

Note

Before you start reading this document, please make sure you have read “Tutorial.pdf”.

About

Freenove provides open source electronic products and services.

Freenove is committed to helping customers learn programming and electronic knowledge, quickly implement product prototypes, realize their creativity and launch innovative products. Our services include:

- Kits for learning programming and electronics
- Kits compatible with Arduino®, Raspberry Pi®, micro:bit®, etc.
- Kits for robots, smart cars, drones, etc.
- Components, modules and tools
- Design and customization

To learn more about us or get our latest information, please visit our website:

<http://www.freenove.com>

Copyright

All the files provided in the ZIP file are released under [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/). You can find a copy of the license in the ZIP file.



It means you can use these files on your own derived works, in part or completely. But not for commercial use.

Freenove® brand and logo are trademarks of Freenove Creative Technology Co., Ltd. Must not be used without permission.



Other registered trademarks and their owners appearing in this document:

Arduino® is a trademark of Arduino LLC (<https://www.arduino.cc/>).

Raspberry Pi® is a trademark of Raspberry Pi Foundation (<https://www.raspberrypi.org/>).

micro:bit® is a trademark of Micro:bit Educational Foundation (<https://www.microbit.org/>).

Contents

Note	i
Contents	l
Preface	1
Processing Software	1
First Use	3
Chapter 1 Voltmeter.....	5
Project 1.1 Voltmeter	5
Project 1.2 Dual-Channel Voltmeter	10
Chapter 2 Oscilloscope	13
Project 2.1 Oscilloscope.....	13
Chapter 3 Control 2D and 3D Figures.....	17
Project 3.1 Ellipse	17
Project 3.2 Box 3D	20
Chapter 4 Snake Game	22
Project 4.1 Snake Game	22
Project 4.2 Snake Game 3D	26
Chapter 5 Pong Game	28
Project 5.1 Pong Game	28
Project 5.2 Pong Game 3D	32
What's Next?	34

Preface

Processing is an easy-to-use, free and open-source software for writing graphical programs to run on a computer.

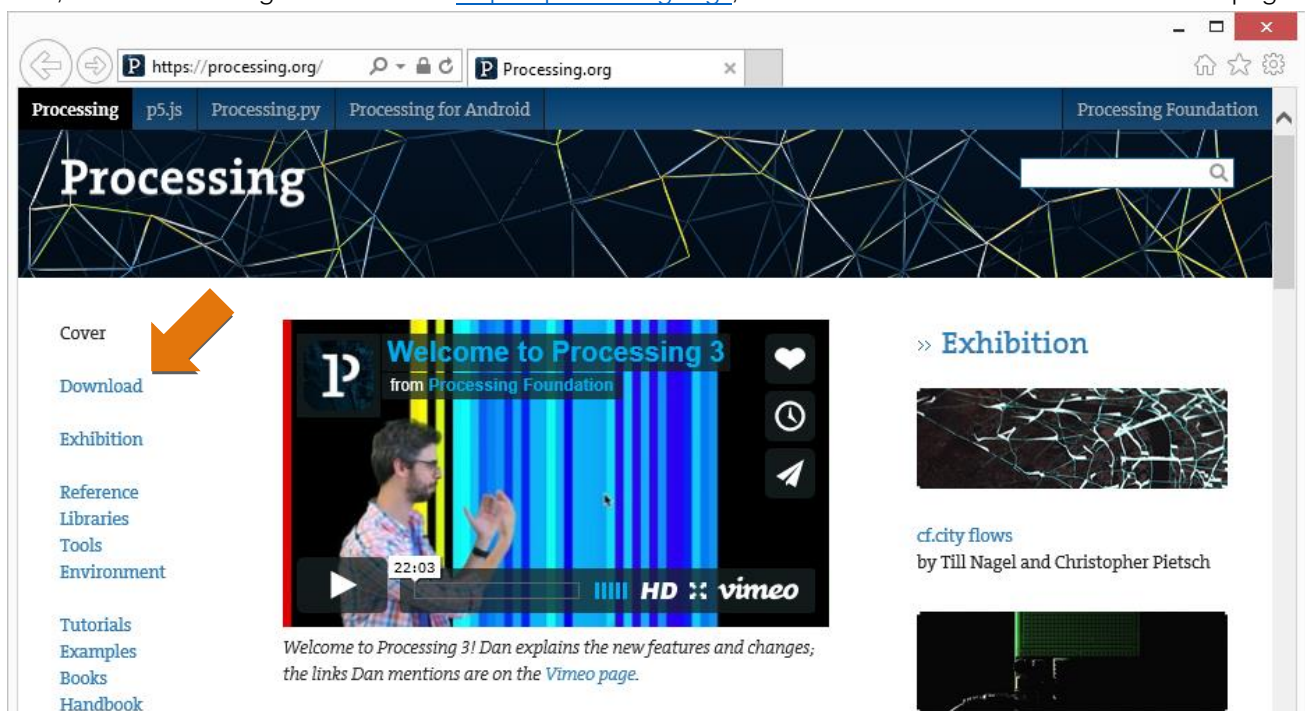
This document will show you how to use Processing to write programs to communicate with a control board. By this means, we can make virtual instruments, game consoles and other projects.

Processing Software

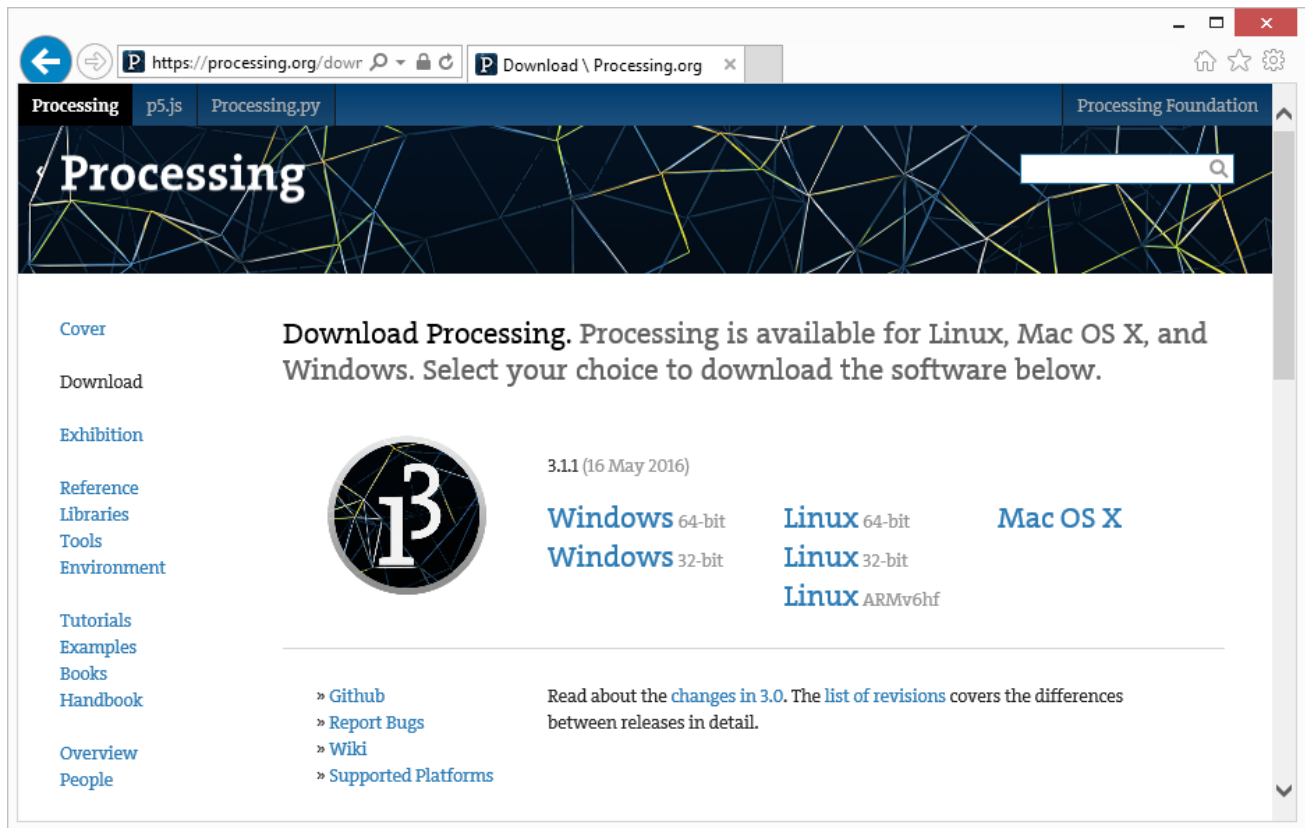
Processing software / Processing Development Environment (PDE) makes it easy to write programs.

Processing software uses Java programming language by default. Do not worry if you don't know Java, because we provide complete code. You can learn Java later if you are interested in it.

First, install Processing software. Visit <https://processing.org/>, click "Download" to enter the download page.



Click the appropriate link for your operating system.



Installation on each machine is straightforward:

- On Windows, you'll have a .zip file. Double-click it, and drag the folder inside to a location on your hard disk. It could be Program Files or simply the desktop, but the important thing is for the processing folder to be pulled out of that .zip file. Then double-click processing.exe to start.
- The Mac OS X version is also a .zip file. Double-click it and drag the Processing icon to the Applications folder. If you're using someone else's machine and can't modify the Applications folder, just drag the application to the desktop. Then double-click the Processing icon to start.
- The Linux version is a .tar.gz file, which should be familiar to most Linux users. Download the file to your home directory, then open a terminal window, and type:

```
tar xvfz processing-xxxx.tgz
```

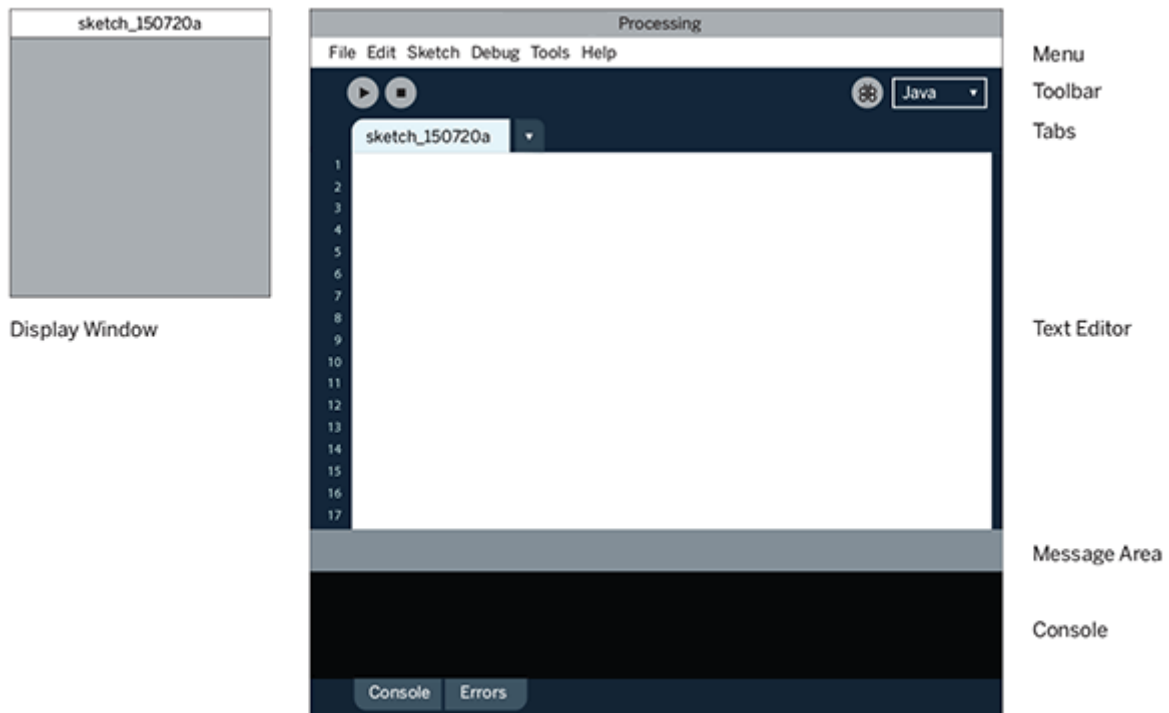
(Replace xxxx with the rest of the file's name, which is the version number.) This will create a folder named processing-2.0 or something similar. Then change to that directory:

```
cd processing-xxxx
```

and run it:

```
./processing
```

With any luck, the main Processing window will now be visible. Everyone's setup is different, so if the program didn't start, or you're otherwise stuck, visit the [troubleshooting page](#) for possible solutions.



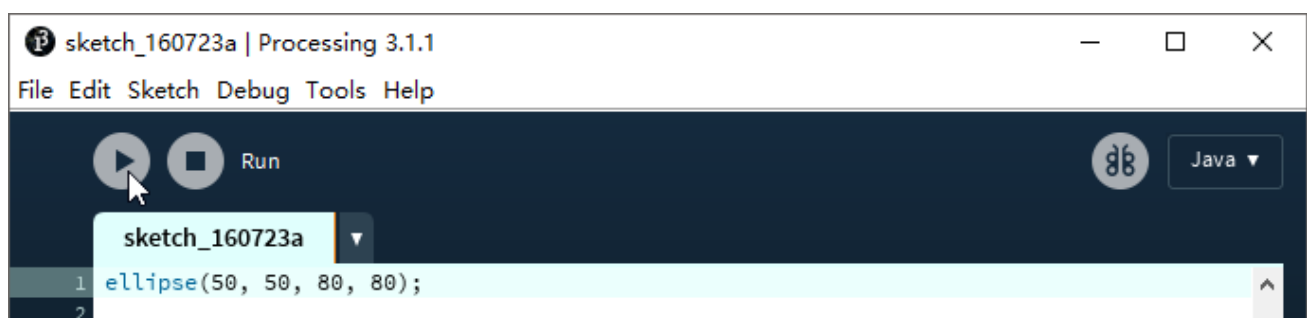
You're now running the Processing Development Environment (or PDE). There's not much to it; the large area is the Text Editor, and there's a row of buttons across the top; this is the toolbar. Below the editor is the Message Area, and below that is the Console. The Message Area is used for one line messages, and the Console is used for more technical details.

