

OPN-2002i SERIES/OPN-3002i SERIES

DATA collector

OPN SDK ユーザーズガイド (C#)

改版履歴

資料管理番号：SI13025

発行管理番号：DM-130705

製 品 名：OPN-2002i Series / OPN-3002i Series

版	日付	変更箇所	内容
Rev 1.0	2013/7/22	—	新規作成

はじめに

本書は、データコレクタ「OPN-3002i」, 「OPN-2002i」を利用した C#向けアプリケーションの開発に利用されることを目的としています。

データコレクタの操作方法や、C#開発で必要となる一般的な技術情報につきましては、それぞれの説明書などを参照してください。

本書の構成

1. 開発環境

本 SDK を利用できる条件や手順について、解説しています。

2. APIの利用

クラスライブラリの主な API について、概要と利用方法をサンプルコードと合わせて解説しています。

コマンドリファレンスについては、「OPN コマンドリファレンス.doc」を参照して下さい。
API の詳細については、本書には記載していません。「OPNTermSDK_API リファレンス_COM.doc」を参照してください。

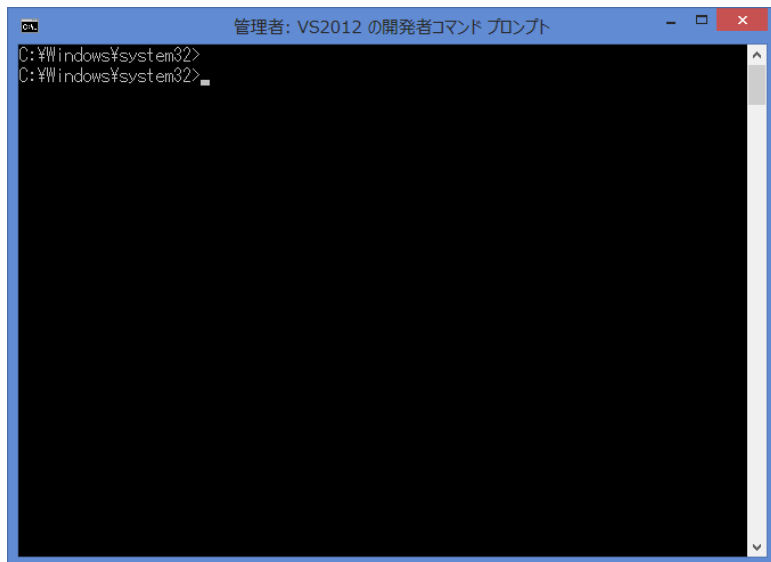
目次

はじめに	2
本書の構成	2
1. 開発環境	2
2. APIの利用	2
目次.....	3
1. 開発環境.....	4
1.1 ビルド環境	4
1.2 実行環境	4
1.3 開発環境でのSDK使用手順.....	4
■COM登録の方法	5
■ファーストAPI作成	6
2. APIの利用	10
■ using OPNTermComDllLib;行を追加する	10
■ WOPN2002iBluetoothServiceの生成と取得	11
■ C#側がデータコレクタからの接続を待ち受ける	11
■ C#側からデータコレクタに接続する	12
■ 切断する	12
■ データコレクタとの通信で発生したイベントを処理する	13
■ 任意のコマンドを実行する	15

また、本書では、開発環境として Visual Studio 2012 を使った場合の手順を解説します。

■COM登録の方法

コマンドプロンプトを起動します。(注意：管理者権限で起動すること)



以下のコマンドを実行して SDK の dll (/sdk/bin/OPNTermComDll.dll) を COM 登録します。

(※太字&斜体部分は、インストール環境によって異なりますので、ご注意ください)

```
regsvr32 C:\Yopto_opn2002_sdk_win.opnterm¥OPNTerm¥Debug¥OPNTermComDll.dll
```



