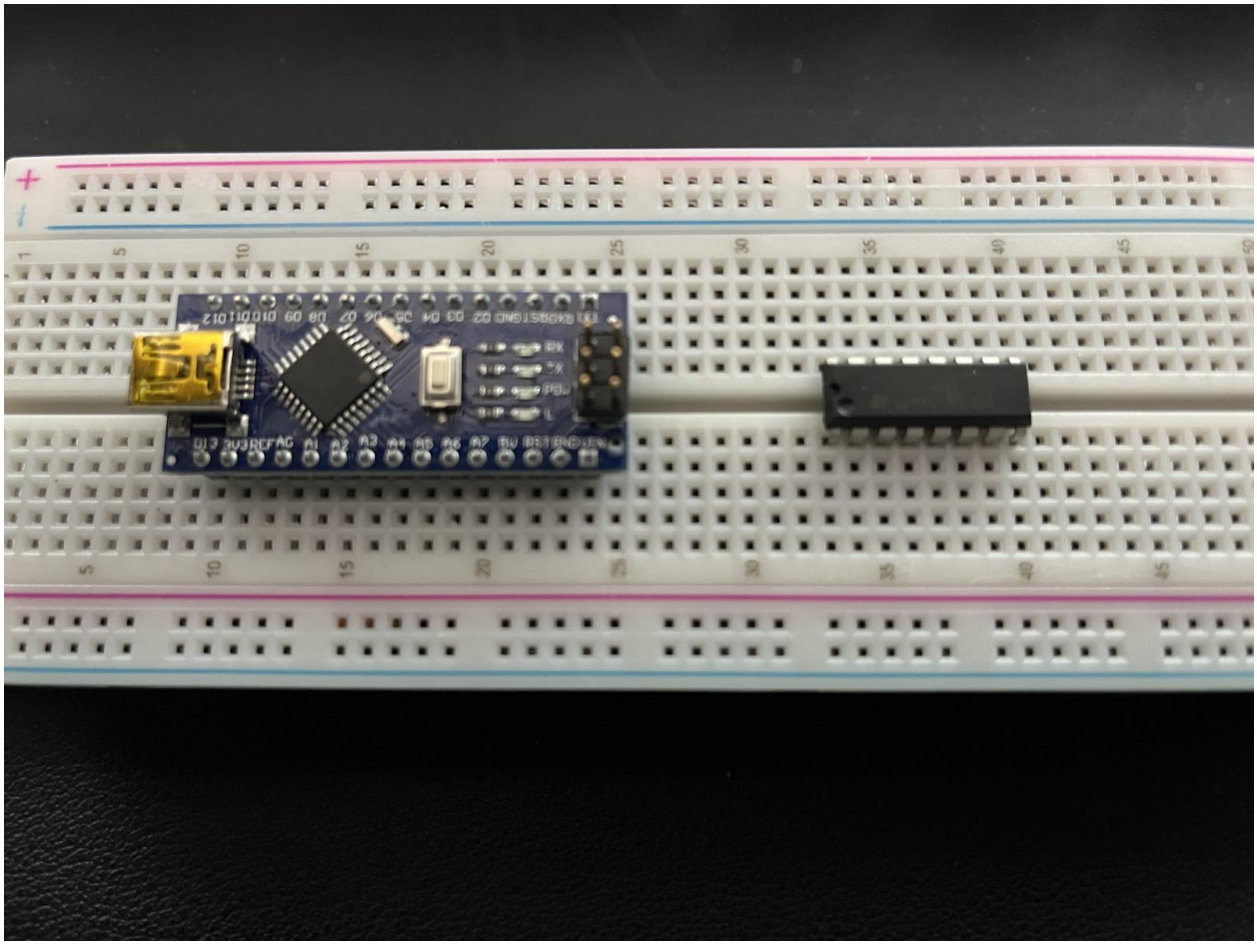
Servo Control with Potentiometer Code:

```
#include <Servo.h>  // add servo library


Servo myservo;  // create servo object to control a servo


int potpin = 0;  // analog pin used to connect the potentiometer

int val;    // variable to read the value from the analog pin


void setup() {

 myservo.attach(9);  // attaches the servo on pin 9 to the servo object

}


void loop() {

 val = analogRead(potpin);            // reads the value of the potentiometer
(value between 0 and 1023)

 val = map(val, 0, 1023, 0, 180);     // scale it to use it with the servo
(value between 0 and 180)

 myservo.write(val);                  // sets the servo position according to
the scaled value

 delay(15);                           // waits for the servo to get there

}
```

