

Delphi Xe10 用
Uni Usb Host 通信
コンポーネント v1.08

[Windows/Android Usb Host アプリケーション用]

(Bulk 通信用)

内容

1. Uni Usb Host 通信コンポーネント使用許諾契約.....	2
2. <i>Window 用ドライバディスクの作成</i>	3
2-1. WINDDK を入手し、インストールします。参考 Version7.60.....	3
2-2. INF ファイル群を作成します。.....	3
2-2-1. ファイルをコピーします。.....	3
2-2-2. inf ファイルの作成.....	4
2-2-3. inf ファイルの内容を変更します。.....	5
3. Delphi へコンポーネント登録(コンパイル済みユニット版).....	6
3-1. プロジェクトを開くで uniusbsample を選びます。.....エラー! ブックマークが定義されていません。	
3-2. コンポーネントをインストールします。.....エラー! ブックマークが定義されていません。	
4. PATH の設定.....	7
4-1. PROJECT 側のパスの設定で、uniusb.dcu のパスを追加します。.....	7
5. <i>Android 時の Manifest ファイルの改変</i>	9
5-1. Manifest ファイルを下記のリンクを参照し改変します。.....	9
5-2. リソースファイルの作成.....	9
5-3. 配置の設定.....	10
6. 通信コンポーネントの使い方.....	11
6-1. プロパティ.....	11
6-2. メソッド.....	11
6-3. イベントメソッド (アンドロイドのみ).....	13
7. 機能説明、機能的制限事項.....	13



マークは有限会社 シー・エス・ディーの登録商標です。

その他の商標はその所有者に帰属します。

「Uni Usb Host 通信 コンポーネント」

copyright (c) 2014-2015 有限会社シー・エス・ディー

e-mail: info2@csd.co.jp

1. Uni Usb Host 通信コンポーネント使用許諾契約

Uni Usb Host コンポーネントの使用に関し、有限会社シー・エス・ディー（以下「CSD」といいます。）と、本件ソフトウェアの全部又は一部をハードディスク等の記憶装置へ保存し、又は本件ソフトウェアに含まれるプログラムをコンピュータ上で実行するお客様との間で締結される契約です。

第1章 契約の成立、効力及び終了

(1) お客様は、本件ソフトウェアの全部又は一部をコンピュータのハードディスク等の記憶装置へ保存したとき、又は 本件ソフトウェアを使用したときは、本契約の締結に同意したものとみなされます。このお客様の同意をもって、本契約は成立し、効力を生じます。

(2) お客様は、保存した本件ソフトウェアの全てを削除することにより、本契約を終了させることができます。

(3) CSD は、独自の判断に基づき、本契約を終了することができます。

(4) お客様は、理由を問わず、本契約の終了について CSD に対し補償金その他いかなる名目での支払いも請求することはできないものとします。

第2章 使用条件及び、禁止事項

(1) CSD は、お客様に対し、Uni Usb Host 通信コンポーネントのコンパイル済ユニットファイルやオブジェクトファイルを提供いたします。お客様は Uni Usb Host コンポーネントのコンパイル済ユニットファイルを用いて御自由に開発及び、配布を行う事ができます。（台数等の制限はございません。）サンプル版は 200 回以上通信をしますとエラー255を発行いたします

(2)CSD は、ソース有償版として、Uni Usb Host 通信コンポーネントのソースファイルを提供いたします。

このソース使用して、お客様は実行ファイルの販売を自由に行う事ができます。

また、お客様が納品時のファイル群の 1 ファイルとして Uni Usb Host コンポーネントのソースファイルを配布する事は可能といたします。ただし、納品後も含め、ソース改変後、改変前にかかわらず、ソース単体での販売、もしくは、不特定多数が閲覧可能な状態でのソース公開、開示は、禁止事項といたします。

第3章 著作権の帰属

本件ソフトウェアの著作権は、全て CSD に帰属します。

第4章 免責

(1) CSD は、お客様、その他の第三者が本件ソフトウェアに関連して直接、間接に蒙ったいかなる損害に対しても、賠償等の一切の責任を負わず、かつ、お客様はこれに対して CSD を免責するものとします。

