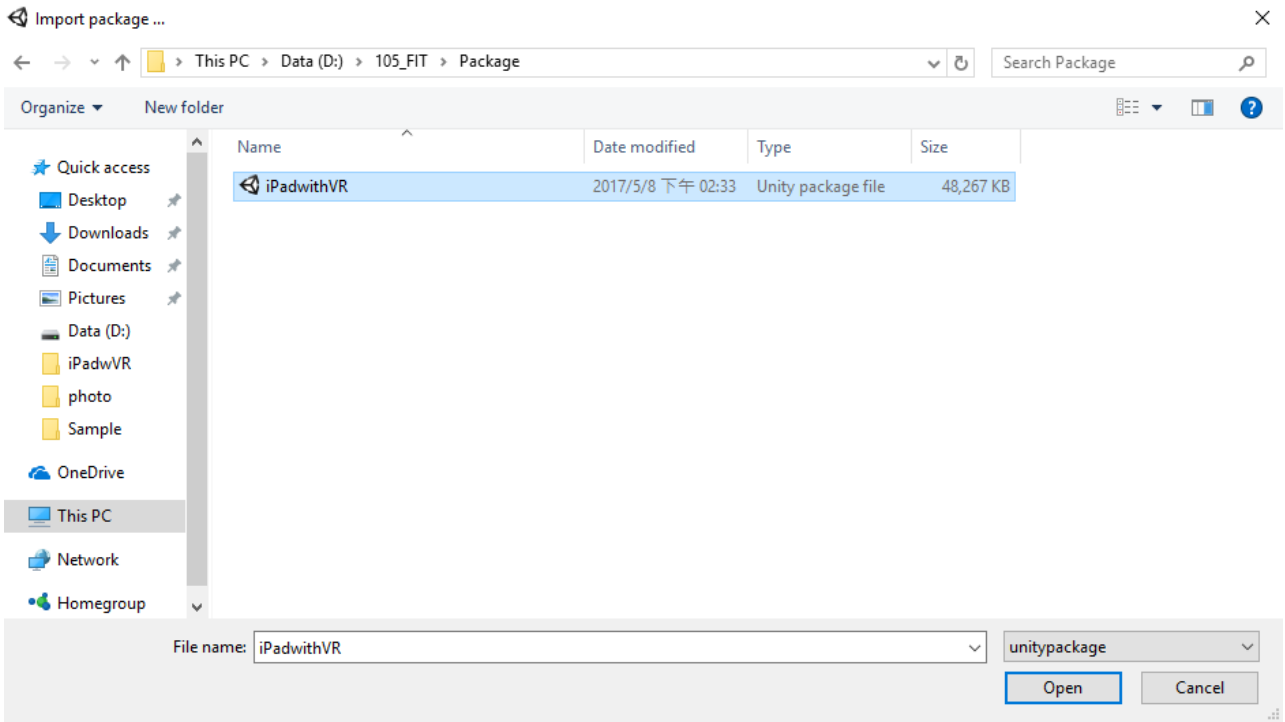
Sample Project tutorial

iPad in VR

1. Attach the controller to the iPad with the magnet mechanism.
2. Open the TUIOpad App and setup the PC's IP and port for OSC Connection.

