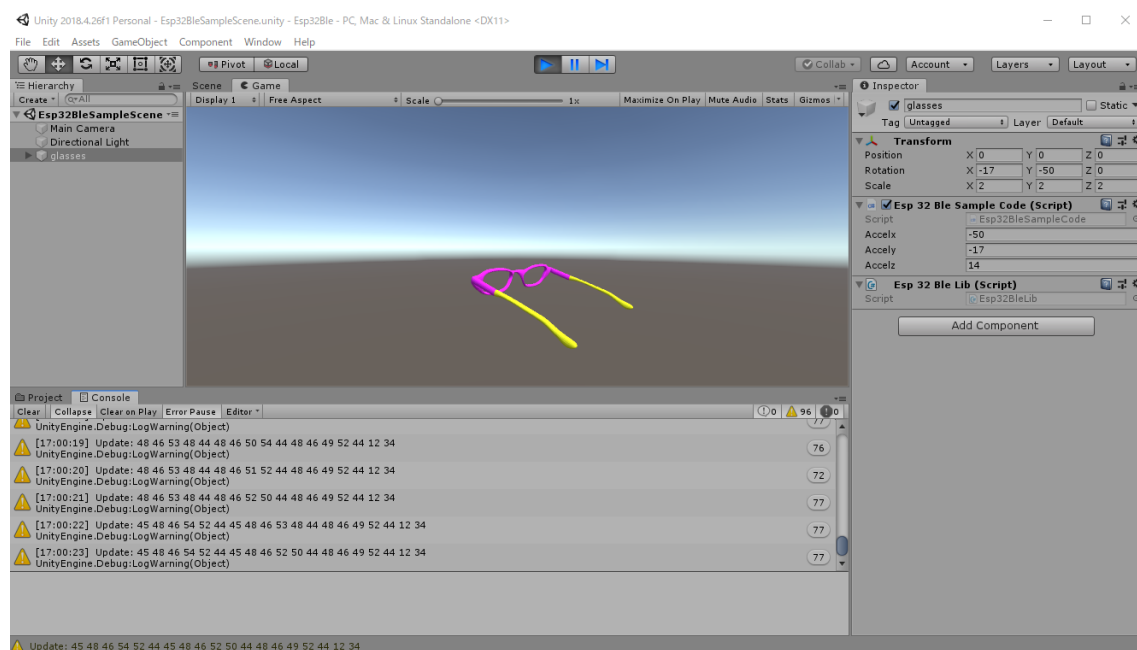


Esp32Ble Quick Start

In the Demo scene, a Unity 3D object 'glasses' tilts using data from the ESP32. The ESP32 generates 3D Object position data 'x, y, z' using the sin function. The ESP32 sends that data from the ESP32 to Unity through Windows PC using the BLE (Bluetooth low energy) interface. The following shows the capture screen when the Demo scene runs.



Demo scene

This Quick Start will quickly get you started with Esp32Ble. You can set up both the Unity script and the ESP32 code for the Demo scene using this Quick Start.

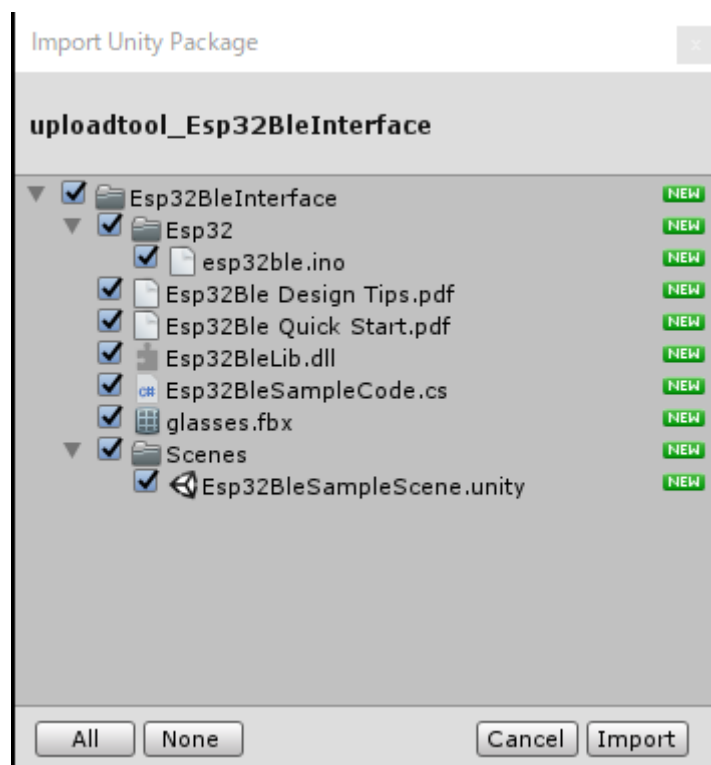
Step-1 Download and set up a package in your unity project

