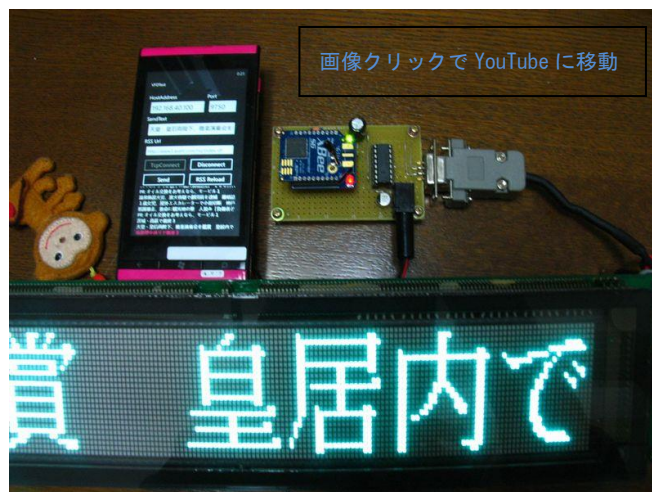


WindowsPhone7.5 から RS-232C シリアルデバイスを制御する。

～VFD を利用した電光掲示板の制作＋自作機器の制御～

2012.04.05 7M4MON



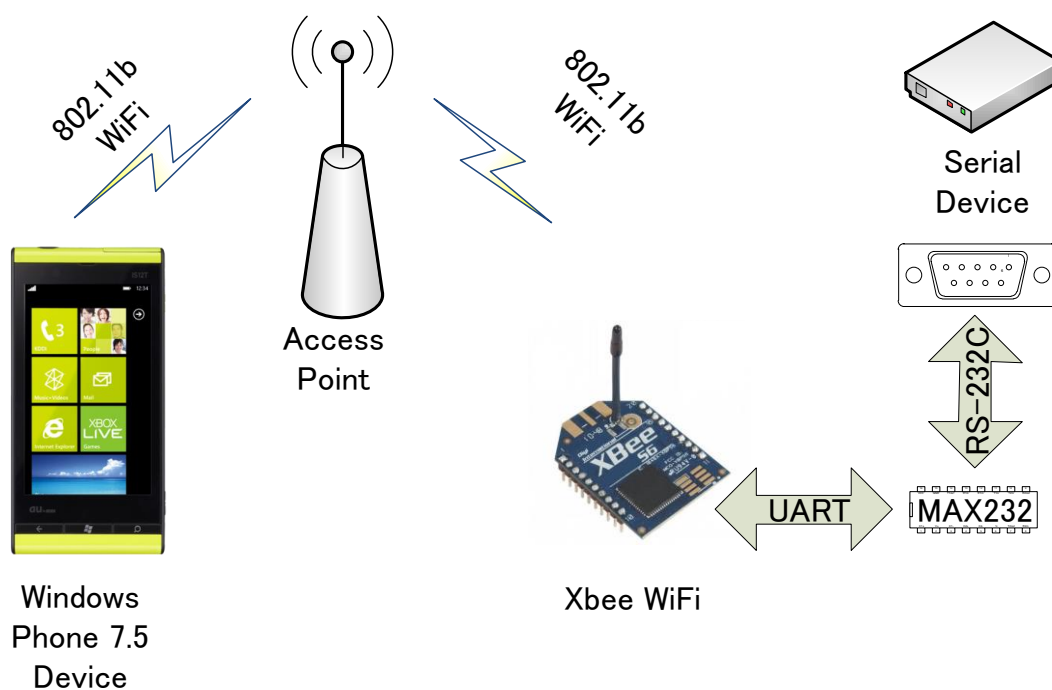
## 1. 概要

Windows Phone 7.5 は Android 2.1 以降と異なり Bluetooth の SPP プロファイルをサポートしていないため簡単に RS-232C シリアルデバイスを制御することはできません。

そこで、Ethernet-UART コンバータである [デザインインターナショナル社](#) の [XBee WiFi](#) を使用して RS-232C シリアルデバイスを制御するシステムを制作しました。

