

*OPN-2002 SERIES / OPN-3002 SERIES*

*DATA collector*

---

OPN SDK (Lib)

APIリファレンスマニュアル

## 改版履歴

資料管理番号 : SI13028

発行管理番号 : DM-130708

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

| 版       | 日付        | 変更箇所 | 内容   |
|---------|-----------|------|------|
| Rev 1.0 | 2013/6/21 | —    | 新規作成 |
|         |           |      |      |
|         |           |      |      |
|         |           |      |      |
|         |           |      |      |
|         |           |      |      |
|         |           |      |      |
|         |           |      |      |
|         |           |      |      |

### ・おことわり

- 本書記載内容は、著作権による所有権を有する場合があります。  
本書の一部または全部を株式会社オプトエレクトロニクスから事前に文書による承諾を得ることなく、いかなる方法によっても無断で複写、写真複写、複製、翻訳することを禁じます。
- 本書の内容に関しては、将来仕様改良などにより、予告無く変更することがあります。  
あらかじめご了承ください。
- 本書は間違いや、誤記の無いように注意して編集しております。  
万一、誤記や記載漏れがあった場合でも、それに起因するお客様の直接、間接の損害、不利益につきましては、責任を負いかねますので、あらかじめご了承ください。
- 製品により、非対応 API や画面構成等が異なる場合があります。

# 目次

|   |    |
|---|----|
| 目次 .....                                    | 1  |
| クラス .....                                   | 3  |
| クラス IOPN2002iBluetoothServiceDelegate ..... | 3  |
| Public メソッド .....                           | 3  |
| 説明 .....                                    | 4  |
| 関数 .....                                    | 4  |
| クラス IOPNBluetoothServiceDelegate .....      | 8  |
| Public メソッド .....                           | 8  |
| 説明 .....                                    | 8  |
| 関数 .....                                    | 9  |
| クラス OPN2002iBluetoothService .....          | 11 |
| Public メソッド .....                           | 11 |
| Static Public メソッド .....                    | 12 |
| Protected メソッド .....                        | 12 |
| コマンド .....                                  | 12 |
| 説明 .....                                    | 12 |
| コンストラクタとデストラクタ .....                        | 13 |
| 関数 .....                                    | 13 |
| クラス OPNBluetoothDevice .....                | 19 |
| Public メソッド .....                           | 19 |
| 説明 .....                                    | 19 |
| コンストラクタとデストラクタ .....                        | 19 |
| 関数 .....                                    | 20 |
| クラス OPNBluetoothHostInfo .....              | 22 |
| Public メソッド .....                           | 22 |
| 説明 .....                                    | 22 |
| コンストラクタとデストラクタ .....                        | 22 |
| 関数 .....                                    | 23 |
| クラス OPNBluetoothService .....               | 24 |
| Public メソッド .....                           | 24 |
| Protected メソッド .....                        | 25 |
| 説明 .....                                    | 26 |
| コンストラクタとデストラクタ .....                        | 26 |
| 関数 .....                                    | 26 |
| クラス OPNSettingChars .....                   | 34 |
| Public メソッド .....                           | 34 |
| Static Public メソッド .....                    | 34 |
| Protected メソッド .....                        | 34 |
| 説明 .....                                    | 34 |
| コンストラクタとデストラクタ .....                        | 35 |
| 関数 .....                                    | 35 |
| クラス OPNSettings .....                       | 37 |
| Public メソッド .....                           | 37 |

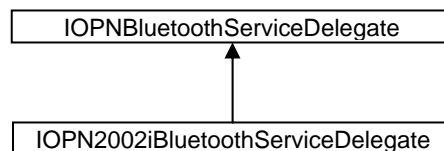
|   |    |
|---|----|
| 説明 .....                                | 46 |
| コンストラクタとデストラクタ .....                    | 46 |
| 関数 .....                                | 46 |
| 構造体 <i>SettingChar</i> .....            | 88 |
| <i>Public</i> 変数 .....                  | 88 |
| 説明 .....                                | 88 |
| 変数 .....                                | 88 |
| ファイル .....                              | 89 |
| <i>OPN2002iBluetoothService.h</i> ..... | 89 |
| 構成 .....                                | 89 |
| 列挙型 .....                               | 89 |
| 列挙型 .....                               | 89 |
| <i>OPNBluetoothDevice.h</i> .....       | 91 |
| 構成 .....                                | 91 |
| <i>OPNBluetoothHostInfo.h</i> .....     | 92 |
| 構成 .....                                | 92 |
| <i>OPNBluetoothService.h</i> .....      | 93 |
| 構成 .....                                | 93 |
| 列挙型 .....                               | 93 |
| 列挙型 .....                               | 93 |
| <i>OPNCommand.h</i> .....               | 94 |
| <i>OPNErrors.h</i> .....                | 95 |
| 列挙型 .....                               | 95 |
| 列挙型 .....                               | 95 |
| <i>OPNSettingChars.h</i> .....          | 96 |
| 構成 .....                                | 96 |
| <i>SettingChar</i> .....                | 96 |
| マクロ定義 .....                             | 96 |
| <i>OPNSettings.h</i> .....              | 97 |
| 構成 .....                                | 97 |

# クラス

## クラス IOPN2002iBluetoothServiceDelegate

OPN3002iBluetoothServiceに対するデリゲータクラス

IOPN2002iBluetoothServiceDelegateに対する継承グラフ



### Public メソッド

- virtual void **OnDidGetFirmware** (OPNBluetoothService \*pService, BYTE \*pData)  
*GetFirmware* の結果を受信したことを通知します。
- virtual void **OnDidReadBarcode** (OPNBluetoothService \*pService, BYTE \*pData)  
*ReadBarcode* の結果を受信したことを通知します。
- virtual void **OnDidGetStoredBarcode** (OPNBluetoothService \*pService, BYTE \*pData)  
*GetStoredBarcode* の結果を受信したことを通知します。
- virtual void **OnDidSaveSetting** (OPNBluetoothService \*pService, BYTE \*pData)  
*SaveSetting* の結果を受信したことを通知します。
- virtual void **OnDidSetDateAndTime** (OPNBluetoothService \*pService, BYTE \*pData)  
*SetDateAndTime* の結果を受信したことを通知します。
- virtual void **OnDidGetDateAndTime** (OPNBluetoothService \*pService, BYTE \*pData)  
*GetDateAndTime* の結果を受信したことを通知します。
- virtual void **OnDidGetBatteryLevel** (OPNBluetoothService \*pService, BYTE \*pData)  
*GetBatteryLevel* の結果を受信したことを通知します。
- virtual void **OnDidGetSetting** (OPNBluetoothService \*pService, BYTE \*pData)  
*GetSetting* の結果を受信したことを通知します。
- virtual void **OnDidInitializeSetting** (OPNBluetoothService \*pService, BYTE \*pData)  
*InitializeSetting* の結果を受信したことを通知します。
- virtual void **OnDidInitializeSettingBluetooth** (OPNBluetoothService \*pService, BYTE \*pData)  
*InitializeSettingBluetooth* の結果を受信したことを通知します。
- virtual void **OnDidEraseStoredBarcode** (OPNBluetoothService \*pService, BYTE \*pData)  
*EraseStoredBarcode* の結果を受信したことを通知します。

- `virtual void OnDidVibratorOn (OPNBluetoothService *pService, BYTE *pData)`  
VibratorOnの結果を受信したことを通知します。
- `virtual void OnDidBeepBuzzer (OPNBluetoothService *pService, BYTE *pData)`  
BeepBuzzerの結果を受信したことを通知します。
- `virtual void OnCompleteCustomCommand (OPNBluetoothService *pService, BYTE *pData)`  
全てのコマンドの送信が完了したことを通知します。

---

## 説明

OPN3002iBluetoothServiceに対するデリゲータクラス

このクラスを継承して各メソッドをオーバーライドすることで、

OPN2002iBluetoothServiceクラスのイベントを受け取ることができます。継承したクラスのインスタンスをOPN3002iBluetoothServiceのSetDelegateメソッドで設定してください。SetDelegateで設定したデリゲータは上書きされます。他のクラスでSetDelegateした後に再びイベントを受け取れるようにするには、SetDelegateをもう一度実行してください。

---

## 関数

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnCompleteCustomCommand  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

全てのコマンドの送信が完了したことを通知します。

引数:

|    |          |                            |
|----|----------|----------------------------|
| in | pService | OPNBluetoothServiceのインスタンス |
| in | pData    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidBeepBuzzer  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

BeepBuzzerの結果を受信したことを通知します。

引数:

|    |          |                            |
|----|----------|----------------------------|
| in | pService | OPNBluetoothServiceのインスタンス |
| in | pData    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidEraseStoredBarcode  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

EraseStoredBarcodeの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidGetBatteryLevel  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

GetBatteryLevelの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidGetDateAndTime  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

GetDateAndTimeの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidGetFirmware  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

GetFirmwareの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidGetSetting  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

GetSettingの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidGetStoredBarcode  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

GetStoredBarcodeの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidInitializeSetting  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

InitializeSettingの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void  
IOPN2002iBluetoothServiceDelegate::OnDidInitializeSettingBluetooth  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

InitializeSettingBluetoothの結果を受信したことを通知します。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |



---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidReadBarcode  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

ReadBarcodeの結果を受信したことを通知します。

引数:

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidSaveSetting  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

SaveSettingの結果を受信したことを通知します。

引数:

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidSetDateAndTime  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

SetDateAndTimeの結果を受信したことを通知します。

引数:

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual void IOPN2002iBluetoothServiceDelegate::OnDidVibratorOn  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

VibratorOnの結果を受信したことを通知します。

引数:

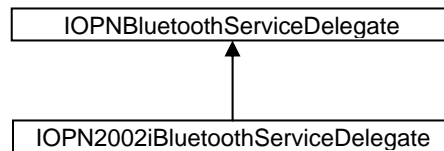
|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

## クラス IOPNBluetoothServiceDelegate

OPNBluetoothServiceに対するデリゲータクラス

IOPNBluetoothServiceDelegateに対する継承グラフ



### Public メソッド

- virtual VOID **OnDidConnectDevice** (OPNBluetoothService \*pService, OPNBluetoothDevice \*pDevice)  
*OPNデバイスが接続されると呼び出されます*
- virtual VOID **OnDidDisconnectDevice** (OPNBluetoothService \*pService, OPNBluetoothDevice \*pDevice)  
*OPNデバイスが切断されると呼び出されます*
- virtual VOID **OnDidDiscoveredDevice** (OPNBluetoothService \*pService, OPNBluetoothDevice pDevice)  
*DoDiscoveryAsync メソッドでデバイスを1台発見すると呼び出されます*
- virtual VOID **OnCompleteDiscovery** (OPNBluetoothService \*pService, INT iDeviceCount)  
*DoDiscoveryAsync メソッドでデバイス探索が完了すると呼び出されます*
- virtual VOID **OnReceiveBarcode** (OPNBluetoothService \*pService, BYTE \*pData)  
*バーコードを受信すると呼び出されます。*
- virtual BOOL **OnRequestPinCode** (OPNBluetoothService \*pService, LPSTR lpszPinCode)  
*認証中にPINコードが必要となる場合に呼び出されます。*

---

### 説明

OPNBluetoothServiceに対するデリゲータクラス

このクラスを継承して各メソッドをオーバーライドすることで、

OPNBluetoothServiceクラスのイベントを受け取ることができます。 継承したクラスのインスタンスをOPNBuetoothServiceのSetDelegateメソッドで 設定してください。 SetDelegateで設定したデリゲータは上書きされます。 他のクラスでSetDelegateした後に再びイベントを受け取れるようにするには、 SetDelegateをもう一度実行してください。

---

## 関数

---

```
virtual VOID IOPNBluetoothServiceDelegate::OnCompleteDiscovery  
(OPNBluetoothService * pService, INT iDeviceCount)[inline], [virtual]
```

DoDiscoveryAsync メソッドでデバイス探索が完了すると呼び出されます

### 引数:

|    |                     |                                   |
|----|---------------------|-----------------------------------|
| in | <i>pService</i>     | 接続時のOPNBluetoothServiceクラスのインスタンス |
| in | <i>iDeviceCount</i> | 発見した端末の台数                         |

---

```
virtual VOID IOPNBluetoothServiceDelegate::OnDidConnectDevice  
(OPNBluetoothService * pService, OPNBluetoothDevice * pDevice)[inline],  
[virtual]
```

OPNデバイスが接続されると呼び出されます

### 引数:

|    |                 |                                   |
|----|-----------------|-----------------------------------|
| in | <i>pService</i> | 接続時のOPNBluetoothServiceクラスのインスタンス |
| in | <i>pDevice</i>  | 接続されたデバイスのインスタンス                  |

---

```
virtual VOID IOPNBluetoothServiceDelegate::OnDidDisconnectDevice  
(OPNBluetoothService * pService, OPNBluetoothDevice * pDevice)[inline],  
[virtual]
```

