

# Delphi 11.3 用 Uni232c 通信 コンポーネント

---

[Windows/Android アプリケーション用]



## 内容

1. Uni232c 通信コンポーネント使用許諾契約.....	2
2. Delphi へコンポーネント登録(コンパイル済みユニット版).....	3
2-1. メニュー=>コンポーネントのインストールを選びます。 .....	3
2-2. 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
2-3. パッケージ名およびパッケージの説明文を入力します。 .....	4
3. サンプルプロジェクトを動かす。 .....	5
4. PATH の設定 .....	5
4-1. パスの追加.....	5
5. Android 時の Manifest ファイルの改変 .....	6
5-1. Manifest ファイルを下記のリンクを参照し改変します。 .....	6
5-2. リソースファイルの作成 .....	7
5-3. 配置の設定.....	7
6. コンポーネントの使い方 .....	8
6-1. プロパティ .....	8
6-2. メソッド .....	9
6-3. イベントメソッド (アンドロイド時のみ) .....	10
7. 各 IC 対応表(アンドロイド時).....	11
8. 確認 IC .....	11
9. 機能説明、機能的制限事項 .....	12



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。

「Uni232c 通信 コンポーネント」

copyright (c) 2015..2023 有限会社シー・エス・ディー

「Uni232c 通信 コンポーネントはダイレクトに USB 通信をします。

従って各メーカーの提供するライブラリーをインポートする必要はありません。」

問い合わせ先

e-mail: info2@csd.co.jp



## 1. Uni232c 通信コンポーネント使用許諾契約

Uni232c 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、Uni232c 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 Uni232c 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。）評価版は CSD の規定の日時まで規定回数に達するまで試用できます。指定の日時を過ぎますと、配布物も含め、動作いたしません。

2)CSD は、有償にて、Uni232c 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御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 Uni232c 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。

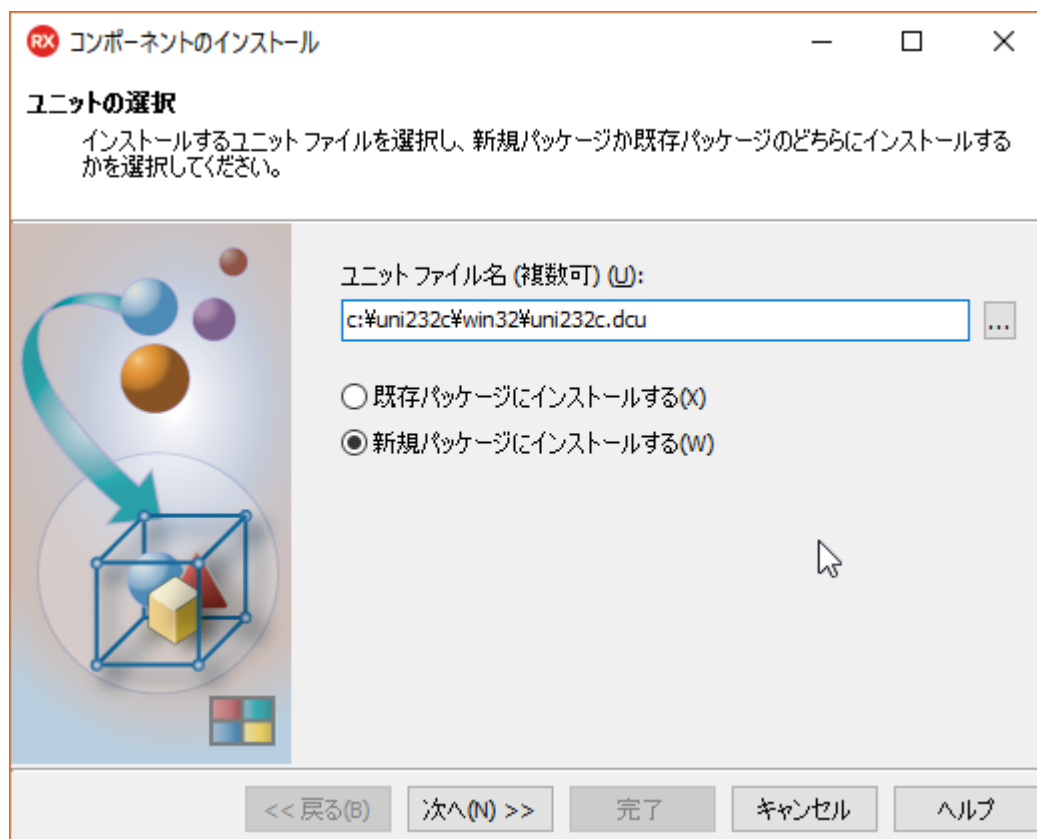
## 2. Delphi へコンポーネント登録(コンパイル済みユニット版)