## 2. 接続

システムの概要図を下に示します。



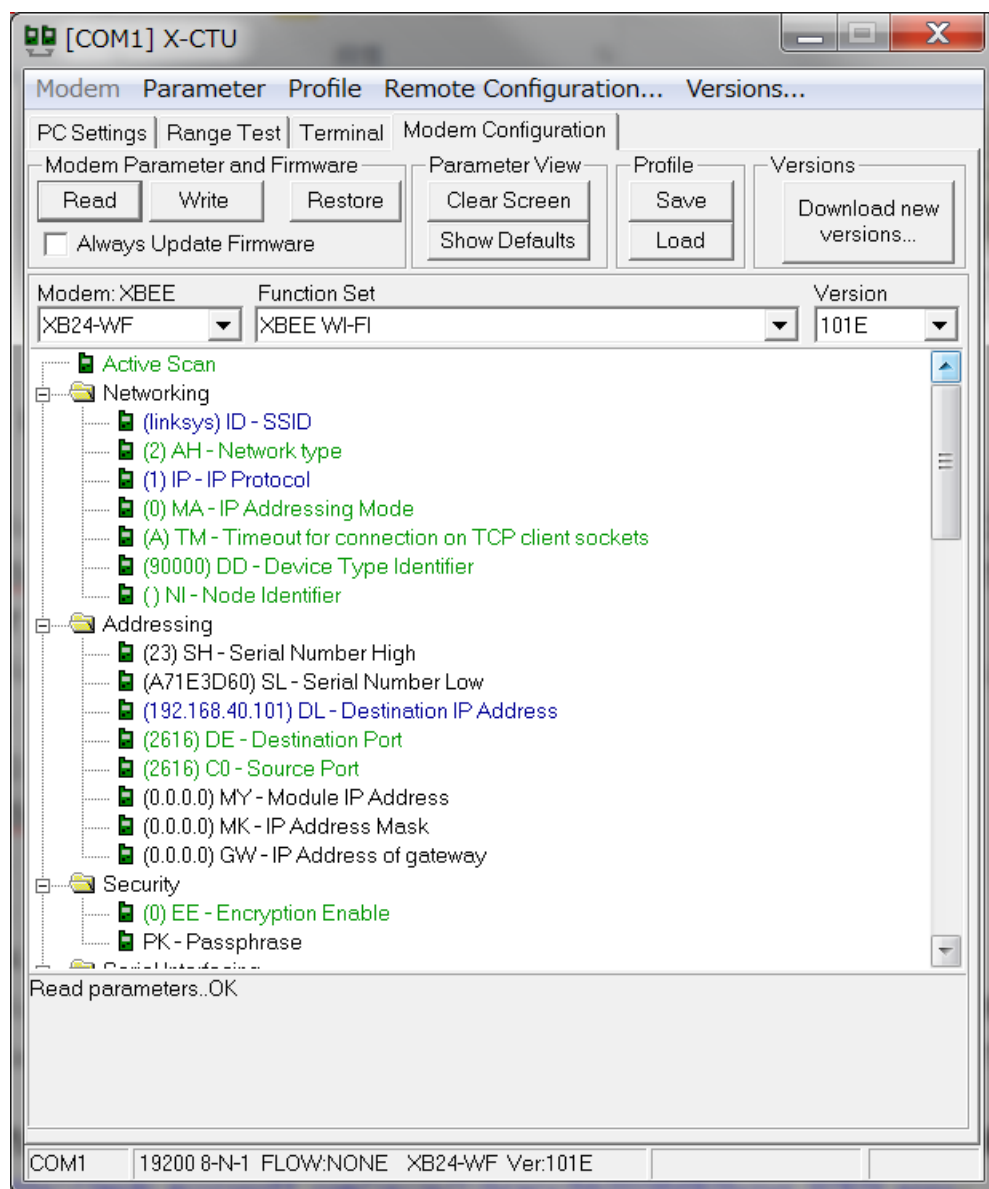
WindowsPhone7.5 端末から TCP Socket 通信を WiFi 経由で行い、XBee WiFi に接続します。

