Future Technology Devices International Ltd.

Technical Note

TN_147 Java D2xx for Android

Document Reference No.:FT_000799

Version 1.2

Issue Date: 2013-09-16

This document describes the installation and use of the FTDI Java D2XX for Android driver for FTxxxx devices in an Android environment.

Future Technology Devices International Limited (FTDI)

Unit 1,2 Seaward Place, Glasgow G41 1HH, United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
E-Mail (Support): support1@ftdichip.com **Web:** http://www.ftdichip.com

Copyright © 2013Future Technology Devices International Limited

Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use.

Table of Contents

Copyright © 2013Future Technology Devices International Limited

1 Introduction

FTDI provides a proprietary D2XX interface for easy communication with FTxxxx devices. The D2XX API is common across several operating systems supported by FTDI, namely Windows, Windows CE, Linux, Mac OS X and Android.

1.1 Android Support

To support the Google Android OS, FTDI has two D2XX solutions for different application scenarios:

1. A Java class using the JNI (Java Native Interface) to access the API of a pre-compiled Linux D2XX library. This solution is applicable to all versions of Android, but requires special root privilege on USB-related device nodes. This is for reusing existing Linux designs in Android applications or creating projects for Android OS predating version 3.2.
2. A Java class library supporting USB Host is available and applicable to Android v3.2or any later series. This library requires no special root access privileges.

The first solution is described in TN_134 FTDI Android D2XX Driver.

This document describes the second solution which can be easily adapted by Android application developers, i.e. the Java D2xx API for Android. The API is packaged in d2xx.jar and is distributed with demo application source code, which can be downloaded from FTDI's website.

1.2 Prerequisites

The following is required to install the FTDI Java D2xx for Android driver:

- An Android device running v3.2 or later OS, with a USB Host or OTG interface;
- An FTDI-based device for testing.
    - FTDI testing was conducted using a Google Nexus 7, and an FT232R-based US232R cable.

To develop an application using the FTDI Java D2XX for Android driver, the development machine needs the Eclipse IDE and an up-to-date Android SDK, including the ADB program and the ADT plug-in for Eclipse. Installation and configuration of these tools is not described in this document but is outlined on the Android developer web site. Please see (http://developer.android.com/sdk/index.html).

Also, the Android device should have USB Debugging enabled to allow access using the ADB utility. To accomplish this, navigate to Settings > Applications > Development and select the USB debugging option.

A summary of the required configuration is provided in the diagram below.

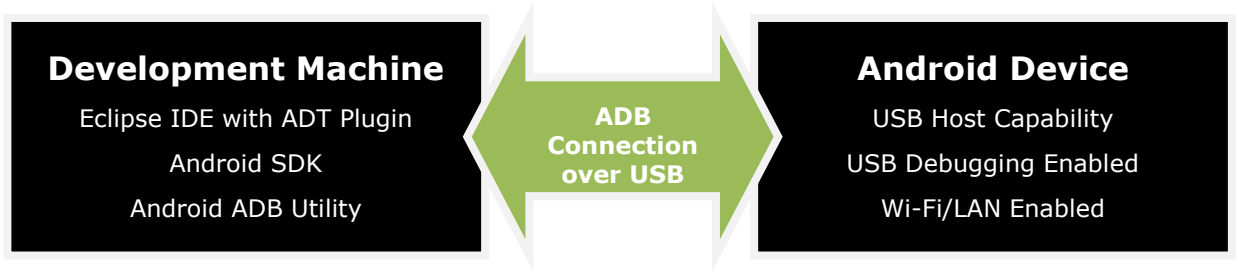

Figure 1: Android Development Configuration

Copyright © 2013Future Technology Devices International Limited

2 Using the FTDI Java D2xx for Android Library

2.1 Introduction and Usage

To support versatile tablet usage scenarios, Google has included a USB Host API in Android since version 3.1. Before version 3.1, an Android application could not access USB devices attached to a system naturally without root access rights. The Android USB Host API removes this limitation, allowing us to utilize the USB gadgets attached to Android Host or OTG port.

FTDI provides a Java class library that adapts to applications so the developer can focus on the desired input and output data. The design goal of the class library is to provide access to all the D2XX functions including, EEPROM functions.