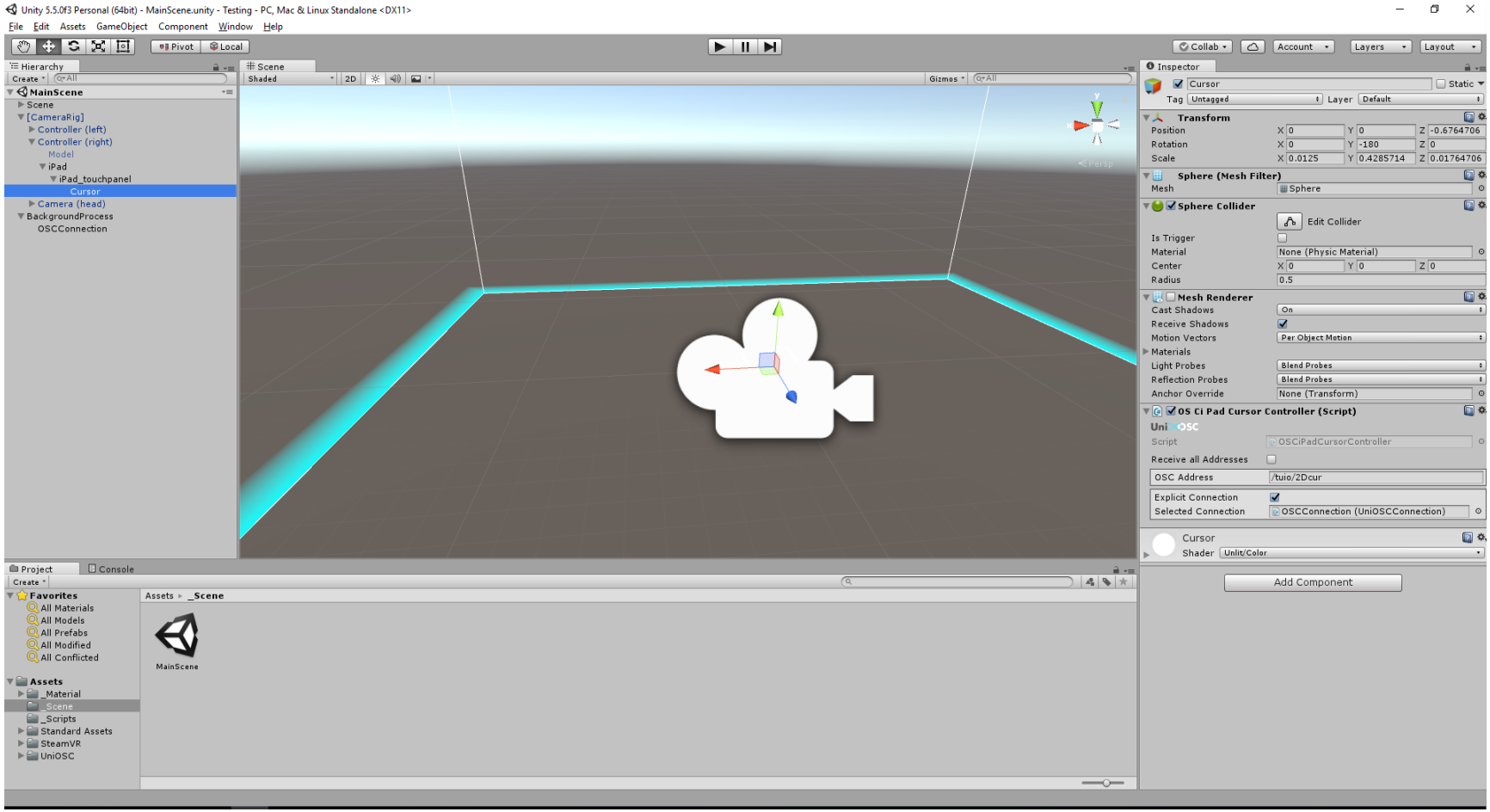
iPad in VR

Import the “iPadwithVR” package to Unity



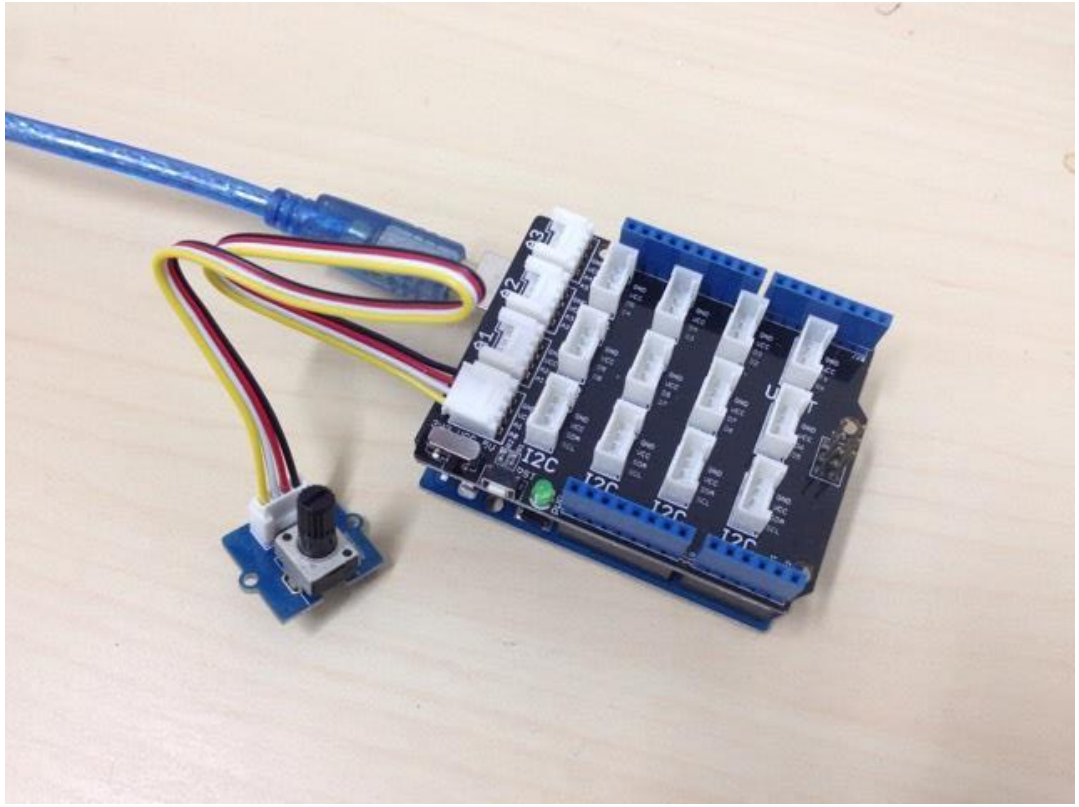
iPad in VR

Open the MainScene in the "_Scene" folder.
The iPad is tracked by the Controller, and the touch point is updated by OSC Connection data.