XBee WiFi の UART 側には RS-232C インターフェース IC が繋がっていて、任意の RS-232C デバイスと通信可能です。

なお、Windows Phone 7.5 の WiFi 接続はアドホックモードをサポートしていないため、アクセスポイントが必要です。

### 3. XBee WiFi の設定

XBee の設定には X-CTU というツールを使います。XBeeWiFi を COM ポートに接続し設定ユーティリティ X-CTU でモデムの情報を読み取ると下記のような画面が出ます。



今回は UDP ではなく TCP の Socket 通信を行うので、IP Protocol を 1-TCP に変更します。

また、Destination IP Address や Bound Rate を接続先に合わせて変更します。

### 4. SocketClientTcp ライブラリ (DLL) の作成

WindowsPhone7.5 から XBee WiFi まで TCP Socket 通信を行うライブラリを制作します。

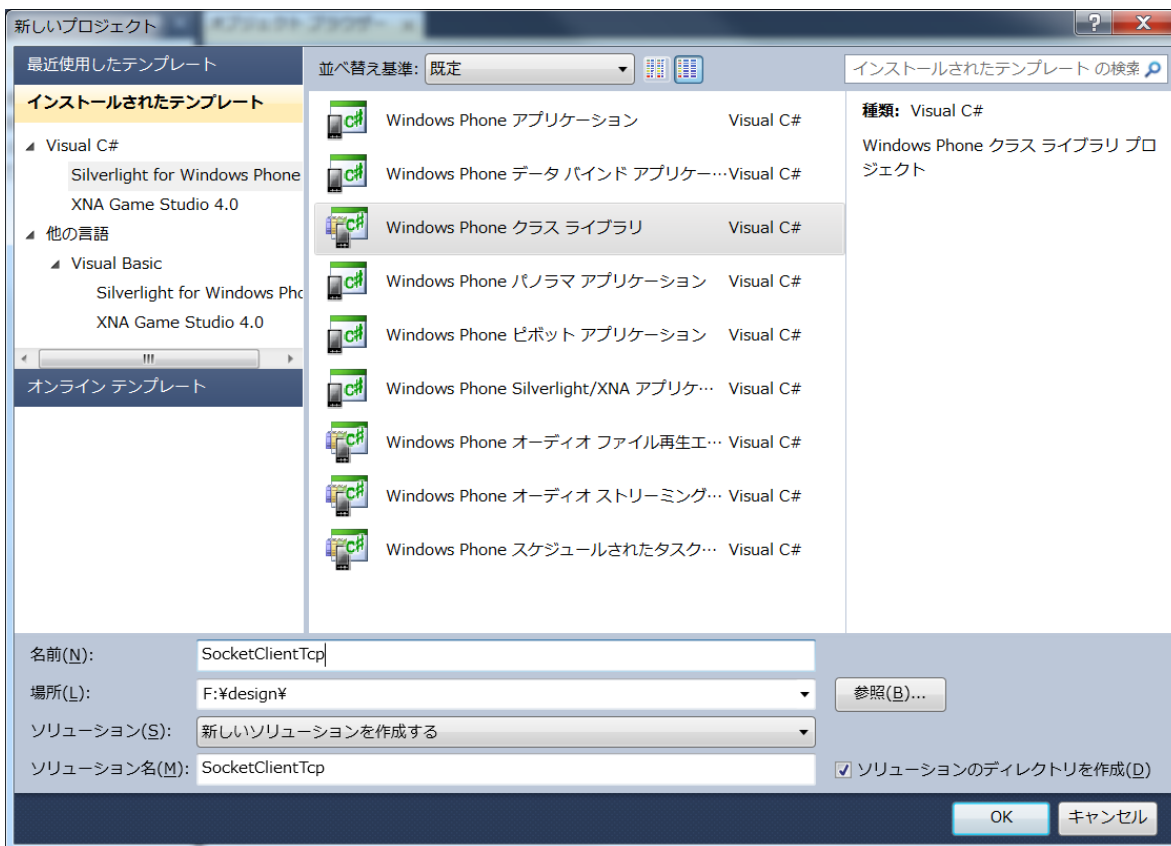
Socket 通信のサンプルコードは、MSDN に掲載されています。

方法: [Windows Phone の TCP ソケット クライアント アプリケーションを作成して使用する](http://msdn.microsoft.com/ja-jp/library/hh202858%28v=vs.92%29.aspx)