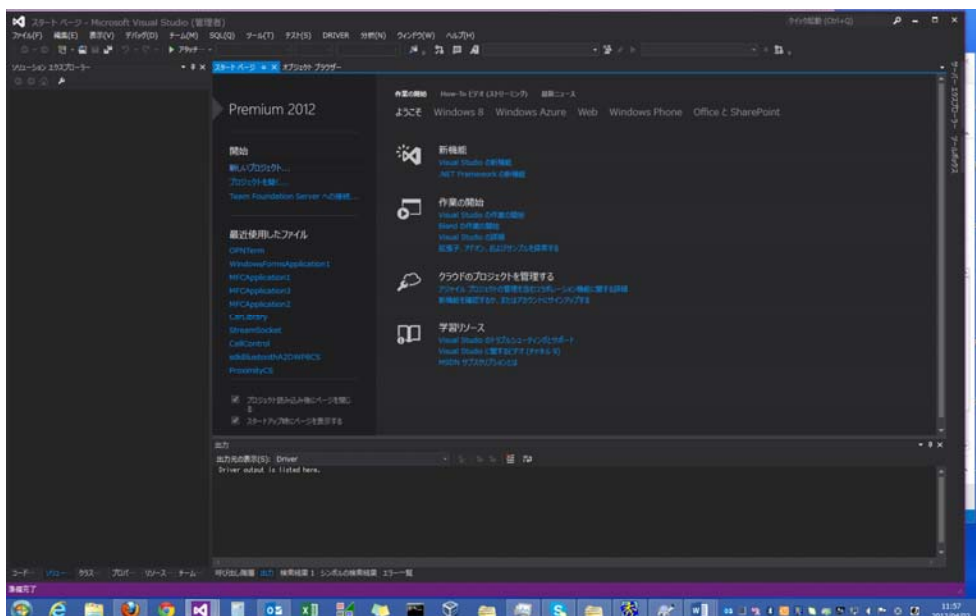
逆に COM 登録を解除するときには以下のコマンドを実行します。

(※太字&斜体部分は、インストール環境によって異なりますので、ご注意ください)

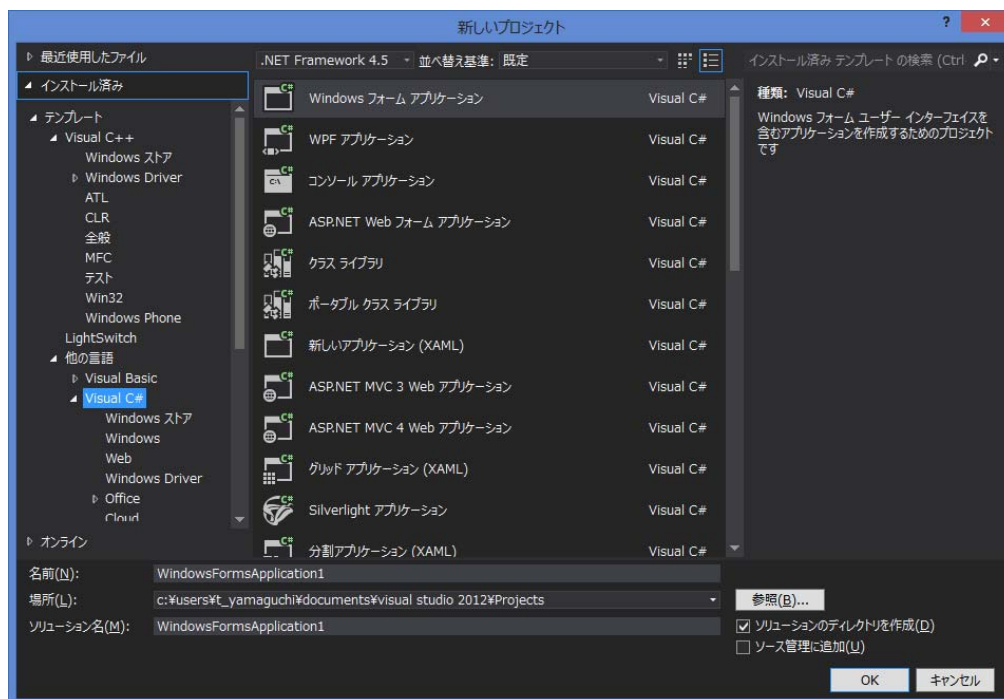
```
regsvr32 /u C:\Yopto_opn2002_sdk_win.opnterm¥OPNTerm¥Debug¥OPNTermComDll.dll
```