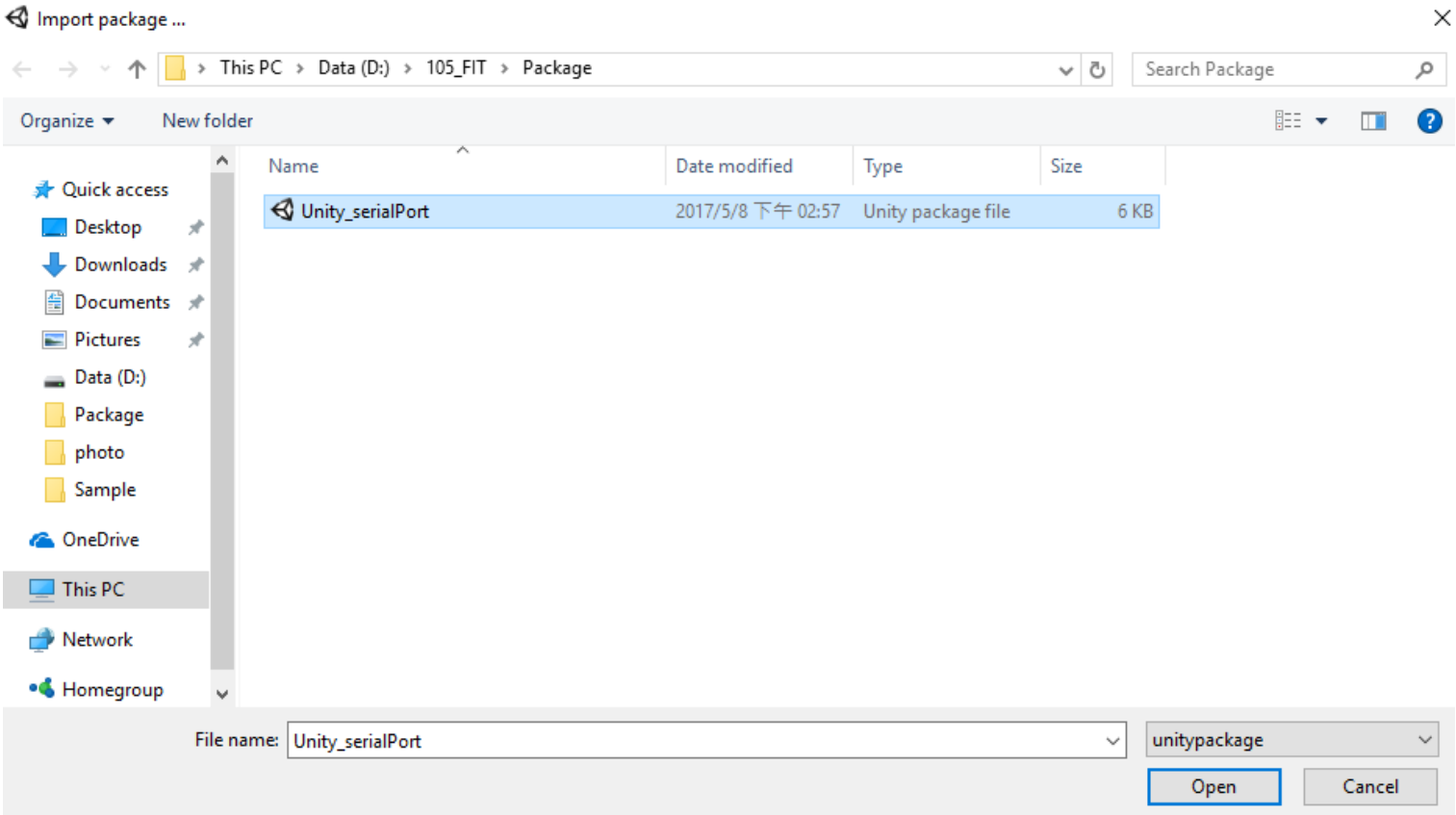
Unity and Arduino (Serial Port)

Connect the knob to Arduino A0 .
Connect the Arduino via Serial Port to Computer.
Upload the “serialport_arduino.ino” code to Arduino.



