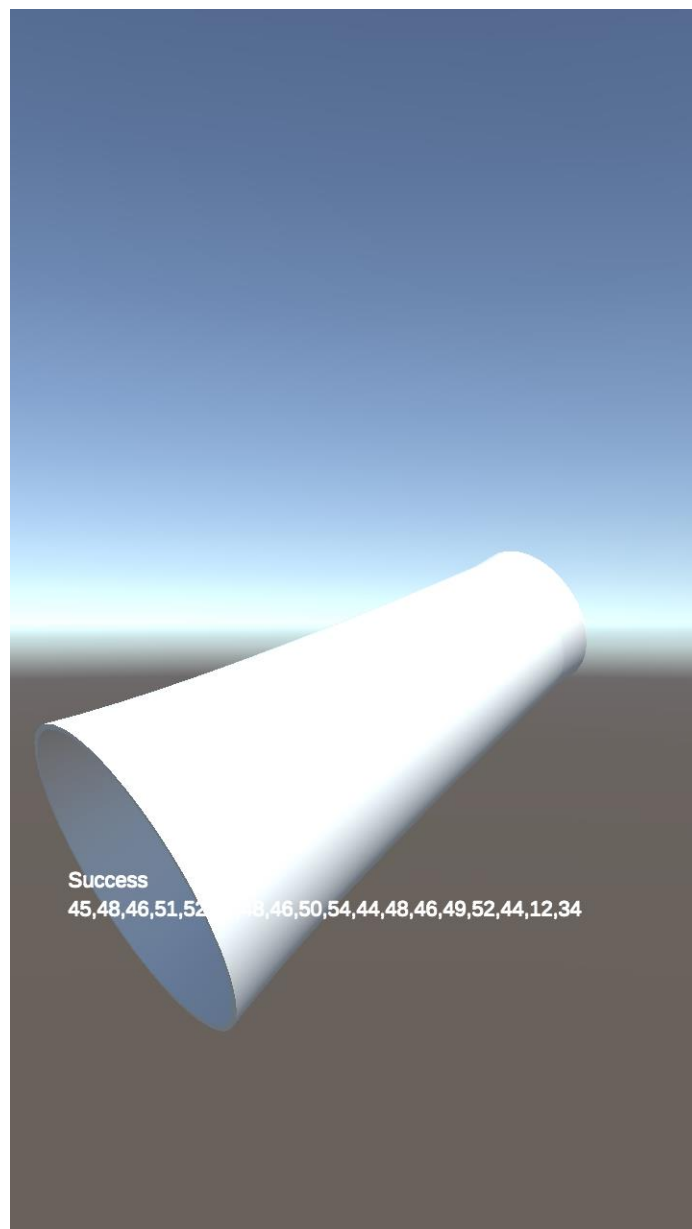


# Esp32BleAndroid Quick Start

In the Demo scene, a Unity 3D object 'tumbler' 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 Android mobile phone using the BLE (Bluetooth low energy) interface. The following shows the capture screen when the Demo scene runs.



[Demo scene](#)

This Quick Start explains how to display the Demo scene on Unity using Esp32BleAndroid. This Quick Start includes some steps for downloading both Unity sample code and the ESP32 code, then set for the Demo scene.

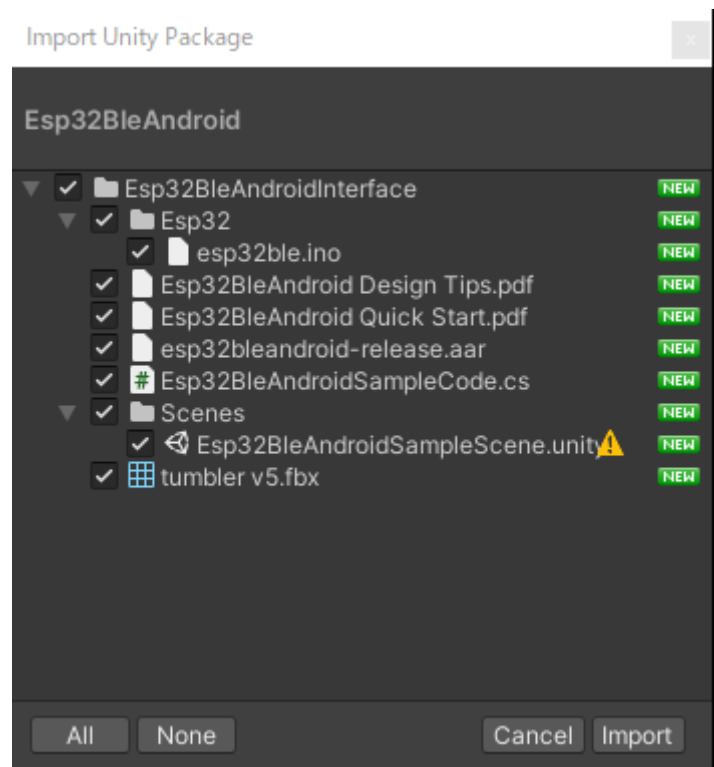
Step-1 Download and set up a package in Unity Project