<http://msdn.microsoft.com/ja-jp/library/hh202858%28v=vs.92%29.aspx>

まず、TcpSocket 通信を行う DLL を作成します。

VisualStudio で WindowsPhone クラスライブラリを新規作成します。



作成したクラスにサンプルプログラムを引用して、TCP Socket 通信を行う DLL を作成します。

上記サンプルの送受信メソッドは、引数・戻り値ともに String 型ですので、バイト配列の送受信は出来ません。そこで、SocketClient クラスを改造しバイト配列を扱えるようにします。

送信側は `public string Send(string data)` 関数を使用し、下記の変更を行います。

- `public string SendByteArray(byte[] data)` に関数名と引数の型を変更します。
- `byte[] payload = Encoding.UTF8.GetBytes(data)` を `byte[] payload = data` に変更します。

#### 説明

Send関数は内部でString型をバイト配列に変換してSocket通信を行っているようです。

よって、String型からByte型に変換している個所を、直接バイト配列を扱うように変更しました。それ以外に変更はありません。

受信側は `public string Receive()` 関数を使用し、下記の変更を行います。

- `public byte[] ReceiveByteArray()` に関数名と関数の型を変更します。
- `string response = "Operation Timeout"` を `byte[] response = null` に変更します。
- `response = Encoding.UTF8.GetString(e.Buffer, e.Offset, e.BytesTransferred)` を `response = e.Buffer` に変更します。
- `response = response.Trim('¥0')` を下記のコードに変更します。

```
int i = response.Length;
while (response[i - 1] == 0 && i > 1) { i--; }
Array.Resize<byte>(ref response, i);
```

戻り値を設定している下記2か所をコメントアウトします。

```
response = e.SocketError.ToString();
response = "Socket is not initialized";
```

## 説明

関数の型と戻り値をバイト配列に変更します。

WindowsPhone7.5のソケット通信の受信バッファの配列は、受信したデータを格納後、0で埋められている仕様になっています。よって、受信バッファをそのまま受信データとして扱うことができず、受信データがバッファのどこで終了したのか判別する必要があります。

今回は、受信バッファをローカルの配列に格納した後、配列の最後から配列の中身が0かどうかを判定し、最初に0以外が現れた個所を受信データの末尾とみなす手法をとりました。

この方法では、受信データの末尾が実際に0だった場合に、埋められた0と区別できずに削除されてしまうという不具合があります（例: 00-01-02-00 → 00-01-02）。ただし、0のみで構成された受信データは、先頭の1バイトである0が戻り値になります（例: 00-00-00 → 00）。受信データの末尾が0x00である場合は別途回避策が必要です。

受信データが存在しなかったり、ソケットが初期化されていなかったりするとNullを返します。

これらの改造を行ったものをビルドし、SocketClientTcp.dllを作成します。

作成したプロジェクトは[こちら](#)でダウンロードできます。

## 5. 実際の応用例

これまでの章でXBee WiFiとSocket通信を行う準備が整いました。

ここから実際のアプリケーションを制作していきます。

ターゲットデバイスはシリアルポート搭載機器なら何でもよかったのですが、今回は双葉電子工業製のVFDモジュールGP1058A02を使用しました。

PC向け制御プログラムは下記で公開されているものを使用させて頂きました。

<http://www.kosaka-lab.com/tips/2010/06/vfd-rsstwitter.php>

これをベースにWindowsPhone向けに下記変更を行いました。

- ・シリアルポートを使用するところを、前章で作成したSocketClientTcp.dllを使用するように変更
- ・ArrayListやXmlDocumentはWindowsPhone向けのフレームワークに用意されていないので代替手段へ変更
- ・文字コードのJIS変換は[atsushieno](#)さん移植の[I18N.CJK](#)を使用
- ・RSS取得部は[okazuki](#)さん作成の[RssReader](#)を参考に作成
- ・メインのコードが長かったのでVFDクラスを新たに作成してコードを移動