First Use

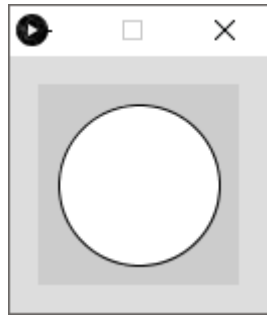
In the editor, type the following:

```
1 ellipse(50, 50, 80, 80);
```

This line of code means "draw an ellipse, with the center 50 pixels over from the left and 50 pixels down from the top, with a width and height of 80 pixels." Click the Run button (the triangle button in the Toolbar).



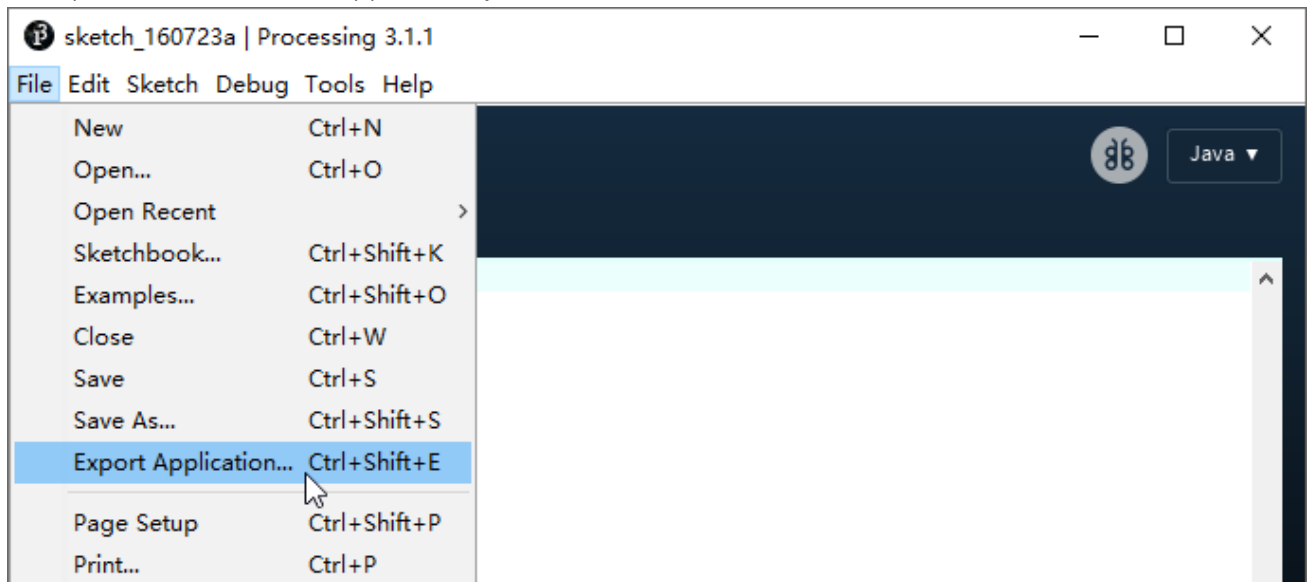
If you've typed everything correctly, you'll see a circle on your screen.



If you didn't type it correctly, the Message Area will turn red and complain about an error. If this happens, make sure that you've copied the example code exactly: the numbers should be contained within parentheses and have commas between each of them, and the line should end with a semicolon.



You can export this sketch to an application to run it directly without opening the Processing. To export the sketch to the application, you must first save it.



So far, we have completed the first use. I believe you have felt the joy of it.

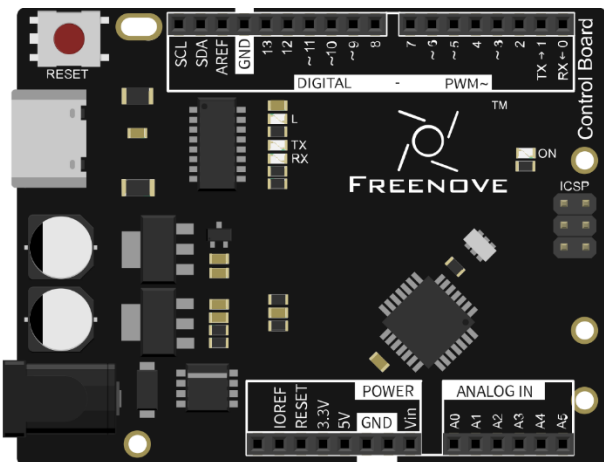
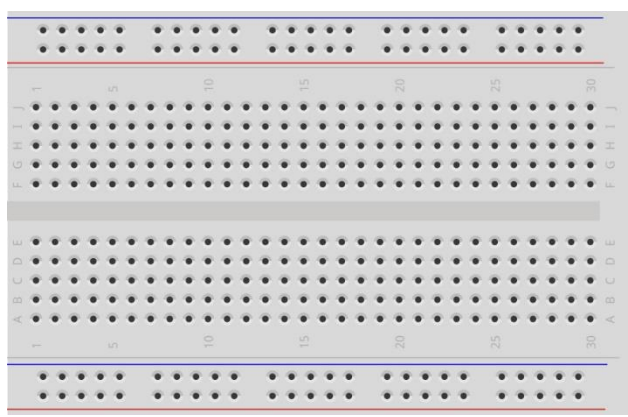

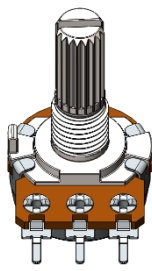

Chapter 1 Voltmeter

In this chapter, we will use a control board and Processing to make a simple voltmeter to understand the mutual communication between them.

Project 1.1 Voltmeter

First, make a simple voltmeter.

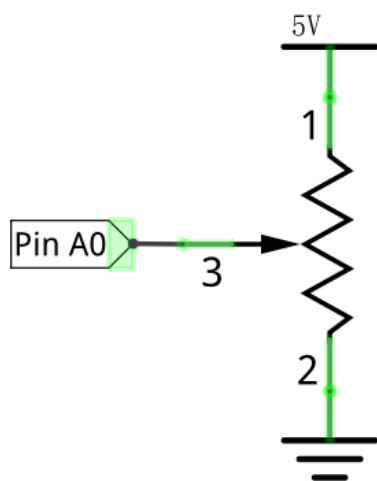
Component List

<p>Control board x1</p>  <p>The image shows a black PCB labeled 'FREENOVE' and 'Control Board'. It features a USB Type-C port, a red push-button labeled 'RESET', and various pin headers. The headers are labeled: 'DIGITAL' (pins 1-13), 'POWER' (IOREF, RESET, 3.3V, 5V, GND, Vin), 'ANALOG IN' (A0-A5), 'TX RX', 'TX~1', 'RX~0', and 'PWM~'. There is also an 'ICSP' header.</p>	<p>Breadboard x1</p>  <p>The image shows a standard white breadboard with a grid of holes. It has power rails on the top and bottom edges, labeled with letters A-J and numbers 1-30.</p>
<p>USB cable x1</p>  <p>The image shows a black USB cable with a USB Type-A connector on one end and a USB Type-C connector on the other.</p>	<p>Rotary potentiometer x1</p>  <p>The image shows a rotary potentiometer with a silver metal shaft and a black plastic body. It has three solder tabs at the bottom.</p>
<p>Jumper M/M x3</p>  <p>The image shows a single green jumper wire with black plastic insulation and metal pins at both ends.</p>	

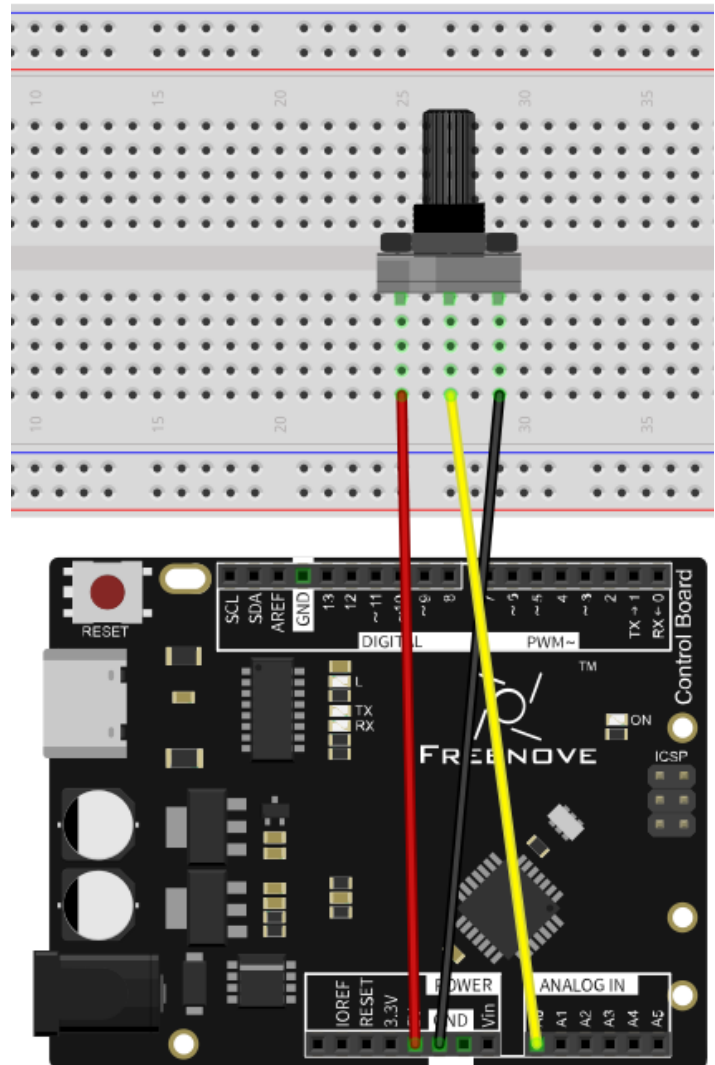
Circuit

Use A0 port on the control board to detect the voltage of the rotary potentiometer.

Schematic diagram



Hardware connection



Communication Protocol

We need to write code for control board and Processing respectively to complete the interaction project of them.

In order to simplify the operation, we have prepared a `SerialDevice` class for Processing to communicate with the control board. To use this class, we need to upload the following sketch to the control board:

Processing\ControlBoard\SerialDevice\SerialDevice.ino.

This sketch only need to be uploaded once and it will be available for the following projects in this tutorial. So the latter projects of this tutorial do not need to upload this code again.

`SerialDevice` class and `SerialDevice.ino` define the communication protocol between them. The features include:

- Recognize the `SerialDevice.ino` uploaded by the control board and establish connection with it automatically. No need to view and set the serial number of the control board connected to the computer, even if there are a number of control boards. It can be connected automatically.
- If `SerialDevice.ino` uploaded by the control board is not connected to computer, the Processing code will not be executed until the connection is done. The Processing sketch does not need to be run again after the connection is done.
- Send data to control board and receive data from it.

Sketch

Before running Processing sketch, make sure that SerialDevice.ino is uploaded to the control board. Processing sketches is stored under the Processing\Processing folder.

Sketch Voltmeter

Use Processing to open Voltmeter.pde and click Run. Then, the following window will pop up and its connection to the control board will be started.