OPNデバイスが切断されると呼び出されます

### 引数:

|    |                 |                                   |
|----|-----------------|-----------------------------------|
| in | <i>pService</i> | 接続時のOPNBluetoothServiceクラスのインスタンス |
| in | <i>pDevice</i>  | 接続されたデバイスのインスタンス                  |

---

```
virtual VOID IOPNBluetoothServiceDelegate::OnDidDiscoveredDevice  
(OPNBluetoothService * pService, OPNBluetoothDevice pDevice)[inline],  
[virtual]
```

DoDiscoveryAsync メソッドでデバイスを1台発見すると呼び出されます

**引数:**

|    |                 |                                   |
|----|-----------------|-----------------------------------|
| in | <i>pService</i> | 接続時のOPNBluetoothServiceクラスのインスタンス |
| in | <i>pDevice</i>  | 発見したデバイスのインスタンス                   |

---

```
virtual VOID IOPNBluetoothServiceDelegate::OnReceiveBarcode  
(OPNBluetoothService * pService, BYTE * pData)[inline], [virtual]
```

バーコードを受信すると呼び出されます。

**引数:**

|    |                 |                            |
|----|-----------------|----------------------------|
| in | <i>pService</i> | OPNBluetoothServiceのインスタンス |
| in | <i>pData</i>    | 受信データ                      |

---

```
virtual BOOL IOPNBluetoothServiceDelegate::OnRequestPinCode  
(OPNBluetoothService * pService, LPSTR lpSzPinCode)[inline], [virtual]
```

認証中にPINコードが必要となる場合に呼び出されます。

**引数:**

|     |                    |                            |
|-----|--------------------|----------------------------|
| in  | <i>pService</i>    | OPNBluetoothServiceのインスタンス |
| out | <i>lpSzPinCode</i> | PINコード文字列を設定します            |

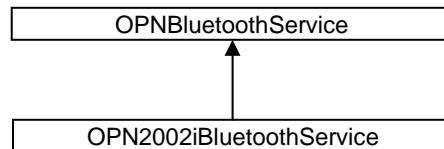
**戻り値:**

成功時TRUE、失敗時はFALSEを指定します。

---

## クラス OPN2002iBluetoothService

OPN2002/OPN2002i、OPN3002/OPN3002iへのAPIを提供するシングルトンクラス  
OPN2002iBluetoothServiceに対する継承グラフ



### Public メソッド

- **APIOPNResult InitializeSetting ( )**  
データコレクタの全設定を初期化します。
- **OPNResult InitializeSettingBluetooth ( )**  
データコレクタのBluetooth設定を初期化します。
- **OPNResult SaveSetting ( )**  
データコレクタの揮発メモリに保存されている設定を、不揮発メモリに保存します。
- **OPNResult ReadBarcode ( )**  
データコレクタでバーコード読み取りを開始します。
- **OPNResult GetStoredBarcodeAll ( )**  
データコレクタのメモリデータを出力します。
- **OPNResult EraseStoredBarcodeAll ( )**  
データコレクタのメモリデータをクリアします。
- **OPNResult SetDateAndTime (const int y, const int M, const int d, const int h, const int m, const int s)**  
データコレクタの日時を設定します。
- **OPNResult GetDateAndTime ( )**  
データコレクタの日時を取得します。
- **OPNResult GetBatteryLevel ( )**  
データコレクタの電池電圧を取得します。
- **OPNResult VibratorOn ( )**  
データコレクタのバイブレーターを振動させます。
- **OPNResult BeepBuzzer ( )**  
データコレクタのブザーを起動させます。
- **OPNResult CancelAllCommand ( )**  
未送信キューのコマンドをすべてクリアします。
- **OPNResult GetFirmwareVersion ( )**  
バージョンを返します。
- **OPNResult GetSetting ( )**  
設定値を返します。

## Static Public メソッド

- **static OPN2002iBluetoothService & GetInstance (void)**  
アプリで唯一のインスタンスを取得します。

## Protected メソッド

- **OPN2002iBluetoothService ()**  
コンストラクタ
- **~OPN2002iBluetoothService ()**  
デストラクタ  
オーバーロード親クラスOPNBluetoothServiceクラスのオーバーロードです。
- **BOOL OnWillReadData (BTHSppConnection \*pConnection, BTHReadCondition \*pReadCondition)**  
データを受信し読込を開始する直前で呼び出されるインターフェイスです。
- **BOOL OnDidReadData (BTHSppConnection \*pConnection, BYTE \*readData, INT readBytes)**  
データの読込が完了した直後で呼び出されるインターフェイスです。

## コマンド

コマンドのデータのキューを操作、送信します。

- **void PushCommand (const char \*command)**  
コマンドをキューに追加します。
- **void PushCommand (const char \*command, OPNApiType type, const int timeOut)**  
コマンドをキューに追加します。
- **OPNResult StartSendCommands ()**  
コマンドの送信を開始します。

---

## 説明

OPN2002/OPN2002i、OPN3002/OPN3002iへのAPIを提供するシングルトンクラス

OPN2002iBluetoothServiceクラスはOPNBluetoothServiceクラスを継承して、OPN2002/OPN2002i、OPN3002/OPN3002iへのAPIを提供します。プロセス内で唯一のインスタンスです (GoFシングルトンパターン)。OPN2002iBluetoothServiceクラスは new を使ってインスタンス化できません。インスタンスを取得するにはGetInstanceメソッドを使用して参照ポインタを取得してください。インスタンスは複製はできません。プロセス終了時に自動的に破棄されます。マルチスレッド安全性はありません。

---

## コンストラクタとデストラクタ

---

**OPN2002iBluetoothService::OPN2002iBluetoothService ()[protected]**

コンストラクタ

---

**OPN2002iBluetoothService::~~OPN2002iBluetoothService ()[protected]**

デストラクタ

---

## 関数

---

**OPNResult OPN2002iBluetoothService::BeepBuzzer ()**

データコレクタのブザーを起動させます。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidBeepBuzzerが呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::CancelAllCommand ()**

未送信キューのコマンドをすべてクリアします。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::EraseStoredBarcodeAll ()**

データコレクタのメモリデータをクリアします。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidEraseStoredBarcodeAll  
が呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::GetBatteryLevel ()**

データコレクタの電池電圧を取得します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidGetBatteryLevel  
が呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::GetDateAndTime ()**

データコレクタの日時を取得します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidGetDateAndTime  
が呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::GetFirmwareVersion ()**

バージョンを返します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidGetFirmwareVersion  
が呼び出されます。

**戻り値:**

結果コード

---

**static OPN2002iBluetoothService& OPN2002iBluetoothService::GetInstance  
(void )[static]**

アプリで唯一のインスタンスを取得します。

**戻り値:**

OPN2002iBluetoothServiceのインスタンス



---

**OPNResult OPN2002iBluetoothService::GetSetting ()**

設定値を返します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidGetGetSetting  
が呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::GetStoredBarcodeAll ()**

データコレクタのメモリデータを出力します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidGetStoredBarcodeAll  
が呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::InitializeSetting ()**

データコレクタの全設定を初期化します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidInitializeSetting  
が呼び出されます。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::InitializeSettingBluetooth ()**

データコレクタのBluetooth設定を初期化します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidInitializeSettingBluetooth  
が呼び出されます。

**戻り値:**

結果コード

---

```
BOOL OPN2002iBluetoothService::OnDidReadData (BTHSppConnection * pConnection,
BYTE * readData, INT readBytes)[protected], [virtual]
```

データの読込が完了した直後で呼び出されるインターフェイスです。

**引数:**

|    |                    |                                  |
|----|--------------------|----------------------------------|
| in | <i>pConnection</i> | BTHSppConnectionクラスのインスタンスへのポインタ |
| in | <i>readData</i>    | 読み込んだデータ                         |
| in | <i>readBytes</i>   | 読み込んだデータのバイト数                    |

**戻り値:**

成功時にTRUE、失敗時はFALSEを返すようにします。

OPNBluetoothService (p.30)を実装しています。

---

```
BOOL OPN2002iBluetoothService::OnWillReadData (BTHSppConnection *
pConnection, BTHReadCondition * pReadCondition)[protected], [virtual]
```

データを受信し読込を開始する直前で呼び出されるインターフェイスです。

**引数:**

|     |                       |                                  |
|-----|-----------------------|----------------------------------|
| in  | <i>pConnection</i>    | BTHSppConnectionクラスのインスタンスへのポインタ |
| out | <i>pReadCondition</i> | 読込条件を指定します。                      |

**戻り値:**

成功時にTRUE、失敗時はFALSEを返すようにします。

OPNBluetoothService (p.31)を実装しています。

---

```
void OPN2002iBluetoothService::PushCommand (const char * command)
```

コマンドをキューに追加します。

独自のコマンドを送信したい場合に使います。 内部的に **PushCommand(const char\*, OPNApiType, const int)** を呼び出します。 API のタイプは OPNApiCustomCommandに設定され、タイムアウトは5秒に設定されます。

**引数:**

|    |                |            |
|----|----------------|------------|
| in | <i>command</i> | BYTE型のコマンド |
|----|----------------|------------|

---

```
void OPN2002iBluetoothService::PushCommand (const char * command, OPNApiType
type, const int timeOut)
```

コマンドをキューに追加します。

**引数:**

|    |                |  |
|----|----------------|--|
| in | <i>command</i> | BYTE型のコマンド                                     |
| in | <i>type</i>    | OPNApiTypeを指定します。通常はOPNApiCustomCommandを指定します。 |
| in | <i>timeOut</i> | タイムアウト時間を秒で指定します。                              |

---

```
OPNResult OPN2002iBluetoothService::ReadBarcode ()
```

データコレクタでバーコード読み取りを開始します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidReadBarcode

が呼び出されます。

**戻り値:**

結果コード

---

```
OPNResult OPN2002iBluetoothService::SaveSetting ()
```

データコレクタの揮発メモリに保存されている設定を、不揮発メモリに保存します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidSaveSetting

が呼び出されます。

**戻り値:**

結果コード

---

```
OPNResult OPN2002iBluetoothService::SetDateAndTime (const int y, const int
M, const int d, const int h, const int m, const int s)
```

データコレクタの日時を設定します。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidSetDateAndTime

が呼び出されます。

**引数:**

|    |          |       |
|----|----------|-------|
| in | <i>y</i> | 年（西暦） |
| in | <i>M</i> | 月     |
| in | <i>d</i> | 日     |
| in | <i>h</i> | 時     |
| in | <i>m</i> | 分     |
| in | <i>s</i> | 秒     |

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::StartSendCommands ()**

コマンドの送信を開始します。

PushCommandメソッドで追加したコマンドをキューから1つずつ送信を開始します。

**戻り値:**

結果コード

---

**OPNResult OPN2002iBluetoothService::VibratorOn ()**

データコレクタのバイブレーターを振動させます。

コマンドの送信が正常に完了した場合、

IOPN2002iBluetoothServiceDelegate::OnDidVibratorOn

が呼び出されます。

**戻り値:**

結果コード

---

## クラス OPNBluetoothDevice

Bluetoothデバイスクラス

### Public メソッド

- **OPNBluetoothDevice** ()  
コンストラクタ
- **OPNBluetoothDevice** (BLUETOOTH\_DEVICE\_INFO bthDevice)  
コンストラクタ (API内部からのみ使用)
- **~OPNBluetoothDevice** ()  
デストラクタ
- **OPNBluetoothDevice** (const **OPNBluetoothDevice** &instance)  
コピーコンストラクタ
- **OPNBluetoothDevice & operator=** (const **OPNBluetoothDevice** &instance)  
代入演算子によるコピー
- **OPNResult GetName** (LPSTR lpszName, INT nLen)  
バーコードコレクタのデバイス名を取得します。
- **OPNResult GetAddress** (LPSTR lpszBthAddr, INT nLen)  
バーコードコレクタのアドレスを取得します。
- **ULONGLONG GetUAddress** ()  
バーコードコレクタのアドレスを取得します。

---

### 説明

Bluetoothデバイスクラス

デバイスの情報を保持します。

---

### コンストラクタとデストラクタ

---

**OPNBluetoothDevice::OPNBluetoothDevice** ()

コンストラクタ

---

**OPNBluetoothDevice::OPNBluetoothDevice** (BLUETOOTH\_DEVICE\_INFO *bthDevice*)

コンストラクタ (API内部からのみ使用)

**引数:**

|    |                  |                          |
|----|------------------|--------------------------|
| in | <i>bthDevice</i> | BLUETOOTH_DEVICE_INFOの実体 |
|----|------------------|--------------------------|

---

**OPNBluetoothDevice::~~OPNBluetoothDevice ()**

デストラクタ

---

**OPNBluetoothDevice::OPNBluetoothDevice (const OPNBluetoothDevice & *instance*)**

コピーコンストラクタ

---

**関数**

---

**OPNResult OPNBluetoothDevice::GetAddress (LPSTR *lpszBthAddr*, INT *nLen*)**

バーコードコレクタのアドレスを取得します。

**引数:**

|     |                    |                          |
|-----|--------------------|--------------------------|
| out | <i>lpszBthAddr</i> | アドレス                     |
| in  | <i>nLen</i>        | <i>lpszBthAddr</i> のバイト長 |