DC Motors with Motor Controller IC

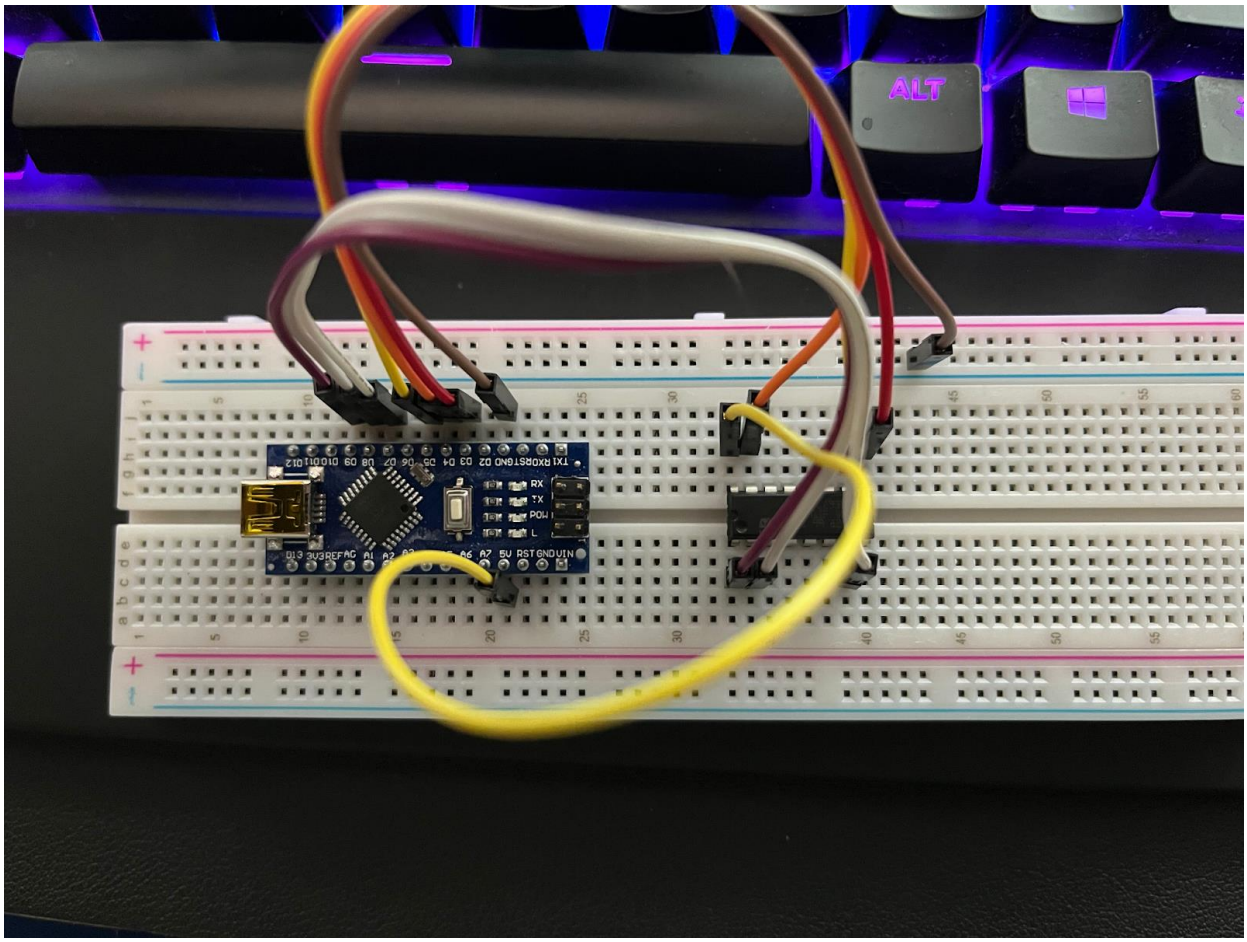
1. Leave the NANO plugged into the same rows and columns (**D10-D24** and **H10-H24**) with the USB connector facing the same direction. Plug in the motor controller IC in **E33-E40** and **F33-F40**.

IMPORTANT: MAKE SURE THE "u-shape" ON THE IC POINTS TOWARDS THE NANO.