Step-2 Write ESP32 code using Arduino IDE

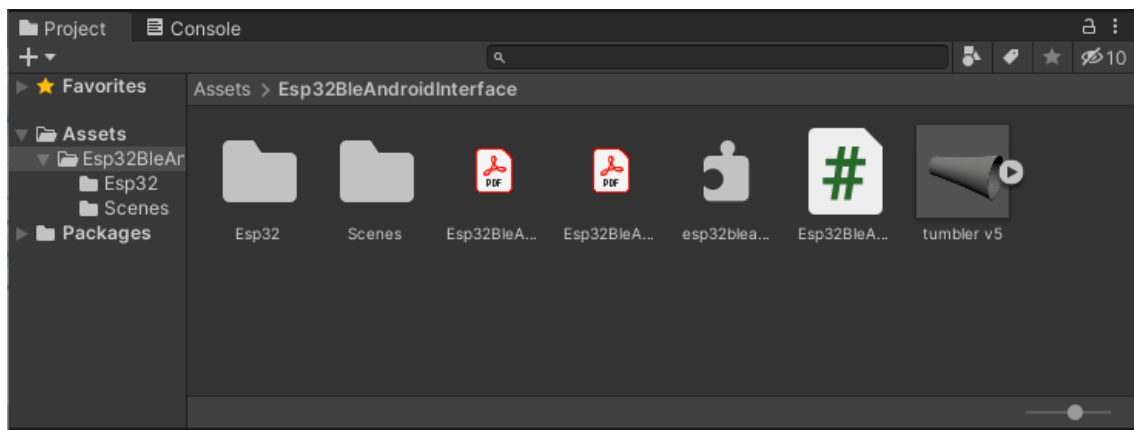
Step-3 Run Demo scene

## Step-1 Download and set up a package in Unity Project

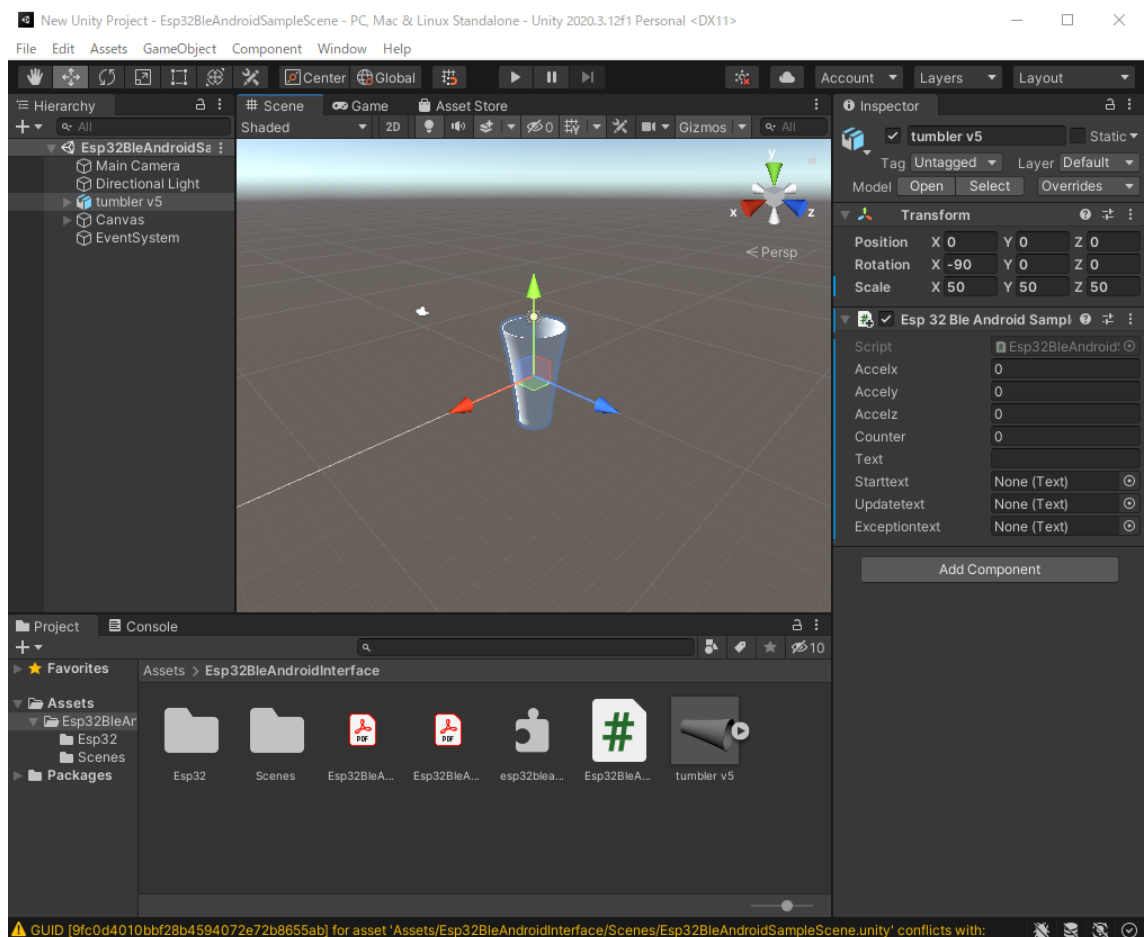
(1) Download Esp32BleAndroid from the Asset Store, and click on the 'Import' button.