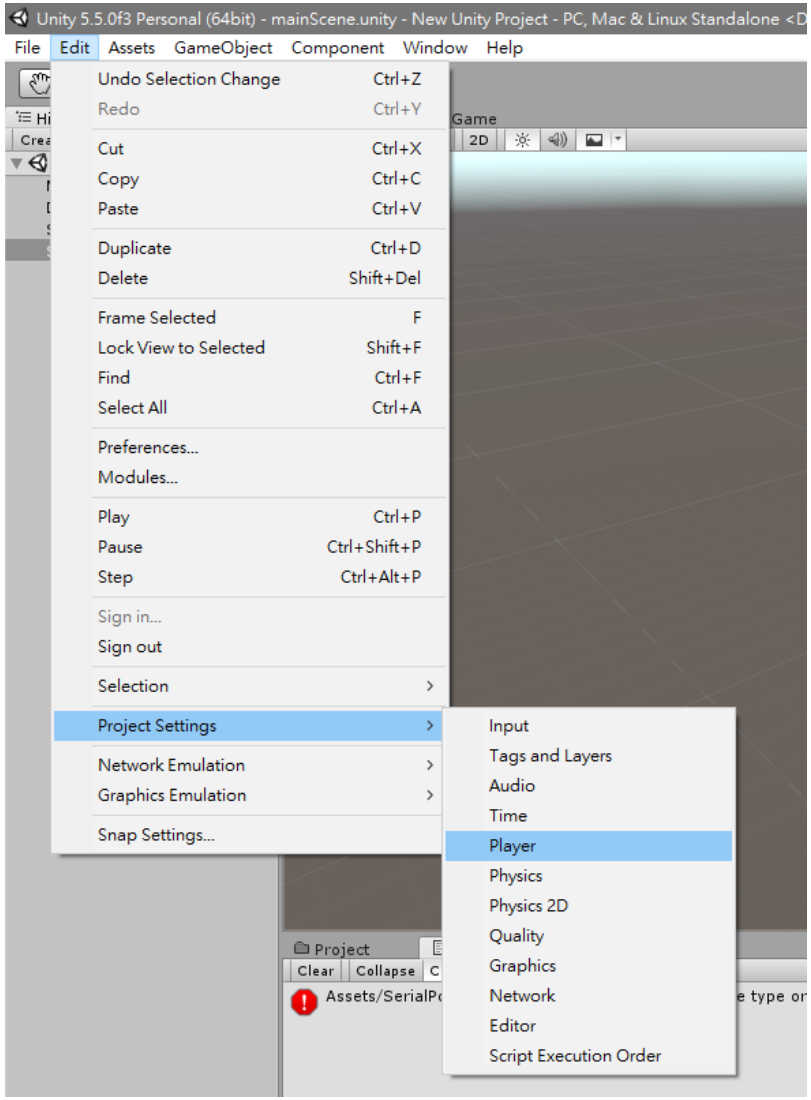
Unity and Arduino (Serial Port)

Import the “Unity_serialPort” package to Unity



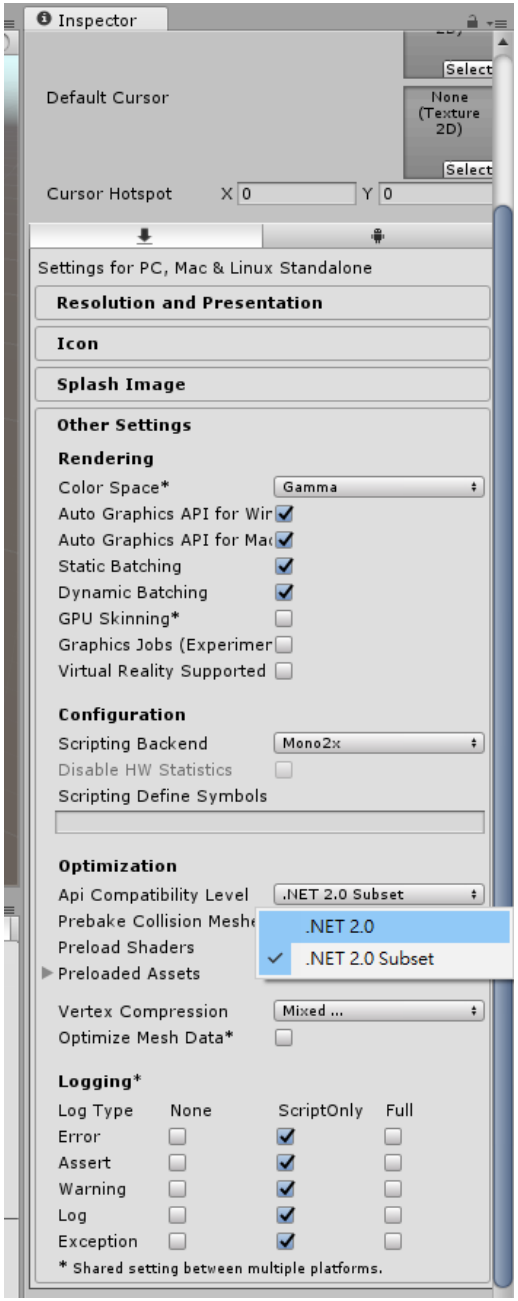
Unity and Arduino (Serial Port)

Edit -> Project Settings -> Player



Unity and Arduino (Serial Port)

Select “Api Compatbility level” to “.NET 2.0”