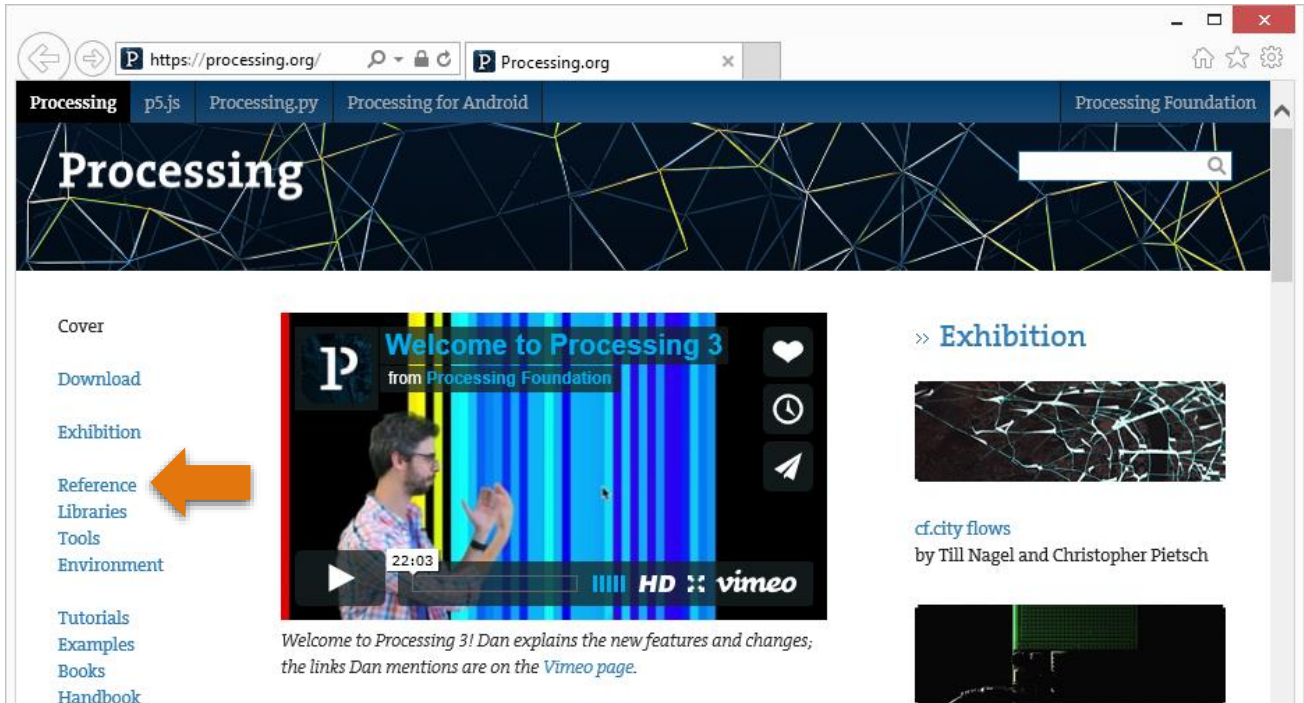
If the control board has not been connected to computer, please connect the control board to your computer. If the connection succeeds, the following will be shown:



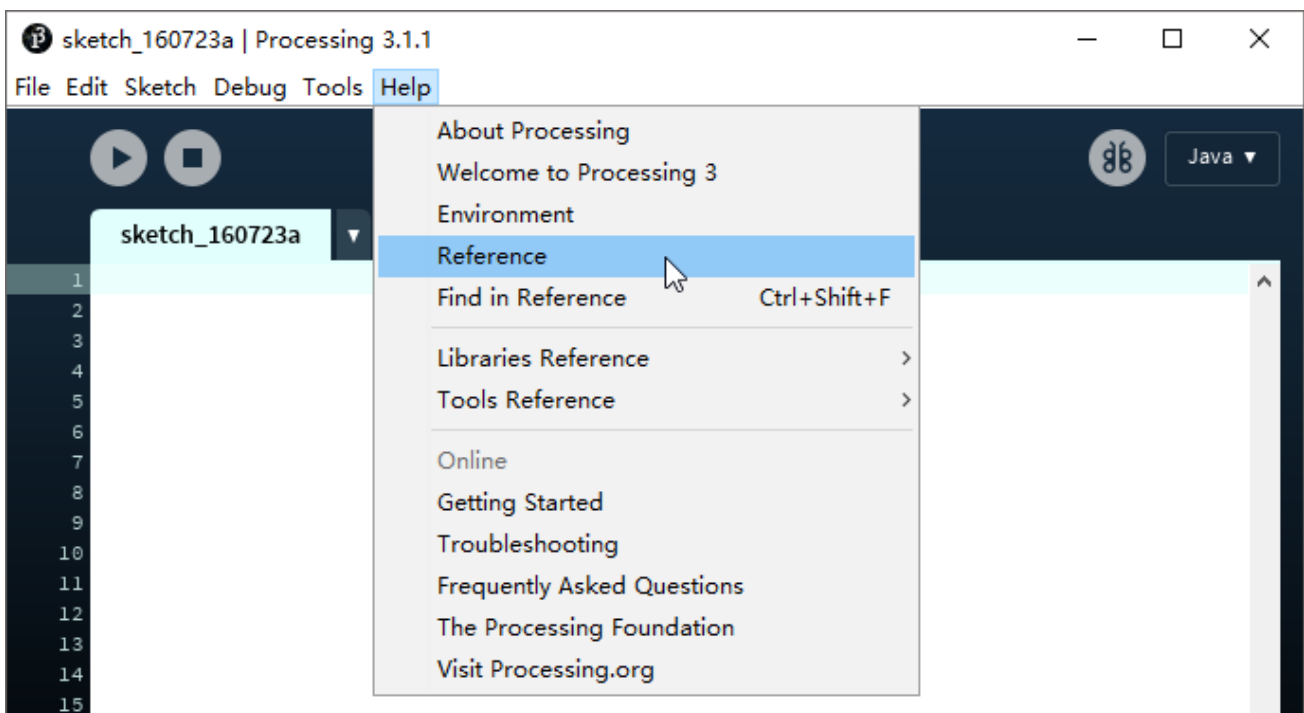
This sketch will obtain analog value from A0 port of control board, and convert it to voltage value to display. You can adjust the potentiometer to observe the change of value, and you can also use the A0 port to measure voltage value of other circuits. Note that the measurement voltage cannot exceed 5V, otherwise it will cause damage to the control board.

Here, Processing sketch code will not be introduced in detail. Readers interested in it can learn it by yourselves.

And as for syntax and standard functions of Processing, you can visit <https://processing.org/> and click Reference to view.



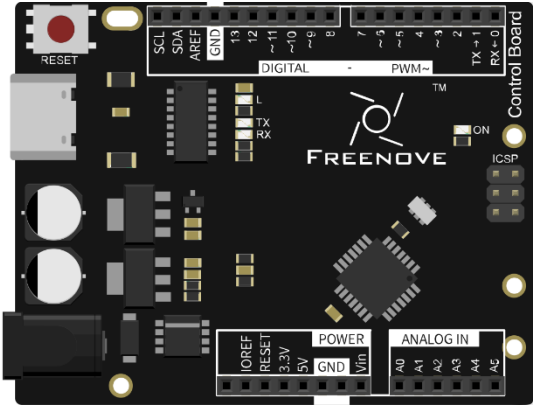
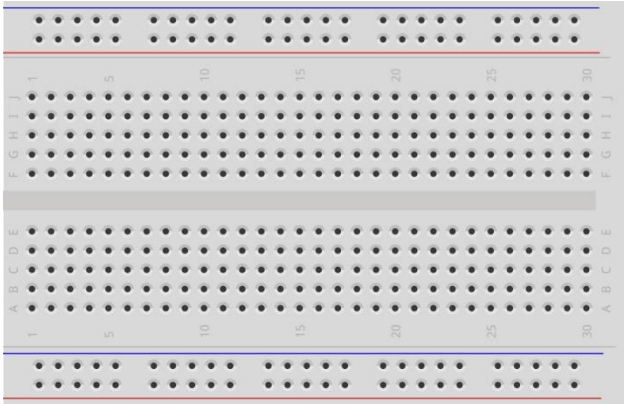


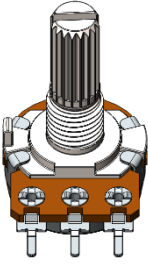
Or in the Processing software menu bar, click Help-Reference to view offline documents.



Project 1.2 Dual-Channel Voltmeter

Now, let's make a dual-channel voltmeter.

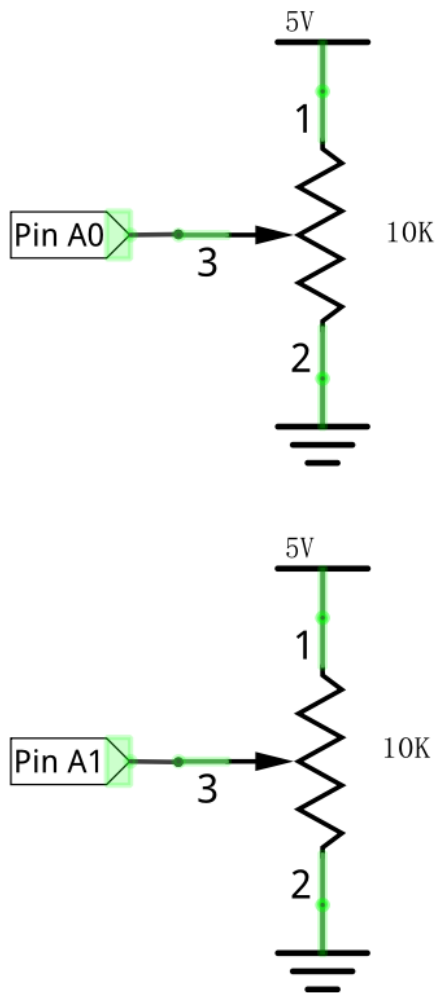
Component List

<p>Control board x1</p>  <p>The image shows a black PCB labeled 'FREENOVE' with various electronic components. It features a USB Type-C port, a reset button, and multiple pin headers for digital, power, and analog connections.</p>	<p>Breadboard x1</p>  <p>The image shows a standard white breadboard with a grid of holes for components. It has power rails on the sides and is labeled with letters A-J and numbers 1-30.</p>
<p>USB cable x1</p>  <p>The image shows a black USB cable with a USB-A connector on one end and a USB-C connector on the other.</p> <p>Jumper M/M x6</p>  <p>The image shows a single green jumper wire with a black plastic cap on one end.</p>	<p>Rotary potentiometer x2</p>  <p>The image shows a rotary potentiometer with a metal shaft and a plastic housing. It has three pins at the bottom for electrical connections.</p>

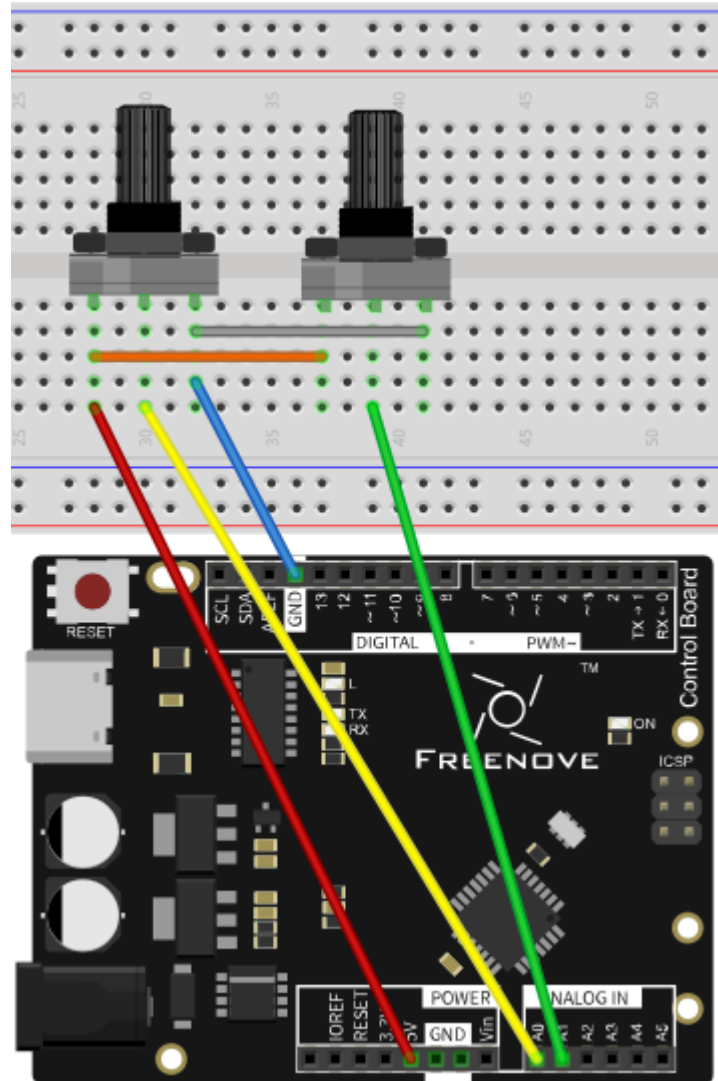
Circuit

Use A0, A1 ports on the control board to detect the voltage of rotary potentiometers.

Schematic diagram



Hardware connection



Sketch

Sketch Voltmeter_Dual_Channel

Use Processing to open Voltmeter_Dual_Channel.pde and click Run. Then, the following window will pop up and its connection to control board will be started.



If you have not yet connect the control board to your computer, please do so. If the connection succeeds, the following will be shown:



This sketch will obtain analog values from A0 and A1 ports of control board, and convert them to voltage value to display. You can adjust the potentiometers to observe the change of value, and you can also use the A0 and A1 ports to measure voltage value of other circuits. Note that the measurement voltage cannot exceed 5V, ortherwise it will cause damage to the control board.

You can export the two Processing sketches in this chapter to the application as common tools.

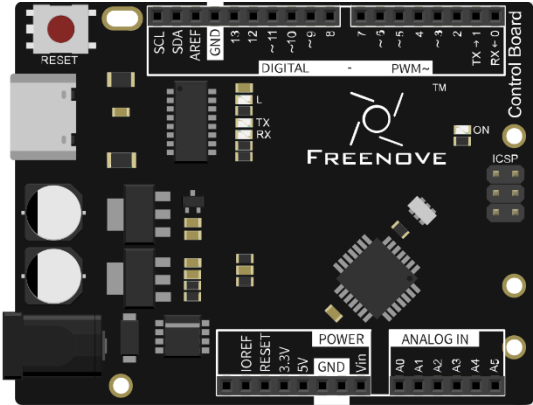
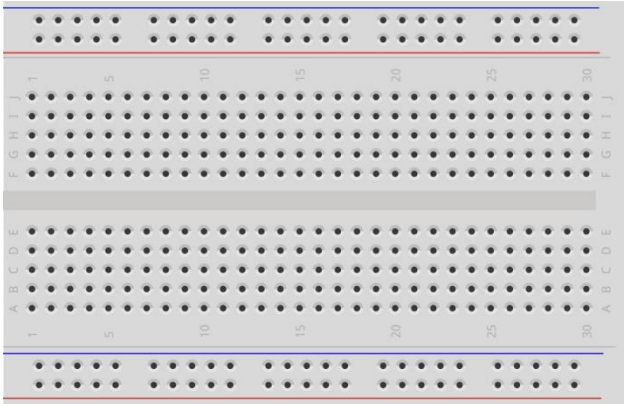

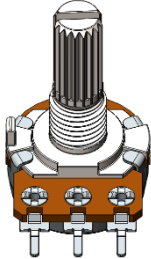

Chapter 2 Oscilloscope

We have implemented a simple virtual instrument voltmeter earlier. In this chapter, we will make a more complex virtual instrument, oscilloscope. Oscilloscope is a widely used electronic measuring instrument. It can get the electrical signals that cannot be observed directly into visible image to facilitate the analysis and study the changing process of various electrical signals.