(2) CSD はお客様に対し、本件ソフトウェアの動作保証、使用目的への適合性の保証、商業性の保証、使用結果についての的確性や信頼性の保証、第三者の権利侵害及び瑕疵担保義務も含め、いかなる責任も一切負いません。CSD がこれらの可能性について事前に知らされていた場合も同様です。

(3) CSD は独自の判断に基づき、本件ソフトウェアの仕様又は内容の変更、修正、配布方法等の変更、対価の設定、変更を予告なくすることができます。

(4) CSD からお客様に提供される本件ソフトウェアにかかる情報についても、直接間接を問わず、本条各項の規定が適用されます。

本契約に関する紛争は、前橋地方裁判所 太田支部を第一審の専属の合意管轄裁判所とします。

2. Window 用ドライバディスクの作成

2-1. WINDDK を入手し、インストールします。参考 Version7.60

(参考 <http://www.microsoft.com/en-us/download/details.aspx?id=11800>)

2-2. INF ファイル群を作成します。

2-2-1. ファイルをコピーします。

(\WinDDK\7600.16385.1\redist\winusb と \WinDDK\7600.16385.1\redist\wdf から
下記の様なフォルダ及びファイル構成でコピーします。

```
├──amd64
|   WdfCoInstaller01009.dll
|   winusbcoinstaller2.dll
|   WUDFUpdate_01009.dll
├──ia64
|   WdfCoInstaller01009.dll
|   winusbcoinstaller2.dll
|   WUDFUpdate_01009.dll
└──x86
    WdfCoInstaller01009.dll
    winusbcoinstaller2.dll
    WUDFUpdate_01009.dll
```

2-2-2. inf ファイルの作成

参考用の inf ファイルから作成する方法を記載いたします。

フォルダーsampleinfo 内にある inf ファイルを 2-2-1.で作成したフォルダーの root にコピーします。

```
|      SampleInfo.Inf
|      └──amd64
|          WdfCoInstaller01009.dll
|          winusbcoinstaller2.dll
|          WUDFUpdate_01009.dll
|      └──ia64
|          WdfCoInstaller01009.dll
|          winusbcoinstaller2.dll
|          WUDFUpdate_01009.dll
└──x86
    WdfCoInstaller01009.dll
    winusbcoinstaller2.dll
    WUDFUpdate_01009.dll
```

2-2-3. inf ファイルの内容を変更します。

[添付の inf ファイルから下記の内容を変更します。

今回は Led Demo としています。赤色の部分は必ず変更します。

2-3-1.バージョン日付、class 名、guid を変更します。

```
[Version]
Signature = "$Windows NT$"
Provider  = %ProviderName%

DriverVer = 31/07/2013,1.0.0.0
Class = LedDemoDeviceClass
ClassGuid= {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx}
```

2-3-2.PID&VID の変更

; ===== Manufacturer/Models sections =====

[Manufacturer]

%ProviderName% = MyDevice_WinUSB,NTx86,NTamd64,NTia64

[MyDevice_WinUSB.NTx86]

%USB\MyDevice.DeviceDesc% =USB_Install, USB\VID_YYYY&PID_XXXX

[MyDevice_WinUSB.NTamd64]

%USB\MyDevice.DeviceDesc% =USB_Install, USB\VID_YYYY&PID_XXXX

[MyDevice_WinUSB.NTia64]

%USB\MyDevice.DeviceDesc% =USB_Install, USB\VID_YYYY&PID_XXXX

; ===== Installation =====

受け側の usb デバイスと同じ Vid 及び pid にします。

2-3-3.デバイスインターフェイス GUID を変更します。(この値はコンポーネント側でも使います。

この値をコンポーネントに StringGuid に設定します{...} まで)

[Dev_AddReg]

HKR,,DeviceInterfaceGUIDs,0x10000,"{yyyyyyyy-yyy-yyy-yyy-yyy-yyyyyyyyyyyy}"

; ===== Strings =====

2-3-4.文字列の変更

[Strings]

ProviderName="Led Demo InterFace"

USB\MyDevice.DeviceDesc="Led Demo InterFace"

WinUSB_SvcDesc="LedDemoInterFace"

DISK_NAME="LedDemoInterFace Install Disk"

ClassName="LedDemoDeviceClass"