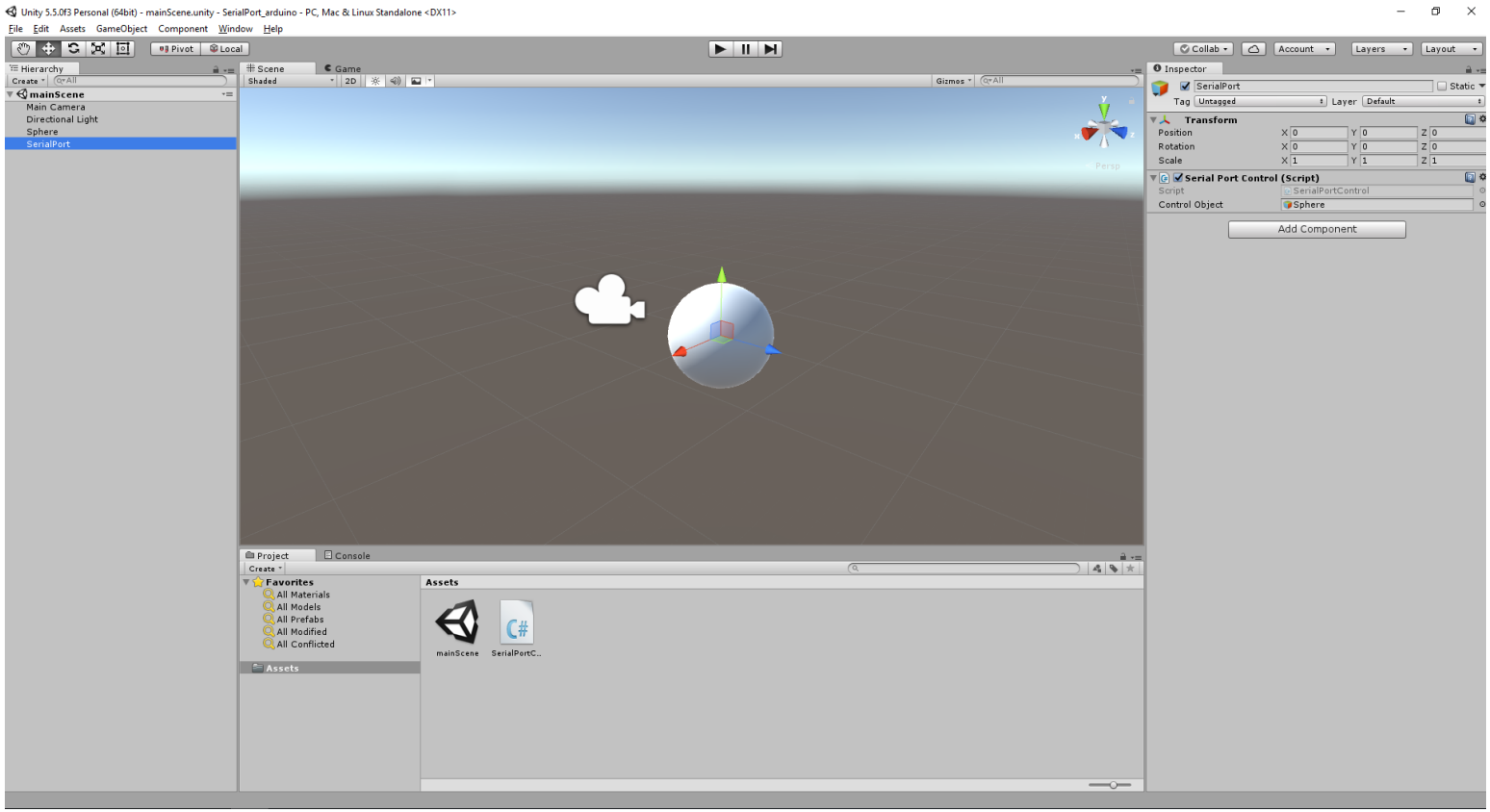
Unity and Arduino (Serial Port)

Remember to change the Serial Port name in the “SerialPortControl.cs” script.

```
lPortControl.cs
erialPortControl                                    OnApplicationQuit()
{
    public bool isRunning = true;
    private SerialPortControl upperClass;

    public SerialPortThread(SerialPortControl upperClass)
    {
        this.upperClass = upperClass;
    }

    public void run()
    {
        SerialPort sp = new SerialPort("COM4", 9600);
        sp.ReadTimeout = 50;
        sp.Open();

        while (isRunning)
        {
```

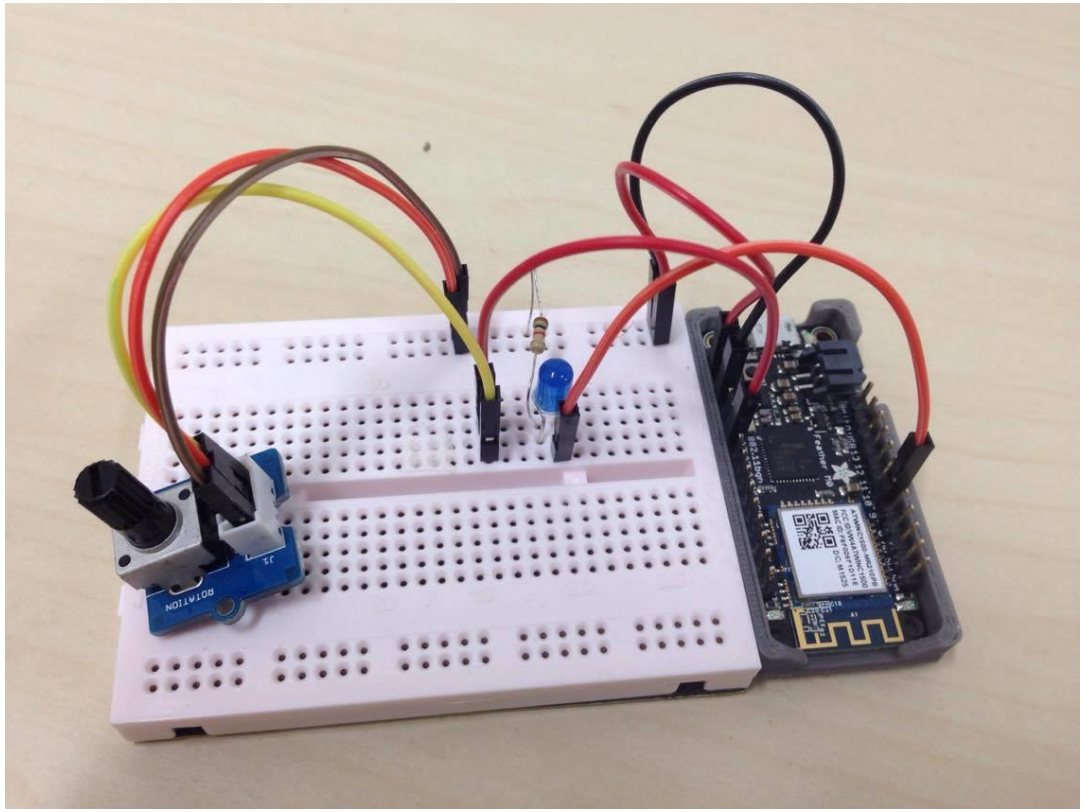
Unity and Arduino (Serial Port)