Project 2.1 Oscilloscope

Now, let's use Processing and control board to achieve an oscilloscope.

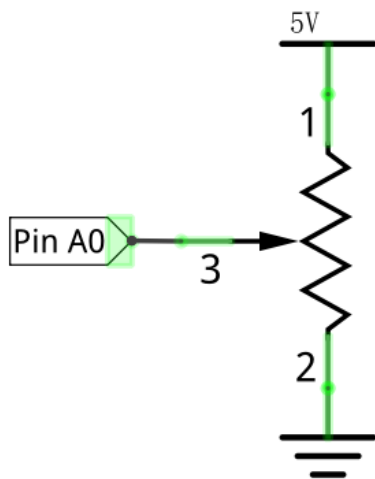
Component List

<p>Control board x1</p> 	<p>Breadboard x1</p> 
<p>USB cable x1</p> 	<p>Rotary potentiometer x1</p> 
<p>Jumper M/M x3</p> 	

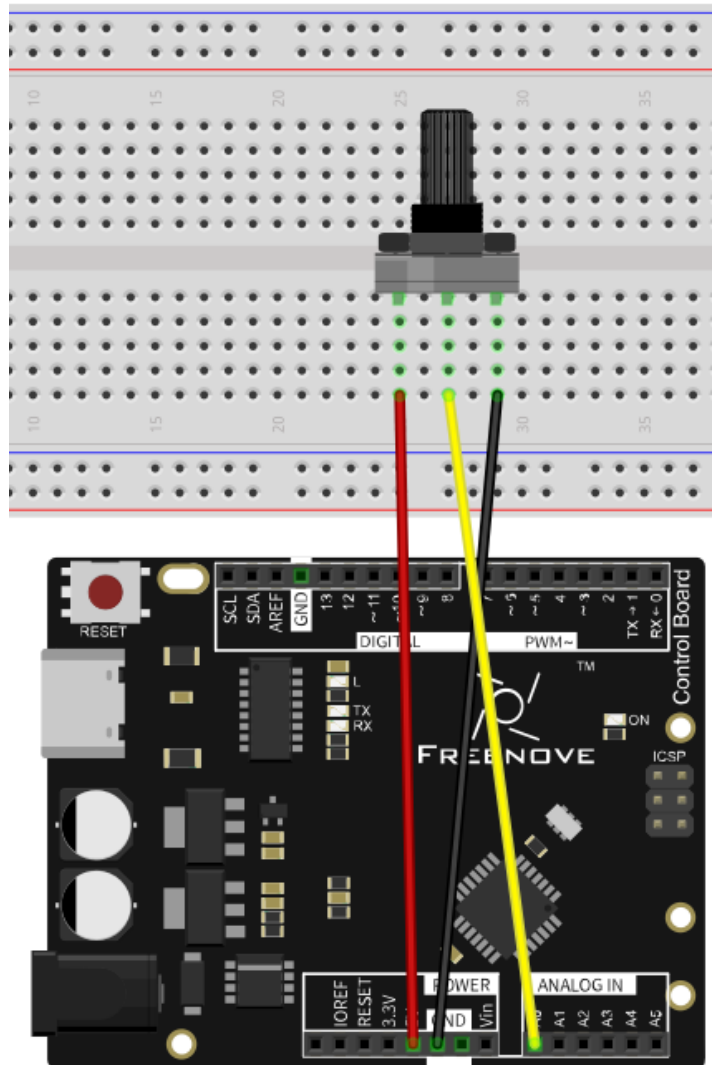
Circuit

Use A0 port on the control board to detect the voltage of rotary potentiometer.

Schematic diagram



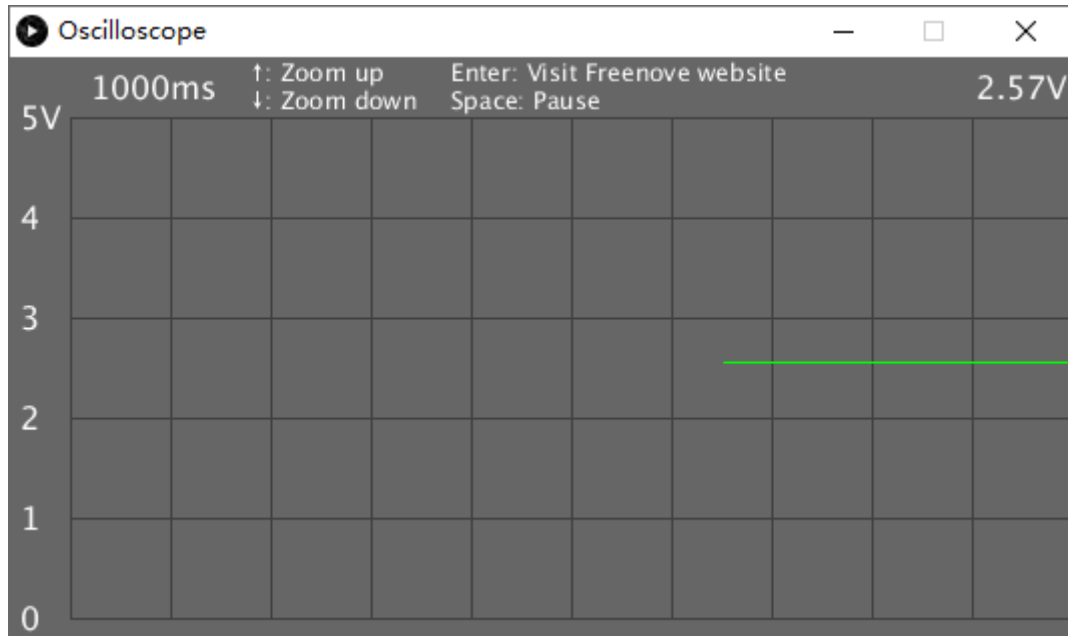
Hardware connection



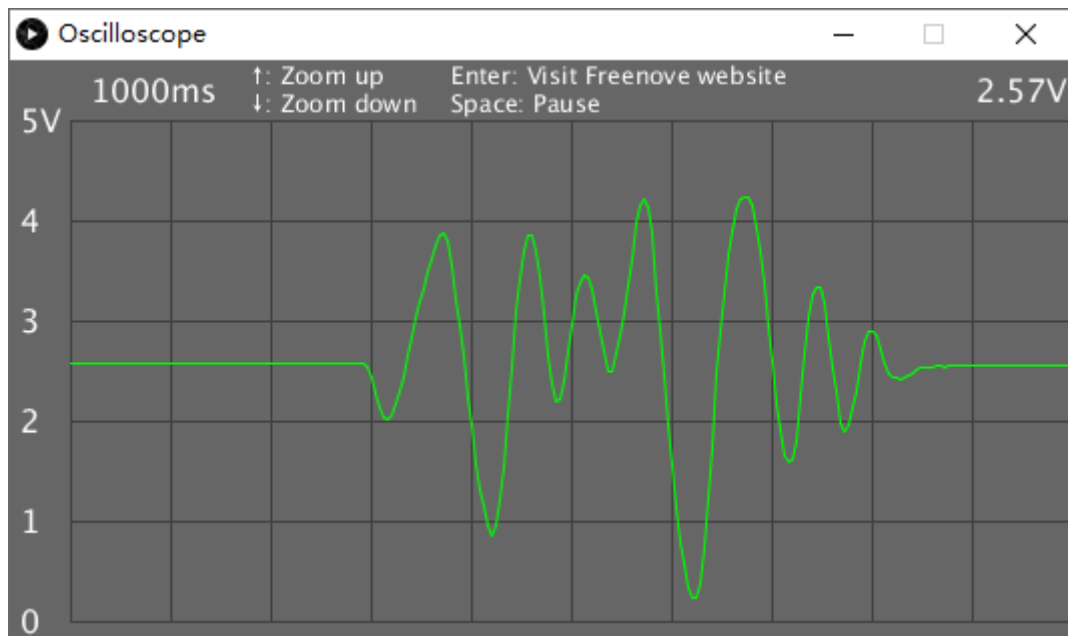
Sketch

Sketch Oscilloscope

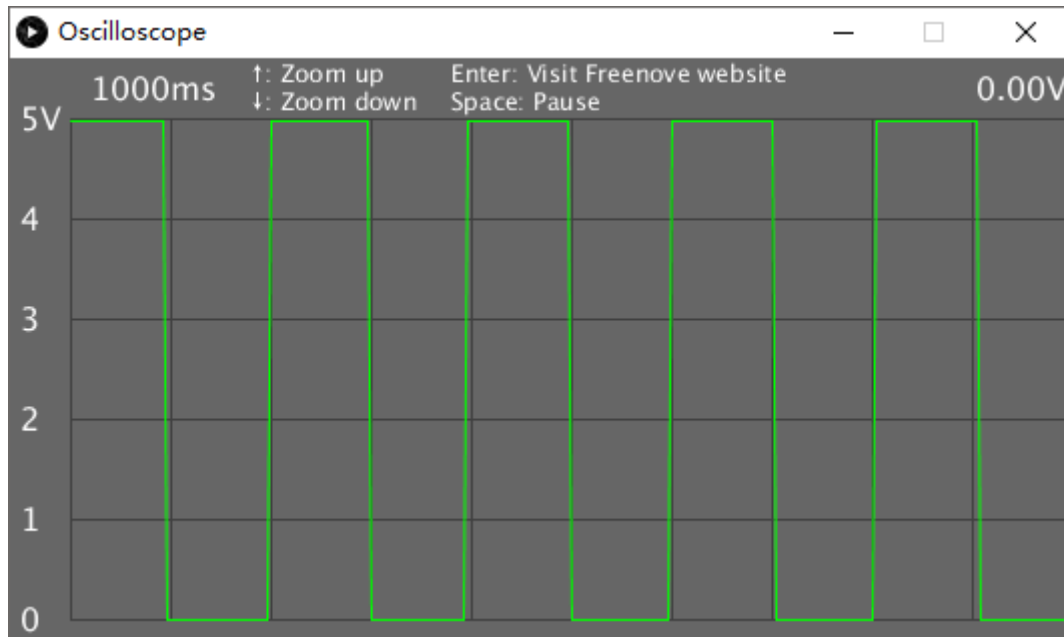
Use Processing to open Oscilloscope.pde and click Run. If the connection succeeds, the following will be shown:



The green line is the waveform collected. Rotate the potentiometer, and you can see changes of the waveform:



Disconnect the A0 port from the potentiometer and connect it to the Pin13 port. The output of Pin13 port is 0.5Hz square wave. As is shown below:



The left side of the software interface is a voltage scale, which is used to indicate the voltage of the waveform. The "1000ms" on top left corner is the time of a square, and you can press "↑" and "↓" key on keyboard to adjust it.

The "0.00V" on top right corner is the voltage value of current signal.

You can press the space bar on keyboard to pause the display of waveform, which is easy to view and analysis.

We believe that with the help of this oscilloscope, you can have a more intuitive understanding of the actual working of some electronic circuits. It will help you complete the project and facilitate troubleshooting. You can export this sketch to an application used as a tool.

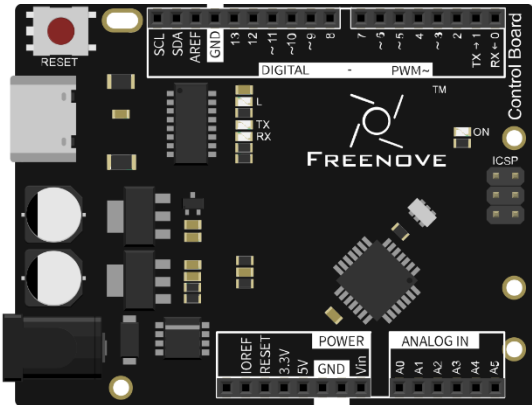
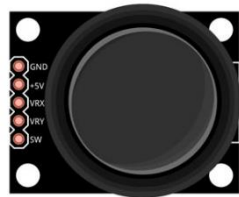


Chapter 3 Control 2D and 3D Figures

In this chapter, we will use the connect board to make Processing program display changes of figures. And we will control 2D and 3D figures respectively.

Project 3.1 Ellipse

First, control a 2D figure.