The D2xx library can be included in an Android application project in Eclipse easily. First, copy the library file, d2xx.jar, to the folder of the project, and add it in "Project"–"Properties". Refer to Figure 2 ~ Figure 4, which show how to add the library file from the "\libs" sub-folder of the project folder.

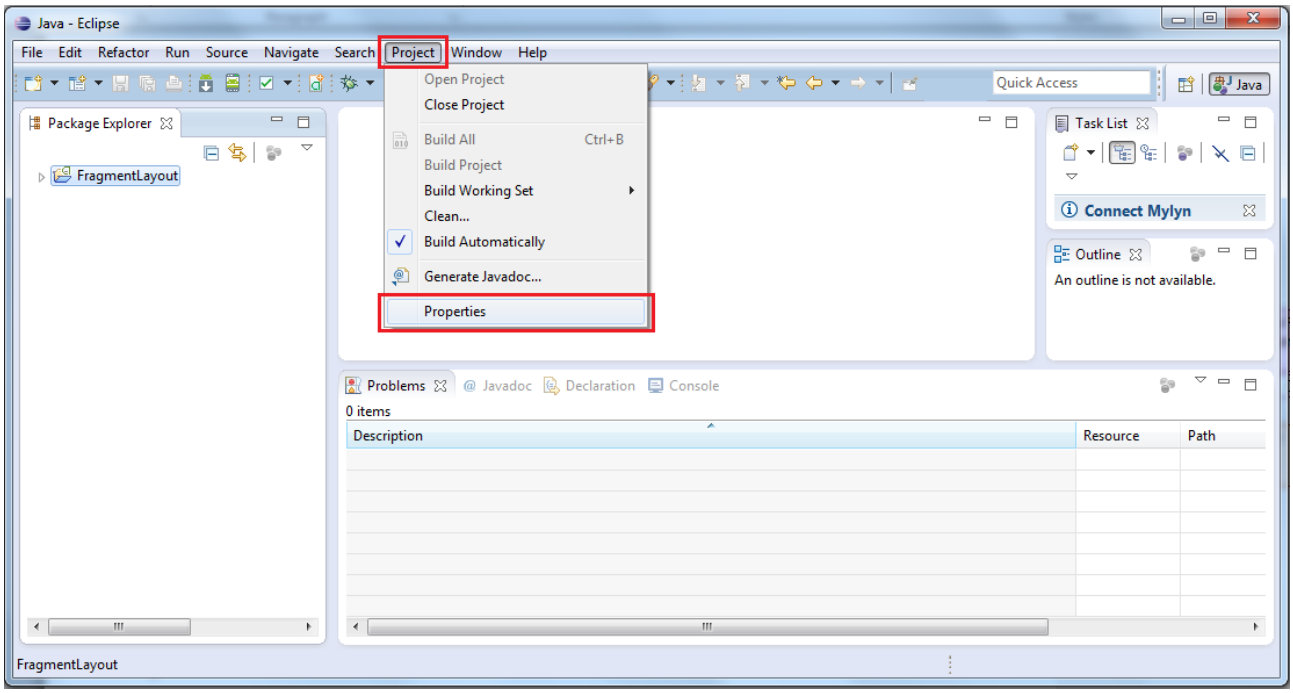

Figure 2: Select "Project" – "Properties"