Step-2 Write ESP32 code using Arduino IDE

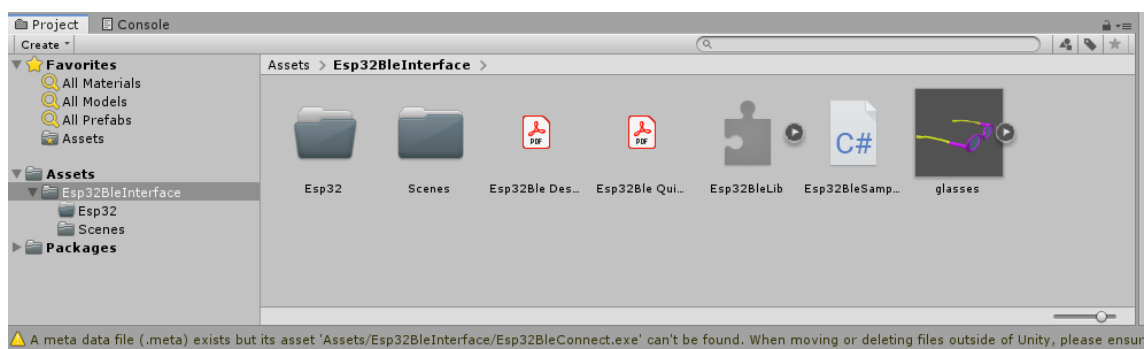
Step-3 Run Demo scene

Step-1 Download and set up a package in your unity project

(1) Download Esp32Ble from the Asset Store, and click on the 'Import' button.



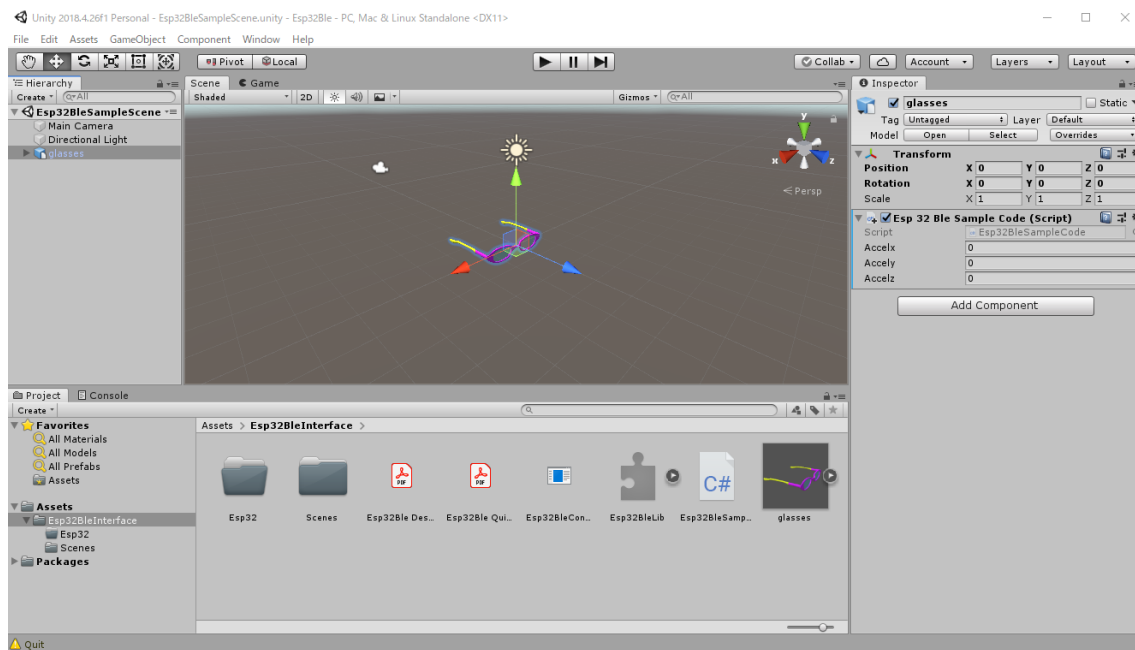
(2) The package is imported under the Assets folder in your Unity project.