**戻り値:**

TRUE : 取得成功/FALSE : 取得失敗

---

**OPNResult OPNBluetoothDevice::GetName (LPSTR *lpszName*, INT *nLen*)**

バーコードコレクタのデバイス名を取得します。

**引数:**

|     |                 |                       |
|-----|-----------------|-----------------------|
| out | <i>lpszName</i> | デバイス名                 |
| in  | <i>nLen</i>     | <i>lpszName</i> のバイト長 |

**戻り値:**

TRUE : 取得成功/FALSE : 取得失敗

---

**ULONGLONG OPNBluetoothDevice::GetUAddress ( )**

バーコードコレクタのアドレスを取得します。

**戻り値:**

バーコードコレクタのアドレス

---

**OPNBluetoothDevice& OPNBluetoothDevice::operator= (const OPNBluetoothDevice  
& *instance*)**

代入演算子によるコピー

---

## クラス OPNBluetoothHostInfo

ホストPCに関する情報を保持します。

### Public メソッド

- **OPNBluetoothHostInfo** ()  
コンストラクタ
- **OPNBluetoothHostInfo** (BLUETOOTH\_RADIO\_INFO bthRadioInfo)  
コンストラクタ (API内部からのみ使用)
- **OPNBluetoothHostInfo** (const OPNBluetoothHostInfo &instance)  
コピーコンストラクタ
- **~OPNBluetoothHostInfo** ()  
デストラクタ
- **OPNBluetoothHostInfo & operator=** (const OPNBluetoothHostInfo &instance)  
代入演算子によるコピー
- **OPNResult GetAddress** (LPSTR lpszBthAddr, INT nLen)  
Bluetoothのアドレスを取得します。

---

### 説明

ホストPCに関する情報を保持します。

---

### コンストラクタとデストラクタ

---

**OPNBluetoothHostInfo::OPNBluetoothHostInfo** ()

コンストラクタ

---

**OPNBluetoothHostInfo::OPNBluetoothHostInfo** (BLUETOOTH\_RADIO\_INFO  
*bthRadioInfo*)

コンストラクタ (API内部からのみ使用)

### 引数:

|    |                     |                         |
|----|---------------------|-------------------------|
| in | <i>bthRadioInfo</i> | BLUETOOTH_RADIO_INFOの実体 |
|----|---------------------|-------------------------|



---

```
OPNBluetoothHostInfo::OPNBluetoothHostInfo (const OPNBluetoothHostInfo &
instance)
```

コピーコンストラクタ

---

```
OPNBluetoothHostInfo::~~OPNBluetoothHostInfo ()
```

デストラクタ

---

## 関数

---

```
OPNResult OPNBluetoothHostInfo::GetAddress (LPSTR lpszBthAddr, INT nLen)
```

Bluetoothのアドレスを取得します。

### 戻り値:

Bluetoothアドレス (XX:XX:XX:XX:XX:XX形式)

---

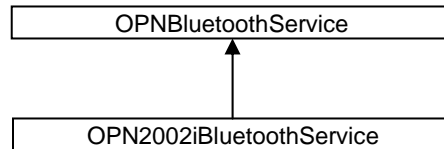
```
OPNBluetoothHostInfo& OPNBluetoothHostInfo::operator= (const
OPNBluetoothHostInfo & instance)
```

代入演算子によるコピー

---

## クラス OPNBluetoothService

OPNデバイスとの接続、データ送受信を管理する抽象クラス  
OPNBluetoothServiceに対する継承グラフ



### Public メソッド

- **VOID SetDelegate** (IOPNBluetoothServiceDelegate \*pDelegate)  
OPNBluetoothServiceのデリゲータを設定します。
  - **VOID SetEnableReadBarcode** (BOOL bEnabled)  
ReadBarcodeの許可・禁止を設定します。
  - **BOOL IsEnableReadBarcode** ()  
現在のReadBarcodeが許可か禁止かを返します。
  - **OPNResult GetConnectedBluetoothDevice** (OPNBluetoothDevice \*pOpnDevice)  
接続済みのデバイス情報を取得します。
  - **OPNResult GetBluetoothHostInfos** (OPNBluetoothHostInfo pHosts[], INT &nCount)  
Bluetoothのホスト情報を取得します。
  - **virtual BOOL OnWillReadData** (BTHSppConnection \*pConnection, BTHReadCondition \*pReadCondition)=0  
データを受信し読込を開始する直前で呼び出されるインターフェイスです。
  - **virtual BOOL OnDidReadData** (BTHSppConnection \*pConnection, BYTE \*readData, INT readBytes)=0  
データの読込が完了した直後で呼び出されるインターフェイスです。  
データの送受信データを送受信するためのメソッドです。
    - **LONG WriteACK** ()  
ACKを送信します。
    - **LONG WriteNAK** ()  
NAKを送信します。
    - **LONG WriteData** (BYTE \*writeData, INT toWriteBytes, INT nTimeout=5000)  
バーコードコレクターにバイナリデータを送信します。
    - **LONG WriteDataReadBarcodeResponseNak** (BYTE \*writeData, INT toWriteBytes, INT nTimeout=5000)  
バーコードコレクターにバイナリデータを送信します。 通信時にReadBarcodeを受信した場合、Nakを返します。
- マスター/スレーブ接続サーバの開始・終了、クライアントの接続・切断メソッドです。
- **OPNResult Connect** (OPNBluetoothDevice opnDevice, BOOL bSecure)  
指定したデバイスにクライアントとして接続します。

- **OPNResult Disconnect ()**  
デバイスを切断します。
- **OPNResult StartServer ()**  
サーバーを開始します。
- **OPNResult StopServer ()**  
サーバーを停止します。
- **BOOL IsConnected ()**  
接続済みの場合はTRUEを返します。
- **BOOL IsServerConnected ()**  
サーバとして接続している場合はTRUEを返します。
- **OPNAutoConnectStatus GetAutoConnectStatus ()**  
スレーブ自動接続が実行中の場合はTRUEを返します。
- **OPNResult GetAutoConnectDevice (OPNBluetoothDevice \*pOpnDevive)**  
自動接続対象のデバイスを返します。
- **OPNResult StartAutoConnect ()**  
スレーブの自動接続を開始します。
- **OPNResult StopAutoConnect ()**  
スレーブの自動接続を終了します。

BluetoothBluetoothデバイス関連のメソッドです。

- **OPNResult IsBluetoothEnabled ()**  
PCがBluetoothをサポートしているかどうかを取得します。
- **BOOL IsDiscoverable ()**  
他のBluetoothデバイスから検索可能な状態かどうか取得します。
- **OPNResult GetPairedDevices (OPNBluetoothDevice devices[], INT &nCount)**  
ペアリング済みのBluetoothデバイスを取得します。
- **OPNResult DoDiscovery (OPNBluetoothDevice devices[], INT &nCount)**  
周辺のBluetoothデバイスを検索します。
- **OPNResult DoDiscoveryAsync ()**  
周辺のBluetoothデバイスを非同期で検索します。
- **VOID CancelDiscovery ()**  
DoDiscoveryAsyncメソッドによるデバイス検索をキャンセルします。

## Protected メソッド

- **OPNBluetoothService ()**  
コンストラクタ
- **OPNBluetoothService (const OPNBluetoothService &instance)**  
コピーコンストラクタ
- **~OPNBluetoothService ()**  
デストラクタ
- **IOPNBluetoothServiceDelegate \* GetDelegate ()**

デリゲータを取得します。

---

## 説明

OPNデバイスとの接続、データ送受信を管理する抽象クラス

OPNBluetoothServiceクラスは、OPN2002x, OPN3002xとのペアリング、スレーブ/マスター接続、データ送受信の機能を搭載した抽象クラスです。継承先でOnWillReadData/OnDidReadDataを実装する必要があります。

---

## コンストラクタとデストラクタ

---

```
OPNBluetoothService::OPNBluetoothService ()[protected]
```

コンストラクタ

---

```
OPNBluetoothService::OPNBluetoothService (const OPNBluetoothService &instance)[protected]
```

コピーコンストラクタ

引数:

|    |          |      |
|----|----------|------|
| in | instance | コピー元 |
|----|----------|------|

---

```
OPNBluetoothService::~OPNBluetoothService ()[protected]
```

デストラクタ

---

## 関数

---

```
VOID OPNBluetoothService::CancelDiscovery ()
```

DoDiscoveryAsyncメソッドによるデバイス検索をキャンセルします。

---

**OPNResult OPNBluetoothService::Connect (OPNBluetoothDevice *opnDevice*, BOOL *bSecure*)**

指定したデバイスにクライアントとして接続します。

**引数:**

|    |                  |  |
|----|------------------|--|
| in | <i>opnDevice</i> | 接続対象のOPNBluetoothDeviceを指定します。         |
| in | <i>bSecure</i>   | セキュア接続の場合TRUE、 インセキュア接続の場合FALSEを指定します。 |

**戻り値:**

エラーを返します。成功の場合は OPN\_SUCCESS を返します。

---

**OPNResult OPNBluetoothService::Disconnect ()**

デバイスを切断します。

**戻り値:**

エラーを返します。成功の場合は OPN\_SUCCESS を返します。

---

**OPNResult OPNBluetoothService::DoDiscovery (OPNBluetoothDevice *devices*[], INT & *nCount*)**

周辺のBluetoothデバイスを検索します。

**引数:**

|     |                |             |
|-----|----------------|-------------|
| out | <i>devices</i> | 取得したデバイスの配列 |
| out | <i>nCount</i>  | 取得したデバイス数   |

**戻り値:**

成功であれば OPN\_SUCCESS を返す。

---

**OPNResult OPNBluetoothService::DoDiscoveryAsync ()**

周辺のBluetoothデバイスを非同期で検索します。

このメソッドは非同期で処理が行われます。 デバイスを1台発見すると、OnDidDiscoveredDevice が呼び出されます。 処理が完了すると、OnCompleteDiscovery が呼び出されます。 既に検索を開始している場合は、検索実行前に検索をキャンセルします。

**戻り値:**

成功であれば OPN\_SUCCESS を返す。

---

```
OPNResult OPNBluetoothService::GetAutoConnectDevice (OPNBluetoothDevice *  
pOpnDevive)
```

自動接続対象のデバイスを返します。

**戻り値:**

成功の場合 OPN\_SUCCESS を返します。

---

```
OPNAutoConnectStatus OPNBluetoothService::GetAutoConnectStatus ()
```

スレーブ自動接続が実行中の場合はTRUEを返します。

**戻り値:**

スレーブ自動接続が実行中の場合はTRUEを返します。 それ以外はFALSEを返します。

---

```
OPNResult OPNBluetoothService::GetBluetoothHostInfos (OPNBluetoothHostInfo  
pHosts[], INT & nCount)
```

Bluetoothのホスト情報を取得します。

**引数:**

|     |        |                             |
|-----|--------|-----------------------------|
| out | pHosts | OPNBluetoothHostInfoの配列     |
| out | nCount | 取得したOPNBluetoothHostInfoの個数 |

**戻り値:**

取得成功 : OPN\_SUCCESS

---

```
OPNResult OPNBluetoothService::GetConnectedBluetoothDevice  
(OPNBluetoothDevice * pOpnDevice)
```

接続済みのデバイス情報を取得します。

**引数:**

|     |            |             |
|-----|------------|-------------|
| out | pOpnDevice | デバイス情報のポインタ |
|-----|------------|-------------|

**戻り値:**

成功ならOPN\_SUCCESSを返します。

---

**IOPNBluetoothServiceDelegate\* OPNBluetoothService::GetDelegate  
( )[protected]**

デリゲータを取得します。

**戻り値:**

IOPNBluetoothServiceDelegateクラスのインスタンスへのポインタ

---

**OPNResult OPNBluetoothService::GetPairedDevices (OPNBluetoothDevice  
devices[], INT & nCount)**

ペアリング済みのBluetoothデバイスを取得します。

**引数:**

|     |                |             |
|-----|----------------|-------------|
| out | <i>devices</i> | 取得したデバイスの配列 |
| out | <i>nCount</i>  | 取得したデバイス数   |

**戻り値:**

成功であれば OPN\_SUCCESS を返す。

---

**OPNResult OPNBluetoothService::IsBluetoothEnabled ( )**

PCがBluetoothをサポートしているかどうかを取得します。

**戻り値:**

サポートが有効であればTRUE、無効であればFALSEを返します。

---

**BOOL OPNBluetoothService::IsConnected ( )**

接続済みの場合はTRUEを返します。

**戻り値:**

接続済みの場合はTRUEを返します。それ以外はFALSEを返します。

---

**BOOL OPNBluetoothService::IsDiscoverable ()**

他のBluetoothデバイスから検索可能な状態かどうか取得します。

**戻り値:**

検索可能な状態であればTRUE。そうでなければFALSE。

---

**BOOL OPNBluetoothService::IsEnableReadBarcode ()**

現在のReadBarcodeが許可か禁止かを返します。

**戻り値:**

TRUE:許可、FALSE:禁止

---

**BOOL OPNBluetoothService::IsServerConnected ()**

サーバとして接続している場合はTRUEを返します。

**戻り値:**

サーバとして接続している場合はTRUEを返します。 それ以外はFALSEを返します。

---

**virtual BOOL OPNBluetoothService::OnDidReadData (BTHSppConnection \* pConnection, BYTE \* readData, INT readBytes)[pure virtual]**