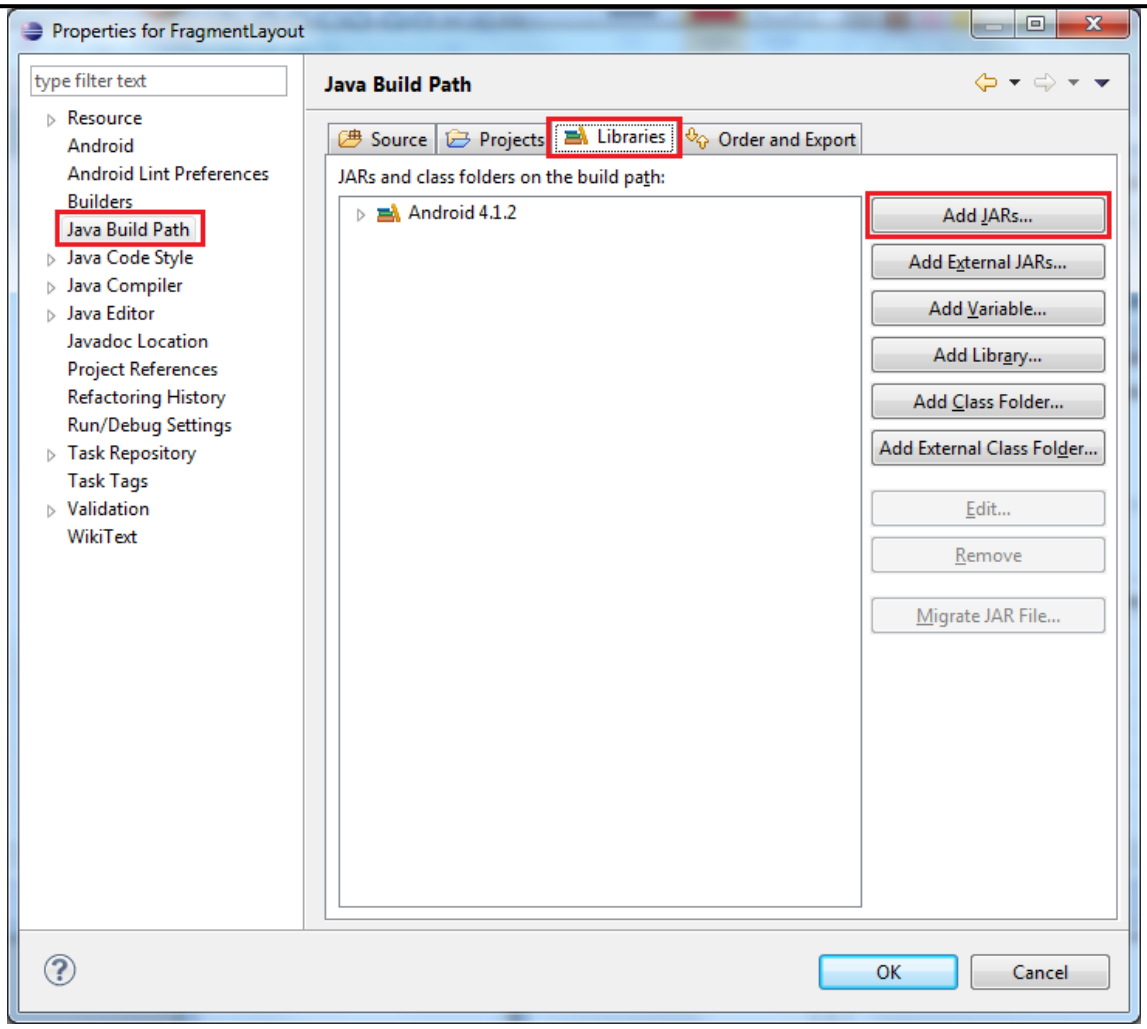

Figure 3: Select "Java Build Path" – "Libraries" – "Add JARs"