(2) Import the package under the Assets folder in Unity project.



- (3) Click the 'Project' tab, choose 'Assets' > 'Esp32BleAndroidInterface' > 'Scenes' folder of the left side menu, double click on the demo scene 'Esp32BleAndroidSampleScene' in the 'Project' tab.



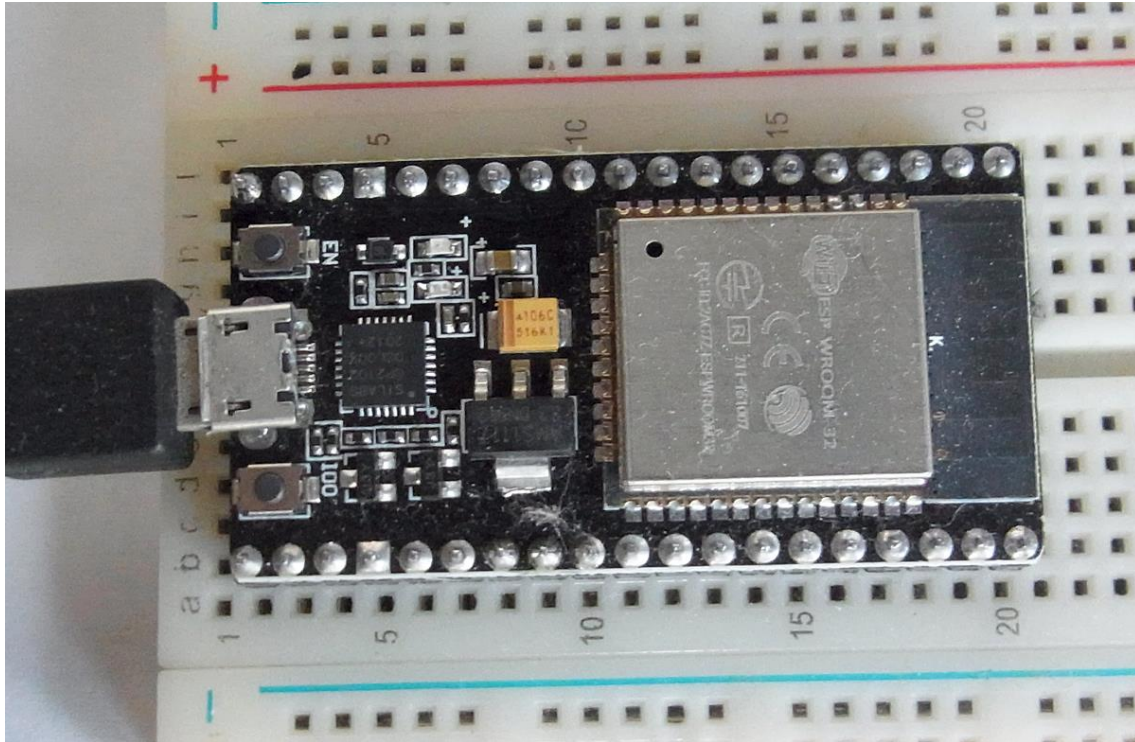
- (4) Connect Android mobile phone by USB cable to Windows PC, build the purchased assets according to [Building games for Android](#), and run it in Android mobile phone.
- (5) Display Esp32BleAndroid from application information in Android mobile phone as follows. Change the Location permission to ON because of the need to enable Bluetooth Low Energy Scanning.