データの読込が完了した直後で呼び出されるインターフェイスです。

**引数:**

|    |                    |                                  |
|----|--------------------|----------------------------------|
| in | <i>pConnection</i> | BTHSppConnectionクラスのインスタンスへのポインタ |
| in | <i>readData</i>    | 読み込んだデータ                         |
| in | <i>readBytes</i>   | 読み込んだデータのバイト数                    |

**戻り値:**

成功時にTRUE、失敗時はFALSEを返すようにします。

OPN2002iBluetoothService (p.16)で実装されています。



---

```
virtual BOOL OPNBluetoothService::OnWillReadData (BTHSppConnection *  
pConnection, BTHReadCondition * pReadCondition)[pure virtual]
```

データを受信し読込を開始する直前で呼び出されるインターフェイスです。

**引数:**

|     |                       |                                  |
|-----|-----------------------|----------------------------------|
| in  | <i>pConnection</i>    | BTHSppConnectionクラスのインスタンスへのポインタ |
| out | <i>pReadCondition</i> | 読込条件を指定します。                      |

**戻り値:**

成功時にTRUE、失敗時はFALSEを返すようにします。

OPN2002iBluetoothService (p.16)で実装されています。

---

```
VOID OPNBluetoothService::SetDelegate (IOPNBluetoothServiceDelegate *  
pDelegate)
```

OPNBluetoothServiceのデリゲータを設定します。

**引数:**

|                  |       |
|------------------|-------|
| <i>pDelegate</i> | デリゲータ |
|------------------|-------|

---

```
VOID OPNBluetoothService::SetEnableReadBarcode (BOOL bEnabled)
```

ReadBarcodeの許可・禁止を設定します。

**引数:**

|    |                 |                      |
|----|-----------------|----------------------|
| in | <i>bEnabled</i> | 許可の場合TRUE、禁止の場合FALSE |
|----|-----------------|----------------------|

---

```
OPNResult OPNBluetoothService::StartAutoConnect ()
```

スレーブの自動接続を開始します。

**戻り値:**

成功の場合 OPN\_SUCCESS を返します。

---

**OPNResult OPNBluetoothService::StartServer ()**

サーバーを開始します。

**戻り値:**

エラーを返します。成功の場合は OPN\_SUCCESS を返します。

---

**OPNResult OPNBluetoothService::StopAutoConnect ()**

スレーブの自動接続を終了します。

**戻り値:**

成功の場合 OPN\_SUCCESS を返します。

---

**OPNResult OPNBluetoothService::StopServer ()**

サーバーを停止します。

**戻り値:**

エラーを返します。成功の場合は OPN\_SUCCESS を返します。

---

**LONG OPNBluetoothService::WriteACK ()**

ACKを送信します。

**戻り値:**

書き込んだデータのバイト数を返します。失敗した場合は-1を返します。

---

**LONG OPNBluetoothService::WriteData (BYTE \* writeData, INT toWriteBytes, INT nTimeout = 5000)**

バーコードコレクターにバイナリデータを送信します。

**引数:**

|     |           |                    |
|-----|-----------|--------------------|
| out | writeData | バーコードコレクターからの受信データ |
|-----|-----------|--------------------|

|    |                     |                            |
|----|---------------------|----------------------------|
| in | <i>toWriteBytes</i> | byteのバッファサイズ               |
| in | <i>nTimeout</i>     | タイムアウト時間 (ms) ※指定しない場合は5秒。 |

**戻り値:**

書き込んだデータのバイト数を返します。失敗した場合は-1を返します。

---

**LONG OPNBluetoothService::WriteDataReadBarcodeResponseNak (BYTE \* *writeData*, INT *toWriteBytes*, INT *nTimeout* = 5000)**

バーコードコレクターにバイナリデータを送信します。 通信時にReadBarcodeを受信した場合、Nakを返します。

**引数:**

|    |                     |                            |
|----|---------------------|----------------------------|
| in | <i>writeData</i>    | 送信するデータ                    |
| in | <i>toWriteBytes</i> | 送信するデータ長                   |
| in | <i>nTimeout</i>     | タイムアウト時間 (ms) ※指定しない場合は5秒。 |

**戻り値:**

書き込んだデータのバイト数を返します。失敗した場合は-1を返します。

---

**LONG OPNBluetoothService::WriteNAK ()**

NAKを送信します。

**戻り値:**

書き込んだデータのバイト数を返します。失敗した場合は-1を返します。

---

## クラス OPNSettingChars

SettingChar構造体を扱いやすくしたヘルパークラス 一覧の取得や、プレフィックス・サフィックスからコマンドを検索することができます。 このクラスはアプリに1つのみ存在する [GoF]Singletonクラスです。

### Public メソッド

- `const SettingChar * GetSettingChars ()`  
文字列設定コマンド一覧を配列として取得します。
- `const SettingChar GetSettingChar (const unsigned char *pfsf)`  
pfsfバイト配列からコマンド構造体を取得します。
- `const char * GetCommandString (const char *str)`  
表示名からコマンドを取得します。 対応するコマンドが見つからない場合はから文字を返します。

### Static Public メソッド

- `static OPNSettingChars & GetInstance ()`  
OPNSettingCharsのインスタンスを取得します。
- `static const SettingChar GetSettingChar (UINT16 pfsf)`  
プレフィックス・サフィックスのビット配列からコマンドの構造体を取得します。 対応するコマンドが見つからない場合は、空のコマンド構造体を返します。

### Protected メソッド

- `OPNSettingChars ()`  
コンストラクタ
- `~OPNSettingChars ()`  
デストラクタ

---

## 説明

SettingChar構造体を扱いやすくしたヘルパークラス 一覧の取得や、プレフィックス・サフィックスからコマンドを検索することができます。 このクラスはアプリに1つのみ存在する [GoF]Singletonクラスです。

---

## コンストラクタとデストラクタ

---

```
OPNSettingChars::OPNSettingChars ()[protected]
```

コンストラクタ

---

```
OPNSettingChars::~~OPNSettingChars ()[protected]
```

デストラクタ

## 関数

---

```
const char* OPNSettingChars::GetCommandString (const char * str)
```

表示名からコマンドを取得します。 対応するコマンドが見つからない場合は空文字を返します。

引数:

|    |     |     |
|----|-----|-----|
| in | str | 表示名 |
|----|-----|-----|

戻り値:

コマンド 対応するコマンドが見つからない場合は空文字を返します。

---

```
static OPNSettingChars& OPNSettingChars::GetInstance ()[static]
```

OPNSettingCharsのインスタンスを取得します。

戻り値:

インスタンス

---

```
static const SettingChar OPNSettingChars::GetSettingChar (UINT16  
pfsf)[static]
```

プレフィックス・サフィックスのビット配列からコマンドの構造体を取得します。 対応するコマンドが見つからない場合は、空のコマンド構造体を返します。

**引数:**

|    |             |                    |
|----|-------------|--------------------|
| in | <i>pfsf</i> | プレフィックス・サフィックスのビット |
|----|-------------|--------------------|

**戻り値:**

コマンドの構造体 対応するコマンドが見つからない場合は、空のコマンド構造体を返します。

---

```
const SettingChar OPNSettingChars::GetSettingChar (const unsigned char * pfsf)
```

*pfsf*バイト配列からコマンド構造体を取得します。

**引数:**

|    |             |                    |
|----|-------------|--------------------|
| in | <i>pfsf</i> | <i>pfsf</i> のバイト配列 |
|----|-------------|--------------------|

**戻り値:**

コマンドの構造体

---

```
const SettingChar* OPNSettingChars::GetSettingChars ()
```

文字列設定コマンド一覧を配列として取得します。

**戻り値:**

SettingCharの配列 (サイズはOPN\_SETTING\_CHAR\_NUM)

---

## クラス OPNSettings

データコレクタから取得したバイトデータの設定値を取得するためのメソッドを提供します。

### Public メソッド

- **OPNSettings** (BYTE \*data)  
コンストラクタ
- **~OPNSettings** ()  
デストラクタ
- void **SetSettingData** (BYTE \*byte)  
GetSetting で取得したデータで初期化します。
- UINT16 **GetFormatVersion** ()  
ファームウェアバージョンを取得します。
- BOOL **JAN\_EAN\_13\_8\_PERMISSION** ()  
EAN-13/8 の読取許可設定を取得します。
- BOOL **JAN\_EAN\_13\_TRANSMIT\_CD** ()  
EAN-13 のCD転送に関する設定を取得します。
- BOOL **WPC\_UPC\_EAN\_JAN\_CD\_CALCULATION** ()  
WPC (UPC、EAN、JAN) のCD計算に関する設定を取得します。
- BOOL **JAN\_EAN\_13\_8\_ADDON2\_PERMISSION** ()  
EAN-13/8 (Addon2) の読取許可設定を取得します。
- BOOL **JAN\_EAN\_13\_8\_ADDON5\_PERMISSION** ()  
EAN-13/8 (Addon5) の読取許可設定を取得します。
- BOOL **JAN\_EAN\_8\_TRANSMIT\_CD** ()  
JAN/EAN-8 のCD転送に関する設定を取得します。
- BOOL **UPC\_A\_E\_PERMISSION** ()  
UPC-A/E の読取許可設定を取得します。
- BOOL **UPC\_A\_TRANSMIT\_CD\_B2** ()  
UPC-A のCD転送に関する設定を取得します。
- BOOL **UPC\_A\_TRANSMIT\_CD\_B4** ()  
UPC-A のCD転送に関する設定を取得します。
- BOOL **UPC\_A\_E\_Addon2\_PERMISSION** ()  
UPC-A/E (Addon2) の読取許可設定を取得します。
- BOOL **UPC\_A\_E\_Addon5\_PERMISSION** ()  
UPC-A/E (Addon5) の読取許可設定を取得します。
- int **UPC\_A\_TRANSMIT** ()  
UPC-A のCD転送に関する設定を取得します。
- BOOL **UPC\_E\_TRANSMIT\_CD\_B2** ()  
UPC-E のCD転送に関する設定を取得します。
- BOOL **UPC\_E\_TRANSMIT\_CD\_B4** ()

