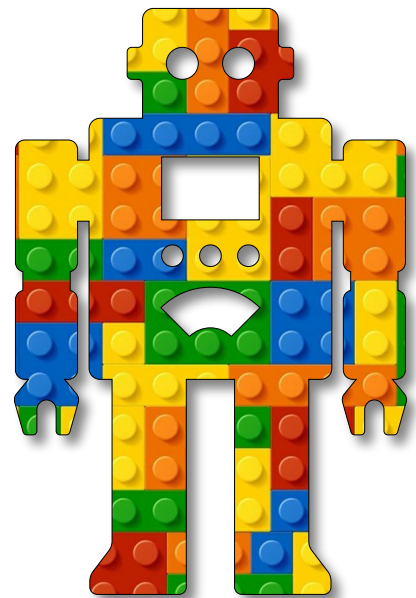


Intelligent Machines Module E Unit #2 slider servo





Module E Unit 2 slider servo

- Sketch E2.1 reading a value
- Sketch E2.2 the port sketch
- Sketch E2.3 the main sketch
- Sketch E2.4 attaching a value
- Sketch E2.5 drawing the servo arm



Introduction to slider servo

We are going to connect our Arduino Nano 33 BLE with p5.js. from there we will control the servo with a slider.



Sketch E2.1 reading a value

Here we will be sending a value (val) through the serial connection with p5.js sketch.

```
#include <Servo.h>
Servo myServo;
int val = 0;

void setup()
{
  myServo.attach(2);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available() > 0)
  {
    val = Serial.read();
    myServo.write(val);
  }
}
```



Notes

The Arduino is in listening mode. waiting for a value (val) to be sent from p5.js.



Sketch E2.2 the port sketch

Make sure that you have everything in the port.js sketch.

port.js

```
const serial = new p5.WebSerial()
let portButton

function navigation()
{
  if (!navigator.serial)
  {
    alert ("WebSerial is not supported. Try Chrome")
  }
  serial.getPorts()
  serial.on("noport", makePortButton)
  serial.on("portavailable", openPort)
  serial.on("requesterror", portError)
  serial.on("close", makePortButton)
}

function makePortButton()
{
  portButton = createButton("choose port")
  portButton.position(10, 10)
  portButton.mousePressed(choosePort)
}

function choosePort()
{
  if (portButton) portButton.show()
  serial.requestPort()
}

function openPort()
{
  serial.open().then(initiateSerial)
  function initiateSerial()
```

```
{
  console.log("port open")
  if (portButton) portButton.hide()
}

function portError(err)
{
  alert("Serial port error: " + err)
}

function portConnect()
{
  console.log("port connected")
  serial.getPorts()
}

function portDisconnect()
{
  serial.close()
  console.log("port disconnected")
}

function closePort()
{
  serial.close()
}
```



Notes

There is nothing new here.



Sketch E2.3 the main sketch

! In sketch.js.

Starting with a fairly basic sketch. We have a slider (remember the LED unit?).

sketch.js

```
let slider

function setup()
{
  createCanvas(400, 400)
  slider = createSlider(0, 180, 0)
  navigation()
}

function draw()
{
  background(220)
}
```



Notes

The slider doesn't do anything yet, we have not attached any functionality to it yet.



Challenge

You could place the slider on the canvas.

