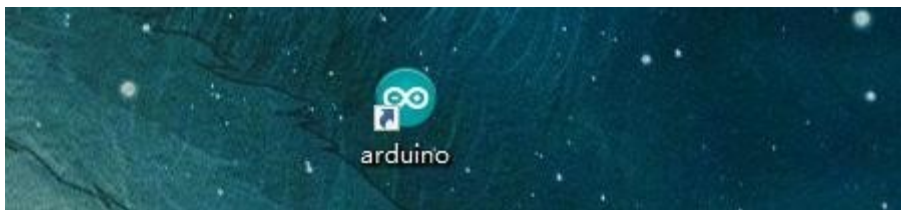


## Lesson 10 Processing controls robotic arm to write and draw

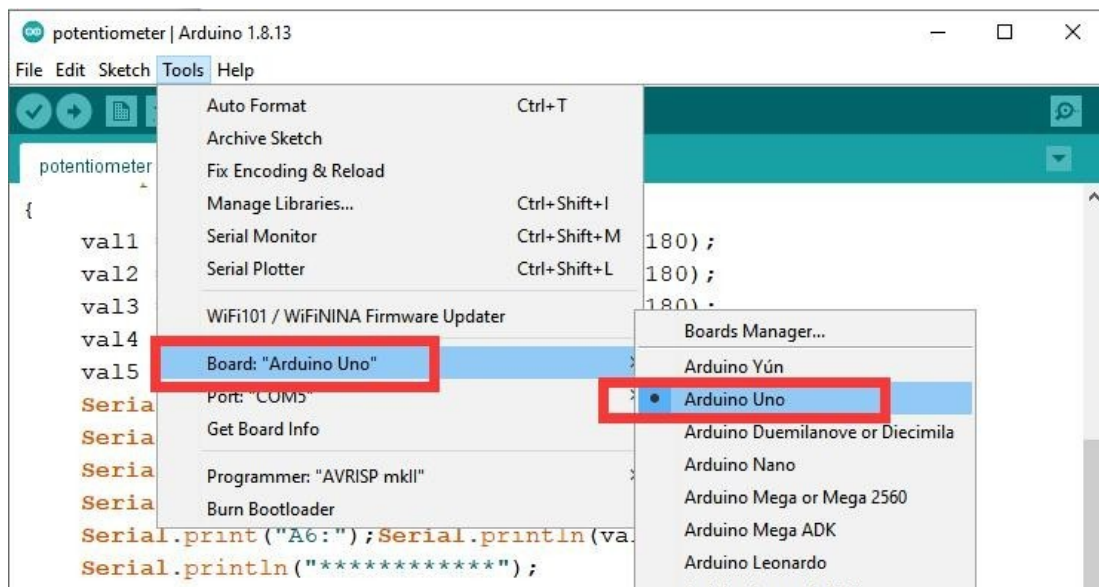
In this course, we will learn how to use Processing software to control robotic arms write and draw.