The Sphere can be control by the knob on the Arduino(A0)



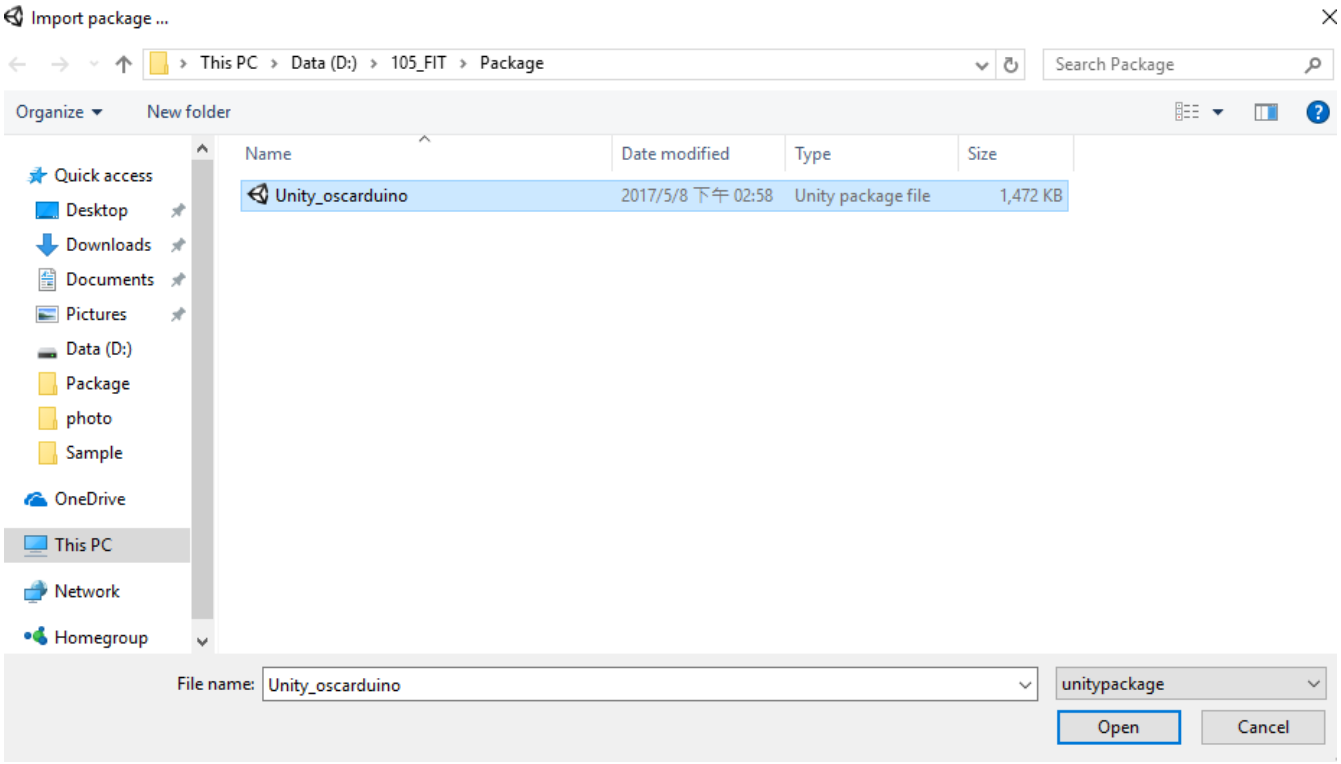
Unity and wireless Arduino

Connect the knob to Arduino A2, and LED to pin10 .
Connect the Arduino via Serial Port to Computer.
Upload the “arduino_osc.ino” code to Arduino.