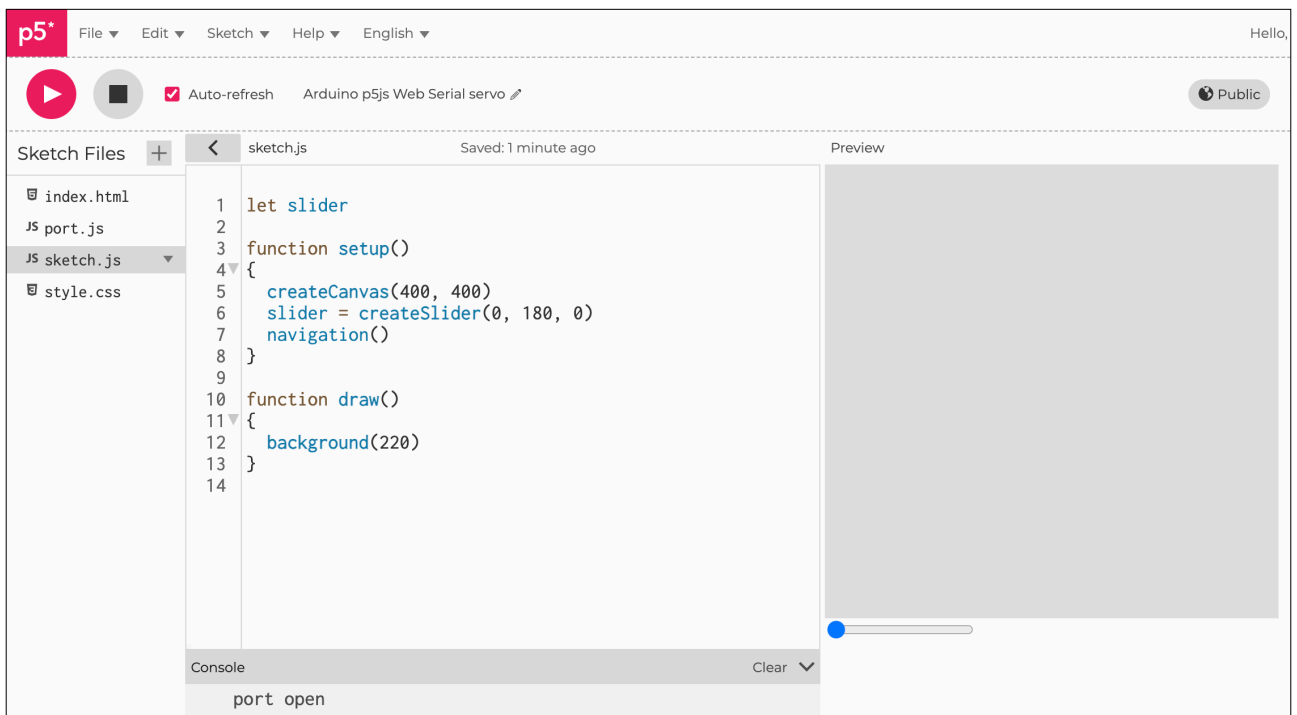


Code Explanation

```
slider = createSlider(0, 180, 0)
```

Creating a slider with value range of 0-180 and a starting value of zero.

Figure E2.3





Sketch E2.4 attaching a value

We want to send the value of the slider to the Arduino Nano 33 BLE.

sketch.js

```
let slider
let val

function setup()
{
  createCanvas(400, 400)
  slider = createSlider(0, 180, 0)
  navigation()
}

function draw()
{
  background(220)
  val = slider.value()
  serial.write(val)
}
```



Notes

When you slide the slider you should see the servo copy the movement from 0° to 180°.



Code Explanation

<code>val = slider.value()</code>	Receives the value from the slider
<code>serial.write(val)</code>	Sends the value from the slider to the Arduino.



Sketch E2.5 drawing the servo arm

Rather than draw the whole thing I have just drawn the rotating arm showing that you can display the movement graphically, also have added the value as text.

sketch.js

```
let slider
let val

function setup()
{
  createCanvas(400, 400)
  slider = createSlider(0, 180, 0)
  navigation()
  angleMode(DEGREES)
}

function draw()
{
  background('darkred')
  val = slider.value()
  serial.write(val)
  translate(width/2, height/2)
  strokeWeight(20)
  stroke('yellow')
  line(0, 0, 100 * sin(val), 100 * cos(val))
  textSize(30)
  noStroke()
  fill('yellow')
  text('angle: ' + val, -150, -150)
}
```



Notes

The arm follows the slider and the servo.



Challenge

1. Draw the rest of the servo.
2. Draw it in 3D.

Figure E2.6