### 10.1 Upload the WritingAndDrawing.ino

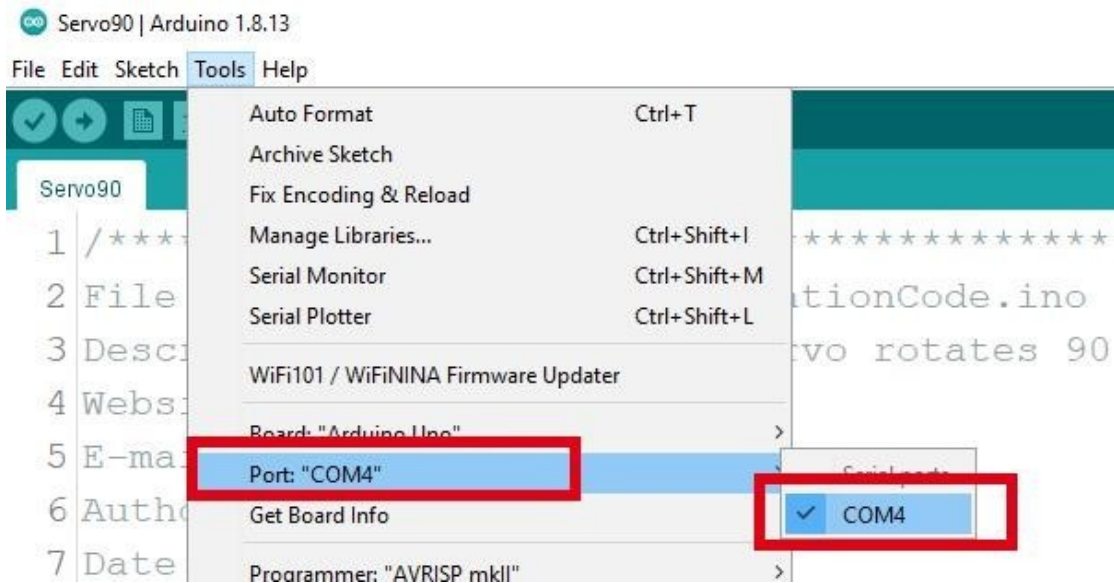
1. Open the Arduino IDE software, as shown below:



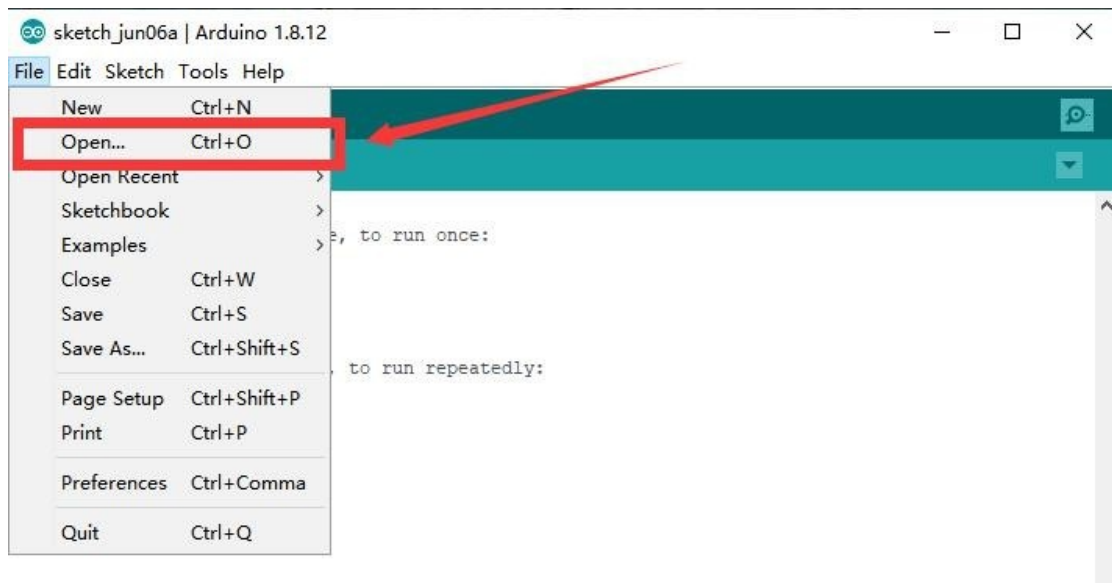
2. In the Tools toolbar, find Board and select Arduino Uno, as shown below:



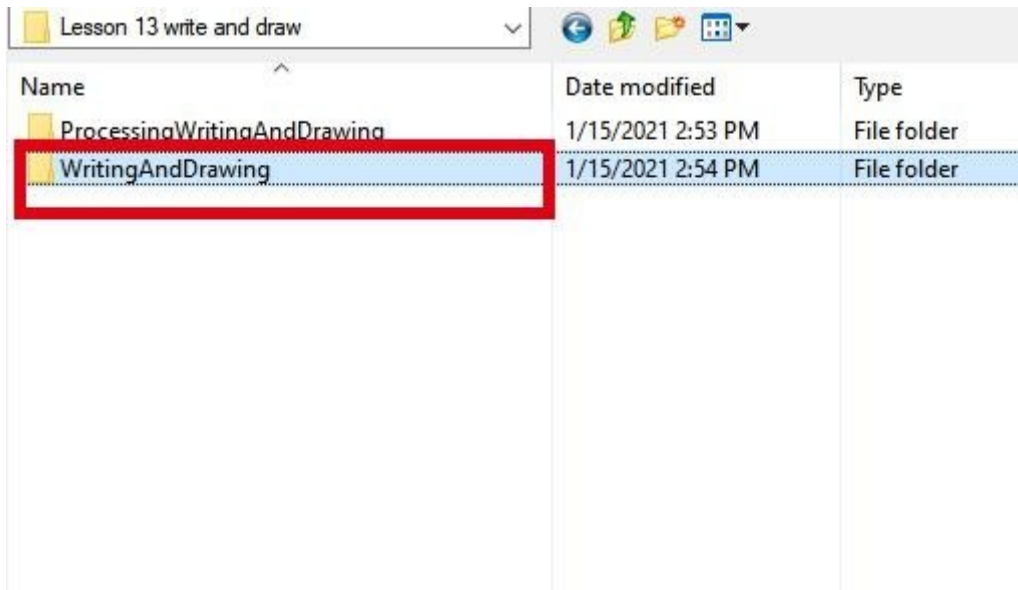
3. In the Tools toolbar, find "Port" and Select the port number of The Adeept Arm Drive Board , as shown below:



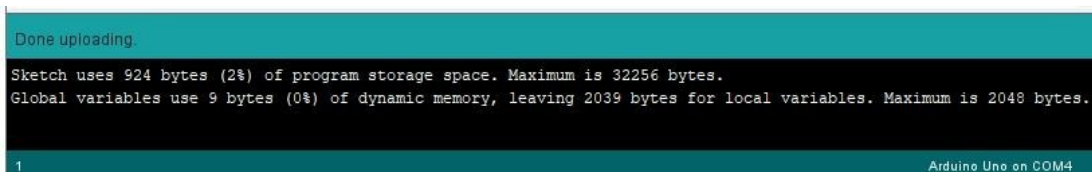
4. Click Open in the File drop-down menu:



5. Find the folder AdeptRoboticArmforArduinoV3\_5 that we provide to the user. Open the folder 02 Course Code in it. Enter the Lesson 13 write and draw→WritingAndDrawing directory. Select WritingAndDrawing.ino. This file is the code program we need in this course. Then click Open.



6. After opening, click to upload the code program to the Adept Arm Drive Board. If there is no error warning in the console below, it means that the Upload is successful.



7. After downloading, close WritingAndDrawing.ino
8. Note that the arm is still connected to the computer with the USB cable. Turn on the power supply.