■ファーストAPI作成

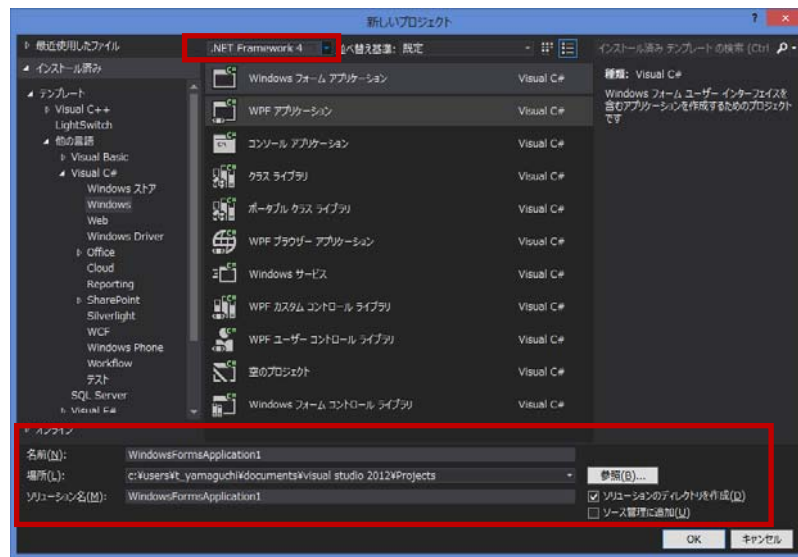
- ① Visual studio 2012 を起動します。



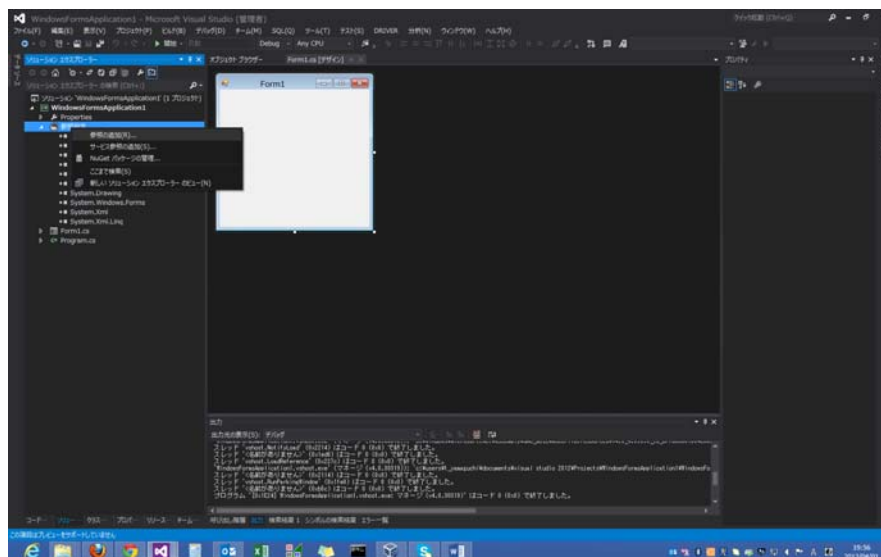
- ② 「ファイル」→「新規作成」→「プロジェクト」を選択し、「Visual C#」を選択し、「Windows フォーム アプリケーション」を選択します。



