

# Esp32Ble Design Tips

Esp32Ble is an adapter interface tool for transfer from Esp32 to Unity via Windows PC using BLE (Bluetooth low energy). By connecting the sensor device (gyro, accelerometer, GPS, etc.) with Esp32 you can input various sense data to Unity. You can also output various data from Unity by connecting the display device (LED, LCD panel, etc.) with Esp32. This design tip explains how to modify for transfer data between Unity and Esp32 based on the purchased Unity script example and the purchased Esp32 code. Esp32Ble transfers array data up to 64 bytes from Unity to Esp32 and array data up to 64 bytes from Esp32 to Unity by changing the Unity script example and the Esp32 code.

## 1. How to transfer data from Esp32 to Unity

### 1.1 How to set up data to Unity in Esp32

### 1.2 How to get data from Esp32 in Unity

## 2. How to transfer data from Unity to Esp32

### 2.1 How to set up data to Esp32 in Unity

### 2.2 How to get data from Unity in Esp32

## 1. How to transfer data from Esp32 to Unity

### 1.1 How to set up data to Unity in Esp32

The Esp32 code 'esp32ble-v2.ino' generates 3D Object 'glasses' tilt data 'x, y, z' using the "SIN" function and convert it to strings using the "sprintf" function. The Esp32 code needs to change the strings to byte array data type using 'saccel.getBytes' function for sending data to Unity when Esp32 connects some sensors device.

```
...

// notify changed value
if (deviceConnected) {
    sprintf(param, "degx:%d degy:%d", degx[value], degy[value]);
    Serial.println(param);

    x = sin(degx[value] / (180 / PI));
    y = sin(degy[value] / (180 / PI));
    dtostrf(x, 4, 2, sx);
```

```

dtostrf(y, 4, 2, sy);

sprintf(formatted_msg, "%s,%s,%s,", sx, sy, z);
Serial.println(formatted_msg);

String saccel = String(formatted_msg);
//Serial.println(saccel);
byte sbyte[saccel.length()];
saccel.getBytes(sbyte, saccel.length() + 1);
//accelCharacteristic.setValue(sbyte, saccel.length());

//    pCharacteristic->setValue((uint8_t*)formatted_msg, i);
pCharacteristic->setValue(sbyte, saccel.length());
pCharacteristic->notify();

...

```

The Esp32 code 'esp32ble-v2.ino' calls the 'setValue' function in the following format for sending data to Unity.

```

void setValue(byte[] sbyte, int length);

```

## 1.2 How to get data from Esp32 in Unity

Unity script example 'Esp32BleSampleCode.cs' receives BLE data from Esp32 as byte array data 'readdata' of return value of 'm\_Esp32BleLib.UpdateRead' function. The byte array data 'readdata' encodes string code using 'System.Text.Encoding.UTF8.GetString' function and, using 'text.Split' function and 'float.Parse' function, the string code is converted to float position data.

```

...

// Update is called once per frame
void Update()
{
    byte[] readdata = new byte[] { };

```

```

//UnityEngine.Debug.LogWarning("Update");
if (!m_Esp32BleLib.UpdateRead(ref readdata))
{
    return;
}
UnityEngine.Debug.LogWarning(" Read1: " + readdata[0] + " " + readdata[1] +
" " + readdata[2]);
UnityEngine.Debug.LogWarning(" Read2: " + readdata.Length);

string text = System.Text.Encoding.UTF8.GetString(readdata);
UnityEngine.Debug.LogWarning(" Read3: " + text);
string[] arr = text.Split(',');
float[] acceldata = new float[3];
acceldata[0] = float.Parse(arr[0]);
acceldata[1] = float.Parse(arr[1]);
acceldata[2] = float.Parse(arr[2]);

UnityEngine.Debug.LogWarning(" Update: " + acceldata[0] + " " +
acceldata[1] + " " + acceldata[2]);

    accelx = acceldata[0] * 100;
    accely = acceldata[1] * 100;
    accelz = acceldata[2] * 100;

    transform.rotation = Quaternion.AngleAxis(accelx, Vector3.up) *
Quaternion.AngleAxis(accely, Vector3.right);
}

...

```

Unity script example 'Esp32BleSampleCode.cs' calls the 'UpdateRead' function in the following format for reading data from Esp32.

```
bool UpdateRead(ref byte[] readdata)
```

## 2. How to transfer data from Unity to Esp32

### 2.1 How to set up data to Esp32 in Unity

Unity script example 'Esp32BleSampleCode.cs' call back 'ButtonClick' function at button 'On' clicked. 'ButtonClick' function inputs data from 'inputFieldno0' / 'inputFieldno1', then calls 'm\_Esp32BleLib.Command' function with the parameter of 'writedata' byte array.

```
...
public void ButtonClick()
{
    UnityEngine.Debug.LogWarning("ButtonClick: ");

    byte[] writedata = new byte[2] { byte.Parse(inputFieldno0.text) ,
    byte.Parse(inputFieldno1.text) };
    UnityEngine.Debug.LogWarning(writedata[0] + " " + writedata[1]);
    m_Esp32BleLib.Command(writedata);
}
...
```

Unity script example 'Esp32BleSampleCode.cs' calls the 'Command' function in the following format for sending byte array data to Esp32.

```
void Command(byte[] writedata);
```

### 2.2 How to get data from Unity in Esp32

The Esp32 code 'esp32ble-v2.ino' calls back 'onWrite' function at received data from Unity. The 'onWrite' function receives Inputfield 'No0' / Inputfield 'No1' of Unity as 2 bytes of data using the pCharacteristic->getValue pCharacteristic->getValue' function.

```
...
```

```
class RxCallbacks: public BLECharacteristicCallbacks {
    void onWrite(BLECharacteristic *pCharacteristic) {
        // std::string rxValue = pCharacteristic->getValue();
        byte row = pCharacteristic->getValue()[0];
        byte col = pCharacteristic->getValue()[1];
        Serial.println("***** receive data! *****");
        Serial.println(row);
        Serial.println(col);
    }
};

...
```

The Esp32 code 'esp32ble-v2.ino' calls the 'getvalue' function for receiving byte array data in the following format from Unity.

```
byte getvalue()[x];
```

x: byte array index

return: byte