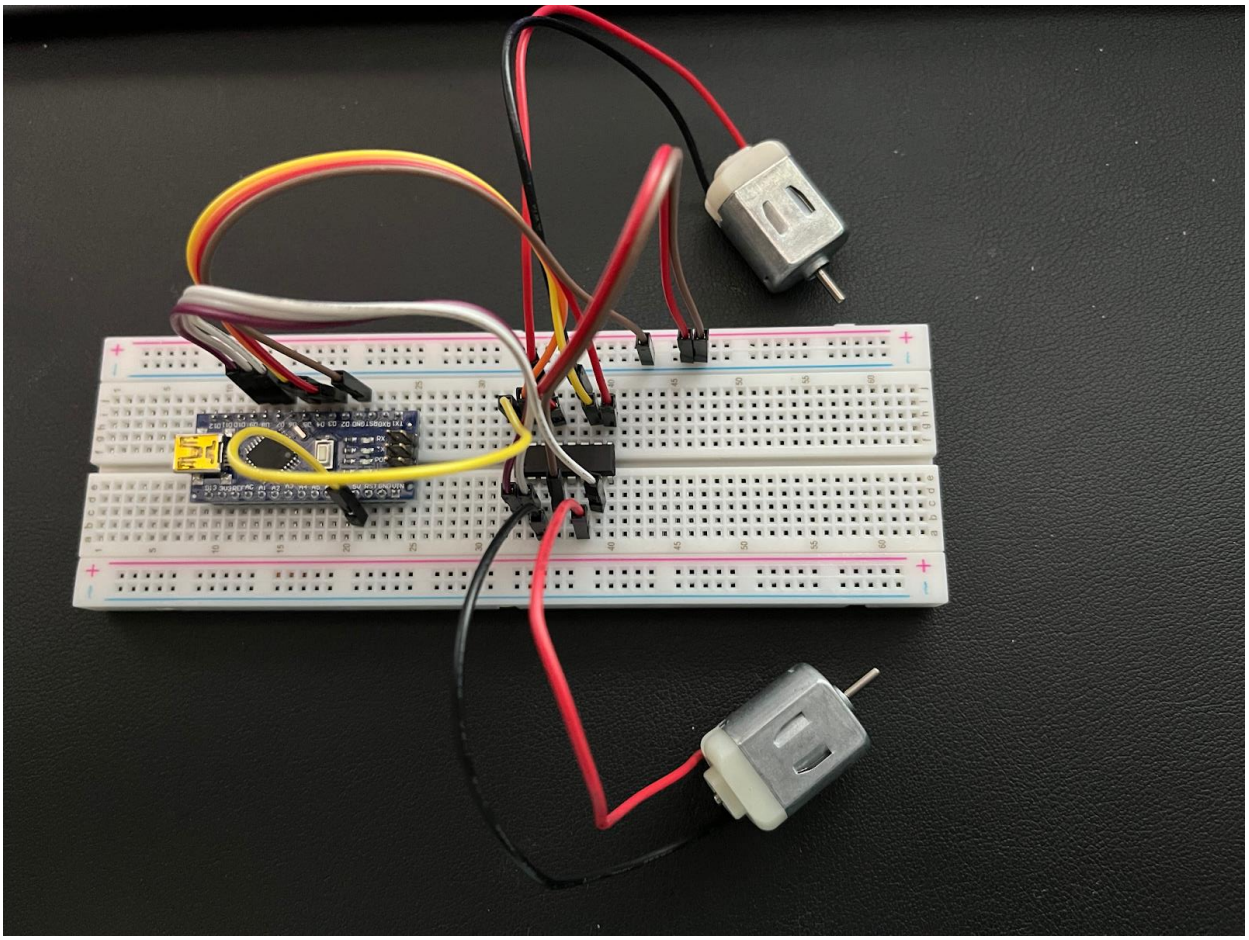


2. Connect wires from NANO to IC

a) J21 → (-) rail
b) J19 → H40
c) J18 → H34
d) J17 → H39
e) J15 → C39
f) J14 → C34
g) J13 → C33
h) B21 → H33



3. GND → IC and connecting the DC motors → IC
   - **a) (-) rail → H36**
   - **b) (-) rail → C36**
   - **c) One motor**
     - **a. Motor wire → A38**
     - **b. Motor wire → A35**
   - **d) Other Motor**
     - **a. Motor wire → J38**
     - **b. Motor wire → J35**