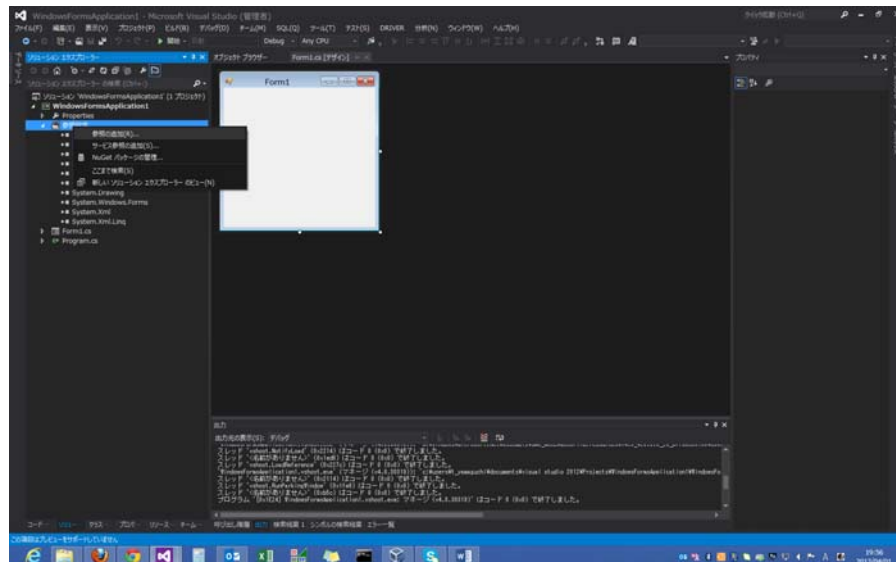
- ③ Framework4.0 に変更し、名前、場所、ソリューション名を入力します。



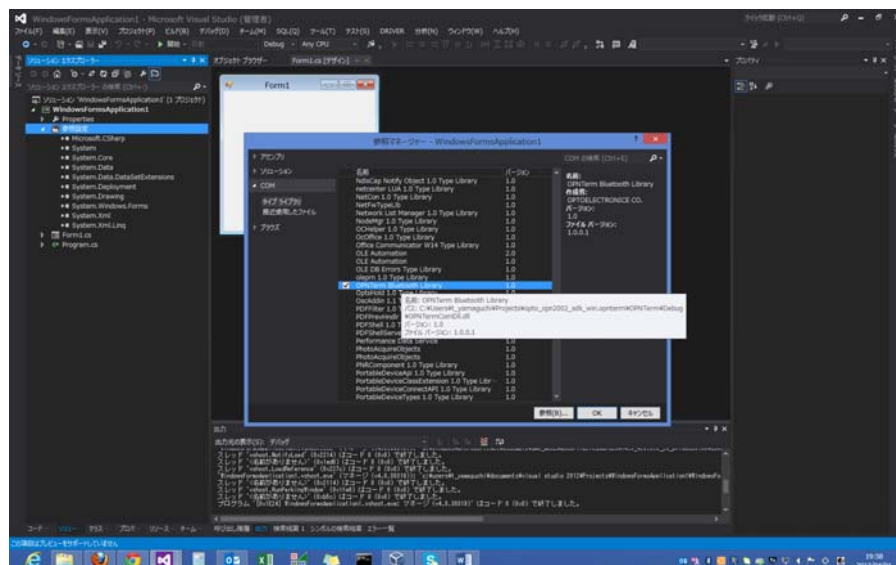
- ④ 「OK」をクリックします。



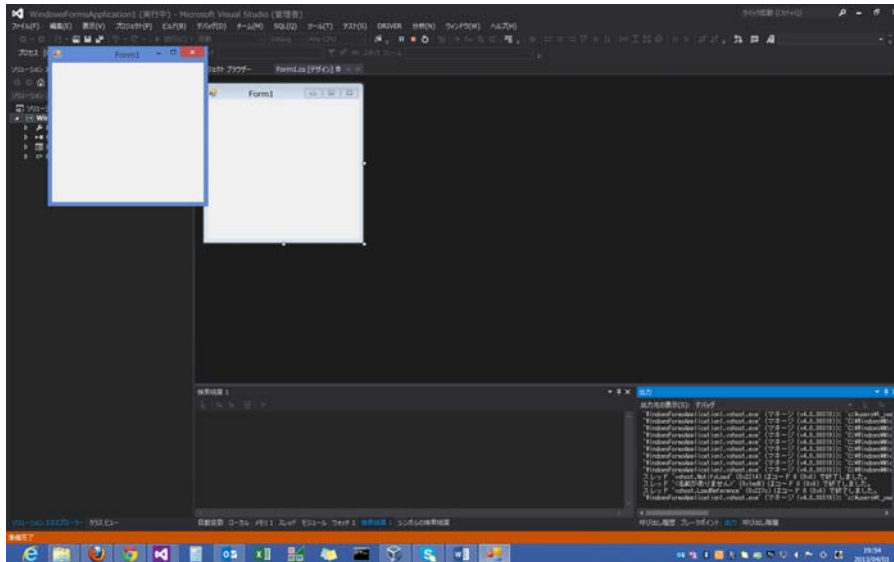
- ⑤ 「参照設定」を右クリックして「参照の追加」を選択します。（注意：事前に COM 登録しておく必要があります。『COM 登録の方法』を参照してください。）