UPC-EのCD転送に関する設定を取得します。

- **int UPC\_E\_TRANSMIT ( )**

UPC-EのCD転送に関する設定を取得します。

- **BOOL CODE\_39\_PERMISSION ( )**

Code39の読取許可設定を取得します。

- **BOOL CODE\_39\_TRANSMIT\_CD ( )**

Code39の読取許可設定を取得します。

- **BOOL CODE\_39\_CALCULATION ( )**

Code39のCD計算に関する設定を取得します。

- **BOOL CODE\_39\_START\_STOP\_CD ( )**

Code39のスタートストップに関する設定を取得します。

- **BOOL NW\_7\_PERMISSION ( )**

NW-7の読取許可設定を取得します。

- **BOOL NW\_7\_TRANSMIT\_CD ( )**

NW-7の読取許可設定を取得します。

- **int NW\_7\_CALCULATION ( )**

NW-7のCD計算に関する設定を取得します。

- **int NW\_7\_START\_STOP\_CD ( )**

NW-7のスタートストップに関する設定を取得します。

- **BOOL INTERLEAVED\_2OF5\_PERMISSION ( )**

Interleaved2of5の読取許可設定を取得します。

- **BOOL INDUSTRIAL\_INTERLEAVED\_2OF5\_TRANSMIT\_CD ( )**

Industrial 2 of 5/Interleaved 2 of 5のCD転送に関する設定を取得します。

- **BOOL INDUSTRIAL\_INTERLEAVED\_2OF5\_CALCULATION ( )**

Industrial 2 of 5/Interleaved 2 of 5のCD計算に関する設定を取得します。

- **BOOL INDUSTRIAL\_2OF5\_PERMISSION ( )**

Industrial2of5の読取許可設定を取得します。

- **BOOL IATA\_PERMISSION ( )**

IATAの読取許可設定を取得します。

- **BOOL IATA\_TRANSMIT\_CD ( )**

IATAのCD転送に関する設定を取得します。

- **int IATA\_CALCULATION ( )**

IATAのCD計算に関する設定を取得します。

- **BOOL CODE\_93\_PERMISSION ( )**

Code93の読取許可設定を取得します。

- **BOOL CODE\_93\_CALCULATION ( )**

Code93のCD計算に関する設定を取得します。

- **BOOL CODE\_128\_PERMISSION ( )**

Code128の読取許可設定を取得します。

- **BOOL CODE\_128\_TRANSMIT\_CD ( )**

Code128のCD転送に関する設定を取得します。



- **BOOL CODE\_128\_EAN\_128\_CALCULATION ( )**  
Code 128/GS1-128のCD計算に関する設定を取得します。
- **int EAN\_128\_PERMISSION ( )**  
Code128の読取許可設定を取得します。
- **BOOL S\_CODE\_PERMISSION ( )**  
S-Codeの読取許可設定を取得します。
- **BOOL MSI\_PLESSEY\_PERMISSION ( )**  
MSI/Plesseyの読取許可設定を取得します。
- **int MSI\_PLESSEY\_TRANSMIT ( )**  
MSI/PlesseyのCD転送に関する設定を取得します。
- **int MSI\_PLESSEY\_CALCULATION ( )**  
MSI/PlesseyのCD計算に関する設定を取得します。
- **BOOL UK\_PLESSEY\_PERMISSION ( )**  
Code128の読取許可設定を取得します。
- **BOOL TELEPEN\_PERMISSION ( )**  
Telepenの読取許可設定を取得します。
- **BOOL MATRIX\_2OF5\_PERMISSION ( )**  
Matrix2of5の読取許可設定を取得します。
- **BOOL TRI\_OPTIC\_PERMISSION ( )**  
Tri-Opticの読取許可設定を取得します。
- **BOOL GS1\_DATABAR\_RSS14\_PERMISSION ( )**  
GS1 DataBar の読取許可設定を取得します。
- **BOOL GS1\_DATABAR\_LIMITED\_PERMISSION ( )**  
GS1 DataBar Limited の読取許可設定を取得します。
- **BOOL GS1\_DATABAR\_EXPANDED\_PERMISSION ( )**  
GS1 DataBar Expanded の読取許可設定を取得します。
- **BOOL GS1\_DATABAR\_TRANSMIT\_CD ( )**  
GS1 DataBarのCD転送に関する設定を取得します。
- **BOOL CODE\_3OF5\_PERMISSION ( )**  
Code 3 of 5 の読取許可設定を取得します。
- **BOOL MICRO\_PDF\_417\_PERMISSION ( )**  
MicroPDF417 の読取許可設定を取得します。
- **BOOL PDF\_417\_PERMISSION ( )**  
PDF417 の読取許可設定を取得します。
- **BOOL CODE\_11\_PERMISSION ( )**  
Code11 の読取許可設定を取得します。
- **BOOL INTELLIGENT\_MAIL\_PERMISSION ( )**  
IntelligentMail の読取許可設定を取得します。
- **BOOL POSTNET\_PERMISSION ( )**  
Postnet の読取許可設定を取得します。
- **BOOL JAPANESE\_POSTAL\_PERMISSION ( )**

Japanese Postal の読取許可設定を取得します。

- **BOOL CODABLOCK\_F\_PERMISSION ( )**  
CodablockF の読取許可設定を取得します。
- **BOOL DATA\_MATRIX\_ECC200\_PERMISSION ( )**  
Data Matrix(ECC200) の読取許可設定を取得します。
- **BOOL DATA\_MATRIX\_ECC000\_140\_PERMISSION ( )**  
Data Matrix(ECC000-140) の読取許可設定を取得します。
- **BOOL AZTEC\_CODE\_PERMISSION ( )**  
Aztec code の読取許可設定を取得します。
- **BOOL AZTEC\_RUNES\_PERMISSION ( )**  
Aztec runes の読取許可設定を取得します。
- **BOOL CHINESE\_SENSIBLE\_CODE\_PERMISSION ( )**  
Chinese Sensible code の読取許可設定を取得します。
- **BOOL QR\_CODE\_PERMISSION ( )**  
QR code の読取許可設定を取得します。
- **BOOL MICRO\_QR\_CODE\_PERMISSION ( )**  
MicroQR code の読取許可設定を取得します。
- **BOOL MAXI\_CODE\_PERMISSION ( )**  
Maxi code の読取許可設定を取得します。
- **BOOL COMPOSITE\_ON\_GS1\_DATABAR\_PERMISSION ( )**  
Composite on GS1 Databar の読取許可設定を取得します。
- **BOOL COMPOSITE\_ON\_UPC\_EAN\_PERMISSION ( )**  
Composite on UPC/EAN の読取許可設定を取得します。
- **int READ\_MODE ( )**  
読み取りモードの設定を取得します。
- **int READABLE\_TIME ( )**  
読み取り時間の設定を取得します。
- **int REDUNDANT\_READING ( )**  
照合回数の設定を取得します。
- **int MULTIPLE\_READ\_RESET\_TIME ( )**  
二度読み防止タイマーの設定を取得します。
- **int ADDON\_WAIT\_TIME ( )**  
アドオンタイマーの設定を取得します。
- **int BUZZER\_VOLUME ( )**  
ブザーボリュームの設定を取得します。
- **int GOOD\_READ\_LED\_ON\_TIME ( )**  
LED点灯時間の設定を取得します。
- **int TRIGGER\_MODE\_ENABLE ( )**  
トリガモードの設定を取得します。
- **int TRIGGER\_REPEAT\_ENABLE ( )**  
トリガリピートの設定を取得します。

- `int BUZZER_ENABLE ()`  
ブザー音の設定を取得します。
- `int BUZZER_TONE ()`  
ブザートーンの設定を取得します。
- `int BUZZER_PERIOD ()`  
ブザー鳴動時間の設定を取得します。
- `int BUZZER_MODE ()`  
ブザー鳴動タイミングの設定を取得します。
- `char * DESTINATION_ADDRESS ()`  
接続相手アドレスの設定を取得します。
- `int COMMAND_RESPONSE ()`  
コマンド応答の設定を取得します。
- `char * PIN_CODE ()`  
PINコードの設定を取得します。
- `int TRIGGER_CONNECTION_ENABLE ()`  
トリガーによる接続処理の設定を取得します。
- `int ADDRESS_BARCODE_CONNECTION_ENABLE ()`  
アドレスバーコード読み取りによる接続処理の設定を取得します。
- `int ACK_NAK_CONTROL ()`  
ACK/NAK制御の設定を取得します。
- `int CONNECTION_MODE ()`  
接続モードの設定を取得します。
- `int SLAVE_CONNECTION_WAIT_TIME ()`  
スレーブ接続待ち時間の設定を取得します。
- `int AUTO_RECONNECTION_TIME ()`  
自動再接続有効時間の設定を取得します。
- `int AUTO_DISCONNECTION_TIME ()`  
自動切断時間の設定を取得します。
- `int TRIGGER_BUTTON_PUSHTIME_FOR_CONNECTION ()`  
トリガ接続長押し時間の設定を取得します。
- `int TRIGGER_BUTTON_PUSHTIME_FOR_DISCONNECTION ()`  
トリガ切断長押し時間の設定を取得します。
- `int ACK_NAK_WAIT_TIME ()`  
ACK/NAK待ち時間の設定を取得します。
- `int OUT_OF_RANGE_MEMORY_ENABLE ()`  
圏外メモリーの設定を取得します。
- `int DATA_COLLECT_MODE_ENABLE ()`  
データコレクトの設定を取得します。
- `int AUTO_CONNECTION_ENABLE ()`  
バーコード読み取り自動接続の設定を取得します。
- `int DISCONNECTION_BUZZER_FROM_DATA_COLLECTOR ()`

データコレクタからの切断時ブザーの設定を取得します。

- **int DISCONNECTION\_BUZZER\_FROM\_DESTINATION ( )**  
接続相手からの切断ブザーの設定を取得します。
- **int MEMORY\_OUTPUT\_METHOD ( )**  
メモリーデータの出力方法の設定を取得します。
- **int USB\_CONNECTION ( )**  
USB接続時COM通信の設定を取得します。
- **int FUNCTION\_BUTTON\_INPUT ( )**  
ファンクション押下時出力の設定を取得します。
- **char \* DEVICE\_NAME ( )**  
Bluetoothデバイス名 (自機) の設定を取得します。
- **char \* DESTINATION\_ADDRESS\_LAST\_TIME ( )**  
前回のスレーブ接続時の相手アドレスの設定を取得します。
- **int INTER\_CHARACTER\_DELAY ( )**  
キャラクタ間ディレイの設定を取得します。
- **int KEYBOARD\_LANGUAGE ( )**  
キーボード言語の設定を取得します。
- **int USE\_NUM\_PAD ( )**  
NumPad使用の設定を取得します。
- **BOOL GOOD\_READ\_VIBRATOR\_ENABLE ( )**  
グッドリードバイブレーターの設定を取得します。
- **BYTE \* UPCA\_PREFIX ( )**  
UPC-Aのプリフィックスを指定します
- **BYTE \* UPCA\_ADDON\_PREFIX ( )**  
UPC-Aアドオンのプリフィックスを指定します
- **BYTE \* UPCE\_PREFIX ( )**  
UPC-Eのプリフィックスを指定します
- **BYTE \* UPCE\_ADDON\_PREFIX ( )**  
UPC-Eアドオンのプリフィックスを指定します
- **BYTE \* EAN13\_PREFIX ( )**  
EAN-13のプリフィックスを指定します
- **BYTE \* EAN13\_ADDON\_PREFIX ( )**  
EAN-13アドオンのプリフィックスを指定します
- **BYTE \* EAN8\_PREFIX ( )**  
EAN-8のプリフィックスを指定します
- **BYTE \* EAN8\_ADDON\_PREFIX ( )**  
EAN-8アドオンのプリフィックスを指定します
- **BYTE \* CODE39\_PREFIX ( )**  
Code-39のプリフィックスを指定します
- **BYTE \* TRI\_OPTIC\_PREFIX ( )**  
Tri-Opticのプリフィックスを指定します

- **BYTE \* NW7\_PREFIX ( )**  
NW-7のプリフィックスを指定します
- **BYTE \* D2OF5\_PREFIX ( )**  
Industrial 2 of 5のプリフィックスを指定します
- **BYTE \* I2OF5\_PREFIX ( )**  
Interleaved 2 of 5のプリフィックスを指定します
- **BYTE \* SCODE\_PREFIX ( )**  
S-Codeのプリフィックスを指定します
- **BYTE \* MATRIX2OF5\_PREFIX ( )**  
Matrix 2 of 5のプリフィックスを指定します
- **BYTE \* CODE93\_PREFIX ( )**  
Code93のプリフィックスを指定します
- **BYTE \* CODE128\_PREFIX ( )**  
Code128のプリフィックスを指定します
- **BYTE \* MSIPLESSEY\_PREFIX ( )**  
MSI/Plesseyのプリフィックスを指定します
- **BYTE \* IATA\_PREFIX ( )**  
IATAのプリフィックスを指定します
- **BYTE \* UKPLESSEY\_PREFIX ( )**  
UK/Plesseyのプリフィックスを指定します
- **BYTE \* TELEPEN\_PREFIX ( )**  
Telepenのプリフィックスを指定します
- **BYTE \* RSS\_PREFIX ( )**  
RSSのプリフィックスを指定します
- **BYTE \* PDF417\_PREFIX ( )**  
PDF417のプリフィックスを指定します
- **BYTE \* MICRO\_PDF417\_PREFIX ( )**  
MicroPDF417のプリフィックスを指定します
- **BYTE \* CODE11\_PREFIX ( )**  
Code11のプリフィックスを指定します
- **BYTE \* CODE3OF5\_PREFIX ( )**  
Code3of5のプリフィックスを指定します
- **BYTE \* EAN128\_PREFIX ( )**  
GS1-128のプリフィックスを指定します
- **BYTE \* COMMON\_PREFIX ( )**  
コモンプリフィックス を指定します
- **BYTE \* INTELLIGENT\_MAIL\_PREFIX ( )**  
Intelligent Mailのプリフィックスを指定します
- **BYTE \* POSTNET\_PREFIX ( )**  
Postnetのプリフィックスを指定します
- **BYTE \* JAPANESE\_POSTAL\_PREFIX ( )**