(3) Download the external plugin 'Esp32BleConnect' below and set it into the 'Assets/ Esp32BleInterface' folder.

- [external plugin 'Esp32BleConnect.exe'](#)

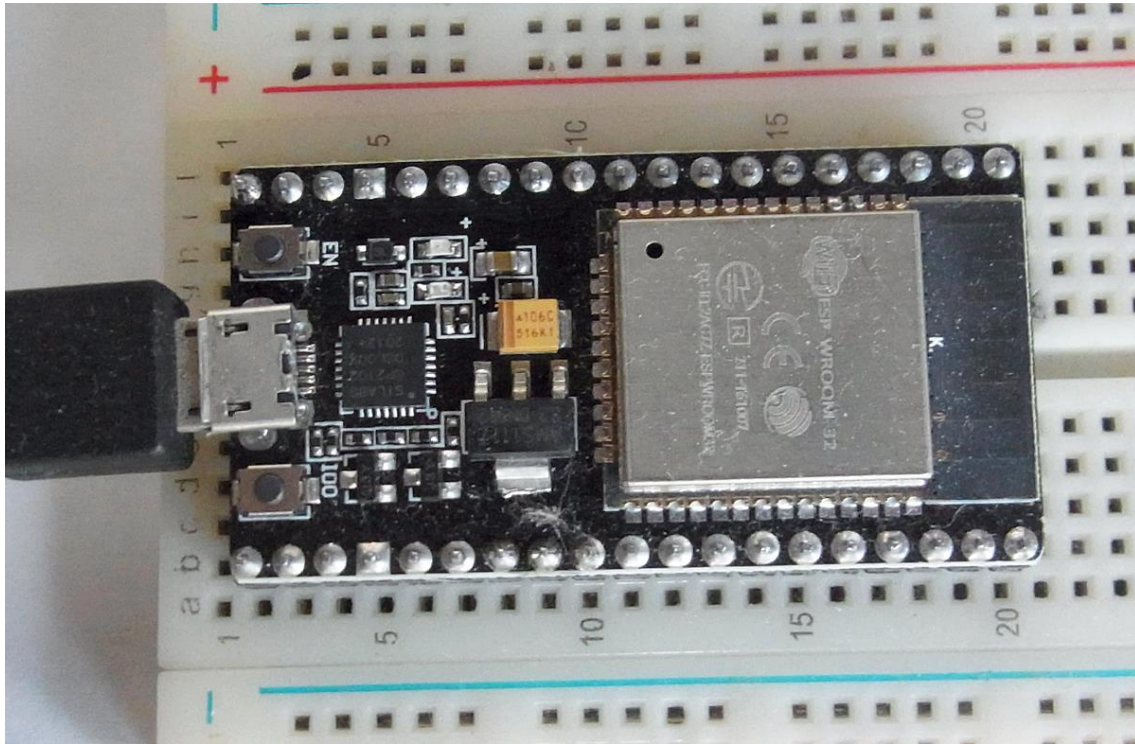
(4) Click the 'Project' tab, choose 'Assets' > 'Esp32BleInterface' > 'Scene' folder of the left side menu, double click on the demo scene 'Esp32BleSampleScene' in the 'Project' window.