Unity and wireless Arduino

Import the “Unity_oscarduino” package to Unity



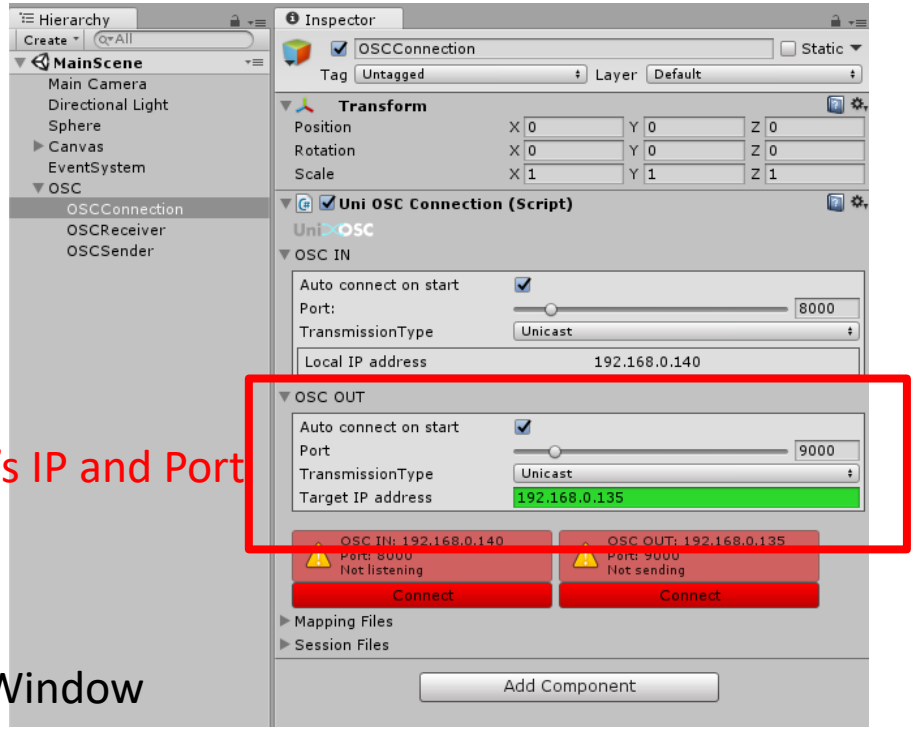
Unity and wireless Arduino

Remember to change the IP Address and Port both in Arduino and Unity.

```
//IP setup
IPAddress sendToUnityPC_Ip(192, 168, 0, 174); // UnityPC's IP
unsigned int sendToUnityPC_Port = 8000; // UnityPC's listening port
unsigned int listenPort = 9000; // local port to listen on

char packetBuffer[255]; //buffer to hold incoming packet
char ReplyBuffer[] = "acknowledged"; // a string to send back
WiFiUDP Udp_send;
WiFiUDP Udp_listen;
```