C:\Uni232c\dcu に解凍します。

### 2-1. メニュー⇒コンポーネントのインストールを選びます。



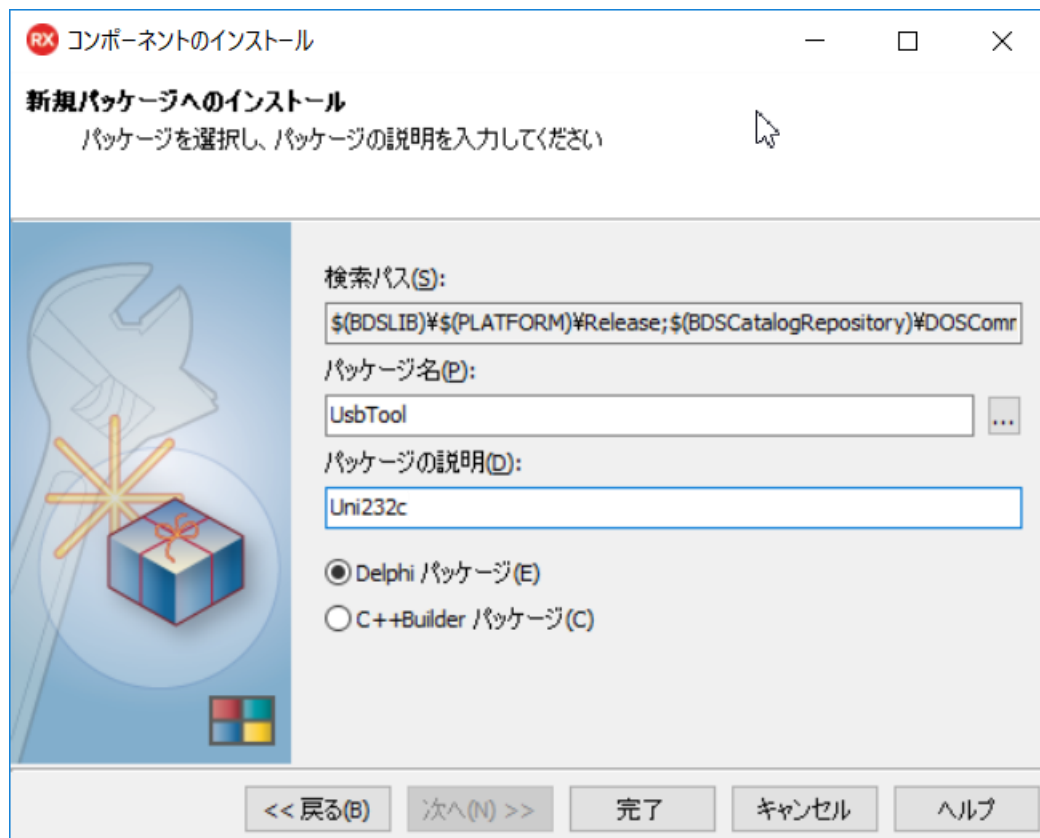
### 2-2. コンポーネントファイルを選択します。(このとき新規パッケージを選択します。)