Plug in USB from NANO to computer and run code

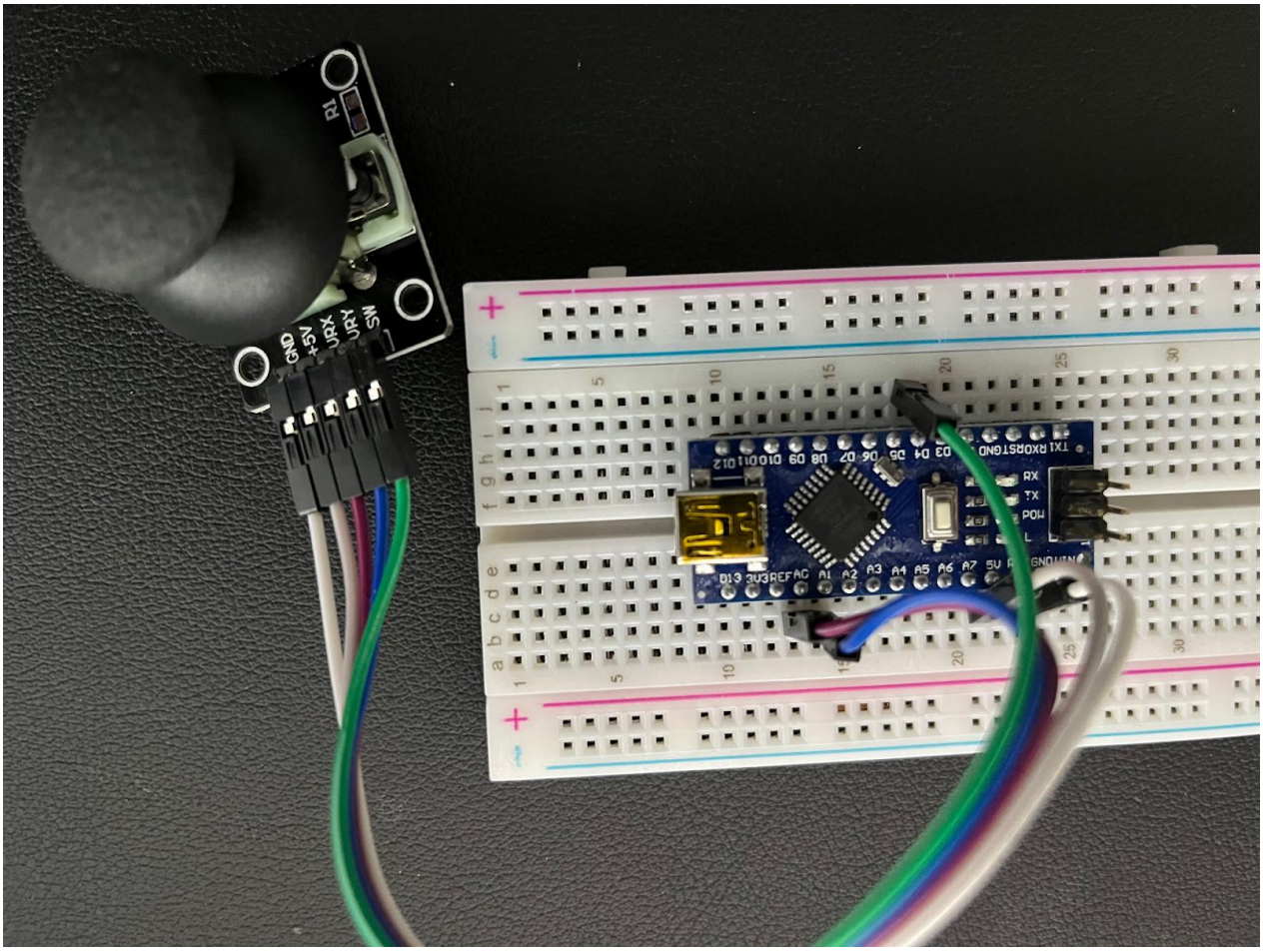
DC Motors with Motor Controller IC Code:

```
// Motor A connections
int enA = 9;
int in1 = 8;
int in2 = 7;
// Motor B connections
int enB = 3;
int in3 = 5;
int in4 = 4;

void setup() {
	// Set all the motor control pins to OUTPUT
	pinMode(enA, OUTPUT);
	pinMode(enB, OUTPUT);
	pinMode(in1, …);
	pinMode(in2, …);
	pinMode(in3, …);
	pinMode(in4, …);

	// Turn off all motors - Initial state
	digitalWrite(in1, LOW);
	digitalWrite(in2, LOW);
	digitalWrite(in3, LOW);
	digitalWrite(in4, LOW);
}

void loop() {
	// We call 2 user-defined function @ a 1 second interval
	directionControl();
	delay(1000);
	speedControl();
	delay(1000);
}

// This function lets you control spinning direction of motors
void directionControl() {
	// Set motors to maximum speed
	// For PWM maximum possible values are 0 to 255
	analogWrite(enA, …);
	analogWrite(enB, …);

	// Turn on motor A & B
	digitalWrite(in1, HIGH);
	digitalWrite(in2, LOW);
	digitalWrite(in3, HIGH);
	digitalWrite(in4, LOW);
	delay(2000);

	// Now change motor directions
	digitalWrite(in1, LOW);
	digitalWrite(in2, HIGH);
	digitalWrite(in3, LOW);
	digitalWrite(in4, HIGH);
	delay(2000);

	// Turn off motors
```

```
	digitalWrite(in1, LOW);
	digitalWrite(in2, LOW);
	digitalWrite(in3, LOW);
	digitalWrite(in4, LOW);
}

// This function lets you control speed of the motors
void speedControl() {
	// Turn on motors
	digitalWrite(in1, LOW);
	digitalWrite(in2, HIGH);
	digitalWrite(in3, LOW);
	digitalWrite(in4, HIGH);

	// Accelerate from zero to maximum speed
	for (int i = 0; i < 256; i++) {
		analogWrite(enA, i);
		analogWrite(enB, i);
		delay(20);
	}

	// Decelerate from maximum speed to zero
	for (int i = 255; i >= 0; --i) {
		analogWrite(enA, i);
		analogWrite(enB, i);
		delay(20);
	}

	// Now turn off motors
	digitalWrite(in1, LOW);
	digitalWrite(in2, LOW);
	digitalWrite(in3, LOW);
	digitalWrite(in4, LOW);
}
```