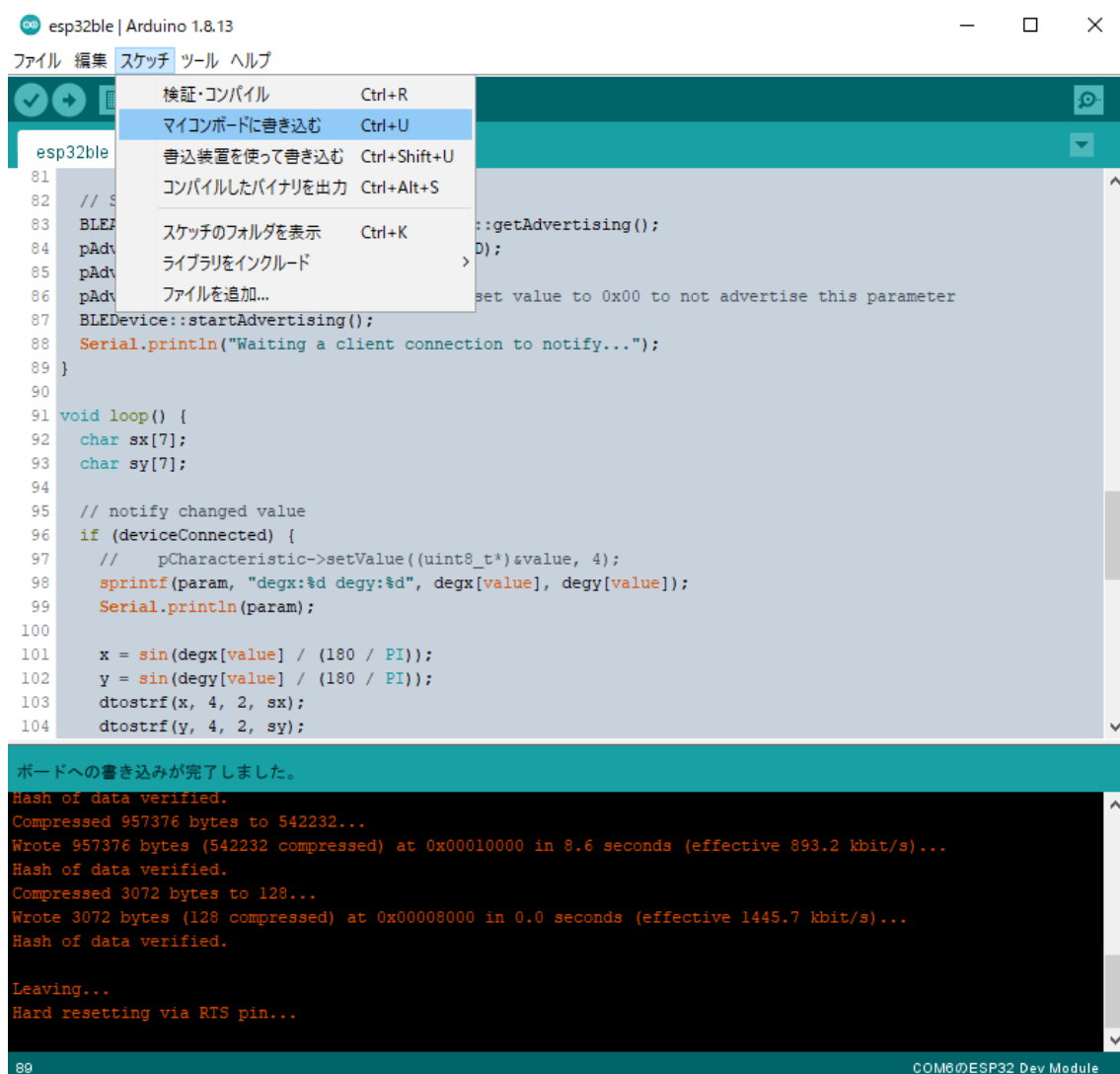
## Step-2 Write ESP32 code using Arduino IDE