- ⑥ 「OPNTerm Bluetooth Library」を選択しチェックして「OK」をクリックします。



⑦ ビルドして実行した結果



2. APIの利用

主な API の使用方法について、サンプルコードを用いて解説します。サンプルコードは、SDK に付属している OPNTermDesktopDotNet を使用します。

■ using OPNTermComDllLib;行を追加する

using OPNTermComDllLib;行を追加します。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Threading;

using OPNTermComDllLib;

namespace OPNTermDesktopDotNet
{
    public partial class Form1 : Form

using OPNTermComDllLib;
```

■ WOPN2002iBluetoothServiceの生成と取得

```
public partial class Form1 : Form
{
    private WOPN2002iBluetoothService kService = new WOPN2002iBluetoothService();
}
```

データコレクタとの Bluetooth 通信は、全て WOPN2002iBluetoothService クラスを介して行います。

本サンプルでは、Form1 の中で、WOPN2002BluetoothService クラスのインスタンスを生成・取得をして、フィールドに保持しています。

■ C#側がデータコレクタからの接続を待ち受ける

```
// 接続ボタン押下
private void ConnectMasterButton_Click(object sender, EventArgs e)
{
    kService.StartServer();
}
```

WOPN2002iBluetoothService#StartServer()メソッドを実行することで、データコレクタへの接続が開始されます。

データコレクタからの接続要求があり、接続が確立した場合、WOPN2002iBluetoothService#OnDidConnectDeviceEvent()メソッドが実行されます(後述)。接続が確立すると、本処理(接続を待ち受ける処理)は自動的に停止します。

■ C#側からデータコレクタに接続する

```
// 接続ボタン押下
private void ConnectSlaveButton_Click(object sender, EventArgs e)
{
    int sl = searchDeviceListBox.SelectedIndex;
    WOPNBluetoothDevice device;
    if (sl >= 0) {
        // PINコードを設定
        kService.SetPinCode("1234");
        device = (WOPNBluetoothDevice)searchdevices.GetValue(sl);
        WOPNBluetoothDevice success = kService.Connect(device, 0);
    }
}
```

WOPN2002iBluetoothService#StartServer()メソッドを実行することで、データコレクタへの接続が開始されます。

接続が成功した場合、WOPN2002iBluetoothService#OnDidConnectDeviceEvent()メソッドが実行されます（後述）。

接続が失敗した場合 WOPN2002iBluetoothService#OnDidDisconnectDeviceEvent()メソッドが実行されます（後述）。

本サンプルでは、Connect(Master)ボタンが押下されたときに、検索結果画面で選択したデバイスを取得し、そのデバイスに接続を試みています。

■ 切断する

```
// 切断ボタン押下
private void DisconnectButton_Click(object sender, EventArgs e)
{
    kService.Disconnect();
}
```

WOPN2002iBluetoothService#Disconnect()メソッドを実行することで、データコレクタとの接続が解除されます。

本サンプルでは、Disconnect ボタンが押下されたときに、本処理を実行しています。

■ データコレクタとの通信で発生したイベントを処理する