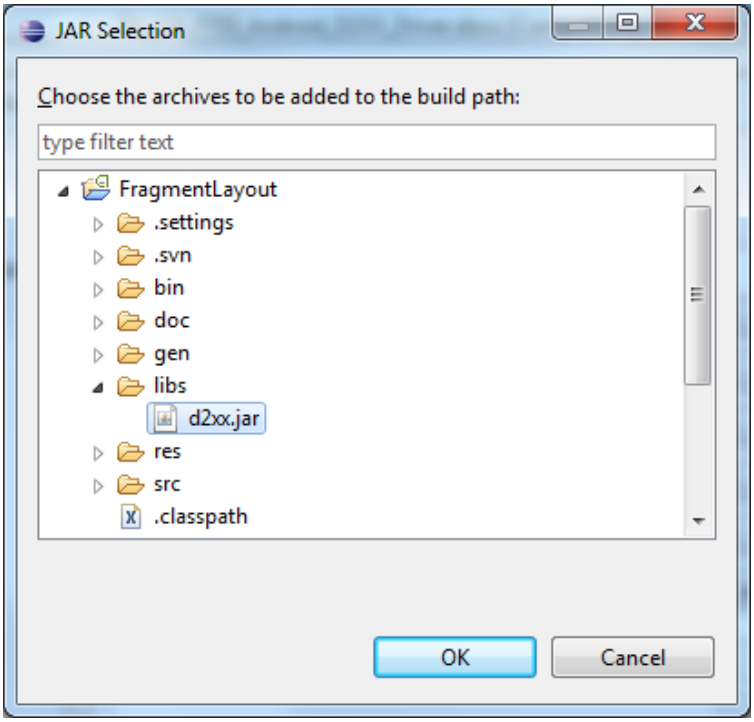

Figure 4: Select the library file – d2xx.jar

Copyright © **2013**Future Technology Devices International Limited

The D2xxManager class allows access to driver-wide information such as the VID and PID combinations to match with the device information list, and provides APIs to open target devices returning corresponding FT_Device objects. The FT_Device object can perform UART, EEPROM, and Bit mode-related operations to control device status or read/write data. When the device is no longer required, the FT_Device object can be closed with the close() method.

The D2xx Java library is fully documented using Javadoc. For information on the D2xx Android library methods, constants and sub-classes, please consult the corresponding Javadoc entry in the sample project's /doc directory.

A sample application demonstrating how to use various methods in the D2xx Java library is shown in section **2.2 Application Examples**. Please refer to "AN_233 Java D2xx for Android API User Manual" for the detail of the demo APIs.

Copyright © **2013**Future Technology Devices International Limited

2.2 Application Examples

2.2.1 Information

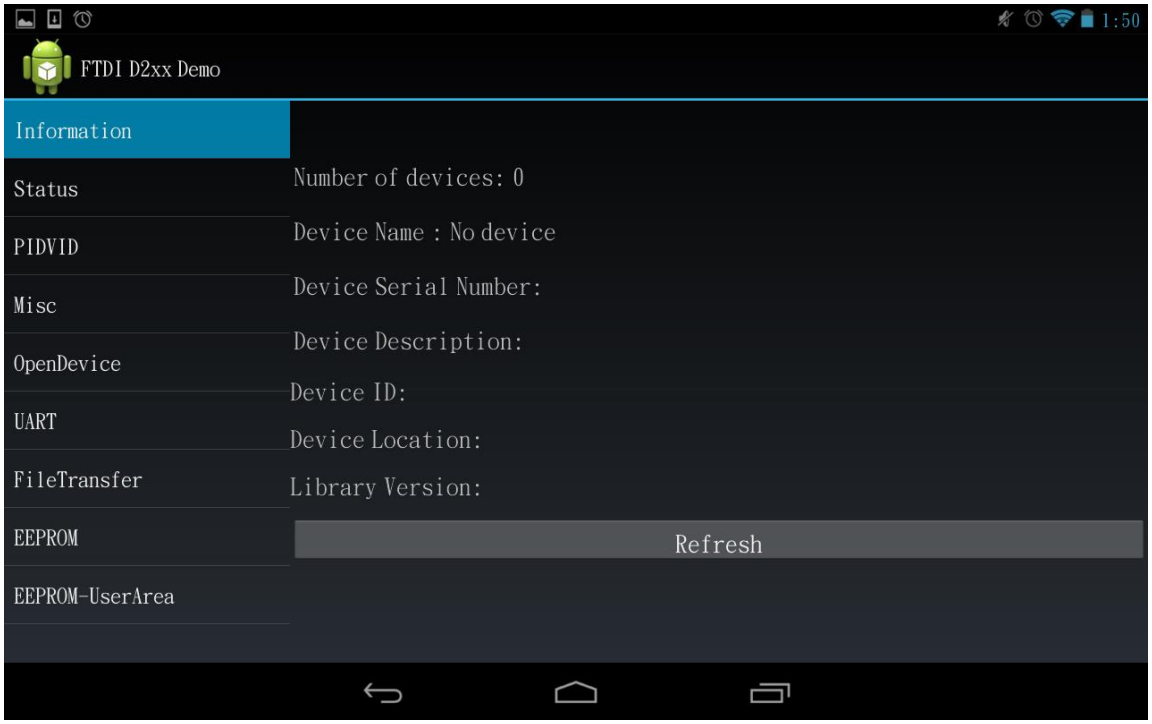

Figure 5: Demo Application Screenshot – Information

This demo application contains a **Refresh** button. Tap **Refresh** to access information about the connected device.

