

SerialPortUtility

これは何

シリアル通信をマルチスレッドに簡単に行うためのライブラリです。SerialPortWriteとかReadをByte配列で行ったりできます。

テスト環境

Unity2018.4.4f1 Windows10 64bit Micro:Bit

サンプルを実行する

MicroBit側の準備

[mbedのサイト](#)でmicrobit用のコンパイラを準備します。

今回は以下のようなコードにしました。

```
1 #include "MicroBit.h"
2 Serial pc(USBTX, USBRX);
3
4 int main()
5 {
6     //ボーレートを設定
7     pc.baud(57600);
8     while(1)
9     {
10         wait(0.5);
11         /*読み込む内容があるときにReceiveを送信する*/
12         while(pc.readable() == 1)
13         {
14             pc.getc();
15             pc.printf("Receive\n");
16         }
17     }
18 }
```

(GitHubにあげるときはここちゃんとしてよう)

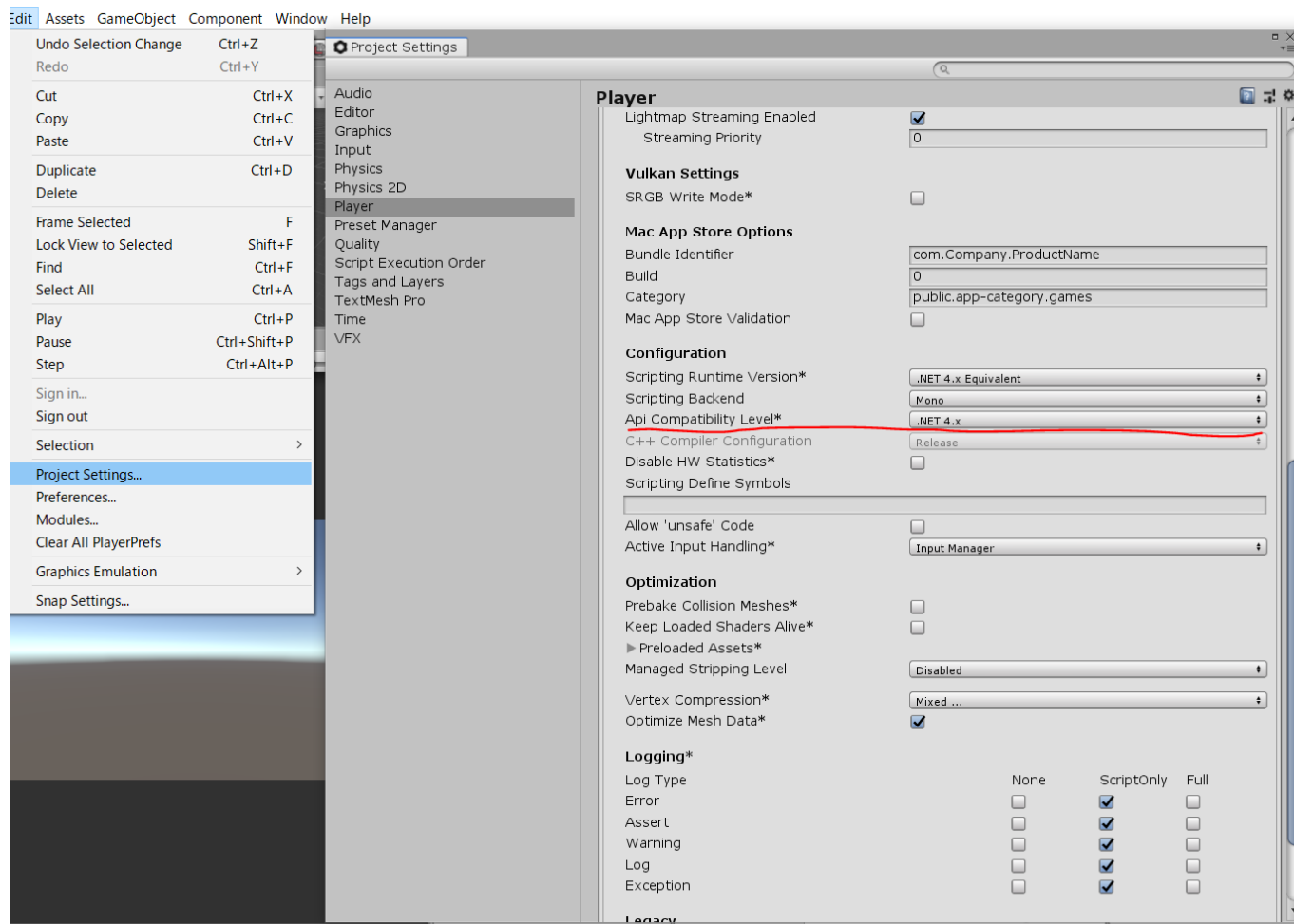
Unity側の準備

Unityの設定を変更する

そのままだとUnityではSerialPortを扱えないので設定を変更します

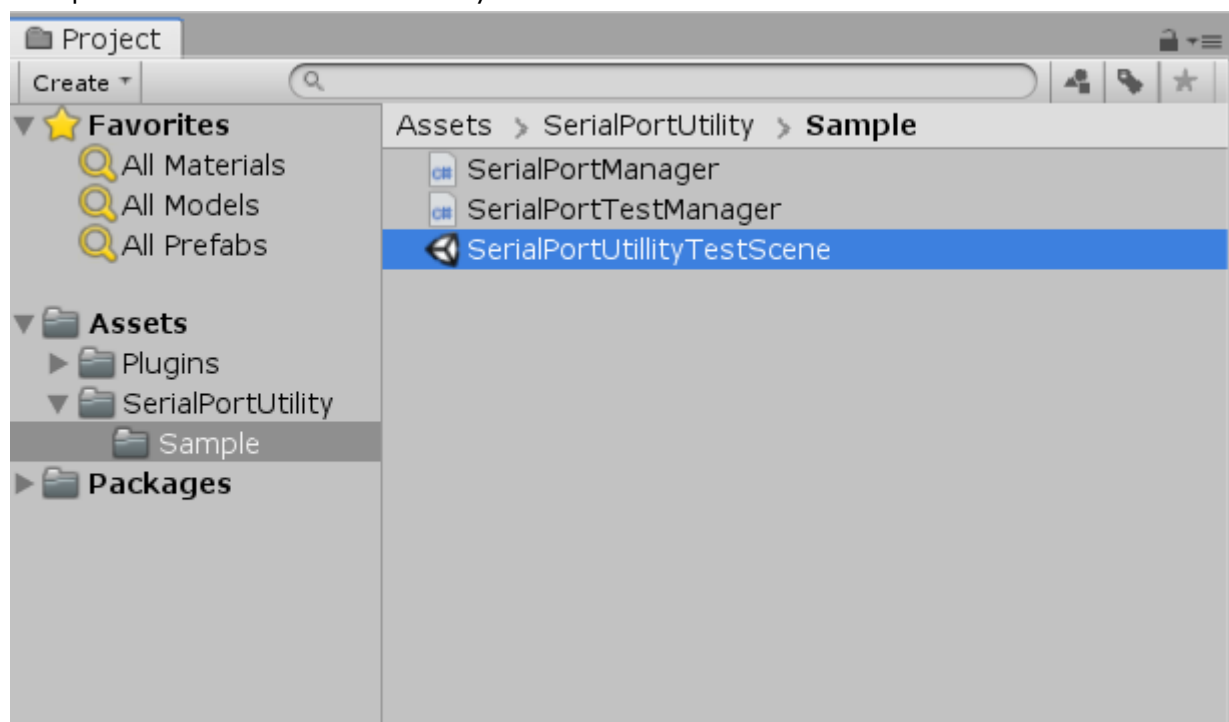
プロジェクトセッティングウィンドウを開き、(UnityのツールバーからEdit>ProjectSettingと選ぶ)