Joystick

1. Connect the Joystick to the NANO
    - **a) Joystick Pin GND → B23**
    - **b) Joystick Pin +5V → B21**
    - **c) Joystick Pin VRx → B13**
    - **d) Joystick Pin VRy → B14**
    - **e) Joystick Pin SW → J18**



Plug in USB from NANO to computer and run code

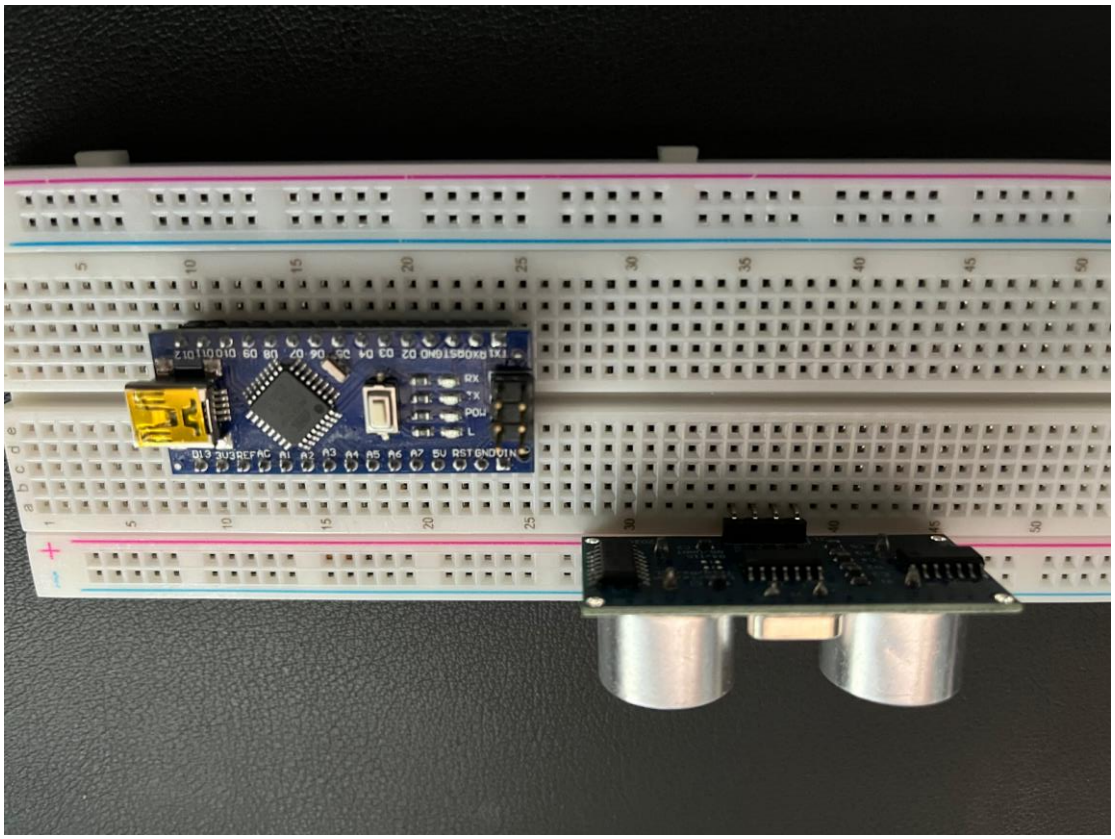
View readings in COM terminal on Arduino IDE