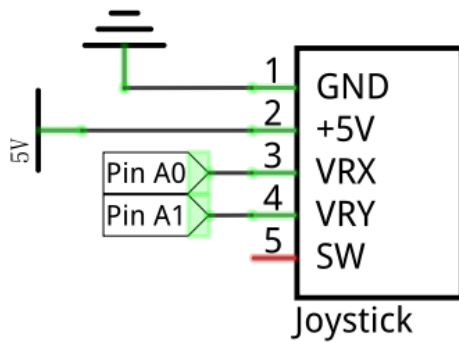
Component List

<p>Connect board x1</p> 	<p>Joystick x1</p> 
<p>USB cable x1</p> 	<p>Jumper F/M x4</p> 

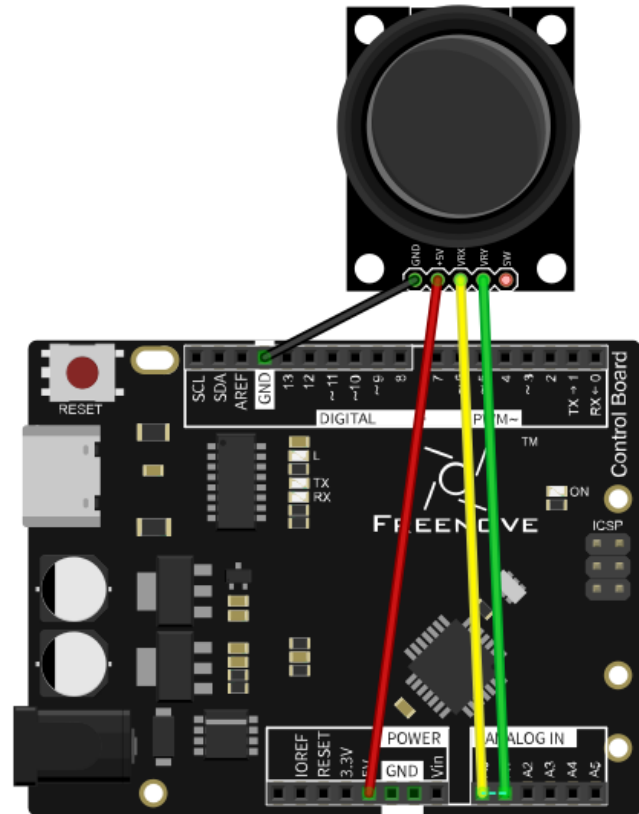
Circuit

Use A0 and A1 ports on connect board to detect the voltage value of two rotary potentiometers inside joystick.

Schematic diagram



Hardware connection



Sketch

Sketch Ellipse

Use Processing to open Ellipse.pde, then click Run. If the connection succeeds, the following will be shown:



Then you can change the shape of the ellipse by shifting the joystick:



Project 3.2 Box 3D

Now control 3D figures.

Component List

The same as previous section.

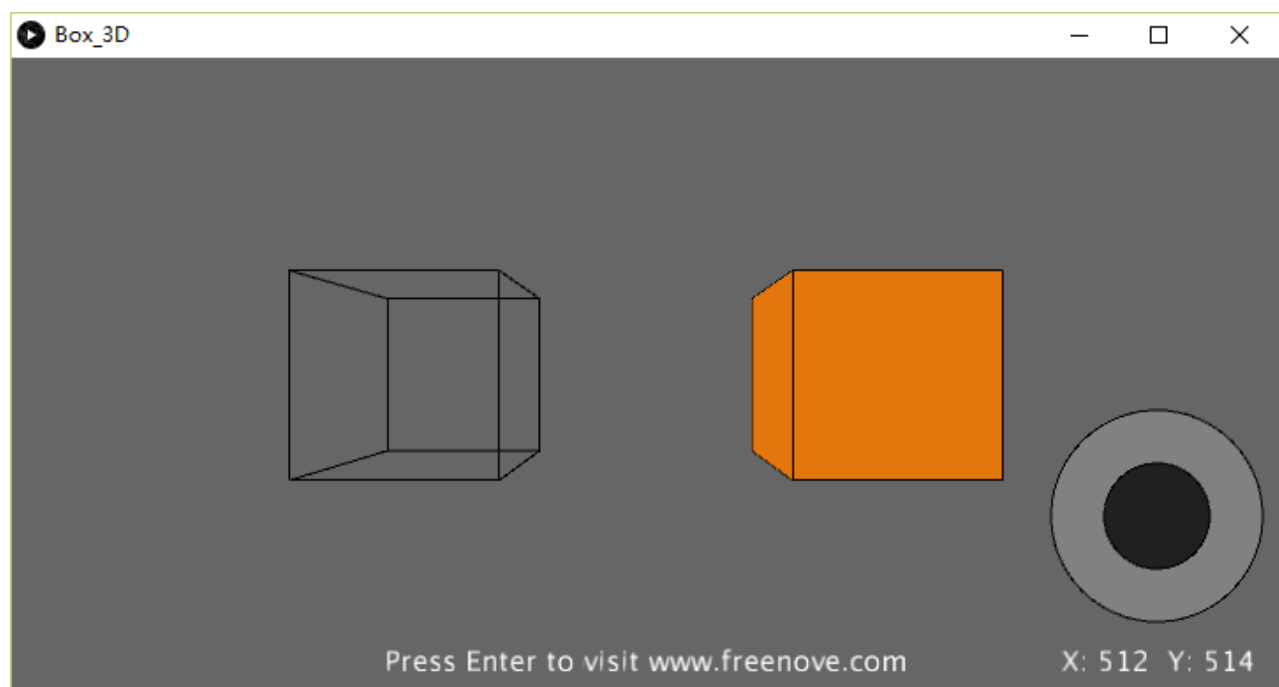
Circuit

The same as previous section.

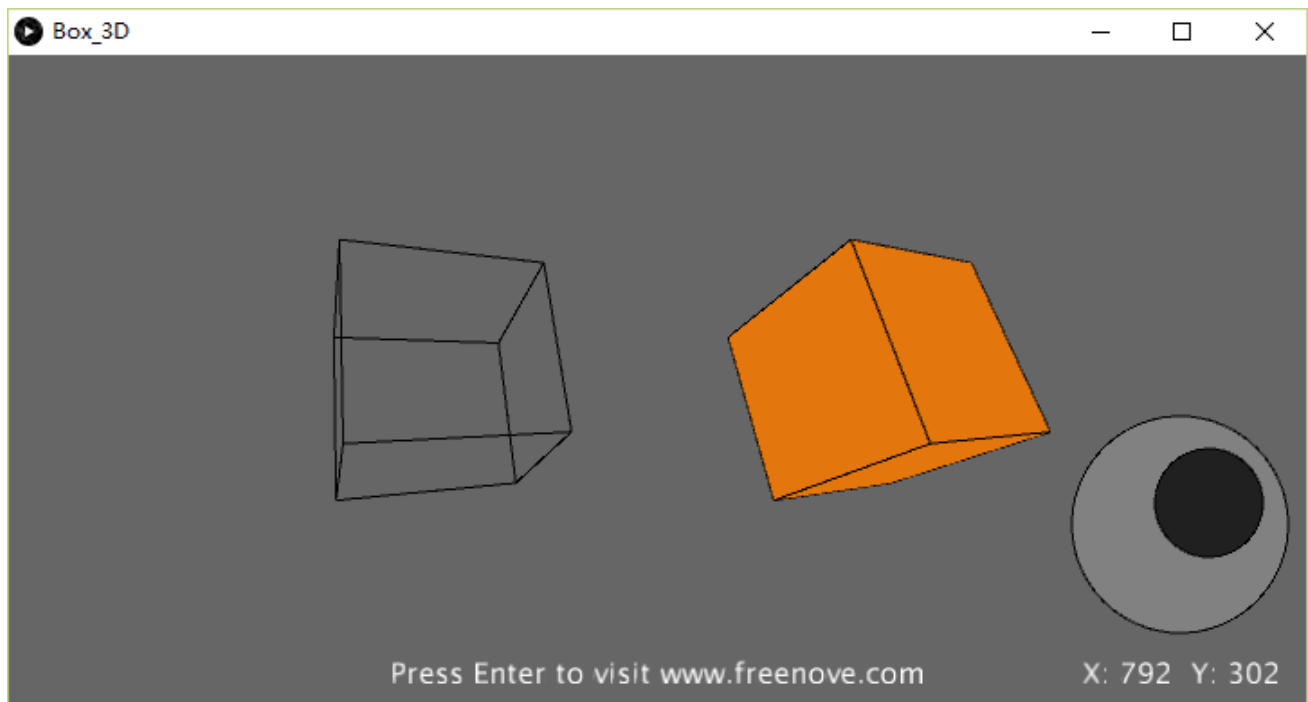
Sketch

Sketch Box_3D

Use Processing to open Box_3D.pde, and click Run. If the connection succeeds, the following will be shown. The left is a 3D box presented by line and the right is a 3D box entity.



Then you can change the space angle of two 3D boxes by shifting the joystick:



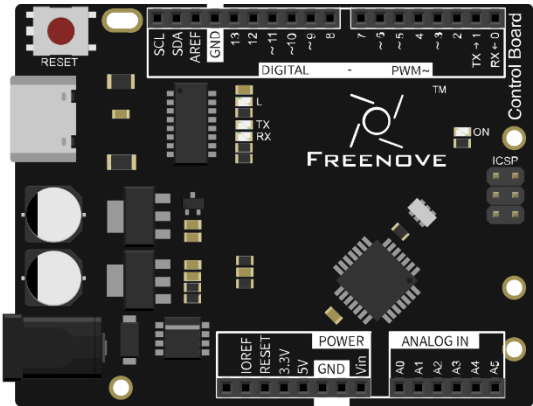
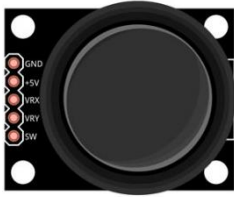


Chapter 4 Snake Game

We have experienced controlling 2D and 3D figures earlier. Now, we use the connect board to play the classic snake game. You will experience both 2D and 3D version.

Project 4.1 Snake Game

First, let's experience the 2D version game.

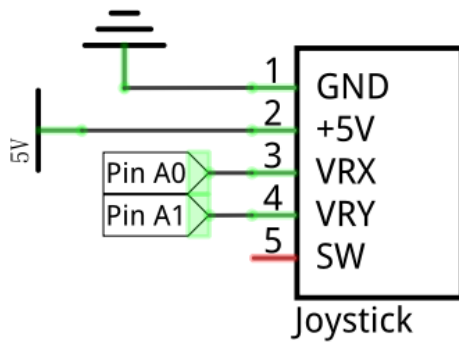
Component List

<p>Connect board x1</p>  <p>The image shows a black Freenove Connect Board. It features a central microcontroller chip, various pins labeled for digital, power, and analog connections, a reset button, and a USB port. The board is populated with several surface components like capacitors and resistors.</p>	<p>Joystick x1</p>  <p>The image shows a small, square joystick module with a black joystick stick in the center. It has four pins on one side, labeled GND, +5V, VCC, and SW.</p>
<p>USB cable x1</p>  <p>The image shows a black USB cable with a standard USB-A connector on one end and a micro-USB connector on the other.</p>	<p>Jumper F/M x4</p>  <p>The image shows a single green jumper wire with a black plastic housing and metal pins at both ends.</p>

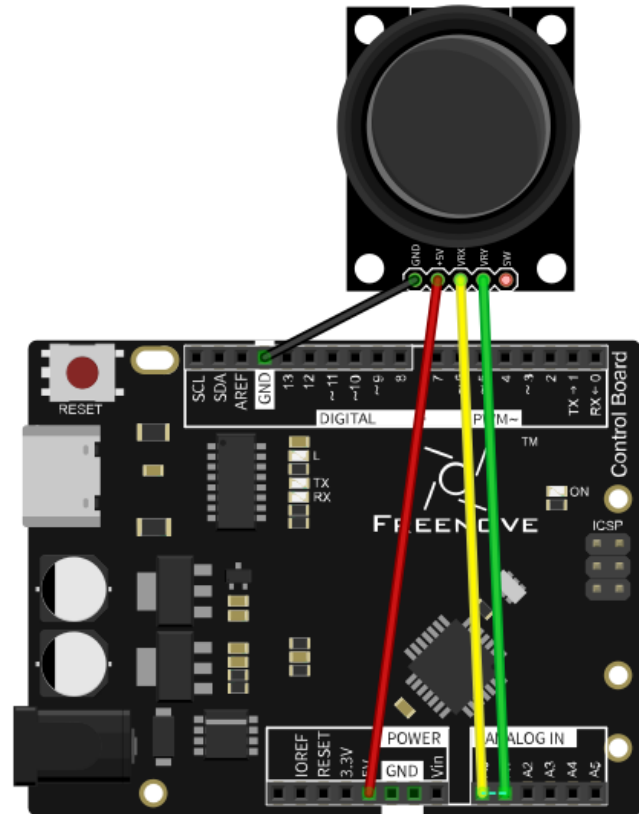
Circuit

Use A0 and A1 ports on connect board to detect the voltage value of two rotary potentiometers inside joystick.