### 2-3. 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。ただし、パッケージ名はファイル名と異なる様にします。  
下記は UsbTool としています。



完了を押して、インストールは完了です。

実行時には、コンポーネントファイルへの Path の設定が必要です

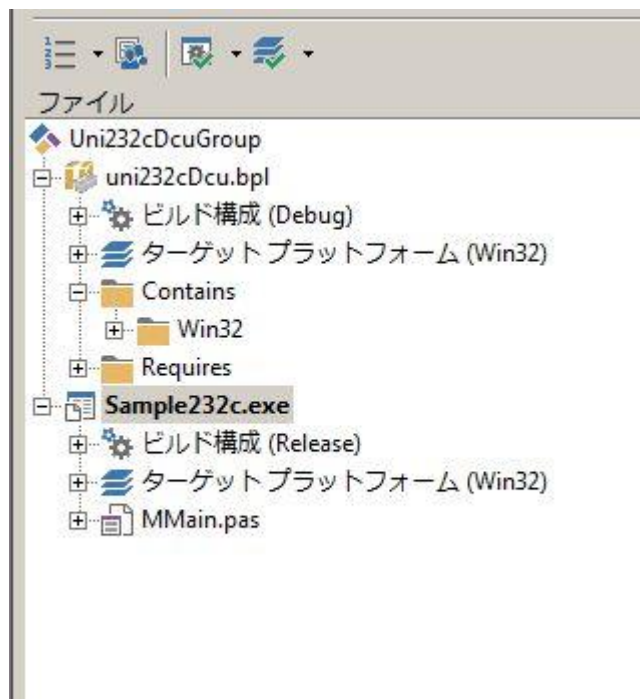
インストールは完了です。実行時、コンパイル時には、コンポーネントファイルへの Path の設定が必要です。



### 3. サンプルプロジェクトを動かす。

既存のプロジェクト追加で Sample232c.dpr を選択し、android もしくは windows で動作確認します。

Windows 時はお使いの 232c のポート番号を、uni232c コンポーネントの port に御使用になる Com ポート番号を設定してください



### 4. PATH の設定

#### 4-1. パスの追加

PROJECT 側のパスの設定で、Uni232c.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。

Android32BIT 時で Debug 時は¥Android¥debug を設定します。

Android32BIT 時で Release 時は¥Android¥release を設定します。

Android64BIT 時で Debug 時は¥Android64¥debug を設定します。

Android64BIT 時で Release 時は¥Android64¥release を設定します。

**注意：**プロジェクト->オプション “Delphi コンパイラ”の部分の検索パスの部分で設定してください。



## 5. Android 時の Manifest ファイルの改変

Test232c ではすでに記載しています。

(Delphi では AndroidManifest.template.xml を編集してください。)

### 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>
```

また、Android 12 でご使用の場合には

```
<activity android:name="com.embarcadero.firemonkey.FMXNativeActivity"
    android:label="%activityLabel%"
    android:configChanges="orientation | keyboardHidden"
    android:exported="true"
    android:launchMode="singleTask">
    ...
```

赤色の部も追加してください。

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>
```