完成したアプリケーションのスクリーンショットを下記に示します。



冒頭の動画では朝日新聞の[朝日新聞デジタル速報ニュース](#)のRSSを取得してGP1058A02に表示させています。

作成したプロジェクトは[こちら](#)でダウンロードできます。

## 6. 終わりに

XBee WiFi を使用することで WindowsPhone からシリアルデバイスを制御することが出来ました。

WindowsPhone7.5 は、洗練された UI & 素晴らしい開発環境 & 過去の VB.net/C# のコードの多くが流用可能と 3 拍子揃ったプラットフォームです。

これを機に WindowsPhone アプリケーションを開発してみたいはかがでしようか？

## 7. おまけ

WindowsPhone7.5 から VisualBasic 言語を使用し、自作機器を制御する



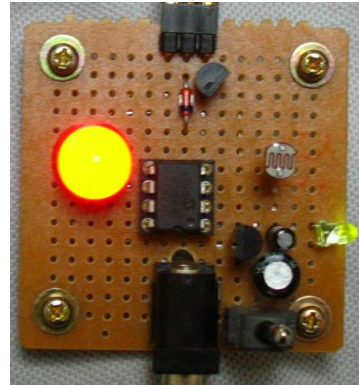
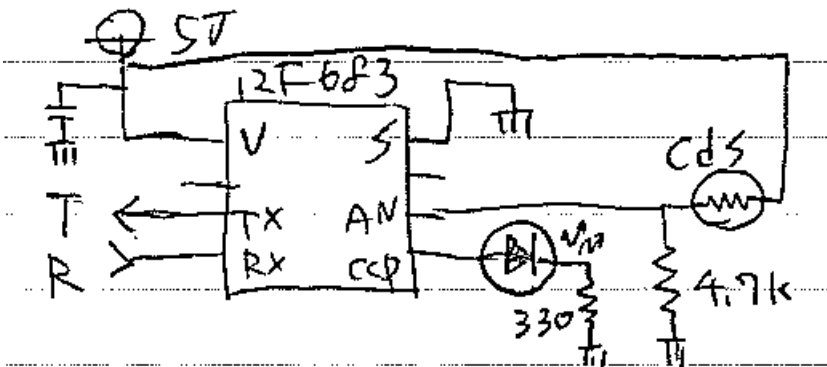
画像クリックで YouTube に移動

4章で作成した SocketClientTcp クラスは DLL 化したので、VisualBasic 言語からでも扱うことができます。この章では Lantronix 社の XPort03R を使用して自作機器をコントロールしてみます。