Unity IP and Port

Local Port to listen on

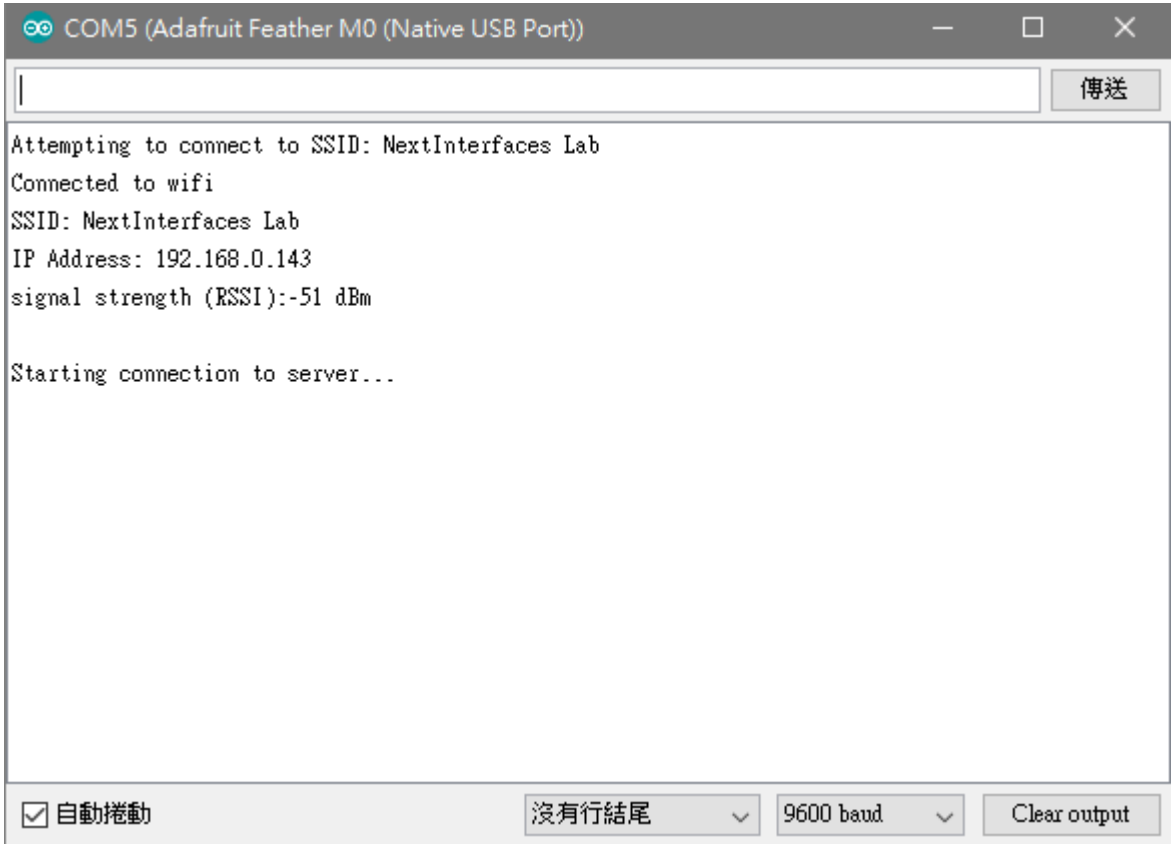


Arduino's IP and Port

You can check up Arduino's IP via the Serial Port Window

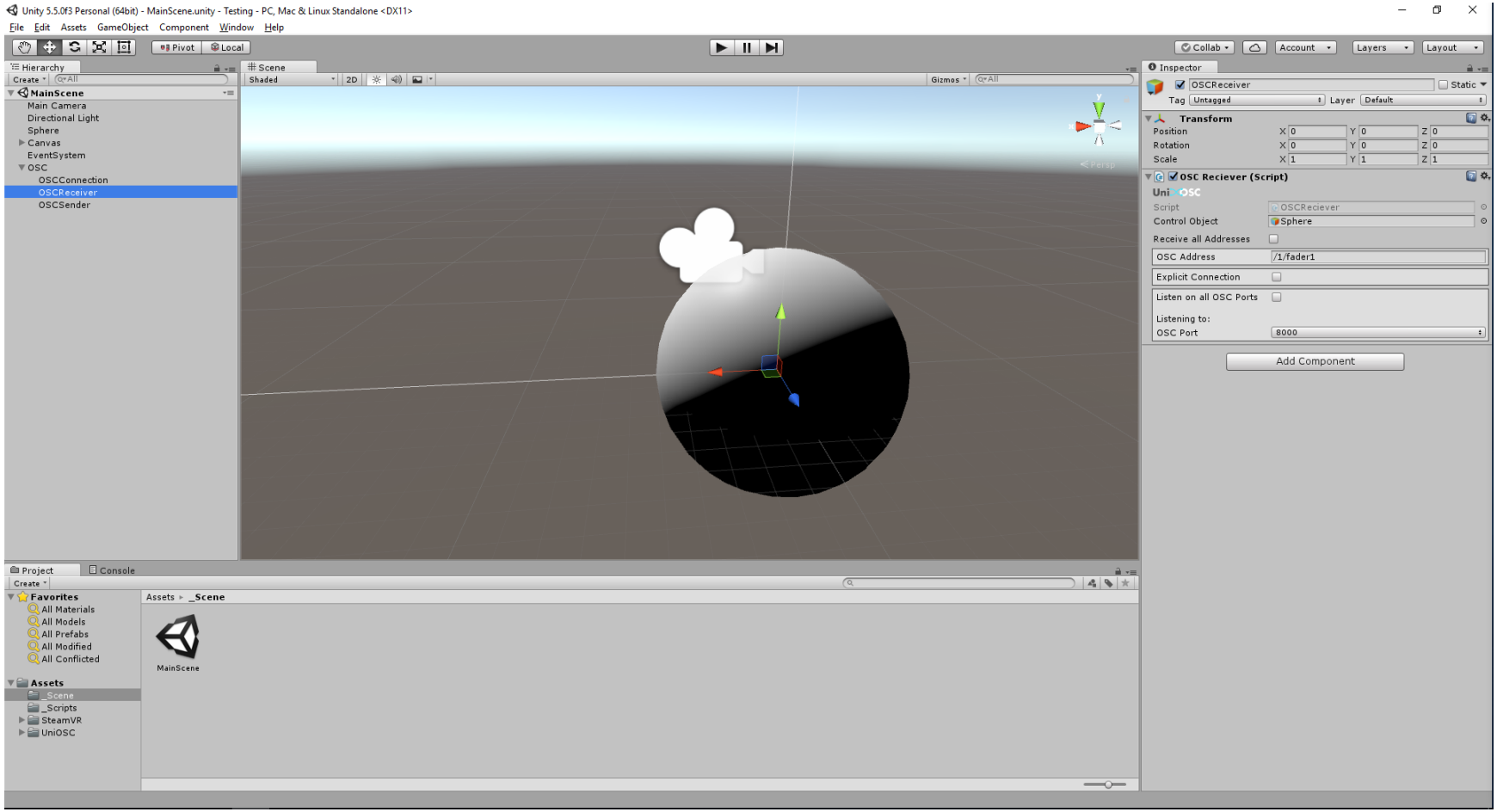
Unity and wireless Arduino

You can check up Arduino’s IP via the Serial Port Window



Unity and wireless Arduino

The Sphere can be control by the knob on the arduino(A2)
And the Slider on Unity can control the arduino led’s lightness.



Unity and wireless Arduino

The Arduino can run wireless after you modify the code and connect it to the battery.

```
void setup() {
    //Configure pins for Adafruit ATWINC1500 Feather
    WiFi.setPins(8,7,4,2);
    //Initialize serial and wait for port to open:
    Serial.begin(9600);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }
    // check for the presence of the shield:
    if (WiFi.status() == WL_NO_SHIELD) {
        Serial.println("WiFi shield not present");
        // don't continue:
        while (true);
    }
    // attempt to connect to Wifi network:
    while ( status != WL_CONNECTED) {
        Serial.print("Attempting to connect to SSID: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
        status = WiFi.begin(ssid, pass);
        // wait 10 seconds for connection:
        delay(10000);
     }
    Serial.println("Connected to wifi");
```

Delete it!