MicroBle Design Tips

MicroBle is an adapter interface tool for transfer from mcro:bit to Unity via Windows PC using BLE (Bluetooth low energy). By connecting the sensor device (gyro, accelerometer, GPS, etc.) with mcro:bit, 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 mcro:bit. This design tip explains how to modify for transfer data between Unity and mcro:bit based on the purchased Unity script example and the purchased micro:bit code. MicroBle transfers array data up to 64 bytes from Unity to micro:bit and array data up to 64 bytes from micro:bit to Unity by changing the Unity script example and the micro:bit code.

1. How to transfer data from mcro:bit to Unity

1.1 How to set up data to Unity in mcro:bit

The mcro:bit code ‘microble-v2.ino’ generates 3D Object ‘glasses’ tilt data ‘x, y, z’ using the ‘accel.update’ acceleration function and convert it to strings using the ‘sprintf’ function. The mcro:bit code needs to change the strings to byte array data type using ‘saccel.getBytes’ function for sending data to Unity when mcro:bit connects some sensors device.

```
    …

void setAccelCharacteristicValue() {
  char acceldata[30];

  accel.update();
  float accelX = (float)accel.getX() * 0.0156;
  float accelY = (float)accel.getY() * 0.0156;
  float accelZ = (float)accel.getZ() * 0.0156;
```

```
   sprintf(acceldata, "accel - X: % 3.4f Y: % 3.4f Z: % 3.4f", accelX , accelY ,
accelZ );
   Serial.println(acceldata);

   sprintf(acceldata, " % d, % d, % d", (int)(accelX * 100000), (int)(accelY * 100000),
(int)(accelZ * 100000));
   Serial.println(acceldata);
   //accelCharacteristic.setValue(accel);

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

    ...
```

The mcro:bit code 'microble-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 mcro:bit in Unity

Unity script example 'MicroBleSampleCode.cs' receives BLE data from mcro:bit as byte array data 'data' of return value of 'm_ mcro:bit BleLib.UpdateRead' function. The byte array data 'data' 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_MicroBleLib.UpdateRead(ref readdata))
      {
            return;
      }
      UnityEngine.Debug.LogWarning(" Read1: " + readdata[0] + " " + readdata[1] +
" " + readdata[2]);
      UnityEngine.Debug.LogWarning(" Read1: " + readdata.Length);

      string text = System.Text.Encoding.UTF8.GetString(readdata);
      UnityEngine.Debug.LogWarning(" Read: " + text);
      string[] arr = text.Split(',');
      float[] acceldata = new float[3];
      acceldata[0] = float.Parse(arr[0]) / 100000;
      acceldata[1] = float.Parse(arr[1]) / 100000;
      acceldata[2] = float.Parse(arr[2]) / 100000;


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

      accelx = acceldata[0] * 400;
      accely = acceldata[1] * 400;
      accelz = acceldata[2] * 400;...
          ...
```

Unity script example 'MicroBleSampleCode.cs' calls the 'UpdateRead' function in the following format for reading data from mcro:bit.

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

2. How to transfer data from Unity to mcro:bit

2.1 How to set up data to mcro:bit in Unity

Unity script example 'MicroBleSampleCode.cs' call back 'ButtonClick' function at button 'On' clicked. 'ButtonClick' function inputs data from Inputfield 'Row' / Inputfield 'Col', then calls 'm_MicroBleLib.Command' function with the parameter of

‘writedata’ byte array.

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

      byte[] writedata = new byte[2] { byte.Parse(inputFieldrow.text) ,
byte.Parse(inputFieldcol.text) };
      UnityEngine.Debug.LogWarning(writedata[0] + " " + writedata[1]);
      m_MicroBleLib.Command(writedata);
}
    …
```

Unity script example ‘MicroBleSampleCode.cs’ calls the ‘Command’ function in the following format for sending byte array data to mcro:bit.

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

2.2 How to get data from Unity in mcro:bit

The mcro:bit code ‘microble-v2.ino’ calls ‘getContolCharacteristicValue’ function at received data from Unity. The ‘getContolCharacteristicValue’ function receives Inputfield ’Row’ / Inputfield ‘Col’ of Unity as 2 bytes of data using the ’controlCharacteristic.value’ function.

```
    …
void getContolCharacteristicValue() {
  Serial.println("********** receive data! **********");
  if (controlCharacteristic.value()) {
     byte row = controlCharacteristic.value()[0];
     byte col = controlCharacteristic.value()[1];
     Serial.println(row);
     Serial.println(col);

     led_pset( col, row, HIGH);
```

```
      delay(200);
      led_pset( col, row, LOW);
   }
}
    ...
```

The mcro:bit code ‘microble-v2.ino’ calls the ‘value’ function for receiving byte array data in the following format from Unity.

```
byte value()[x];

x: byte array index
return: byte
```