上記で、ドライバインストールディスクの作成は完了です。

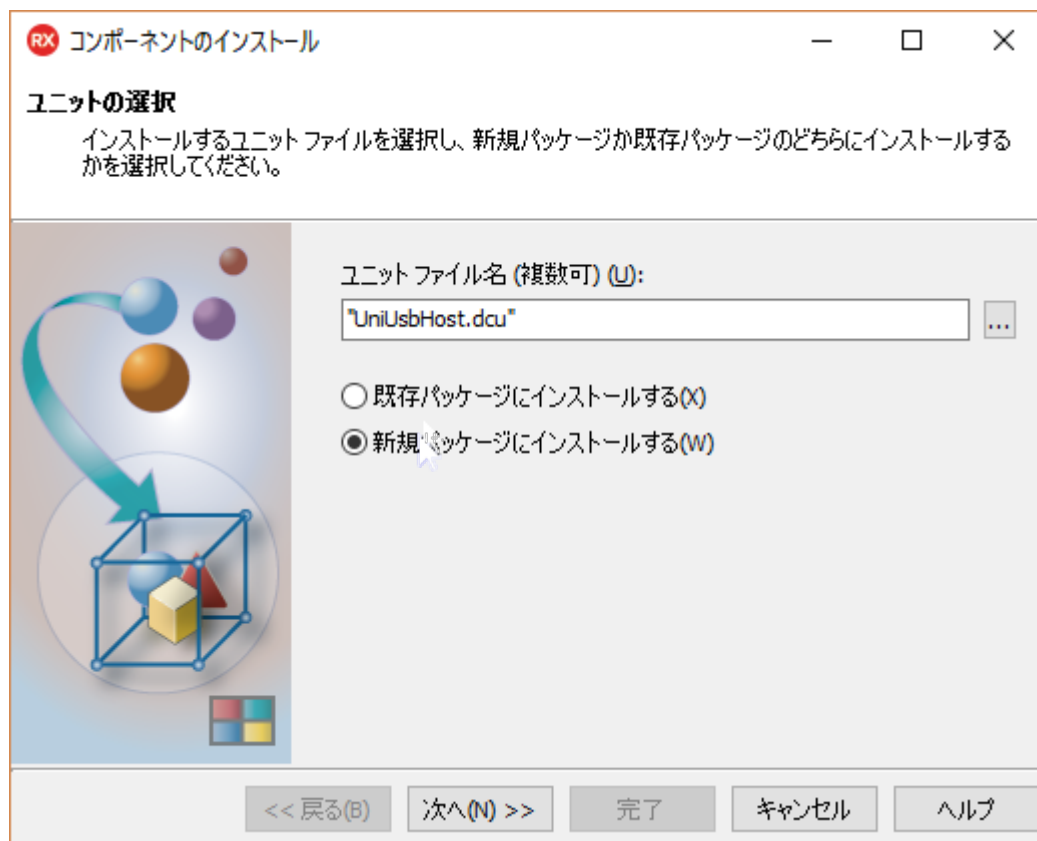
3. Delphi へコンポーネント登録(コンパイル済みユニット版)