Joystick Code:

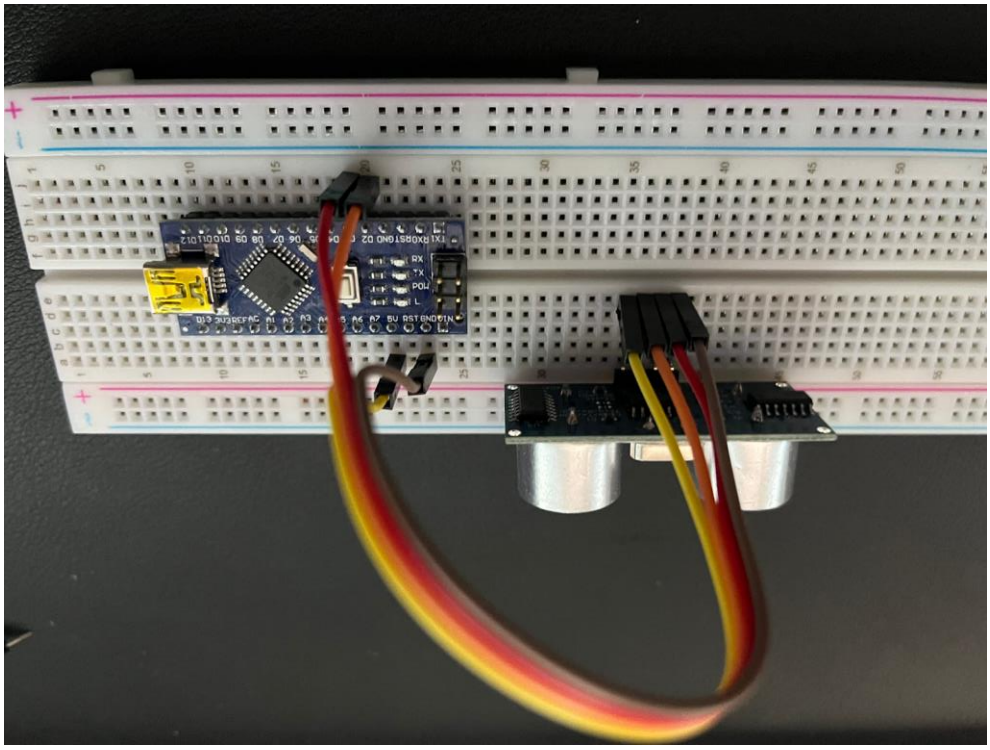
```
// Joystick:
const int VRx = 0; // Connect to Analog Pin 0
const int VRy = …; // Connect to Analog Pin 1
const int SW = …; // Connect to Digital Pin 4

void setup() {
// Joystick:
pinMode(SW, INPUT);
digitalWrite(SW, HIGH);

Serial.begin(9600);
}

void loop() {
// Joystick:
Serial.print("x-axis tilt: ");
Serial.println(analogRead(VRx));
Serial.print("y-axis tilt: ");
Serial.println(analogRead(VRy));
delay(800);
}
```