## 7. 1 自作機器について

以前作成した[パソコン等から外部端子付きの電源をリモートコントロールする治具](#)に、CdS セル（照度センサ）と大型 LED をつける改造を行いました。

回路図、外観図を下記に示します。

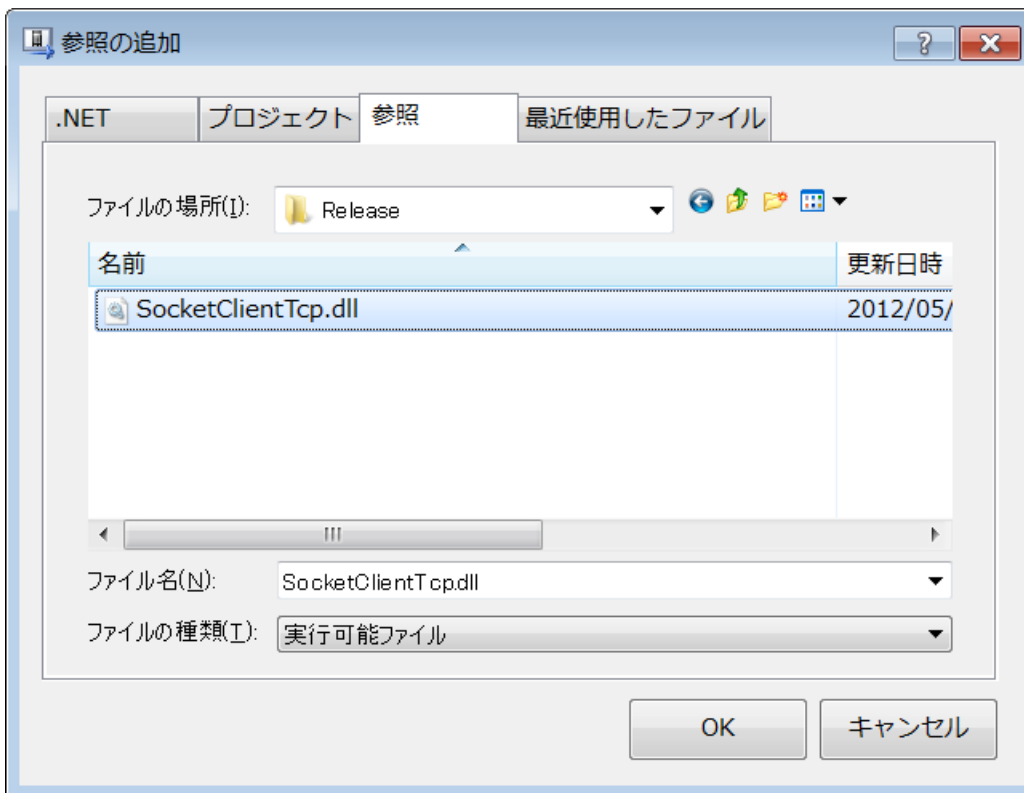


マイコン側のファームウェアは同じものです。 [ソース](#)、[HEX](#)

この機器はバイト列を受け取ると、その値を PWM に設定し、LED の明るさを制御します。また、周囲の明るさを CdS セルで読み取って、値を送信します。

## 7. 2 アプリケーションの作成

まず、4 章で作成した SocketClientTcp.dll を参照に追加します。



次に Import 文で SocketClientTcp クラスを使用できるようにして、

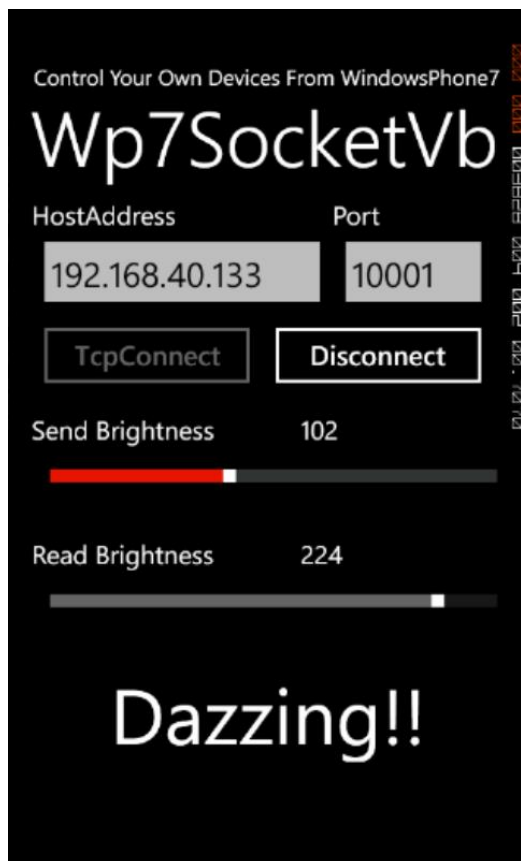
```
Imports SocketClientTcp
```

client というインスタンスを作成すれば、他の関数と同様に使用する事ができます。

```
client = New SocketClientTcp.SocketClient
```



作成したアプリケーションのスクリーンショットを下記に示します。



上のスライダーでLEDの明るさを制御し、読み取った明るさを下のスライダーに表示します。

せっかくの高機能デバイスなので、加速度センサーを利用して端末を振ったことを（シェイクイベントで）検知してLEDを制御したり、起動時にカメラのフラッシュをONにしてCdSセルに近づけた時に「Dazzing!（眩しい!）」と表示したりしています。また、読み取り値にしきい値を持たせて、暗い時は「Dark…」と表示させています。

作成したプロジェクトは[こちら](#)でダウンロードできます。

最後までお読みいただきありがとうございました。