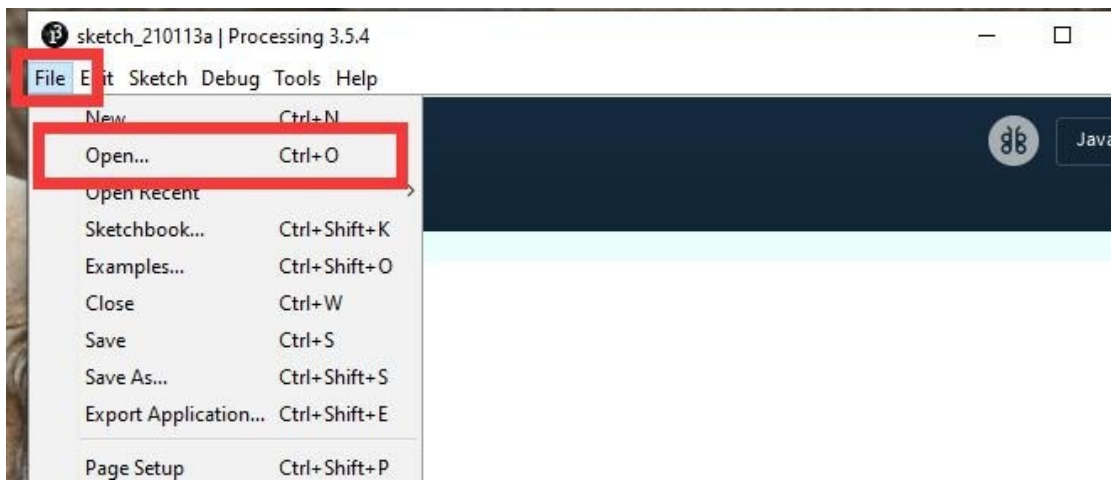
## 10.2 Run the ProcessingWritingAndDrawing.pde

**Note that the arm is still connected to the computer with the USB cable. Turn on the power supply.**

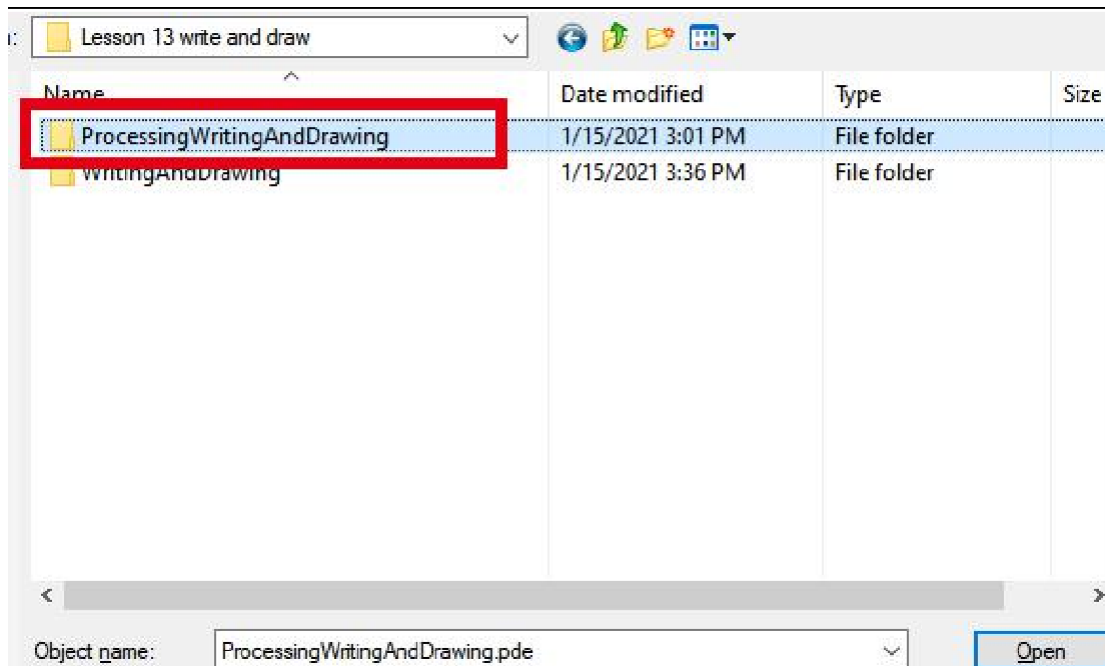
1. Open the Processing software, as shown below:




2. Click Open in the File drop-down menu:



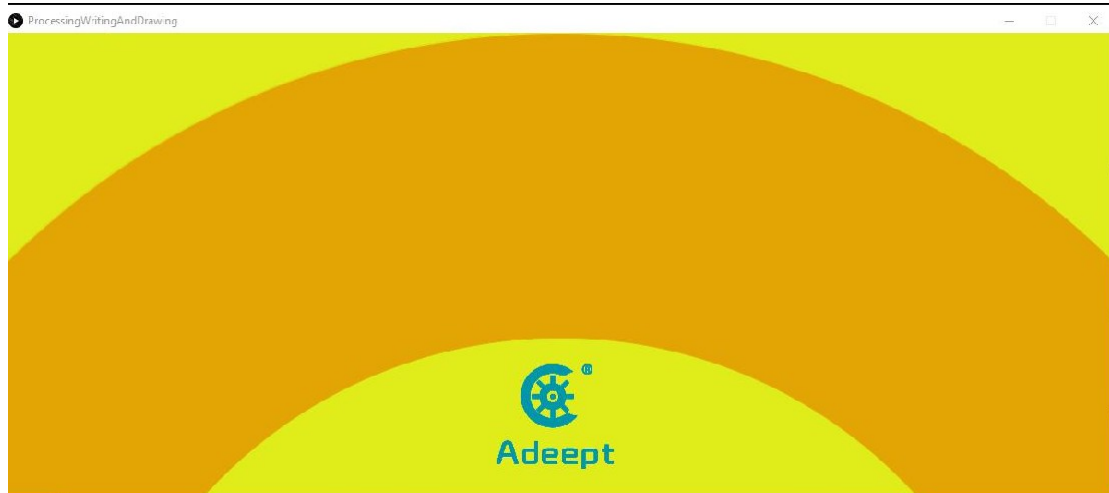
3. Find the folder AdeptRoboticArmforArduinoV3\_5 that we provide to the user. Open the folder 02 Course Code in it. Enter the Lesson 13 write and draw→WritingAndDrawing directory. Select ProcessingWritingAndDrawing.pde. This file is the code program we need in this course. Then click Open.