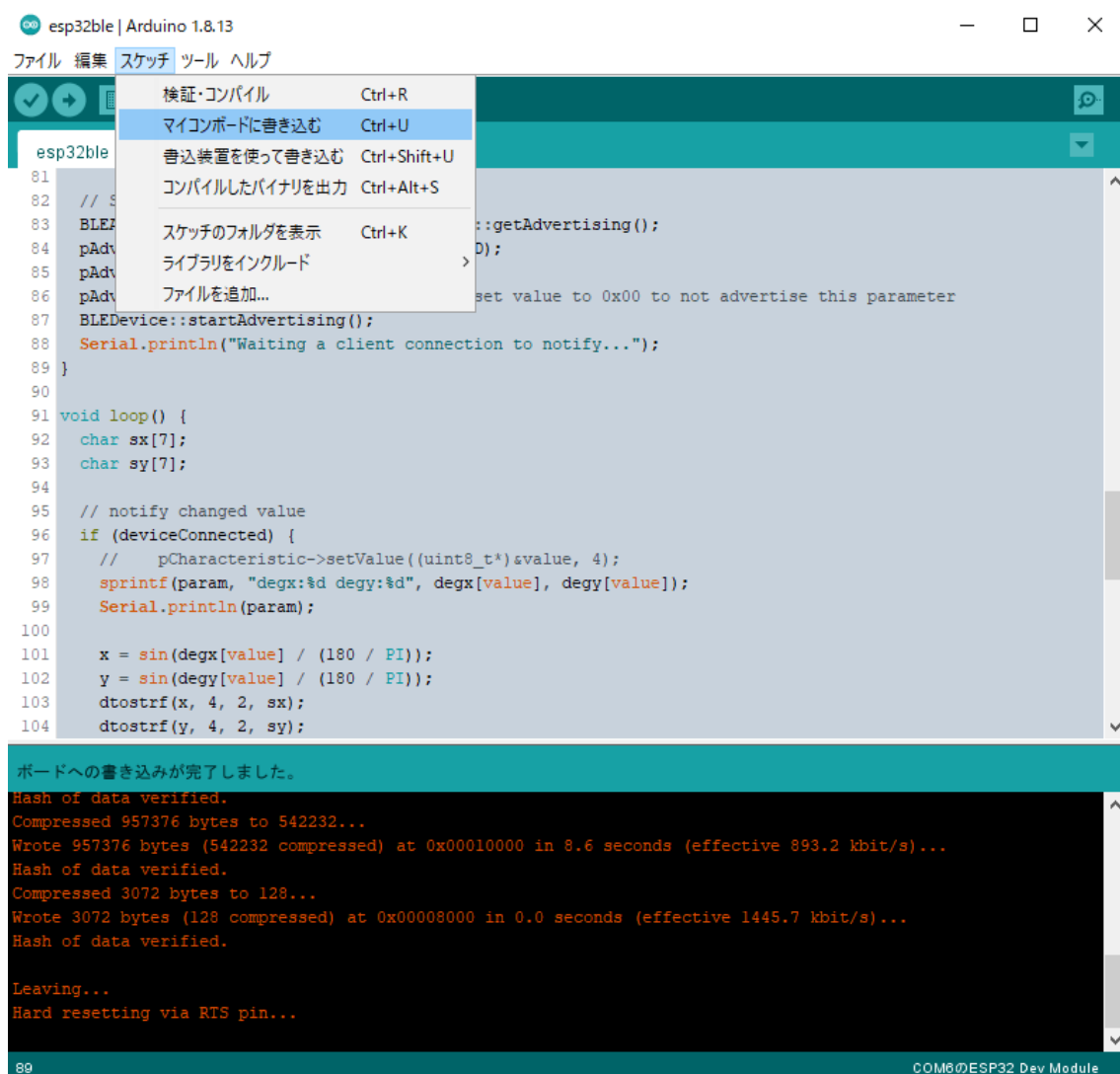
Step-2 Write ESP32 code using Arduino IDE

(1) Connect the ESP32 by Micro USB cable to Windows PC, then build a software development environment for Esp32Ble with the Arduino IDE according to [Arduino IDE for Esp32Ble](#).