- (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

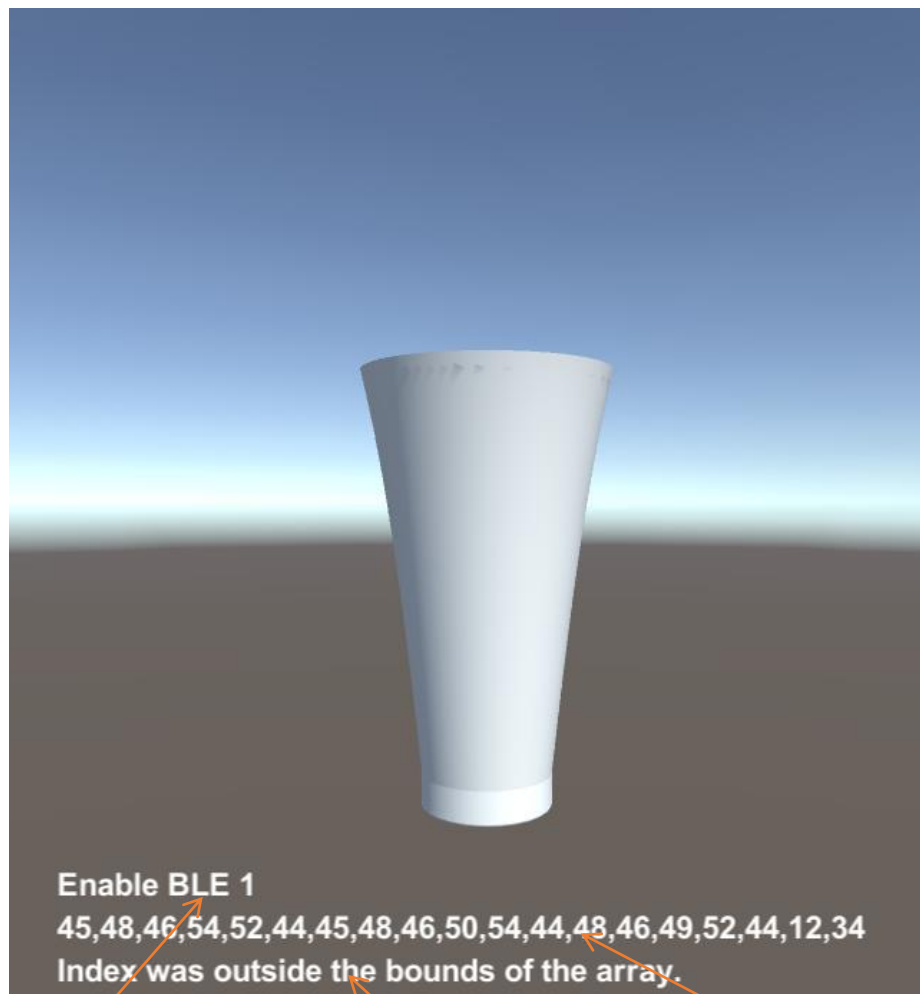


- (2) Compile and write ESP32 code 'esp32ble.ino' of the folder  
'/Assets/Esp32BleAndroidInterface/Esp32' into the ESP32 using the Arduino  
IDE.



## Step-3 Run Demo scene

- (1) The installed Esp32BleAndroid on Android mobile phone displays the game object as follows.