4. After opening, Click " " to run the code, as shown below:

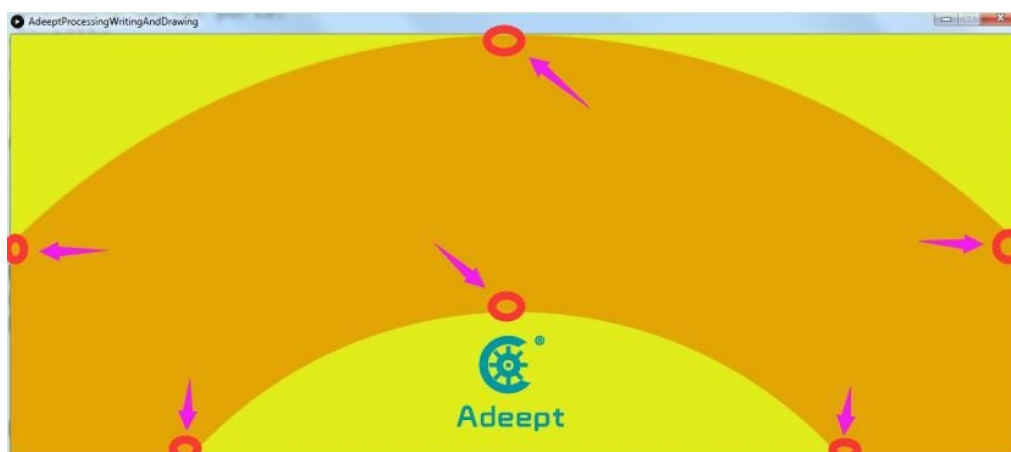


5. Then click "Run" to run the control panel program, the following control interface will appear:

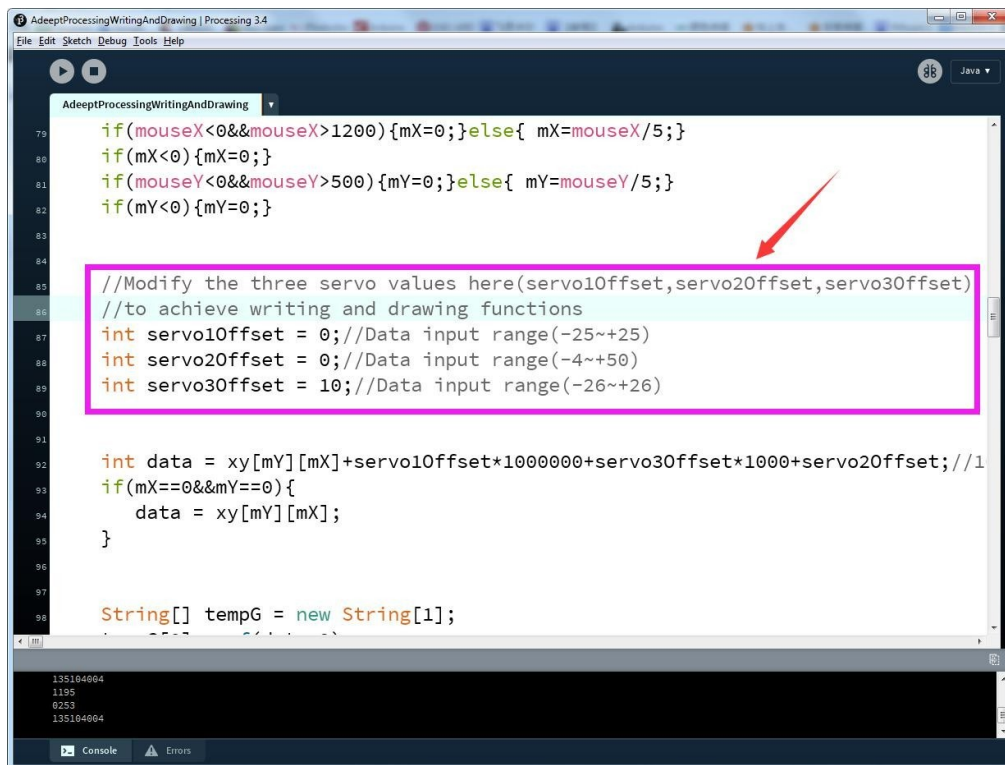


6. Draw or write in the yellow area with the mouse, you will see that the robotic arm paints what we depict on the control panel on the paper. Note that due to errors in the servo, etc., the content depicted by the arm will be slightly biased.

7. Click the dots as shown below:



8. Click on each point to see if the pen tip falls on the paper. If it is not or is suspended in the air, three parameters need to be modified. These three parameters are to fine-tune the offset of the servo. In this experiment, three servo systems need to be adjusted initially. Otherwise, the pen may not be able to write and draw, or produce bad fonts (note that there is a certain error, these three parameters do not need to be precise). In the process of debugging, you will find that there are several sets of data that can make the robot work normally.



```
79  if(mouseX<0&&mouseX>1200){mX=0;}else{ mX=mouseX/5;}
80  if(mX<0){mX=0;}
81  if(mouseY<0&&mouseY>500){mY=0;}else{ mY=mouseY/5;}
82  if(mY<0){mY=0;}
83
84
85  //Modify the three servo values here(servo10ffset,servo20ffset,servo30ffset)
86  //to achieve writing and drawing functions
87  int servo10ffset = 0;//Data input range(-25~+25)
88  int servo20ffset = 0;//Data input range(-4~+50)
89  int servo30ffset = 10;//Data input range(-26~+26)
90
91
92  int data = xy[mY][mX]+servo10ffset*1000000+servo30ffset*1000+servo20ffset;//1
93  if(mX==0&&mY==0){
94      data = xy[mY][mX];
95  }
96
97
98  String[] tempG = new String[1];
99
100
```

Console

```
135104004
1195
0253
135104004
```

9. After modifying the three parameters to make the tip of the pen reach to the paper (do not make the tip press against the paper), click "Run" to run the control panel program. The robotic arm will paint as you write or draw in the dark yellow area with the mouse.