Ultrasonic Range Finder

2. Leave the NANO as is and plug in the rangefinder to A35-A38



2. Connect rangefinder to NANO
   a. **E38 → A23**
   b. **E37 → J19**
   c. **E36 → J20**
   d. **E35 → A21**



Plug in USB from NANO to computer and run code

View readings in COM terminal on Arduino IDE

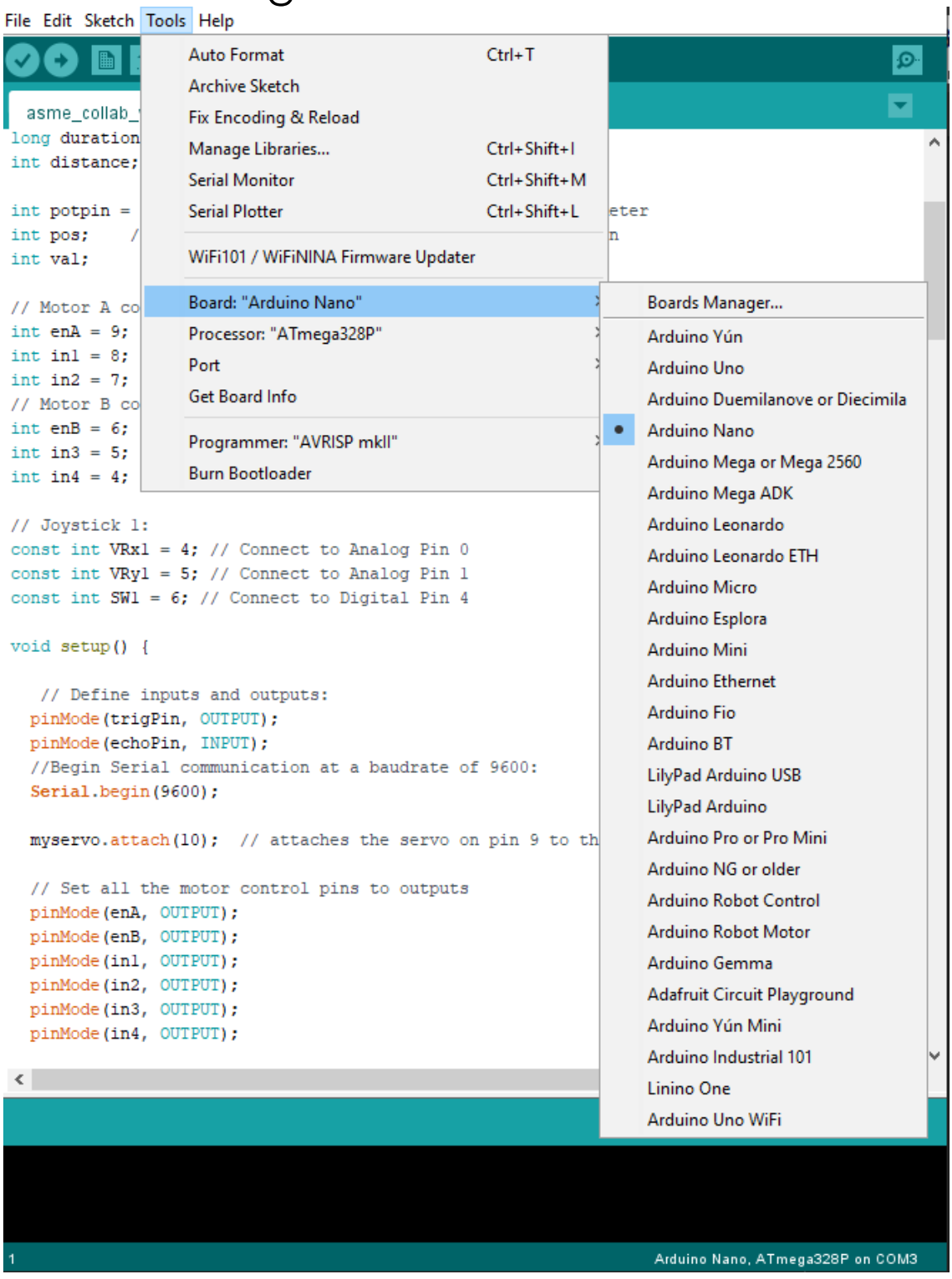
Ultrasonic Range Finder Code:

```
// Define Trig and Echo pin:

#define trigPin … // Set trig pin to Analog Pin 2

#define echoPin … // Set echo pin to Analog Pin 3

// Define variables:

long duration;

int distance;

void setup() {

 // Define inputs and outputs:

 pinMode(trigPin, …); // Set trig pin to OUTPUT

 pinMode(echoPin, …); // Set trig pin to INPUT

 //Begin Serial communication at a baud rate of 9600:

 Serial.begin(9600);

}

void loop() {

 digitalWrite(trigPin, LOW);

 delayMicroseconds(5);

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);
 // Read the echoPin, pulseIn() returns the duration (length of the pulse) in
microseconds:

 duration = pulseIn(echoPin, HIGH);

 // Calculate the distance:

 distance = duration*0.034/2;

 // Print the distance on the Serial Monitor

 Serial.print("Distance = ");

 Serial.print(distance);
```