In this demo, these APIs are used:

Class	API Name
D2xxManager	createDeviceInfoList
D2xxManager	getDeviceInfoList
D2xxManager	getLibraryVersion

Note: In later demo applications, duplicated APIs are not listed again.

Copyright © **2013**Future Technology Devices International Limited

2.2.2 Status

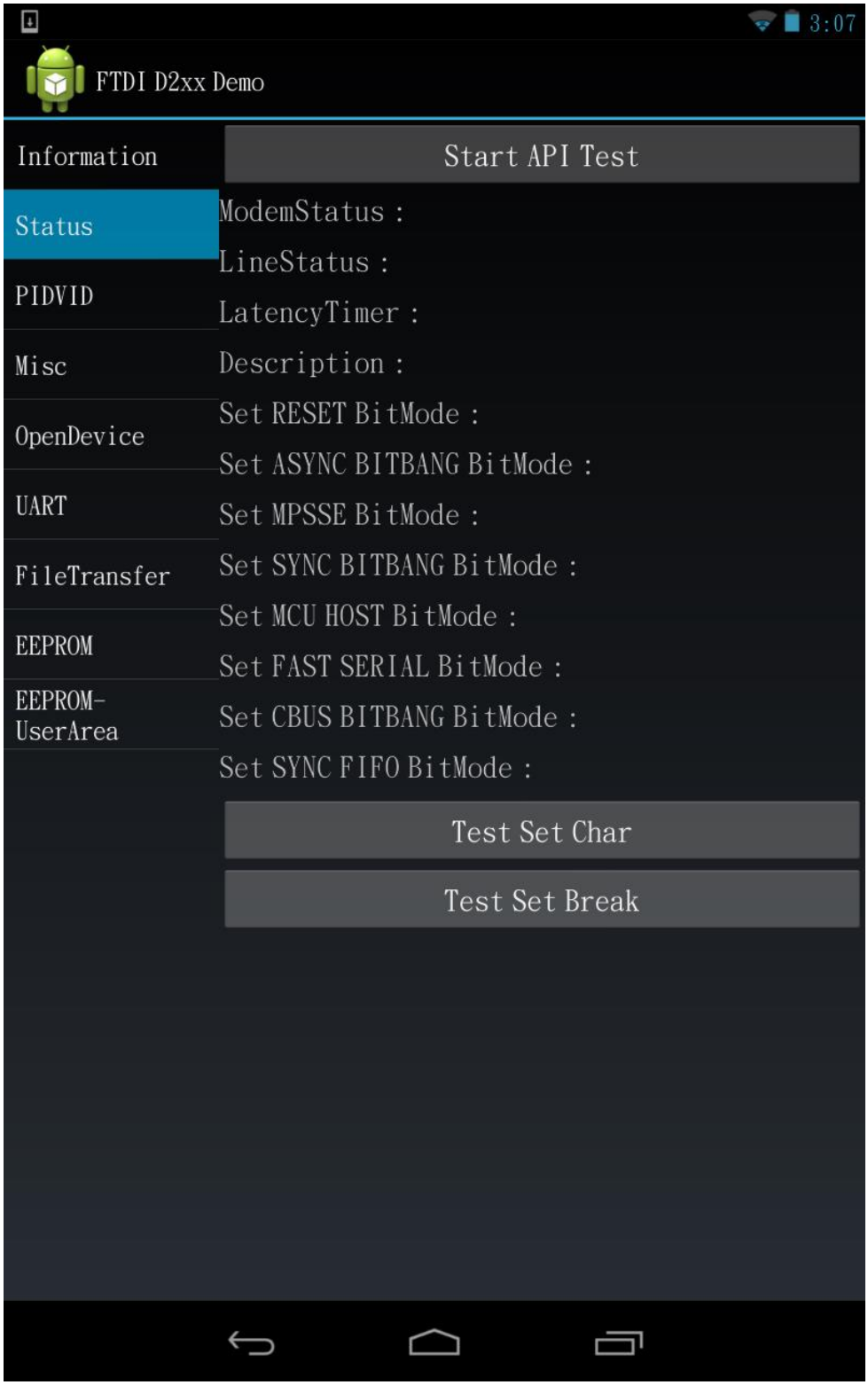

Figure 6: Demo Application Screenshot – Status

This demo application contains **Start API Test**, **Test Set Char** and **Test Set Break** buttons.