```
public partial class Form1 : Form
{
    private WOPN2002iBluetoothService kService = new WOPN2002iBluetoothService();
    private Array searchdevices; // デバイス検索結果一覧
    public Form1(){
        InitializeComponent();
        kService.OnDidGetFirmwareEvent += kService_OnDidGetFirmwareEvent;
        kService.OnReceiveBarcodeEvent += kService_OnReceiveBarcodeEvent;
        kService.OnDidConnectDeviceEvent += kService_OnDidConnectDeviceEvent;
        kService.OnDidDisconnectDeviceEvent += kService_OnDidDisconnectDeviceEvent;
        kService.OnRequestPinCodeEvent += kService_OnRequestPinCodeEvent;
    }
    void kService_OnDidDisconnectDeviceEvent(){
        MessageBox.Show("接続されました。");
    }
    void kService_OnDidConnectDeviceEvent(){
        MessageBox.Show("切断されました。");
    }
}
```

WOPN2002iBluetoothService とデータコレクタが通信をする中で、「切断」「接続完了」のイベントが発生することがあります。これらのイベントを受信して処理するためには、コンストラクタなどでイベントを予め登録しておく必要があります。

本サンプルでは、Form1 クラスが当該イベントをハンドルしています。

データ受信

```
void kService_OnReceiveBarcodeEvent(object data){
    ...
}
```

データコレクタで、バーコードを読んだときなどに発生するイベントを処理します。本サンプルでは、取得したバーコードデータをエディットボックスに追加しています。

OnDidConnectDeviceEvent: 接続完了

```
void kService_OnDidConnectDeviceEvent()  
{  
    MessageBox.Show("接続されました。");  
}
```

データコレクタとの SPP 接続が完了したとき発生する「接続完了」イベントを処理します。

本サンプルでは、MessageBox を使ってユーザーに「接続が完了したこと」を通知しています。

OnDidDisconnectDeviceEvent: 切断

```
void kService_OnDidDisconnectDeviceEvent()  
{  
    MessageBox.Show("切断されました。");  
}
```

データコレクタとの通信が切断されたとき発生する「切断」イベントを処理します。

本サンプルでは、MessageBox を使ってユーザーに「切断されたこと」を通知しています。

OnRequestPinCodeEvent: Pinコード要求

```
bool kService_OnRequestPinCodeEvent(out string pszPinCode)  
{  
    pszPinCode = "1234";  
    return true;  
}
```

データコレクタで PIN コードが必要になると OnRequestPinCodeEvent イベントが呼び出されます。送信する PIN コードを引数 pszPinCode に設定し、true を返します。返り値に false を指定すると認証要求がキャンセルされ、Connect 関数から WOPN_AUTH_FAILED が戻ります。

■ API 標準のコマンドを実行する

```
// バージョンボタン押下
private void GetFirmwareVerButton_Click(object sender, EventArgs e)
{
    kService.GetFirmwareVersion();
}
```

データコレクタに関するコマンドの中には、WOPN2002iBluetoothService クラスのメソッドとして、提供されているものがあります。

本サンプルでは、バージョンボタンが押下されたときに、WOPN2002iBluetoothService クラスの getFirmwareVersion()メソッドを実行しています。

■ 任意のコマンドを実行する

```
// ブザーON
private void BuzzerOnButton_Click(object sender, EventArgs e)
{
    kService.PushCommand("P3");
    kService.PushCommand("B");
    kService.StartSendCommands();
}
```

WOPN2002iBluetoothService クラスのメソッドとして、提供されていないメソッドを実行する場合は、WOPN2002iBluetoothService# PushCommand ()メソッドを実行します。PushCommand ()メソッドの引数に、実行したいコマンド文字列を指定した後、WOPN2002iBluetoothService#StartCommands()を実行することで、メソッドが非同期で実行されます。

本サンプルでは、Buzzer ON ボタンが押下されたときに、WOPN2002iBluetoothService クラスの PushCommand()で"B"コマンドを送信しています。

利用できるコマンドについては、「3. コマンドリファレンス」を参照してください。

弊社製品名: OPN-2002i Series
OPN-3002i Series
発行管理番号: DM-130705
管理番号: SI13025

株式会社オプトエレクトロニクス

E-Mail:sales@opto.co.jp URL:<http://www.opto.co.jp>