*Japanese postal*のプリフィックスを指定します

- **BYTE \* CODABLOCK\_F\_PREFIX ( )**

*CodablockF*のプリフィックスを指定します

- **BYTE \* DATA\_MATRIX\_ECC200\_ECC000\_140\_PREFIX ( )**

*Data Matrix(ECC200,ECC000-140)*のプリフィックスを指定します

- **BYTE \* AZTEC\_CODE\_RUNES\_PREFIX ( )**

*Aztec code/Aztec runes*のプリフィックスを指定します

- **BYTE \* CHINESE\_SENSIBLE\_CODE\_PREFIX ( )**

*Chinese Sensible code*のプリフィックスを指定します

- **BYTE \* QR\_MICRO\_QR\_PREFIX ( )**

*QR/MicroQR*のプリフィックスを指定します

- **BYTE \* MAXI\_CODE\_PREFIX ( )**

*Maxi code*のプリフィックスを指定します

- **BYTE \* COMPOSITE\_ON\_GS1\_DATABAR\_ON\_UPC\_EAN\_PREFIX ( )**

*Composite (on GS1Databar,UPC/EAN)*のプリフィックスを指定します

- **BYTE \* UPCA\_SUFFIX ( )**

*UPC-A*のサフィックスを指定します

- **BYTE \* UPCA\_ADDON\_SUFFIX ( )**

*UPC-Aアドオン*のサフィックスを指定します

- **BYTE \* UPCE\_SUFFIX ( )**

*UPC-E*のサフィックスを指定します

- **BYTE \* UPCE\_ADDON\_SUFFIX ( )**

*UPC-Eアドオン*のサフィックスを指定します

- **BYTE \* EAN13\_SUFFIX ( )**

*EAN-13*のサフィックスを指定します

- **BYTE \* EAN13\_ADDON\_SUFFIX ( )**

*EAN-13アドオン*のサフィックスを指定します

- **BYTE \* EAN8\_SUFFIX ( )**

*EAN-8*のサフィックスを指定します

- **BYTE \* EAN8\_ADDON\_SUFFIX ( )**

*EAN-8アドオン*のサフィックスを指定します

- **BYTE \* CODE39\_SUFFIX ( )**

*Code-39*のサフィックスを指定します

- **BYTE \* TRI\_OPTIC\_SUFFIX ( )**

*Tri-Optic*のサフィックスを指定します

- **BYTE \* NW7\_SUFFIX ( )**

*NW-7*のサフィックスを指定します

- **BYTE \* D2OF5\_SUFFIX ( )**

*Industrial 2 of 5*のサフィックスを指定します

- **BYTE \* I2OF5\_SUFFIX ( )**

*Interleaved 2 of 5*のサフィックスを指定します

- **BYTE \* SCODE\_SUFFIX ( )**  
S-Codeのサフィックスを指定します
- **BYTE \* MATRIX2OF5\_SUFFIX ( )**  
Matrix2of5のサフィックスを指定します
- **BYTE \* CODE93\_SUFFIX ( )**  
Code93のサフィックスを指定します
- **BYTE \* CODE128\_SUFFIX ( )**  
Code128のサフィックスを指定します
- **BYTE \* MSIPLESSEY\_SUFFIX ( )**  
MSI/Plesseyのサフィックスを指定します
- **BYTE \* IATA\_SUFFIX ( )**  
IATAのサフィックスを指定します
- **BYTE \* UKPLESSEY\_SUFFIX ( )**  
UK/Plesseyのサフィックスを指定します
- **BYTE \* TELEPEN\_SUFFIX ( )**  
Telepenのサフィックスを指定します
- **BYTE \* RSS\_SUFFIX ( )**  
RSSのサフィックスを指定します
- **BYTE \* PDF417\_SUFFIX ( )**  
PDF417のサフィックスを指定します
- **BYTE \* MICRO\_PDF417\_SUFFIX ( )**  
MicroPDF417のサフィックスを指定します
- **BYTE \* CODE11\_SUFFIX ( )**  
Code11のサフィックスを指定します
- **BYTE \* CODE3OF5\_SUFFIX ( )**  
Code3of5のサフィックスを指定します
- **BYTE \* EAN128\_SUFFIX ( )**  
GS1-128のサフィックスを指定します
- **BYTE \* COMMON\_SUFFIX ( )**  
コモンサフィックスを指定します
- **BYTE \* INTELLIGENT\_MAIL\_SUFFIX ( )**  
Intelligent Mailのサフィックスを指定します
- **BYTE \* POSTNET\_SUFFIX ( )**  
Postnetのサフィックスを指定します
- **BYTE \* JAPANESE\_POSTAL\_SUFFIX ( )**  
Japanese postalのサフィックスを指定します
- **BYTE \* CODABLOCK\_F\_SUFFIX ( )**  
CodablockFのサフィックスを指定します
- **BYTE \* DATA\_MATRIX\_ECC200\_ECC000\_140\_SUFFIX ( )**  
Data Matrix(ECC200,ECC000-140)のサフィックスを指定します
- **BYTE \* AZTEC\_CODE\_RUNES\_SUFFIX ( )**

*Aztec code/Aztec runes*のサフィックスを指定します

- **BYTE \* CHINESE\_SENSIBLE\_CODE\_SUFFIX ( )**

*Chinese Sensible code*のサフィックスを指定します

- **BYTE \* QR\_MICRO\_QR\_SUFFIX ( )**

*QR/MicroQR*のサフィックスを指定します

- **BYTE \* MAXI\_CODE\_SUFFIX ( )**

*Maxi code*のサフィックスを指定します

- **BYTE \* COMPOSITE\_ON\_GS1\_DATABAR\_ON\_UPC\_EAN\_SUFFIX ( )**

*Composite (on GS1Databar,UPC/EAN)*のサフィックスを指定します

---

## 説明

データコレクタから取得したバイトデータの設定値を取得するためのメソッドを提供します。

[GoF]Singletonパターンを使っているため、インスタンスは1つだけ生成されます。インスタンスの取得には、GetInstance(void)を使用してください。 コピーコンストラクタがprivate化されているので、インスタンスの複製ができないようになっています。ポインタではなく参照を返すので、deleteは不要です。 マルチスレッド安全性はありません。 プロセス終了時に、破棄されます。

---

## コンストラクタとデストラクタ

---

**OPNSettings::OPNSettings (BYTE \* data)**

コンストラクタ

---

**OPNSettings::~~OPNSettings ( )**

デストラクタ

---

## 関数

---

**int OPNSettings::ACK\_NAK\_CONTROL ( )**

ACK/NAK制御の設定を取得します。



**戻り値:**

0:無し 1:有り 2:有り (NoResponse)

---

**int OPNSettings::ACK\_NAK\_WAIT\_TIME ( )**

ACK/NAK待ち時間の設定を取得します。

**戻り値:**

5:100ms 25:500ms 50:1s

---

**int OPNSettings::ADDON\_WAIT\_TIME ( )**

アドオンタイマーの設定を取得します。

**戻り値:**

0:なし  
13:250ms  
25:500ms  
38:750ms

---

**int OPNSettings::ADDRESS\_BARCODE\_CONNECTION\_ENABLE ( )**

アドレスバーコード読み取りによる接続処理の設定を取得します。

**戻り値:**

0:無効 1:有効

---

**int OPNSettings::AUTO\_CONNECTION\_ENABLE ( )**

バーコード読み取り自動接続の設定を取得します。

**戻り値:**

0:無効 1:有効

---

**int OPNSettings::AUTO\_DISCONNECTION\_TIME ( )**

自動切断時間の設定を取得します。

**戻り値:**

0:無効  
30000:10分  
60000:20分  
90000:30分  
120000:40分  
150000:50分  
180000:60分  
3000:1分  
6000:2分  
9000:3分  
12000:4分  
15000:5分  
18000:6分  
21000:7分  
24000:8分  
27000:9分  
500:10秒  
1000:20秒  
1500:30秒  
2000:40秒  
2500:50秒

---

**int OPNSettings::AUTO\_RECONNECTION\_TIME ( )**

自動再接続有効時間の設定を取得します。

**戻り値:**

0:無効  
3000:1分  
6000:2分  
9000:3分  
12000:4分  
15000:5分  
18000:6分  
21000:7分  
24000:8分  
27000:9分  
30000:10分  
33000:11分  
36000:12分  
39000:13分

42000:14分  
45000:15分

---

**BOOL OPNSettings::AZTEC\_CODE\_PERMISSION ()**

Aztec code の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::AZTEC\_CODE\_RUNES\_PREFIX ()**

Aztec code/Aztec runesのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::AZTEC\_CODE\_RUNES\_SUFFIX ()**

Aztec code/Aztec runesのサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::AZTEC\_RUNES\_PERMISSION ()**

Aztec runes の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::BUZZER\_ENABLE ()**

ブザー音の設定を取得します。

**戻り値:**

0:無効 1:有効

---

**int OPNSettings::BUZZER\_MODE ()**

ブザー鳴動タイミングの設定を取得します。

**戻り値:**

0:転送前ブザー 1:転送後ブザー

---

**int OPNSettings::BUZZER\_PERIOD ()**

ブザー鳴動時間の設定を取得します。

**戻り値:**

5:100ms  
10:200ms  
20:400ms  
2:50ms

---

**int OPNSettings::BUZZER\_TONE ()**

ブザートーンの設定を取得します。

**戻り値:**

0:単音  
1:高低  
2:低高  
3:4.5KHz  
4:2.2KHz-2KHz

---

**int OPNSettings::BUZZER\_VOLUME ()**

ブザーボリュームの設定を取得します。

**戻り値:**

127:最大

32:大  
8:中  
1:小

---

**BOOL OPNSettings::CHINESE\_SENSIBLE\_CODE\_PERMISSION ( )**

Chinese Sensible code の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::CHINESE\_SENSIBLE\_CODE\_PREFIX ( )**

Chinese Sensible codeのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CHINESE\_SENSIBLE\_CODE\_SUFFIX ( )**

Chinese Sensible codeのサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::CODABLOCK\_F\_PERMISSION ( )**

CodablockF の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::CODABLOCK\_F\_PREFIX ( )**

CodablockFのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CODABLOCK\_F\_SUFFIX ( )**

CodablockFのサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::CODE11\_PREFIX ( )**

Code11のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CODE11\_SUFFIX ( )**

Code11のサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::CODE128\_PREFIX ( )**

Code128のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CODE128\_SUFFIX ( )**

Code128のサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::CODE39\_PREFIX ( )**

Code-39のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CODE39\_SUFFIX ( )**

Code-39のサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::CODE3OF5\_PREFIX ( )**

Code 3 of 5のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CODE3OF5\_SUFFIX ( )**

Code 3 of 5のサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::CODE93\_PREFIX ( )**

Code93のプリフィックスを指定します

---

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::CODE93\_SUFFIX ( )**

Code93のサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::CODE\_11\_PERMISSION ( )**

Code11 の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::CODE\_128\_EAN\_128\_CALCULATION ( )**

Code 128/GS1-128のCD計算に関する設定を取得します。

**戻り値:**

YES:CD計算する NO:CD計算しない

---

**BOOL OPNSettings::CODE\_128\_PERMISSION ( )**

Code128の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::CODE\_128\_TRANSMIT\_CD ( )**

Code128のCD転送に関する設定を取得します。



**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BOOL OPNSettings::CODE\_39\_CALCULATION ( )**

Code39のCD計算に関する設定を取得します。

**戻り値:**

YES:CD計算する NO:CD計算しない

---

**BOOL OPNSettings::CODE\_39\_PERMISSION ( )**

Code39の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::CODE\_39\_START\_STOP\_CD ( )**

Code39のスタートストップに関する設定を取得します。

**戻り値:**

YES:転送する NO:転送しない

---

**BOOL OPNSettings::CODE\_39\_TRANSMIT\_CD ( )**

Code39の読取許可設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BOOL OPNSettings::CODE\_3OF5\_PERMISSION ( )**

Code 3 of 5 の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::CODE\_93\_CALCULATION ()**

Code93のCD計算に関する設定を取得します。

**戻り値:**

YES:CD計算する NO:CD計算しない

---

**BOOL OPNSettings::CODE\_93\_PERMISSION ()**

Code93の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::COMMAND\_RESPONSE ()**

コマンド応答の設定を取得します。

**戻り値:**

1:有り (ACK/NAK) 0:BT33(0-7)の設定に従う

---

**BYTE\* OPNSettings::COMMON\_PREFIX ()**

コモンプリフィックス を指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::COMMON\_SUFFIX ()**

コモンサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::COMPOSITE\_ON\_GS1\_DATABAR\_ON\_UPC\_EAN\_PREFIX ( )**

Composite (on GS1Databar,UPC/EAN)のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::COMPOSITE\_ON\_GS1\_DATABAR\_ON\_UPC\_EAN\_SUFFIX ( )**

Composite (on GS1Databar,UPC/EAN)のサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::COMPOSITE\_ON\_GS1\_DATABAR\_PERMISSION ( )**

Composite on GS1 Databar の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::COMPOSITE\_ON\_UPC\_EAN\_PERMISSION ( )**

Composite on UPC/EAN の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::CONNECTION\_MODE ( )**

接続モードの設定を取得します。

**戻り値:**

0:SPPマスター  
1:SPPスレーブ  
2:HID

---

**BYTE\* OPNSettings::D2OF5\_PREFIX ( )**

Industrial 2 of 5のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::D2OF5\_SUFFIX ( )**

Industrial 2 of 5のサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::DATA\_COLLECT\_MODE\_ENABLE ( )**

データコレクトの設定を取得します。

**戻り値:**

0:無効 1:有効

---

**BOOL OPNSettings::DATA\_MATRIX\_ECC000\_140\_PERMISSION ( )**

Data Matrix(ECC000-140) の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::DATA\_MATRIX\_ECC200\_ECC000\_140\_PREFIX ( )**

Data Matrix(ECC200,ECC000-140)のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::DATA\_MATRIX\_ECC200\_ECC000\_140\_SUFFIX ( )**

Data Matrix(ECC200,ECC000-140)のサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::DATA\_MATRIX\_ECC200\_PERMISSION ( )**

Data Matrix(ECC200) の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**char\* OPNSettings::DESTINATION\_ADDRESS ( )**