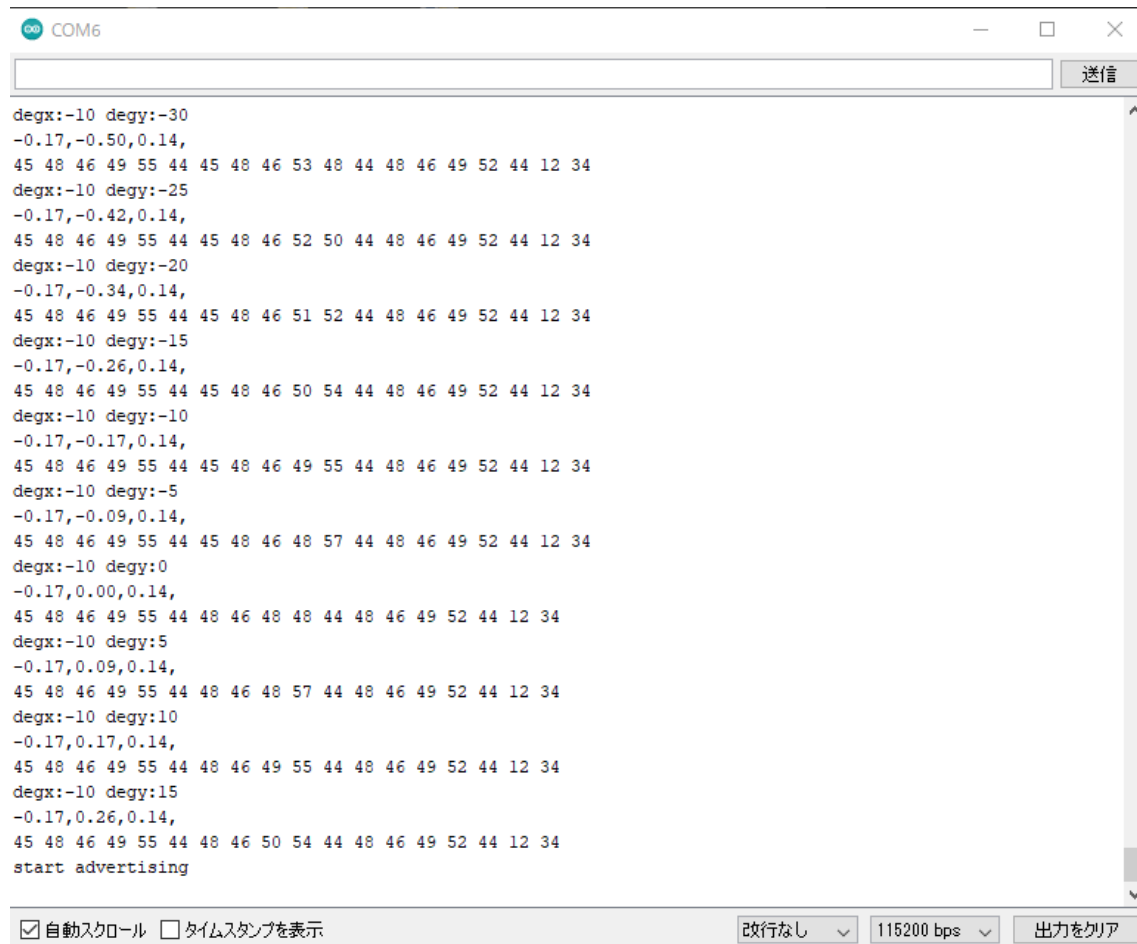
Result of Start Process

Exception Message

Received ESP32 data

Result of Start Process	'Success' : Success start process. 'Enable BLE 1' or 'Enable BLE 2' : disable location permission. Refer to Step-1 (5).
Received ESP32 data	Display the content of data when Unity receives it from ESP32.
Exception Message	Display the content of the exception when the C# script issues the exception.

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



```
degx:-10 degy:-30
-0.17,-0.50,0.14,
45 48 46 49 55 44 45 48 46 53 48 44 48 46 49 52 44 12 34
degx:-10 degy:-25
-0.17,-0.42,0.14,
45 48 46 49 55 44 45 48 46 52 50 44 48 46 49 52 44 12 34
degx:-10 degy:-20
-0.17,-0.34,0.14,
45 48 46 49 55 44 45 48 46 51 52 44 48 46 49 52 44 12 34
degx:-10 degy:-15
-0.17,-0.26,0.14,
45 48 46 49 55 44 45 48 46 50 54 44 48 46 49 52 44 12 34
degx:-10 degy:-10
-0.17,-0.17,0.14,
45 48 46 49 55 44 45 48 46 49 55 44 48 46 49 52 44 12 34
degx:-10 degy:-5
-0.17,-0.09,0.14,
45 48 46 49 55 44 45 48 46 48 57 44 48 46 49 52 44 12 34
degx:-10 degy:0
-0.17,0.00,0.14,
45 48 46 49 55 44 48 46 48 48 44 48 46 49 52 44 12 34
degx:-10 degy:5
-0.17,0.09,0.14,
45 48 46 49 55 44 48 46 48 57 44 48 46 49 52 44 12 34
degx:-10 degy:10
-0.17,0.17,0.14,
45 48 46 49 55 44 48 46 49 55 44 48 46 49 52 44 12 34
degx:-10 degy:15
-0.17,0.26,0.14,
45 48 46 49 55 44 48 46 50 54 44 48 46 49 52 44 12 34
start advertising
```