Tap **Start API Test** to call several APIs to obtain the related information.

Test Set Char and **Test Set Break** buttons will show the test results on the buttons directly – appended with Pass or Fail– when they are tapped.

Copyright © **2013**Future Technology Devices International Limited

In this demo, these APIs are used:

Class	API Name
FT_Device	getDeviceInfo
FT_Device	getLatencyTimer
FT_Device	getLineStatus
FT_Device	getModemStatus
FT_Device	setBitMode
FT_Device	setBreakOff
FT_Device	setBreakOn
FT_Device	setChars

2.2.3 PIDVID

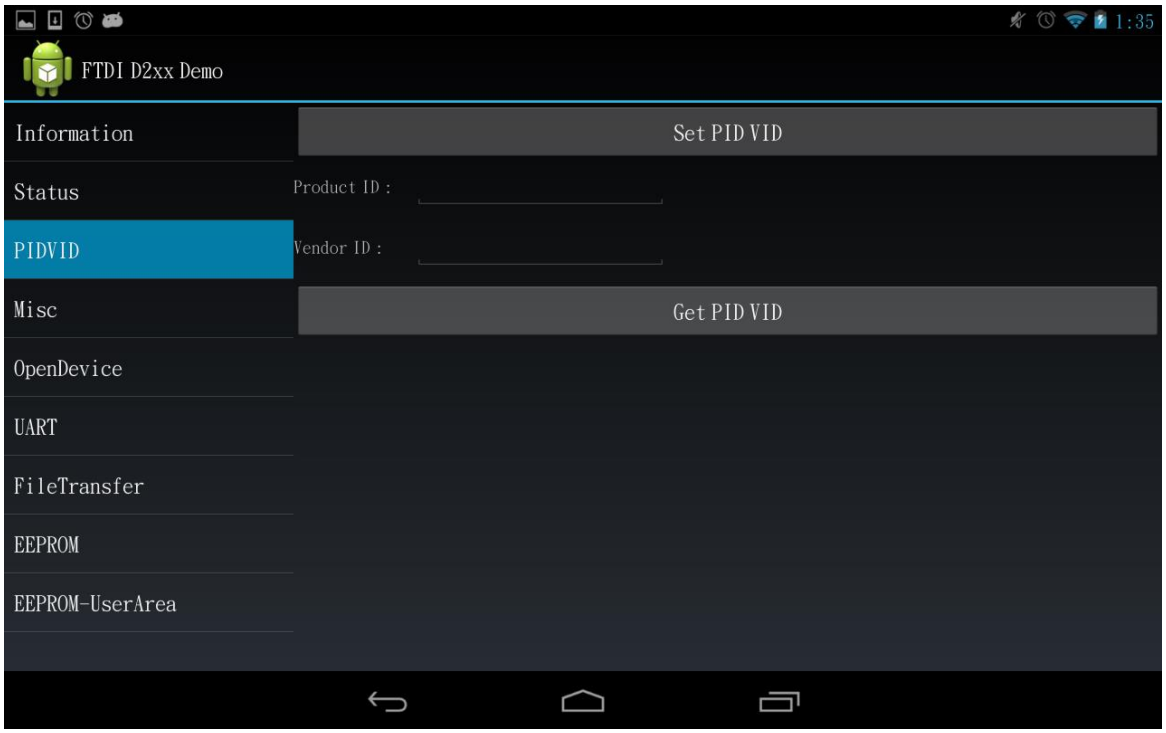

Figure 7: Demo Application Screenshot – PIDVID

This demo application contains two buttons – **Set PID VID** and **Get PID VID**.

Get PIDVID lists all the devices (in terms of Product and Vendor IDs) that would be recognised and supported by D2XX.

To specify an additional device, enter the corresponding Product and Vendor ID, then tap **Set PIDVID**.