C:\Yamsho\usbdcu に解凍します。

3-1. メニュー=>コンポーネントのインストールを選びます。

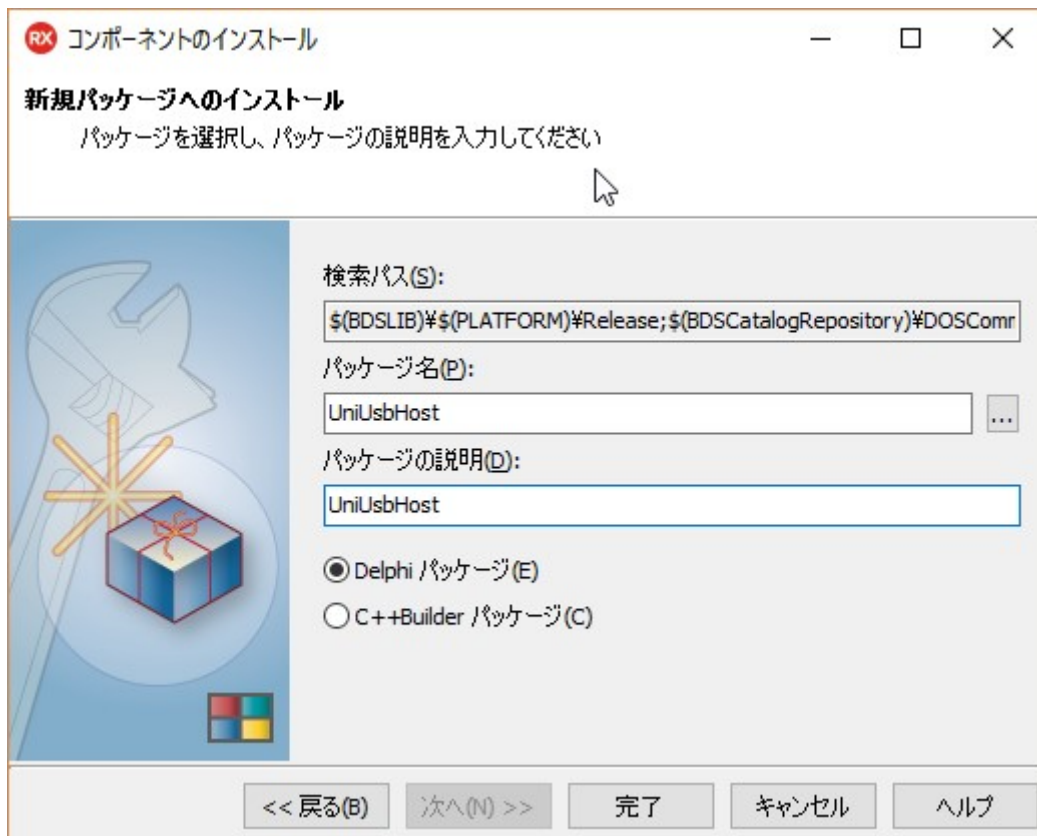


3-2. コンポーネントファイルを選択します。(このとき新規パッケージを選択します。)



3-3. パッケージ名およびパッケージの説明文を入力します。

内容は任意です。ここではとりあえず UniUsbHost としています。



完了を押して、インストールは完了です。

4. PATH の設定

4-1. PROJECT 側のパスの設定で、uniusbhost.dcu のパスを追加します。

Win32 時は¥win32 を設定します。

Win64 時は¥win64 を設定します。

Android 時で Debug 時は¥Android¥debug を設定します。

Android 時で Release 時¥Android¥release を設定します。

TestPrj.exe のプロジェクト オプション (Win32 - Release)

ターゲット(T): Release 構成 - 32 ビット Windows プラットフォーム 適用(A)... 保存(S)...

DCP 出力ディレクトリ	
パッケージの出力ディレクトリ	
ユニット スコープ名	Winapi;System.Win;Data.Win;Datasnaps.Win;Web.Win;S
ユニットのエイリアス	
ユニットの出力ディレクトリ	.\$\$(Platform)\$(Config)
出力ディレクトリ	.\$\$(Platform)\$(Config)
条件定義	RELEASE
検索パス	C:\XE5 ...