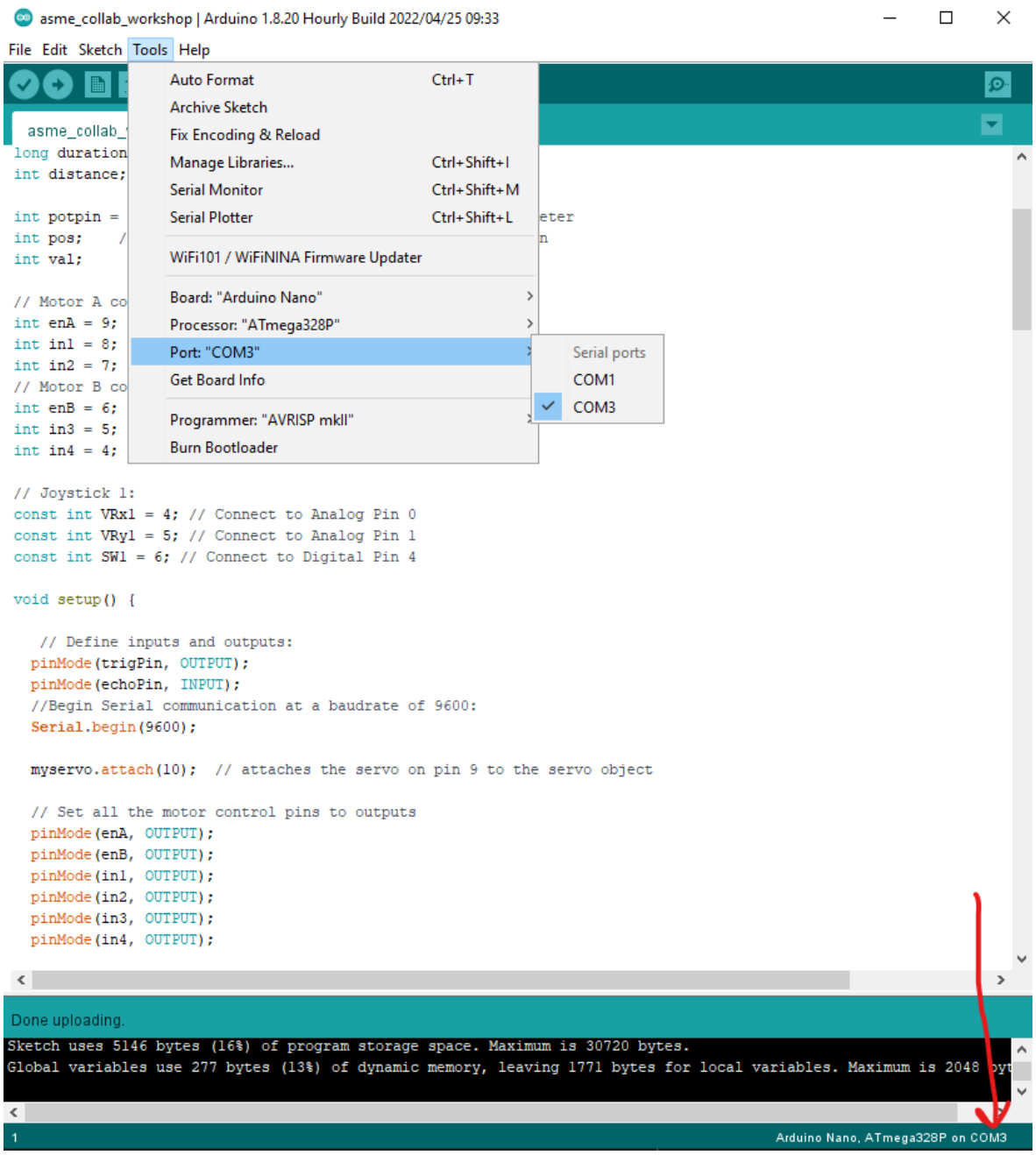
```
 Serial.println(" cm");
 delay(1000);
}
```

How to Display the Serial Monitor

Make sure that when you click on the "Tools" tab the section labeled "Board" has the correct Arduino being used. Which here is the NANO.



Once the NANO is connected the computer will read which COM terminal it is on at the bottom right of the IDE (shown by the red arrow in the image) Click on the “Tools” tab and select the correct COM terminal.



Click on the “Tools” tab and select “Serial Monitor”. Another window will open where you can view the readings of the joystick live.