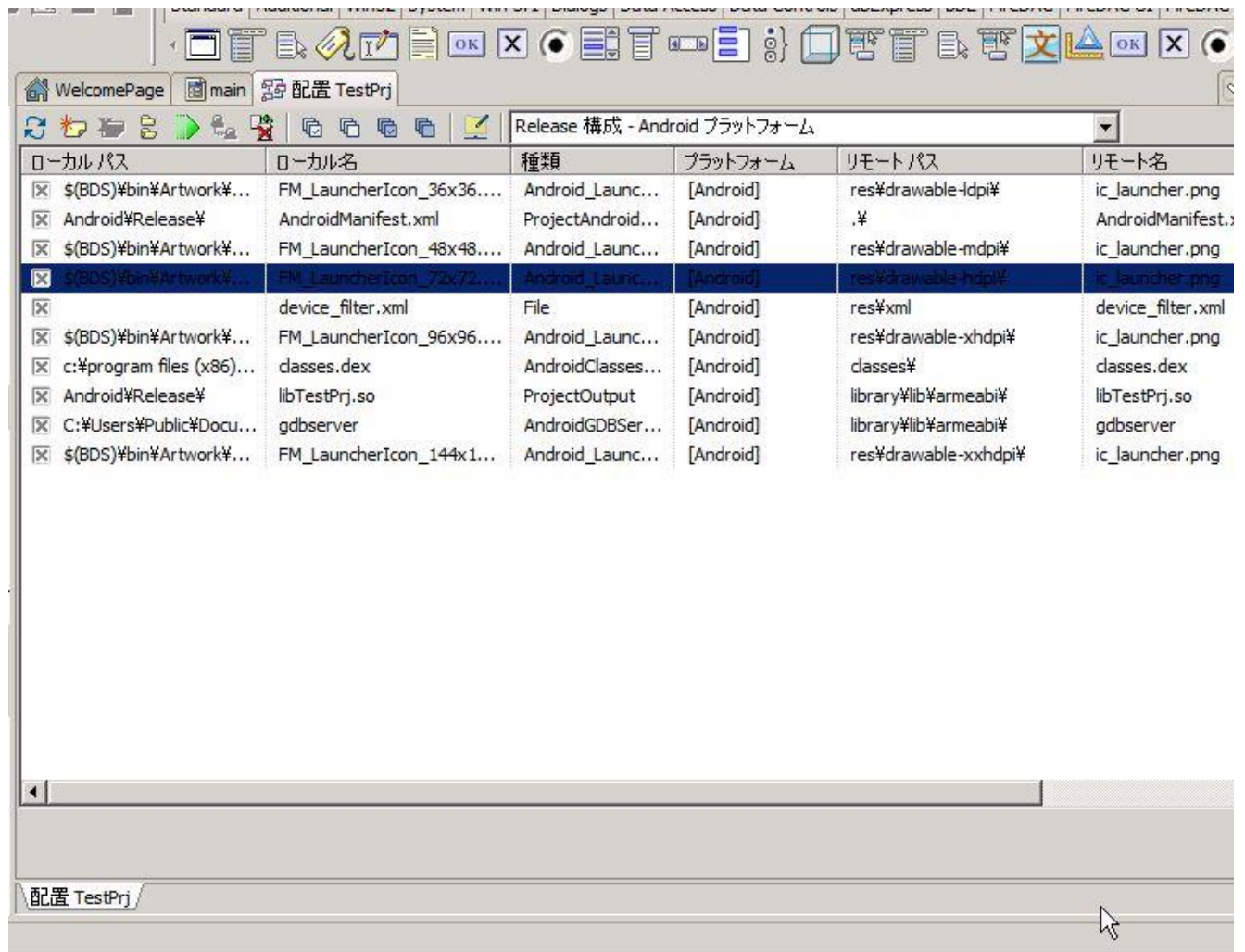
```
  <usb-device vendor-id="YYYY (DEC)" product-id="XXXX (DEC)" />
```

```
</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. コンポーネントの使い方

### 6-1. プロパティ

1.BaudRate **ボーレートの通信速度を設定します。**

EX) 256000,128000,115200, 57600,56000,38400, 19200,14400,9600 等、  
(各 IC の許される範囲で設定ください。)

2.ByteSize **データビット長を設定します。**

EX) (Bit7, Bit8)

3.FlowControl **フローコントロールを設定します。**

CtrlNone,CtrlRtsCts,CtrlDtrDsr,CtrlXonXoff

EX) (CtrlDtrDsr,CtrlXonXoff は Ic によって制御できない場合もあります。)

4.ParityBits **パリティビットを設定します。**

EX) (ParityNone, ParityOdd, ParityEven, ParityMark, ParitySpace)

5.Stop Bit **ストップビットを設定します。**

EX) (Stopbit1, Stopbit2)

6.WaitTime **アンドロイド時の USB 送信時に使用する Wait Time 設定です**

EX) 100

7.Port **Windows 時の com ポート番号を指定します。**

EX) 1 2 3 etc

8.CdcProductId1, CdcProductId2 **アンドロイド時の CDC 認識 Product Id です。**

CDC として認識する Product Id です。 \$ffff を共に設定した場合は、Productid 及び、venderid のチェックは無効となり、すべて cdc 扱いとなります。 1 と 2 で最大 2 つまで認識できます。

9.CdcVenderId1, CdcVenderId2 **アンドロイド時の CDC 認識 VenderId です。**

CDC として認識する VenderId 0 は無効となります。 1 と 2 で最大 2 つまで認識できます。



## 6-2. メソッド

1.Open 232C 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 232C 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 Write( Size : cardinal ; Buffer : PByte ):Integer;

引数：サイズ、書込サイズ(64 以下を設定)です。 Buffer 書込バッファのポインタです。

戻り値が 0 以上であれば、成功です。戻り値は、書込んだ数が戻ります。

4.Read Read( Size : cardinal ; Buffer : PByte ):Integer;

引数：サイズ、読み込みサイズ(64 以下を設定)です。 Buffer 読み込みバッファのポインタです。

戻り値が 0 以上であれば、成功です。1 以上で読み込数が戻ります。

5.GetModemStatus GetModemStatus():Integer

戻り値が 0 以上であれば正常に正常に入力です。

0bit 目を起点として、CTS が 4bit 目、DSR が 5bit 目、Ri が 6bit 目、RTCD が 7bit 目に入り戻ります。

6.SetModemStatus SetModemStatus(State: Word):Integer