接続相手アドレスの設定を取得します。

**戻り値:**

接続相手アドレス

---

**char\* OPNSettings::DESTINATION\_ADDRESS\_LAST\_TIME ( )**

前回のスレーブ接続時の相手アドレスの設定を取得します。

**戻り値:**

前回のスレーブ接続時の相手アドレス

---

**char\* OPNSettings::DEVICE\_NAME ( )**

Bluetoothデバイス名（自機）の設定を取得します。

---

**戻り値:**

Bluetoothデバイス名

---

**int OPNSettings::DISCONNECTION\_BUZZER\_FROM\_DATA\_COLLECTOR ()**

データコレクタからの切断時ブザーの設定を取得します。

**戻り値:**

0:無効 1:有効

---

**int OPNSettings::DISCONNECTION\_BUZZER\_FROM\_DESTINATION ()**

接続相手からの切断ブザーの設定を取得します。

**戻り値:**

0:無効 1:有効

---

**BYTE\* OPNSettings::EAN128\_PREFIX ()**

GS1-128のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::EAN128\_SUFFIX ()**

GS1-128のサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::EAN13\_ADDON\_PREFIX ()**

EAN-13アドオンのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::EAN13\_ADDON\_SUFFIX ( )**

EAN-13アドオンのサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::EAN13\_PREFIX ( )**

EAN-13のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::EAN13\_SUFFIX ( )**

EAN-13のサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::EAN8\_ADDON\_PREFIX ( )**

EAN-8アドオンのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::EAN8\_ADDON\_SUFFIX ( )**

EAN-8アドオンのサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::EAN8\_PREFIX ()**

EAN-8のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::EAN8\_SUFFIX ()**

EAN-8のサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::EAN128\_PERMISSION ()**

Code128の読取許可設定を取得します。

**戻り値:**

2: 許可 if Possible  
1: 許可 GS1-128 Only  
0: 禁止 Code 128として出力

---

**int OPNSettings::FUNCTION\_BUTTON\_INPUT ()**

ファンクション押下時出力の設定を取得します。

**戻り値:**

0x09:HT  
0x0A:LF  
0x0D:CR  
0x18:CAN  
0x1B:ESC



---

**UINT16 OPNSettings::GetFormatVersion ()**

ファームウェアバージョンを取得します。

**戻り値:**

ファームウェアバージョン

---

**int OPNSettings::GOOD\_READ\_LED\_ON\_TIME ()**

LED点灯時間の設定を取得します。

**戻り値:**

0:無効  
10:200ms  
20:400ms  
30:600ms

---

**BOOL OPNSettings::GOOD\_READ\_VIBRATOR\_ENABLE ()**

グッドリードバイブレーターの設定を取得します。

**戻り値:**

YES:有効 NO:無効

---

**BOOL OPNSettings::GS1\_DATABAR\_EXPANDED\_PERMISSION ()**

GS1 DataBar Expanded の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::GS1\_DATABAR\_LIMITED\_PERMISSION ()**

GS1 DataBar Limited の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::GS1\_DATABAR\_RSS14\_PERMISSION ( )**

GS1 DataBar の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::GS1\_DATABAR\_TRANSMIT\_CD ( )**

GS1 DataBarのCD転送に関する設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BYTE\* OPNSettings::I2OF5\_PREFIX ( )**

Interleaved 2 of 5のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::I2OF5\_SUFFIX ( )**

Interleaved 2 of 5のサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::IATA\_CALCULATION ( )**

IATAのCD計算に関する設定を取得します。

**戻り値:**

- 1: C/D計算する (CPN+FORM SERIAL)
- 2: C/D計算する (FORM SERIAL)
- 3: C/D計算する (ALL DATA)
- 0: C/D計算しない

---

**BOOL OPNSettings::IATA\_PERMISSION ()**

IATAの読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::IATA\_PREFIX ()**

IATAのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::IATA\_SUFFIX ()**

IATAのサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::IATA\_TRANSMIT\_CD ()**

IATAのCD転送に関する設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BOOL OPNSettings::INDUSTRIAL\_2OF5\_PERMISSION ()**

Industrial 2 of 5の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::INDUSTRIAL\_INTERLEAVED\_2OF5\_CALCULATION ( )**

Industrial 2 of 5/Interleaved 2 of 5のCD計算に関する設定を取得します。

**戻り値:**

YES:CD計算する NO:CD計算しない

---

**BOOL OPNSettings::INDUSTRIAL\_INTERLEAVED\_2OF5\_TRANSMIT\_CD ( )**

Industrial 2 of 5/Interleaved 2 of 5のCD転送に関する設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BOOL OPNSettings::INTELLIGENT\_MAIL\_PERMISSION ( )**

IntelligentMail の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::INTELLIGENT\_MAIL\_PREFIX ( )**

Intelligent Mailのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::INTELLIGENT\_MAIL\_SUFFIX ( )**

Intelligent Mailのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::INTER\_CHARACTER\_DELAY ( )**

キャラクタ間ディレイの設定を取得します。

**戻り値:**

0:0ms  
10:10ms  
20:20ms  
30:30ms  
40:40ms  
50:50ms  
60:60ms  
70:70ms  
80:80ms  
90:90ms  
100:100ms  
150:150ms  
200:200ms  
250:250ms  
300:300ms  
350:350ms  
400:400ms  
450:450ms  
500:500ms  
550:550ms  
600:600ms

---

**BOOL OPNSettings::INTERLEAVED\_2OF5\_PERMISSION ( )**

Interleaved 2 of 5の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::JAN\_EAN\_13\_8\_ADDON2\_PERMISSION ( )**

EAN-13/8 (Addon2) の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::JAN\_EAN\_13\_8\_ADDON5\_PERMISSION ( )**

EAN-13/8 (Addon5) の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::JAN\_EAN\_13\_8\_PERMISSION ( )**

EAN-13/8の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::JAN\_EAN\_13\_TRANSMIT\_CD ( )**

EAN-13のCD転送に関する設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BOOL OPNSettings::JAN\_EAN\_8\_TRANSMIT\_CD ( )**

JAN/EAN-8のCD転送に関する設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**BOOL OPNSettings::JAPANESE\_POSTAL\_PERMISSION ( )**

Japanese Postal の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::JAPANESE\_POSTAL\_PREFIX ()**

Japanese postalのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::JAPANESE\_POSTAL\_SUFFIX ()**

Japanese postalのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::KEYBOARD\_LANGUAGE ()**

キーボード言語の設定を取得します。

**戻り値:**

0x01:USA(Default)  
0x02:UK  
0x03:French  
0x04:German  
0x05:Italian  
0x09:Belgian  
0x0A:Spanish  
0x0B:Portuguese  
0x0C:Dutch  
0x0D:Swedish  
0x0E:Finnish  
0x0F:Swiss German  
0x10:Swiss French  
0x11:Japanese  
0x12:Danish  
0x13:Norwegian  
0x15:Czech

---

**BYTE\* OPNSettings::MATRIX2OF5\_PREFIX ( )**

Matrix 2 of 5のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::MATRIX2OF5\_SUFFIX ( )**

Matrix 2 of 5のサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::MATRIX\_2OF5\_PERMISSION ( )**

Matrix 2 of 5の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::MAXI\_CODE\_PERMISSION ( )**

Maxi code の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::MAXI\_CODE\_PREFIX ( )**

Maxi codeのプリフィックスを指定します

**戻り値:**

プリフィックス



---

**BYTE\* OPNSettings::MAXI\_CODE\_SUFFIX ( )**

Maxi codeのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::MEMORY\_OUTPUT\_METHOD ( )**

メモリーデータの出力方法の設定を取得します。

**戻り値:**

0:接続時即出力  
1:ファンクションキー押下かコマンドで出力  
:データ出力コマンド

---

**BYTE\* OPNSettings::MICRO\_PDF417\_PREFIX ( )**

MicroPDF417のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::MICRO\_PDF417\_SUFFIX ( )**

MicroPDF417のサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::MICRO\_PDF\_417\_PERMISSION ( )**

MicroPDF417 の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::MICRO\_QR\_CODE\_PERMISSION ()**

MicroQR code の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::MSI\_PLESSEY\_CALCULATION ()**

MSI/PlesseyのCD計算に関する設定を取得します

**戻り値:**

- 1:C/D計算する CD1 only (Mod10)
  - 2:C/D計算する CD's (Mod10/Mod10)
  - 3:C/D計算する CD's (Mod10/Mod11)
  - 4:C/D計算する CD's (Mod11/Mod10)
  - 0:C/D計算しない
- 

**BOOL OPNSettings::MSI\_PLESSEY\_PERMISSION ()**

MSI/Plesseyの読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::MSI\_PLESSEY\_TRANSMIT ()**

MSI/PlesseyのCD転送に関する設定を取得します。

**戻り値:**

- 1:CD転送する CD1
- 2:CD転送する CD1 and CD2
- 0:CD転送しない

---

**BYTE\* OPNSettings::MSIPLESSEY\_PREFIX ( )**

MSI/Plesseyのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::MSIPLESSEY\_SUFFIX ( )**

MSI/Plesseyのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::MULTIPLE\_READ\_RESET\_TIME ( )**

二度読み防止タイマーの設定を取得します。

**戻り値:**

0: 無限  
3: 50ms  
5: 100ms  
10: 200ms  
15: 300ms  
20: 400ms  
25: 500ms  
30: 600ms

---

**BYTE\* OPNSettings::NW7\_PREFIX ( )**

NW-7のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::NW7\_SUFFIX ( )**

NW-7のサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::NW\_7\_CALCULATION ( )**

NW-7のCD計算に関する設定を取得します。

**戻り値:**

- 1: CD計算する Mod10/W1,2 spec1
  - 2: CD計算する Mod16
  - 3: CD計算する 7 check
  - 4: CD計算する Mod11
  - 0: CD計算しない
- 

**BOOL OPNSettings::NW\_7\_PERMISSION ( )**

NW-7の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::NW\_7\_START\_STOP\_CD ( )**

NW-7のスタートストップに関する設定を取得します。

**戻り値:**

- 1: 転送する ABCD/TN\*E
- 2: 転送する abcd/tn\*e
- 3: 転送する ABCD
- 4: 転送する abcd
- 5: 転送する DC1DC2DC3DC4
- 0: 転送しない

---

**BOOL OPNSettings::NW\_7\_TRANSMIT\_CD ( )**

NW-7の読取許可設定を取得します。

**戻り値:**

YES:CD転送する NO:CD転送しない

---

**int OPNSettings::OUT\_OF\_RANGE\_MEMORY\_ENABLE ( )**

圏外メモリーの設定を取得します。

**戻り値:**

0:無効 1:有効

---

**BYTE\* OPNSettings::PDF417\_PREFIX ( )**

PDF417のプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::PDF417\_SUFFIX ( )**

PDF417のサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::PDF\_417\_PERMISSION ( )**

PDF417 の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**char\* OPNSettings::PIN\_CODE ( )**

PINコードの設定を取得します。

**戻り値:**

PINコード

---

**BOOL OPNSettings::POSTNET\_PERMISSION ( )**

Postnet の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::POSTNET\_PREFIX ( )**

Postnetのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::POSTNET\_SUFFIX ( )**

Postnetのサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::QR\_CODE\_PERMISSION ( )**

QR code の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::QR\_MICRO\_QR\_PREFIX ( )**

QR/MicroQRのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::QR\_MICRO\_QR\_SUFFIX ( )**

QR/MicroQRのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::READ\_MODE ( )**

読み取りモードの設定を取得します。

**戻り値:**

- 1: 単発読み
  - 2: 複数読み
  - 3: 連続読み
- 

**int OPNSettings::READABLE\_TIME ( )**

読み取り時間の設定を取得します。

**戻り値:**

- 0 : 無限
- 1 : 0秒
- 50 : 1秒
- 100 : 2秒
- 150 : 3秒
- 200 : 4秒
- 250 : 5秒
- 300 : 6秒
- 350 : 7秒

400:8秒  
450:9秒  
※返り値 x 10 = 読み取り時間 (ミリ秒)

---

**int OPNSettings::REDUNDANT\_READING ()**

照合回数の設定を取得します。

**戻り値:**

0: 1回読取 0回照合  
1: 2回読取 1回照合  
2: 3回読取 2回照合  
3: 4回読取 3回照合  
4: 5回読取 4回照合  
5: 6回読取 5回照合  
6: 7回読取 6回照合  
7: 8回読取 7回照合  
8: 9回読取 8回照合  
9:10回読取 9回照合  
10:11回読取 10回照合  
11:12回読取 11回照合  
12:13回読取 12回照合  
13:14回読取 13回照合  
14:15回読取 14回照合  
15:16回読取 15回照合

---

**BYTE\* OPNSettings::RSS\_PREFIX ()**

RSSのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::RSS\_SUFFIX ()**

RSSのサフィックスを指定します

**戻り値:**

サフィックス



---

**BOOL OPNSettings::S\_CODE\_PERMISSION ( )**

S-Codeの読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::SCODE\_PREFIX ( )**

S-Codeのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::SCODE\_SUFFIX ( )**