PlayerのApi Compatibility Levelを.NET4.xに変更してください。

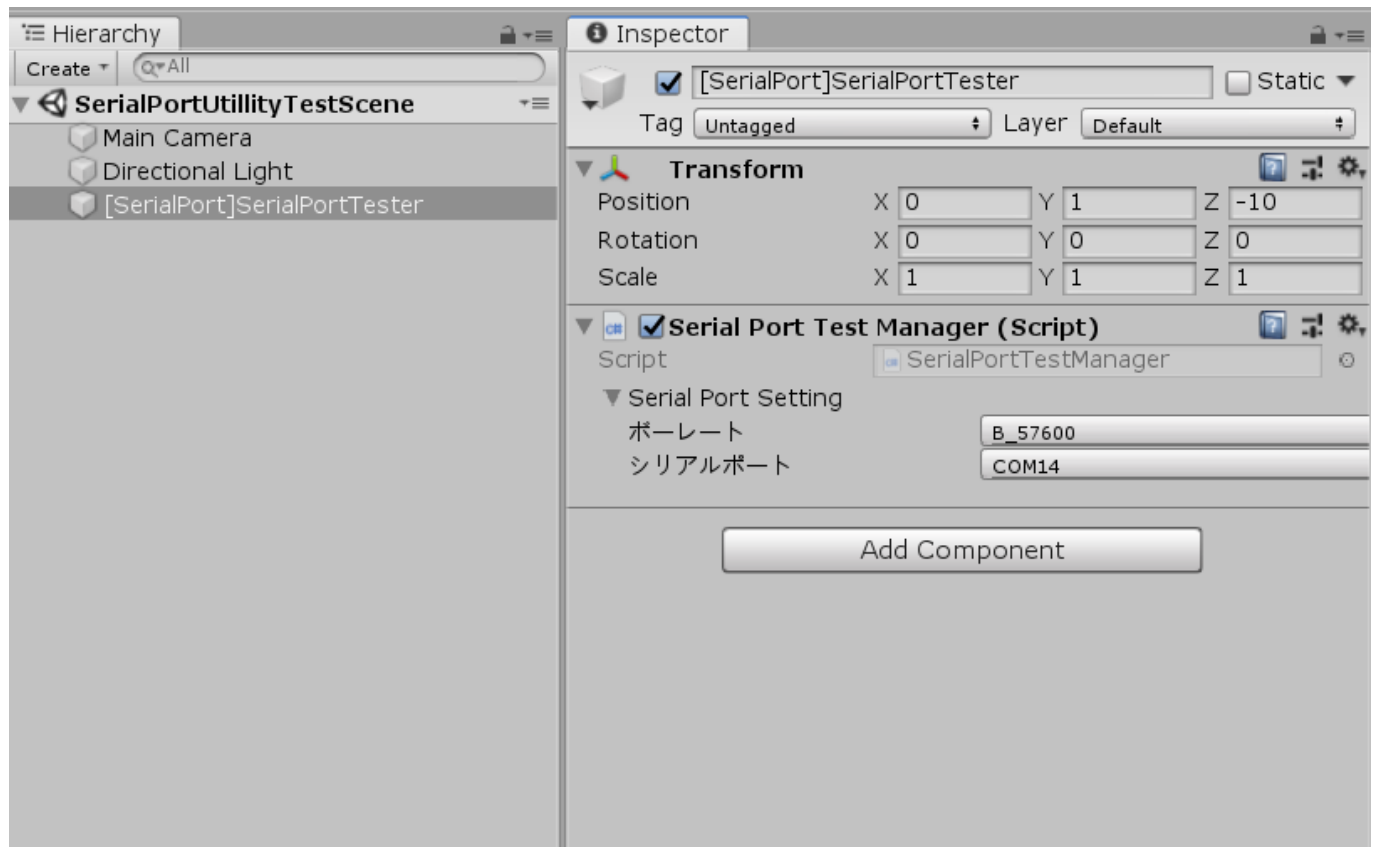


サンプルシーンを開く

Sampleフォルダ内のSerialPortUtilitySceneを開いてください。



Hierarchy上でSerialPortTesterを選択し、ボーレートと通信したい機器が認識されているシリアルポートを設定します。



Unityでの設定は以上です。

実行する

実行後、Aキーを押し、Recieved!が4回Debug.Logに表示されれば成功です

具体的な使い方

SerialPortReaderクラス

シリアル通信をUnity側で受信するためのクラスです。
1byteずつ読み、読むたびにアクションを発火します。
SerialPortReader.OnStreamReadにコールバックを追加してください。

SerialPortWriterクラス

シリアル通信をUnityから送信するためのクラスです。
SerialPortWriter.WriteSerialPort()の引数に乗つけたByte配列を1byteずつ送信します。