ESP32

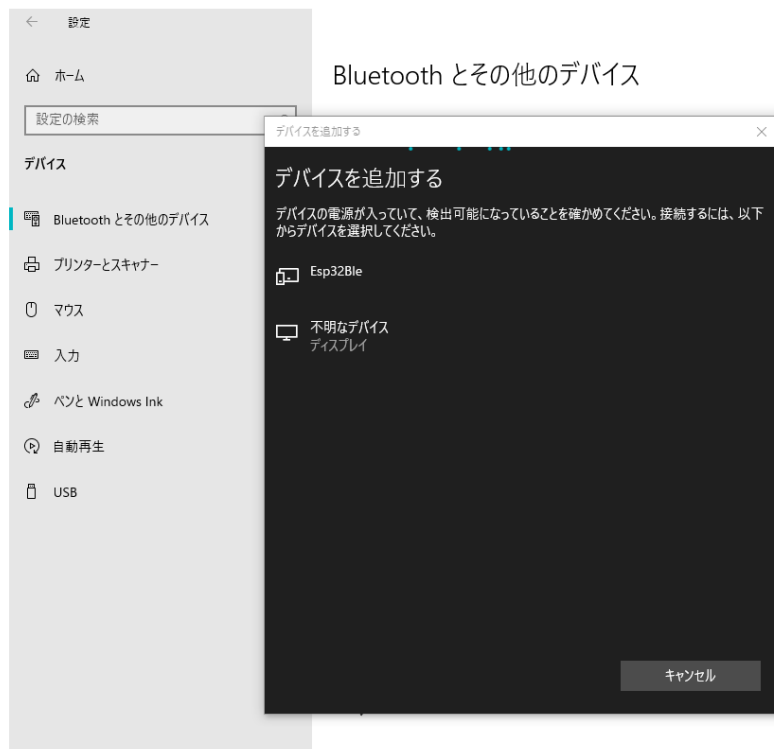


- (2) Compile and write ESP32 code 'esp32ble.ino' of the folder '/Assets /Esp32BleInterface/esp32ble' into the ESP32 using the Arduino IDE.

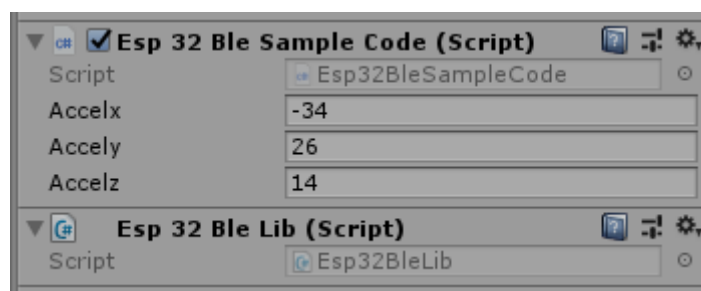


Step-3 Run Demo scene

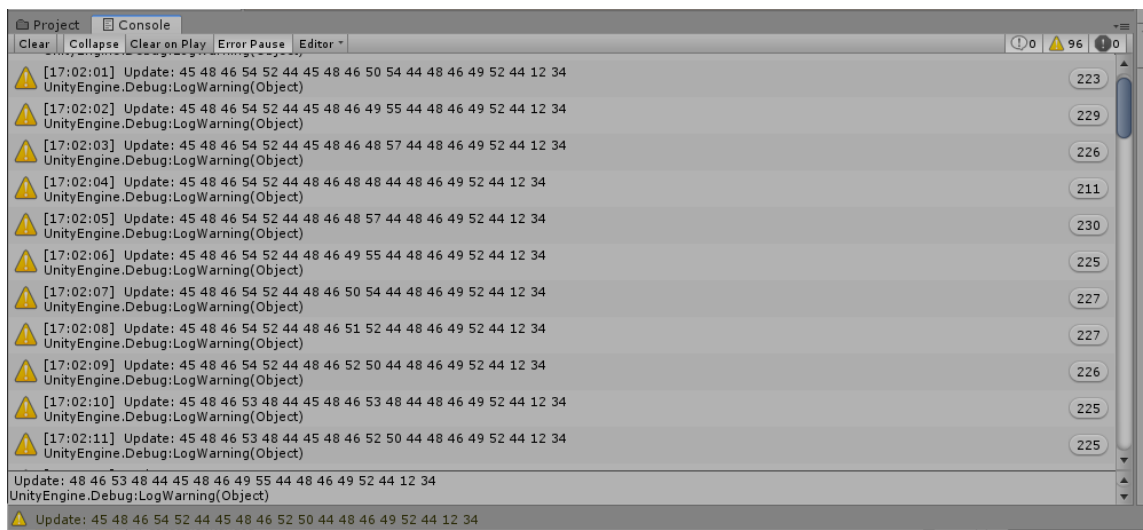
- (1) In Windows PC with Bluetooth low Energy Adapter, go to 'Start', choose 'Settings' > 'Devices' > 'Bluetooth and other devices settings'. Click 'Add Bluetooth or other device' to start searching for the device.
- (2) Like below, confirm the Bluetooth device 'Esp32Ble'.



- (3) Hit the 'Play' button on Unity. The Unity 3D object 'glasses' tilts using data generated by ESP32 using the sin function. The 'Accelx, y, z' in the 'Esp32Ble Sample Code' of the 'Inspector' tab displays the received data 'x, y, z' from the ESP32 as follows.



- (4) The 'console' tab of Unity displays the data from ESP32 as follows.



- (5) In ESP32, run the ESP32 code "esp32ble.ino". The data sent Unity is displayed in the serial monitor of Arduino IDE as follows.