Schematic diagram



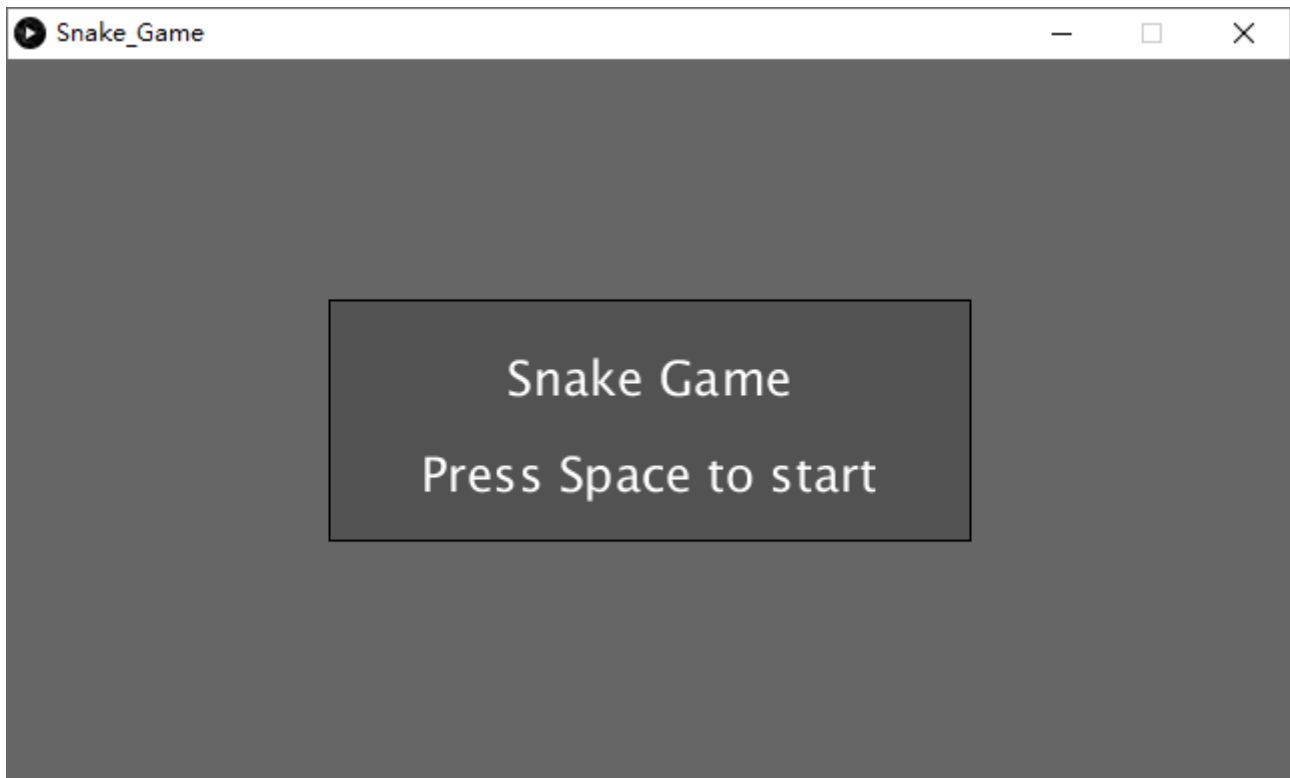
Hardware connection



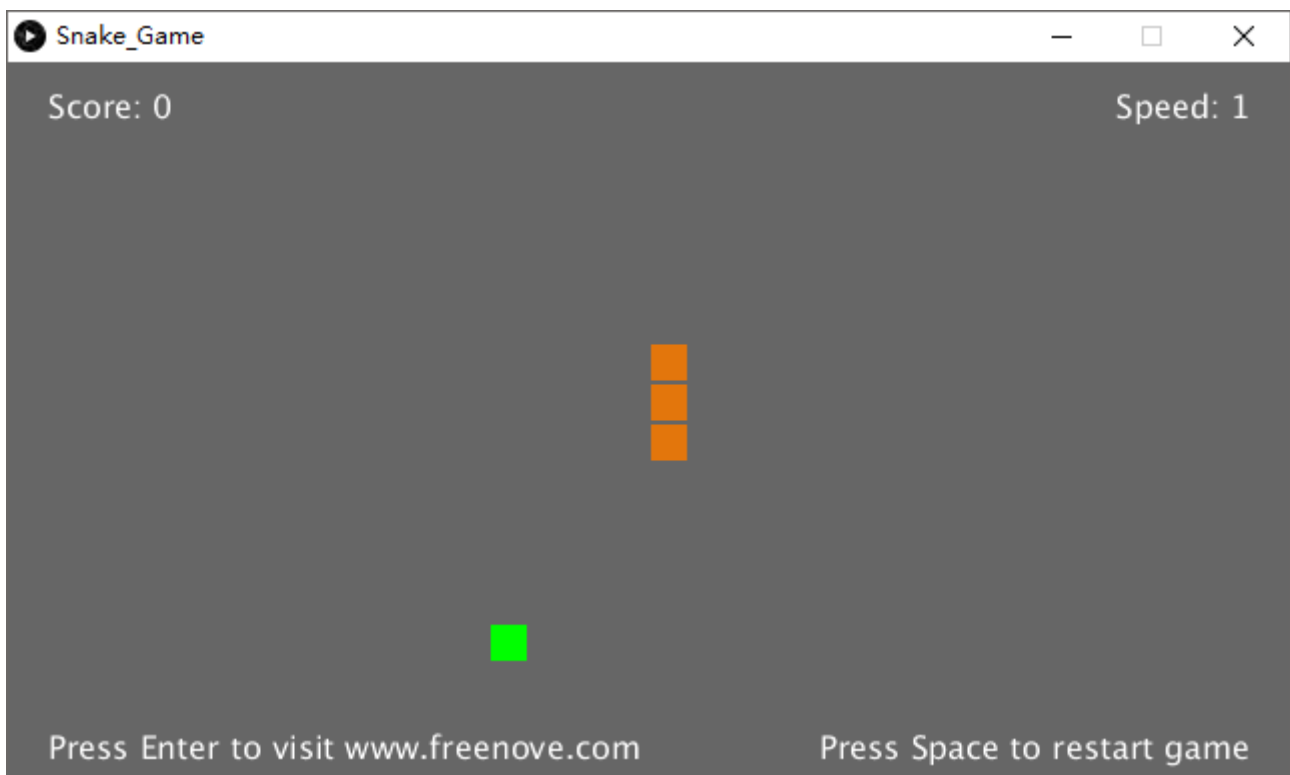
Sketch

Sketch Snake_Game

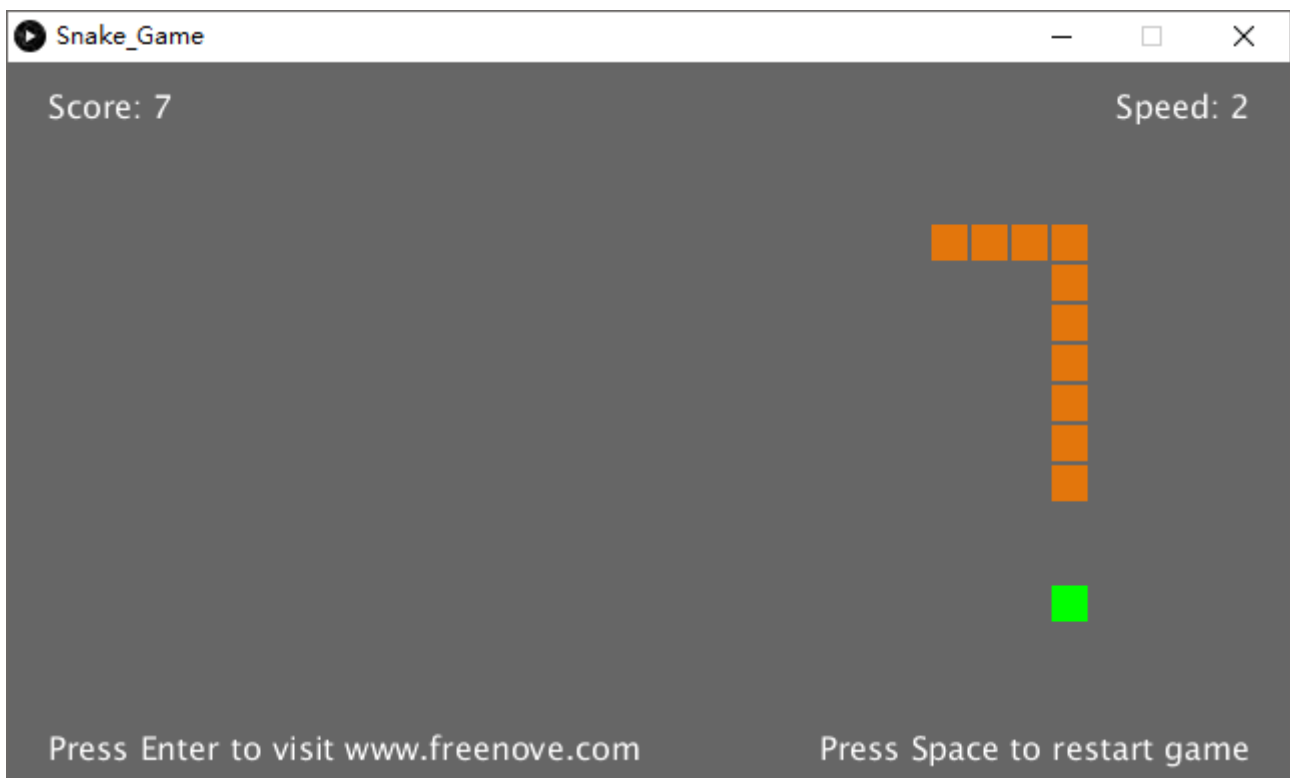
Use Processing to open Snake_Game.pde and click Run. If the connection succeeds, the following will be shown:



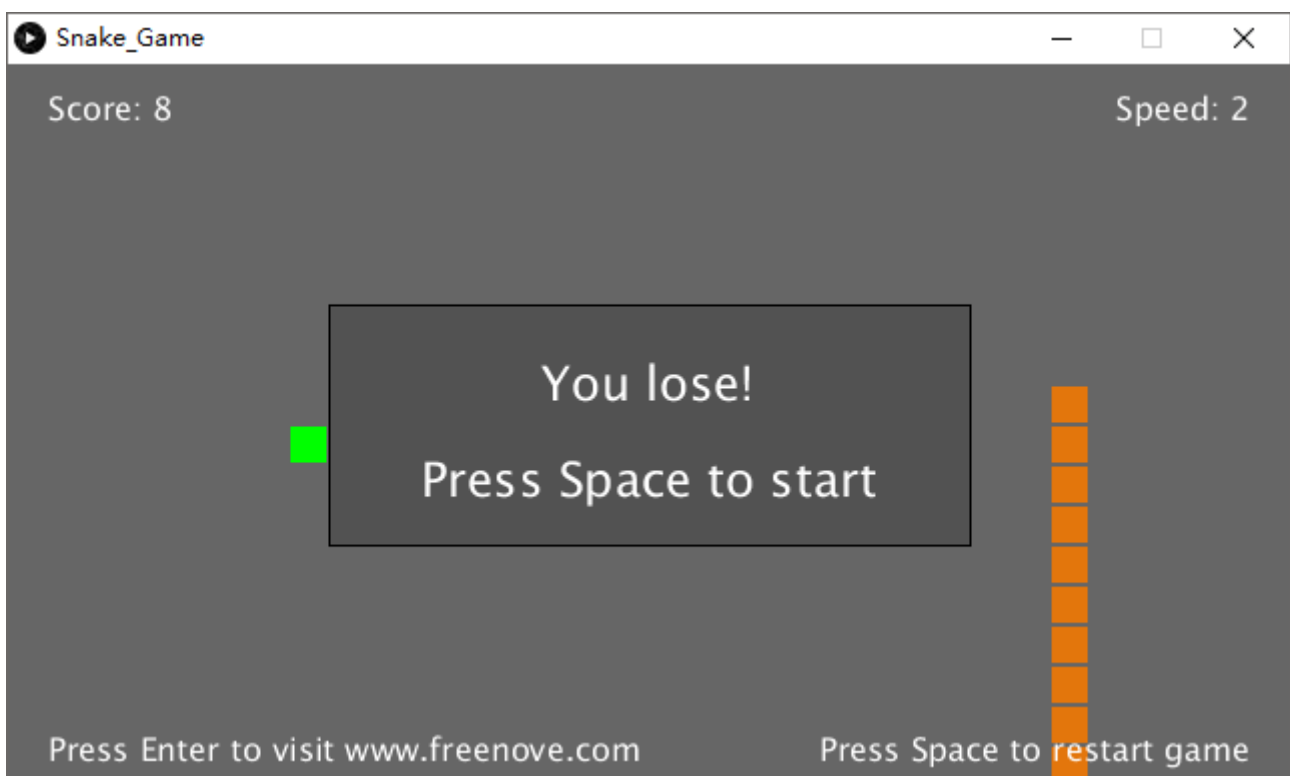
Press the space bar on keyboard to start the game:



Shift the joystick to control the snake's action. The game rules are the same as the classic snake game:



When you lose the game, press space bar to restart the game:



Additionally, you can restart the game by pressing the space bar at any time.

Project 4.2 Snake Game 3D

Now, let's experience the 3D version game.

Component List

The same as previous section.