OK キャンセル ヘルプ

5. Android 時の Manifest ファイルの改変

5-1. Manifest ファイルを下記のリンクを参照し改変します。

<http://developer.android.com/guide/topics/connectivity/usb/host.html>

色の付いた部分を追加します。

```
<manifest ...>
    <uses-feature android:name="android.hardware.usb.host" />
    <uses-sdk android:minSdkVersion="12" />
    ...
    <application>
        <activity ...>
            ...
            <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
                android:resource="@xml/device_filter" />
            ...
            ...
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

次にリソースファイルを device_filter.xml 作成します。

5-2. リソースファイルの作成

上記 1.で指定しました。リソースファイル *device_filter.xml* を作成し、

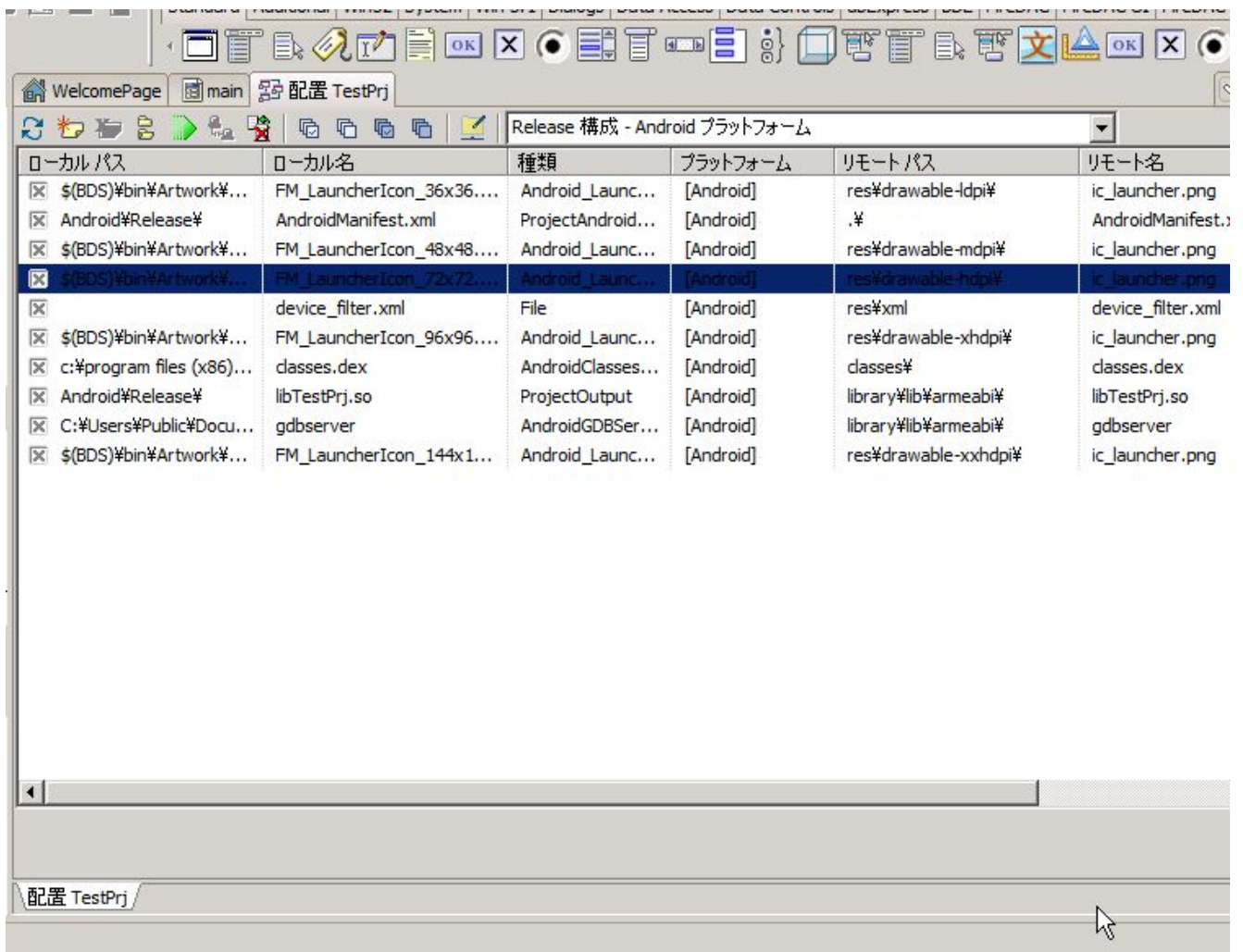
対応のベンダー ID とプロダクト ID 10 進数で USB デバイスの ID に変更します。