戻り値が 0 以上であれば正常に出力できました、0bit 目を起点として、DTS が 1bit 目、RTS が 2bit 目となります。（IC によって正常に出力できない IC もあります。）

7.SetModemBreak SetModemBreak (waitms: Cardinal):Integer;

戻り値が 0 以上であれば正常に出力、引数としてミリ秒オーダーの値をいれます。

その時間だけブレイク信号を送信

8.GetVersion GetVersion():String

コンポーネントのバージョンを文字列で出力します。

EX) 'Uni232c Version 1.03'



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

引数：各メソッドの返値を入れると文字列が出力されます。

返値については以下の文字列になっています。

表 1.返値及び文字列変換時変換値 (Android 時)

値	意味	文字列
0	正常	NO Error
-2	既に USB OPEN 済み	Already Usb Open
-3	USB が接続されていない	Please Usb Connect
-4	デバイスがオープンできない。	Can not Open Device
-5	デバイスがオープンできない。	Can not Open Device
-6	デバイスがオープンできない。	Can not Open Device
-7	使用期限終了	The end of the expiration date
-8	Write Buffer Error	Write Buffer Error
-9	Read Buffer Error	Read Buffer Error
-10	使用制限回数 over	Trial 50 Times
その他	不明のエラー	UnKnown Error

### 6-3. イベントメソッド (アンドロイド時のみ)

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. 各 IC 対応表(アンドロイド時)

	フロー制御				備考
	NONE	CTS/RTS	DSR/DTR	XON/XOFF	
FTDI 232XX 231XX Etc..	可	可	可	可	
Prolific PL2303 REV.X	可	可	不可	可	DSR/DTRはREV.D以降のチップでは対応されている模様。
SILICON LABS CP230X	可	可	可	可	
南京沁恒电子有限公司 CH34X	可	可	可	不可	ハード的に XON/XOFF が無いと思われる。DSR/DTR ポート直コントロールに問題あり
Usb cdc	可	不可	不可	不可	

表 1.IC 対応表

**注意：WINDOWS 時は WINDOWS にインストールしているドライバーの動作に従います。**

## 8. 確認 IC

弊社では、FTDI FT232RL、Prolific PL2303 Rev.HX 及び RevHxD、SLICON LABS CP2102、Wch340G、PSOC 4200、MICROCHIP USB CD 使用して確認しております。

PL2303 では FlowControl Dtr/Dsr は IC よって機能の有無があります。  
よって FlowControl Dtr/Dsr を設定をしないでください。

必ずお使いになる。IC で必ず動作確認の上、リリースしてください。  
また、動作させたい IC 等がありましたら、御相談ください。



## 9. 機能説明、機能的制限事項

**USB シリアル変換 IC を直接接続した、1 台のみをサポートします。**

**(USB ハブ不使用)**

## 10. リリース note

Ver 112:C++ ビルダで起動後エラーになる問題に対応。

Ver 112a:Android 64bit 版を付属