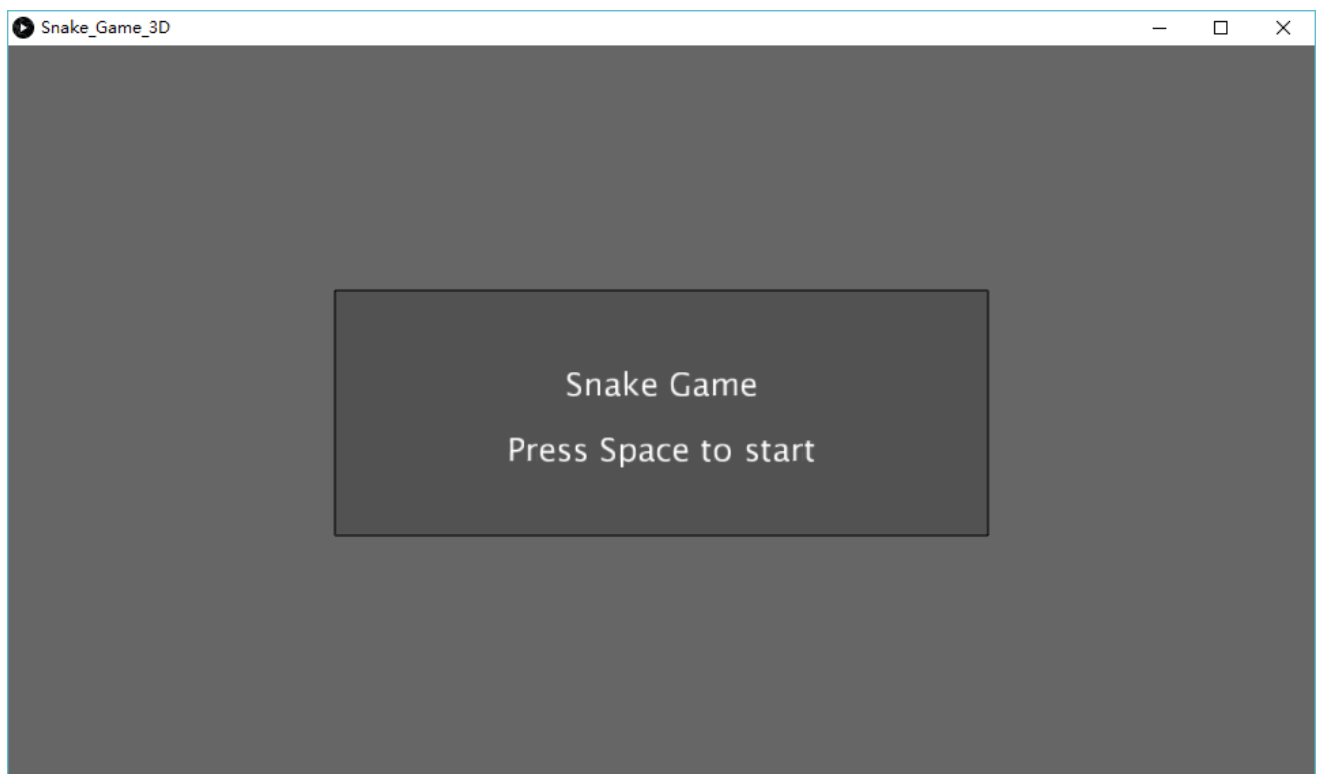
Circuit

The same as previous section.

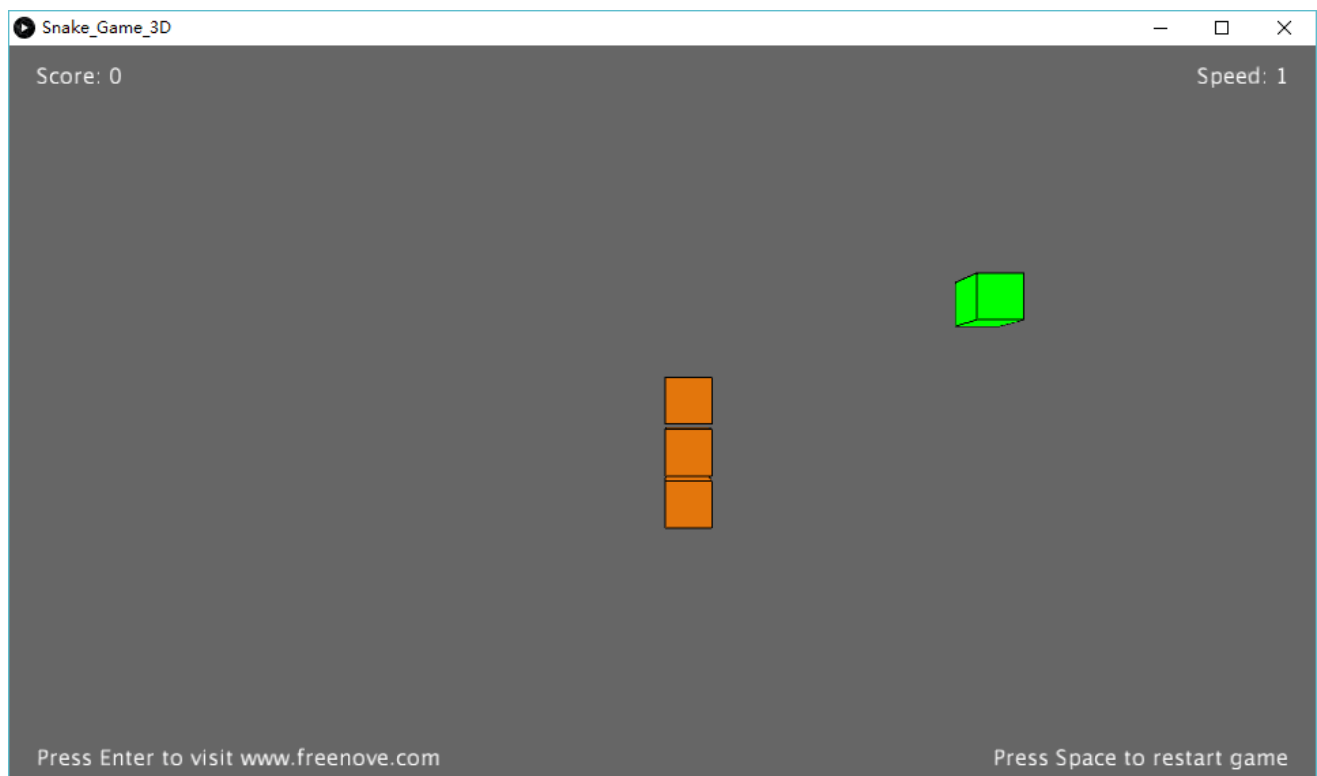
Sketch

Sketch Snake_Game_3D

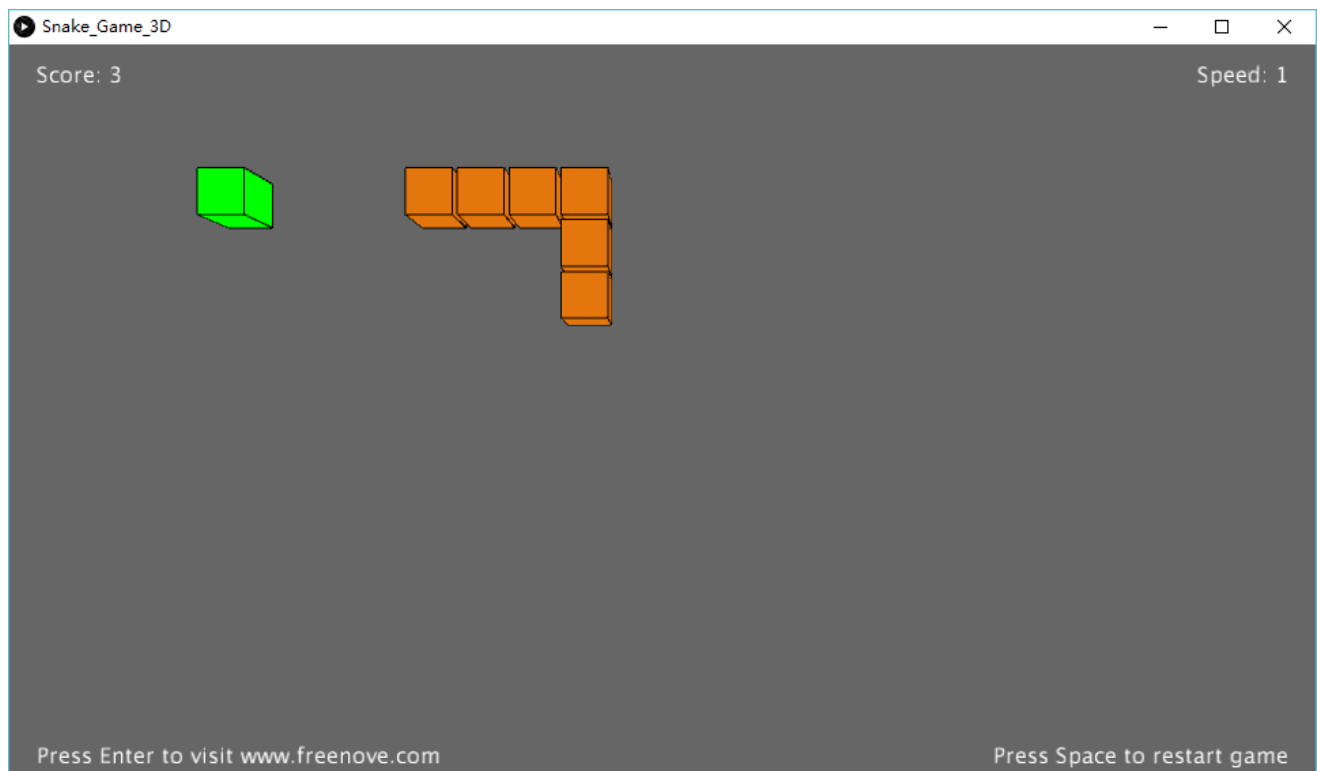
Use Processing to open Snake_Game_3D.pde and click Run. If the connection succeeds, the following will be shown:



Press the space bar on keyboard to start the game:



Shift the joystick to control the snake's action. The game rules are the same as the classic snake game:



The rest of operation is the same as the 2D version.

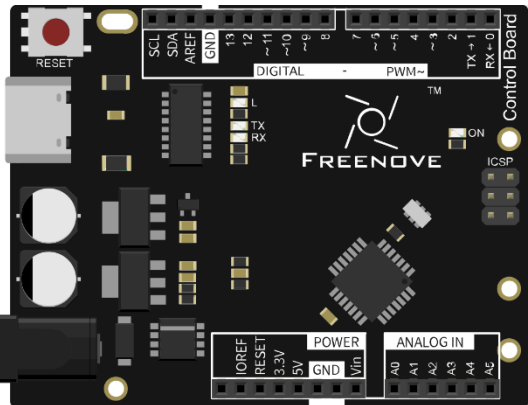
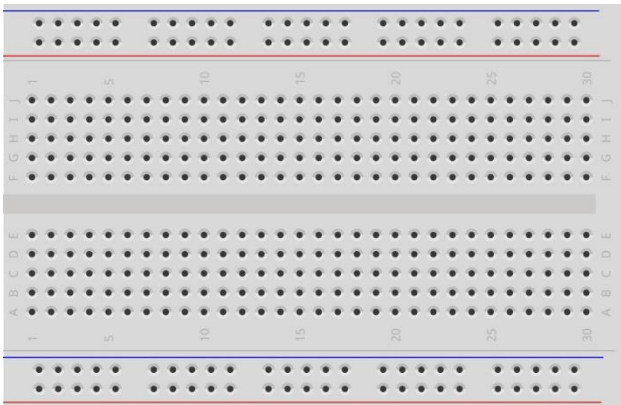

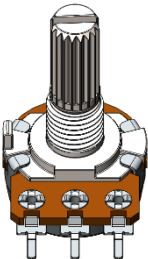

Chapter 5 Pong Game

We have experienced a single-player game Snake before. Now, let's use connect board to play a classic two-player pong game. You will experience both 2D and 3D version.

Project 5.1 Pong Game

First, let's experience the 2D version game.

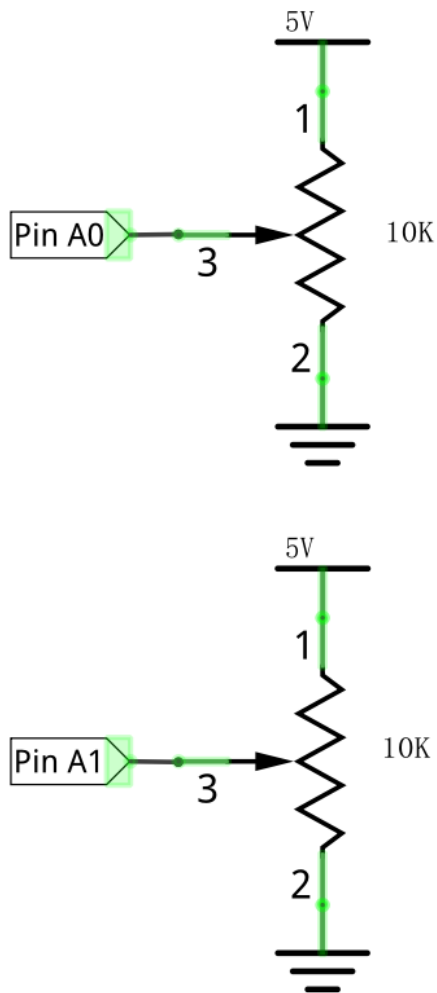
Component List

<p>Connect board x1</p>  <p>The image shows a black Freenove Connect Board. It features a central microcontroller, various pins labeled (SCL, SDA, AREF, GND, DIGITAL, TX, RX, TX+, RX+, TX-, RX-), and a reset button. The board is labeled 'FREENOVE' and 'Control Board'.</p>	<p>Breadboard x1</p>  <p>The image shows a standard white breadboard with a grid of holes. It has labels for columns (A, B, C, D, E, F, G, H, I, J) and rows (1, 5, 10, 15, 20, 25, 30).</p>
<p>USB cable x1</p>  <p>The image shows a black USB cable with a standard USB-A connector on one end and a micro-USB connector on the other.</p>	<p>Rotary potentiometer x2</p>  <p>The image shows a rotary potentiometer with a metal shaft and a red body. It has three pins at the bottom.</p>
<p>Jumper M/M x6</p>  <p>The image shows a single green jumper wire with a black plastic cap on one end.</p>	