```
version="1.0" encoding="utf-8"?>
<resources>
    <usb-device vendor-id="YYYY (DEC)" product-id="XXXX (DEC)" class="255" subclass="66" protocol="1" />
</resources>
```

例 vendor-id="100" product-id="0001"

5-3. 配置の設定

プロジェクト配置にて、device_filter.xml 追加し、リモートパスを res¥xml とします。



6. 通信コンポーネントの使い方

6-1. プロパティ

1.GuidString .デバイスインターフェイス GUID を文字列設定します。

EX) '{01234567-89ab-cdef-0123-456789abcdef}'

2.ProductId .プロダクト ID 4文字を指定します。

EX) '012A' (注意プロパティは文字列ですが、16進数です。)

3.TAG 自由にお使いください。

4.VenderId .ベンダーID 4文字を指定します。

EX) 'ABCD' (注意プロパティは文字列ですが、16進数です。)

4.WaitTime アンドロイド時の Wait Time 設定です

EX) 100

5 BulkPipe[0..7] パイプのインフォメーション格納エリア、この要素を指定して、入出力を行います。

(実行時専用のプロパティです。Open 後にリードしてください。)

Open メソッド実行後、BulkPipe の内容にて最上位 bit が立っていれば、入力としてエンドポイント番号となり、立っていなければ、出力としてのエンドポイント番号となります。

例：BulkPipe の内容

Index	0	1	2	3	4	5	6	7
内容	\$81	\$02	\$00	\$00	\$00	\$00	\$00	\$00

サンプルのソフトの場合 Index0 が Endpoint1 /Index 1 が Endpoint2 となります。

6-2. メソッド

1.Open USB のポートをオープンします。返値はバイトです。0 で成功です。

2.Close USB のポートをクローズします。返値はバイトです。0 で成功です。

3. Write(PiPeNo : Byte ; Size : Cardinal ; const Buffer : PByte) 返値はバイトです。 0 で成功です。

引数 PipeNo: BulkPipe の要素番号を指定します。 Size 送信サイズを指定します。

BUFFER バッファのポインタをします。

注意：通信間があまりにも早いとチップセットにより通信がうまくできない場合があります。

その場合は待ち時間 sleep(5)等を入れ調整します。

4.Read (PiPeNo : Byte ; Size : Cardinal Buffer : PByte) 返値はバイトです。 0 で成功です。

引数 PipeNo: BulkPipe の要素番号を指定します。 Size 送信サイズを指定します。

BUFFER バッファのポインタをします。

注意：通信間があまりにも早いとチップセットにより通信がうまくできない場合があります。

その場合は待ち時間 sleep(5)等を入れ調整します。

5.Error2Str(OpenErrorCode: BYTE)エラー内容を文字列に変換します。

引数：各メソッドの返回值を入れると文字列が出力されます。

返回值については以下の文字列になっています。

表 1.返回值及び文字列変換時変換値(windows 時)

値	意味	文字列
0	正常	NO Error
1	既に USB OPEN 済み	Already Usb Open
2	USB ドライバーがインストールされていない。	Please Usb Driver Install
3	USB が接続されていない	Please Usb Connect
4	GUID が異なる。	Please Check Guid
5	USB ポートが OPEN できない。	Can not Open Usb
6	USB が初期化できない。	Can not Initate Usb
7	USB のインフォメーションがロードできない。	Can not Read Usb Infomation
8	USB I/F 設定ができない。	Can not Read Usb IF Setting
9	USB ポートが OPEN されていない。	Please Open Usb Port
10	PIPE/ENDPOINT 番号をチェックしてください。	Check PipeNo/EndPoint [Write]
11	USB ポートが OPEN されていない。	Please Open Usb Port
12	PIPE/ENDPOINT 番号をチェックしてください。	Check PipeNo/EndPoint [Read]
その他	不明のエラー	UnKnown Error
255	規定回数オーバー	UnKnown Error

表 2.返回值及び文字列変換時変換値 (Android 時)

値	意味	文字列
0	正常	NO Error
1	既に USB OPEN 済み	Already Usb Open
2		
3	USB が接続されていない	Please Usb Connect
4		
5	USB が許可されない。	Can not Open Usb
6	USB が初期化できない。	Can not Initate Usb
7		
8		
9	USB ポートが OPEN されていない。	Please Open Usb Port
10	規定バイト数送信できない。	Transmission size error
11	USB ポートが OPEN されていない。	Please Open Usb Port
12	規定バイト数受信ができない。	Reception size error
その他	不明のエラー	UnKnown Error
255	規定回数数オーバー	UnKnown Error

6-3. イベントメソッド (アンドロイドのみ)

アンドロイドのみですが、USB の挿抜きのイベントです。

1. OnUsbAttach
USB の挿入イベントです。
2. OnUsbDettach
USB の抜きイベントです。

7. 機能説明、機能的制限事項

1. WINDOS 側の認識は、複数台可能な設計となっています。

アンドロイド側は単一な機器のみをサポートしております。

よって、アンドロイド側は、直接接続した、1 台のみ

をサポートします。(USB ハブ不使用)