S-Codeのサフィックスを指定します

**戻り値:**

サフィックス

---

**void OPNSettings::SetSettingData (BYTE \* *byte*)**

GetSettingで取得したデータで初期化します。

**引数:**

|             |                    |
|-------------|--------------------|
| <i>byte</i> | GetSettingで取得したデータ |
|-------------|--------------------|

---

**int OPNSettings::SLAVE\_CONNECTION\_WAIT\_TIME ( )**

スレーブ接続待ち時間の設定を取得します。

**戻り値:**

1500:30秒  
3000:1分

6000:2分  
9000:3分  
12000:4分

---

**BOOL OPNSettings::TELEPEN\_PERMISSION ( )**

Telepenの読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::TELEPEN\_PREFIX ( )**

Telepenのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::TELEPEN\_SUFFIX ( )**

Telepenのサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::TRI\_OPTIC\_PERMISSION ( )**

Tri-Opticの読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BYTE\* OPNSettings::TRI\_OPTIC\_PREFIX ( )**

Tri-Opticのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::TRI\_OPTIC\_SUFFIX ( )**

Tri-Opticのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::TRIGGER\_BUTTON\_PUSHTIME\_FOR\_CONNECTION ( )**

トリガ接続長押し時間の設定を取得します。

**戻り値:**

0: トリガ接続無効  
50: 1秒  
100: 2秒  
150: 3秒  
200: 4秒  
250: 5秒  
300: 6秒  
350: 7秒  
400: 8秒  
450: 9秒

---

**int OPNSettings::TRIGGER\_BUTTON\_PUSHTIME\_FOR\_DISCONNECTION ( )**

トリガ切断長押し時間の設定を取得します。

**戻り値:**

0: トリガ切断無効 50: 1秒  
100: 2秒  
150: 3秒  
200: 4秒  
250: 5秒  
300: 6秒  
350: 7秒  
400: 8秒

450:9秒

---

```
int OPNSettings::TRIGGER_CONNECTION_ENABLE ()
```

トリガーによる接続処理の設定を取得します。

**戻り値:**

0:無効 1:有効

---

```
int OPNSettings::TRIGGER_MODE_ENABLE ()
```

トリガモードの設定を取得します。

**戻り値:**

0:トリガ無効 1:トリガ有効

---

```
int OPNSettings::TRIGGER_REPEAT_ENABLE ()
```

トリガリピートの設定を取得します。

**戻り値:**

0:無効 1:有効

---

```
BOOL OPNSettings::UK_PLESSEY_PERMISSION ()
```

Code128の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

```
BYTE* OPNSettings::UKPLESSEY_PREFIX ()
```

UK/Plesseyのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::UKPLESSEY\_SUFFIX ( )**

UK/Plesseyのサフィックスを指定します

**戻り値:**

サフィックス

---

**BOOL OPNSettings::UPC\_A\_E\_Addon2\_PERMISSION ( )**

UPC-A/E (Addon2) の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::UPC\_A\_E\_Addon5\_PERMISSION ( )**

UPC-A/E (Addon5) の読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**BOOL OPNSettings::UPC\_A\_E\_PERMISSION ( )**

UPC-A/Eの読取許可設定を取得します。

**戻り値:**

YES:許可 NO:禁止

---

**int OPNSettings::UPC\_A\_TRANSMIT ( )**

UPC-AのCD転送に関する設定を取得します。

**戻り値:**

- 1: 1 3桁 (先頭0とCD付き)
- 2: 1 2桁 (先頭0なし)
- 3: 1 2桁 (CDなし)
- 0: 1 1桁 (先頭0とCDなし)

---

**BOOL OPNSettings::UPC\_A\_TRANSMIT\_CD\_B2 ()**

UPC-AのCD転送に関する設定を取得します。

**戻り値:**

xsw7 2bit目

---

**BOOL OPNSettings::UPC\_A\_TRANSMIT\_CD\_B4 ()**

UPC-AのCD転送に関する設定を取得します。

**戻り値:**

xsw7 4bit目

---

**int OPNSettings::UPC\_E\_TRANSMIT ()**

UPC-EのCD転送に関する設定を取得します。

**戻り値:**

- 1: 8桁 (先頭0とCD付き)
- 2: 7桁 (先頭なし)
- 3: 7桁 (CDなし)
- 0: 6桁 (先頭0とCDなし)

---

**BOOL OPNSettings::UPC\_E\_TRANSMIT\_CD\_B2 ()**

UPC-EのCD転送に関する設定を取得します。

**戻り値:**

xsw8 2bit目

---

**BOOL OPNSettings::UPC\_E\_TRANSMIT\_CD\_B4 ( )**

UPC-EのCD転送に関する設定を取得します。

**戻り値:**

xsw8 4bit目

---

**BYTE\* OPNSettings::UPCA\_ADDON\_PREFIX ( )**

UPC-Aアドオンのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::UPCA\_ADDON\_SUFFIX ( )**

UPC-Aアドオンのサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::UPCA\_PREFIX ( )**

UPC-Aのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::UPCA\_SUFFIX ( )**

UPC-Aのサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::UPCE\_ADDON\_PREFIX ( )**

UPC-Eアドオンのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::UPCE\_ADDON\_SUFFIX ( )**

UPC-Eアドオンのサフィックスを指定します

**戻り値:**

サフィックス

---

**BYTE\* OPNSettings::UPCE\_PREFIX ( )**

UPC-Eのプリフィックスを指定します

**戻り値:**

プリフィックス

---

**BYTE\* OPNSettings::UPCE\_SUFFIX ( )**

UPC-Eのサフィックスを指定します

**戻り値:**

サフィックス

---

**int OPNSettings::USB\_CONNECTION ( )**

USB接続時COM通信の設定を取得します。

**戻り値:**

0:無効 1:有効



---

**int OPNSettings::USE\_NUM\_PAD ( )**

NumPad使用の設定を取得します。

**戻り値:**

0xFF:Numパッドを使用しない  
0x00:Numパッドを使用する

---

**BOOL OPNSettings::WPC\_UPC\_EAN\_JAN\_CD\_CALCULATION ( )**

WPC (UPC、EAN、JAN) のCD計算に関する設定を取得します。

**戻り値:**

YES:CD計算する NO:CD計算しない

---

## 構造体 SettingChar

文字列設定コマンドの構造体

### Public 変数

- `const char * name`
- `const char * command`
- `UINT16 pfsf`

---

### 説明

文字列設定コマンドの構造体

---

### 変数

---

`const char* SettingChar::command`

コマンド

---

`const char* SettingChar::name`

プレフィックス・サフィックスの表示内容

---

`UINT16 SettingChar::pfsf`

プレフィックス・サフィックス

# ファイル

## OPN2002iBluetoothService.h

### 構成

- `class IOPN2002iBluetoothServiceDelegate`  
OPN3002iBluetoothServiceに対するデリゲータクラス
- `class OPN2002iBluetoothService`  
OPN2002/OPN2002i、OPN3002/OPN3002iへのAPIを提供するシングルトンクラス

### 列挙型

- `enum OPNApiType { OPNApiNone, OPNApiInitializeSetting, OPNApiInitializeSettingBluetooth, OPNApiGetFirmwareVersion, OPNApiSaveSetting, OPNApiReadBarcode, OPNApiGetStoredBarcodeAll, OPNApiEraseStoredBarcodeAll, OPNApiSetDataAndTimeStart, OPNApiSetDataAndTimeEnd, OPNApiSetDataAndTimeCommand, OPNApiGetDataAndTime, OPNApiGetDataAndTime2, OPNApiGetBatteryLevel, OPNApiGetSetting, OPNApiVibratorOn, OPNApiBeepBuzzer, OPNApiCustomCommand, OPNApiCancel }`

各APIを示す列挙体

---

### 列挙型

---

`enum OPNApiType`

各APIを示す列挙体

列挙型の値

**`OPNApiNone`** APIの指定なし

**`OPNApiInitializeSetting`** API InitializeSetting.

**`OPNApiInitializeSettingBluetooth`** API InitializeSettingBluetooth.

**`OPNApiGetFirmwareVersion`** API GetFirmwareVersion.

**`OPNApiSaveSetting`** API SaveSetting.

**`OPNApiReadBarcode`** API ReadBarcode.

**`OPNApiGetStoredBarcodeAll`** API GetStoredBarcodeAll.

**`OPNApiEraseStoredBarcodeAll`** API EraseStoredBarcodeAll.

**`OPNApiSetDataAndTimeStart`** API SetDataAndTimeStart.

**`OPNApiSetDataAndTimeEnd`** API SetDataAndTimeEnd.

**`OPNApiSetDataAndTimeCommand`** API SetDataAndTimeCommand.

**OPNApiGetDataAndTime** API GetDataAndTime 日付の取得  
(OPN2002iの場合は日付・時刻の取得)

**OPNApiGetDataAndTime2** API GetDataAndTime2 時刻の取得  
(OPN3002iのみ)

**OPNApiGetBatteryLevel** API GetBatteryLevel.

**OPNApiGetSetting** API GetSetting.

**OPNApiVibratorOn** API VibratorOn.

**OPNApiBeepBuzzer** API BeepBuzzer.

**OPNApiCustomCommand** CustomCommand カスタムコマンド

**OPNApiCancel** CancelAllCommand コマンドをキャンセルするAPI.

---

## OPNBluetoothDevice.h

### 構成

- `class OPNBluetoothDevice`  
*Bluetooth* デバイス クラス

## OPNBluetoothHostInfo.h

### 構成

- `class OPNBluetoothHostInfo`  
ホストPCに関する情報を保持します。

## OPNBluetoothService.h

### 構成

- `class IOPNBluetoothServiceDelegate`  
OPNBluetoothServiceに対するデリゲータクラス
- `class OPNBluetoothService`  
OPNデバイスとの接続、データ送受信を管理する抽象クラス

### 列挙型

- `enum OPNAutoConnectStatus { OPN_AUTO_CONNECT_STATUS_STOP = 0, OPN_AUTO_CONNECT_STATUS_RUNNING, OPN_AUTO_CONNECT_STATUS_CONNECTING, OPN_AUTO_CONNECT_STATUS_CONNECTED }`  
スレーブの自動接続の状態を示す定数
- 

### 列挙型

---

`enum OPNAutoConnectStatus`

スレーブの自動接続の状態を示す定数

列挙型の値

`OPN_AUTO_CONNECT_STATUS_STOP`  
`OPN_AUTO_CONNECT_STATUS_RUNNING`  
`OPN_AUTO_CONNECT_STATUS_CONNECTING` 接続試行中  
`OPN_AUTO_CONNECT_STATUS_CONNECTED` 接続済

---

OPNCommand.h



## OPNErrors.h

### 列挙型

- `enum OPNResult { OPN_SUCCESS = 1, OPN_CONNECTION_ERROR = -1, OPN_ARGUMENT_ERROR = -2, OPN_CONNECT_TIMEOUT = -3, OPN_ACCEPT_TIMEOUT = -4, OPN_AUTH_FAILED = -5, OPN_DISABLED_BLUETOOTH = -6, OPN_ERROR_CONVERT_TO_STR = -7, OPN_NOT_FOUND = -8, OPN_ERROR_DRIVER_INVALID = -9, OPN_ERROR_INTERNAL = -10 }`

APIの処理結果を返すための定数

---

### 列挙型

---

`enum OPNResult`

APIの処理結果を返すための定数

列挙型の値

`OPN_SUCCESS` 処理成功

`OPN_CONNECTION_ERROR` 接続失敗

`OPN_ARGUMENT_ERROR` 引数が不正

`OPN_CONNECT_TIMEOUT` connectタイムアウト

`OPN_ACCEPT_TIMEOUT` acceptタイムアウト

`OPN_AUTH_FAILED` 認証失敗

`OPN_DISABLED_BLUETOOTH` Bluetoothアダプタ無効

`OPN_ERROR_CONVERT_TO_STR` 文字列への変換に失敗

`OPN_NOT_FOUND` 見つからない

`OPN_ERROR_DRIVER_INVALID` 適切なドライバーが見つからない

`OPN_ERROR_INTERNAL` 内部エラー

---

## OPNSettingChars.h

### 構成

- `struct SettingChar`  
文字列設定コマンドの構造体
- `class OPNSettingChars`

### *SettingChar*

構造体を扱いやすくしたヘルパークラス 一覧の取得や、プレフィックス・サフィックスからコマンドを検索することができます。 このクラスはアプリに1つのみ存在する[GoF]Singletonクラスです。

### マクロ定義

- `#define OPN_SETTING_CHAR_NUM 142`  
*OPNSettingCharsの個数*

---

### マクロ定義

---

```
#define OPN_SETTING_CHAR_NUM 142

    OPNSettingCharsの個数
```

---

## OPNSettings.h

### 構成

- **class OPNSettings**

データコレクタから取得したバイトデータの設定値を取得するためのメソッドを提供します。

弊社製品名: OPN-2002i Series  
OPN-3002i Series  
発行管理番号: DM-130708  
管理番号: SI13028

---

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>