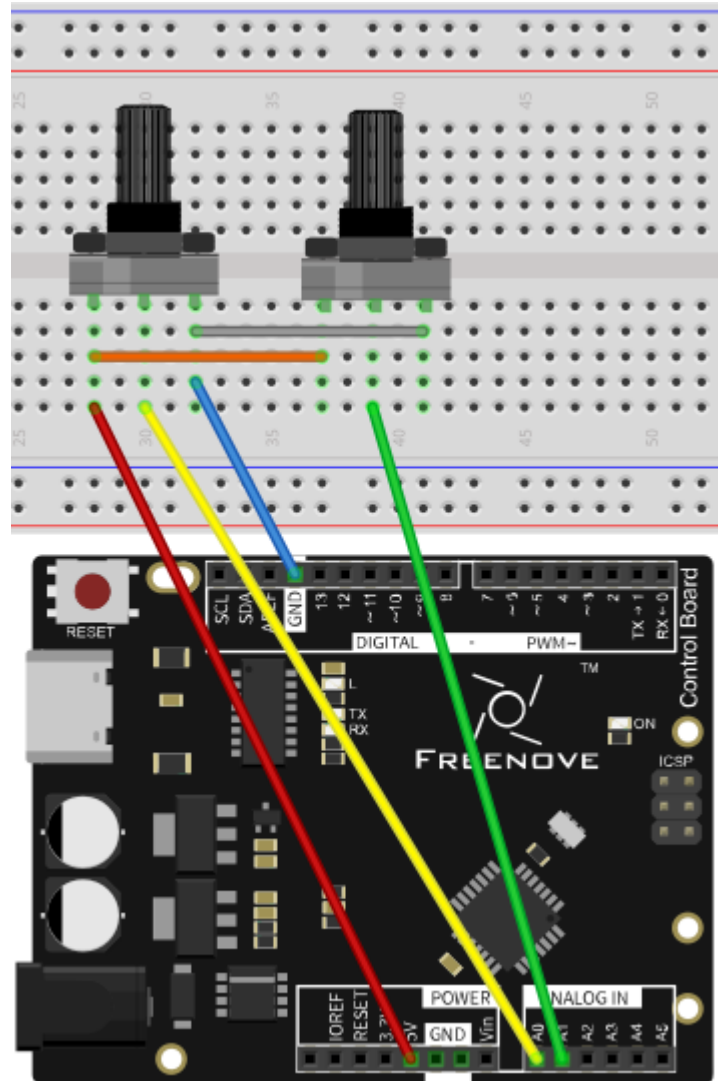
Circuit

Use A0, A1 ports on connect board to detect the voltage of rotary potentiometers.

Schematic diagram



Hardware connection



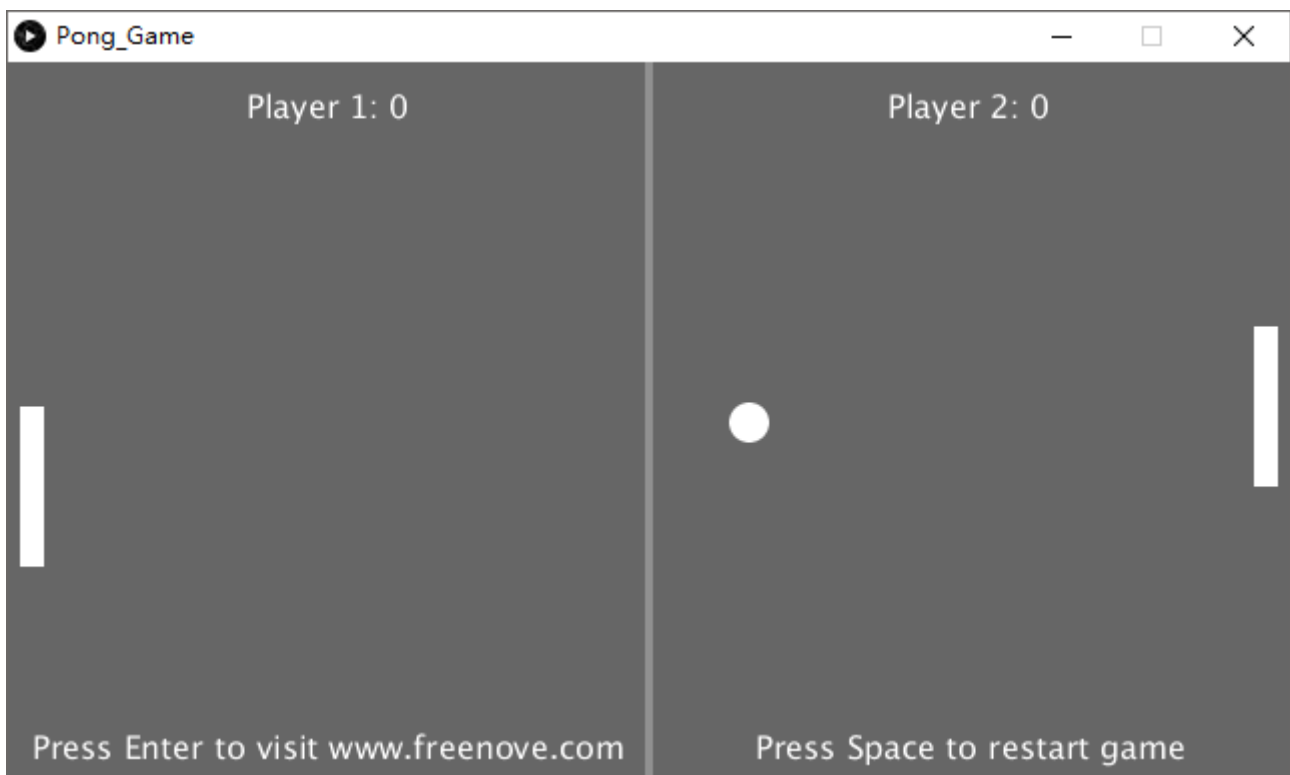
Sketch

Sketch Pong_Game

Use Processing to open Pong_Game and click Run. If the connection succeeds, the following will be shown:



Now you can try to rotate the potentiometer to control the movement of paddle without ball. Press space bar to start the game:



Use potentiometer to control the movement of paddle to hit the ball back. The game rules are the same as classic pong game:



The game will be over when one side reaches three points. Pressing the space bar can restart the game:



Additionally, you can restart the game by pressing the space bar at any time.

Project 5.2 Pong Game 3D

Now, let's experience the 3D version game.

Component List

The same as previous section.

Circuit

The same as previous section.

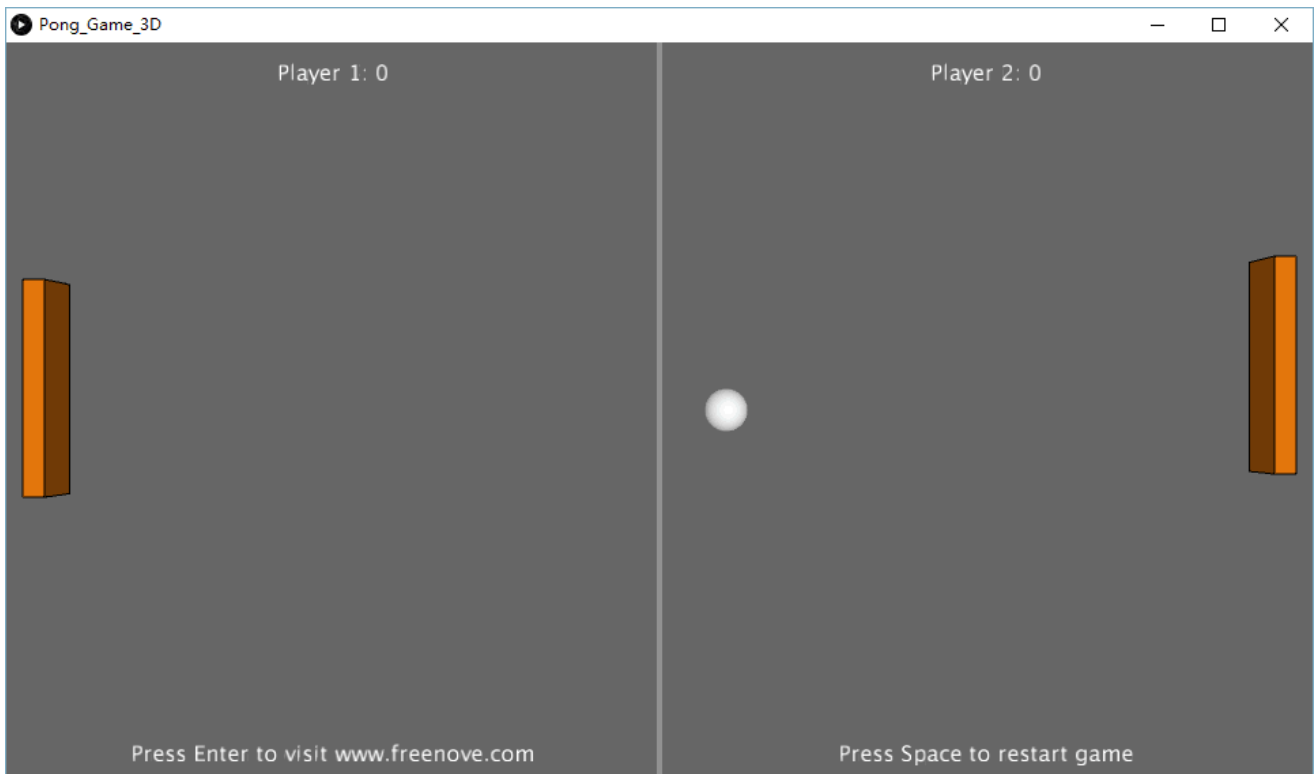
Sketch

Sketch Pong_Game_3D

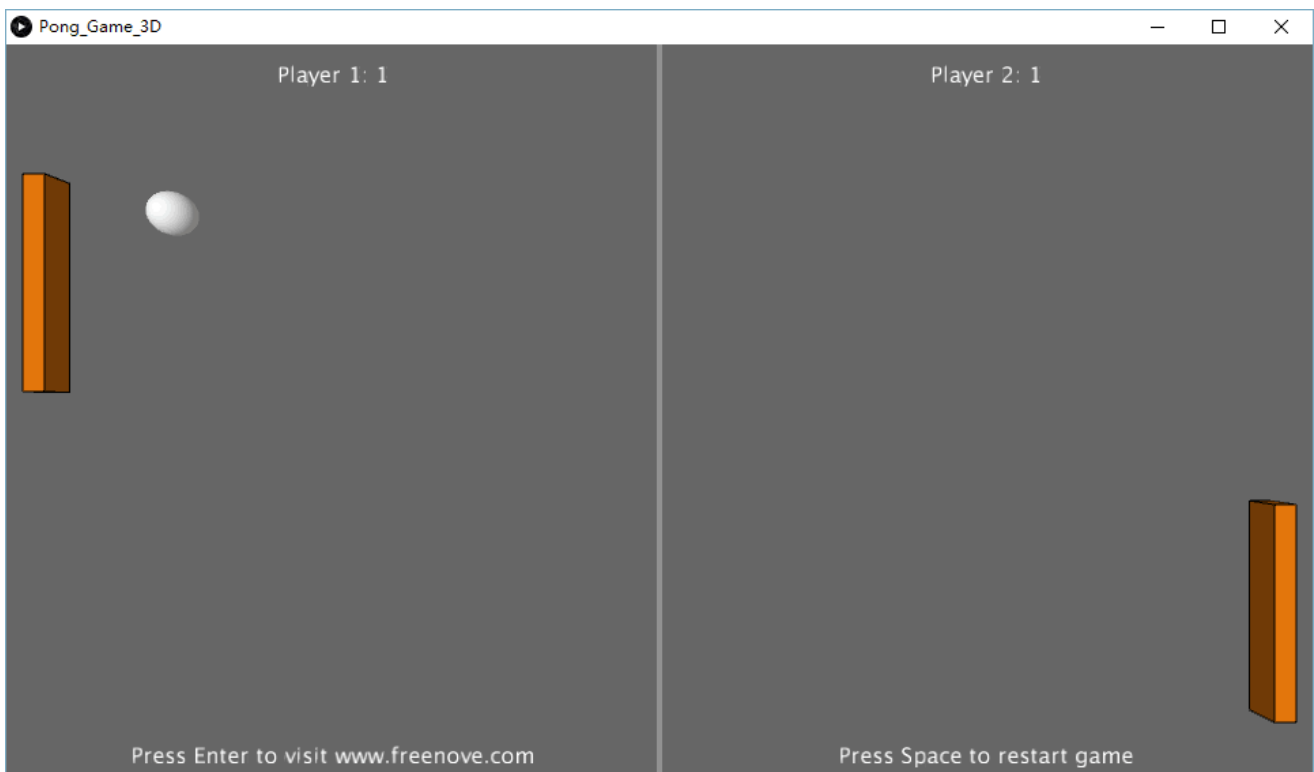
Use Processing to open Pong_Game_3D.pde and click Run. If the connection succeeds, the following will be shown:



Now you can try to rotate the potentiometer to control the movement of paddle without ball. Press space bar to start the game:



Use potentiometer to control the movement of paddle to hit the ball back. The game rules are the same as classic pong game:



The rest of operation is the same as the 2D version.

What's Next?

THANK YOU for participating in this learning experience!

We have reached the end of this tutorial. If you find errors, omissions or you have suggestions and/or questions about this tutorial or component contents of this kit, please feel free to contact us:

support@freenove.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, micro:bit, robots, smart cars and other interesting products, please visit our website:

<http://www.freenove.com/>

We will continue to launch fun, cost-effective, innovative and exciting products.

Thank you again for choosing Freenove products.