In this demo, these APIs are used:

Class	API Name
D2xxManager	getVIDPID
D2xxManager	setVIDPID

Copyright © **2013**Future Technology Devices International Limited

2.2.4 Misc

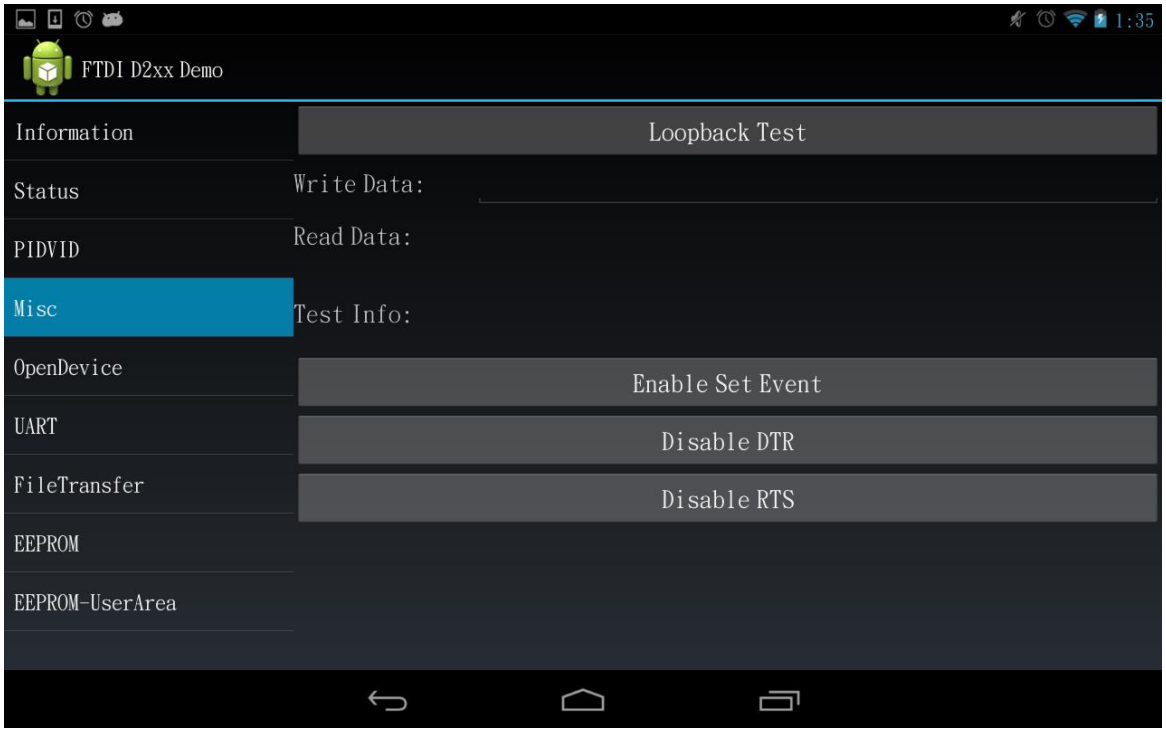

Figure 8: Demo Application Screenshot – Misc

This demo application contains:

1. Loopback demo:

There is a **Loopback Test** button and an input area for **Write Data**.

To run this test, you must connect an RS232 'loop-back' device, which has pairs of pins connected together (Rx to Tx, RTS to CTS and DSR to DTR).

Enter some words in the **Write Data** area and tap **Loopback Test**. Shortly, the same data should appear in the Read Data area.

1. Event demo:

Tap **Enable Set Event** to set event notification which waits RX event, and then redo the actions in loopback demo. When there is some data sent and received, the event would be triggered. The test results are displayed in the Test Info area.

1. DTR/RTS Enable/Disable Demo:

The 2 functions, **Disable DTR** and **Disable RTS** are for disabling/enabling DTR and RTS. The test results are displayed in the Test Info area.

Copyright © **2013**Future Technology Devices International Limited

In this demo, these APIs are used:

Class	API Name
FT_Device	clrDtr
FT_Device	clrRts
FT_Device	getEventStatus
FT_Device	getQueueStatus
FT_Device	read(*1)
FT_Device	setDtr
FT_Device	setEventNotification
FT_Device	setRts
FT_Device	write(*2)

Note:

*1 : read without length parameter

*2 : write without length parameter

Copyright © 2013Future Technology Devices International Limited

2.2.5 OpenDevice

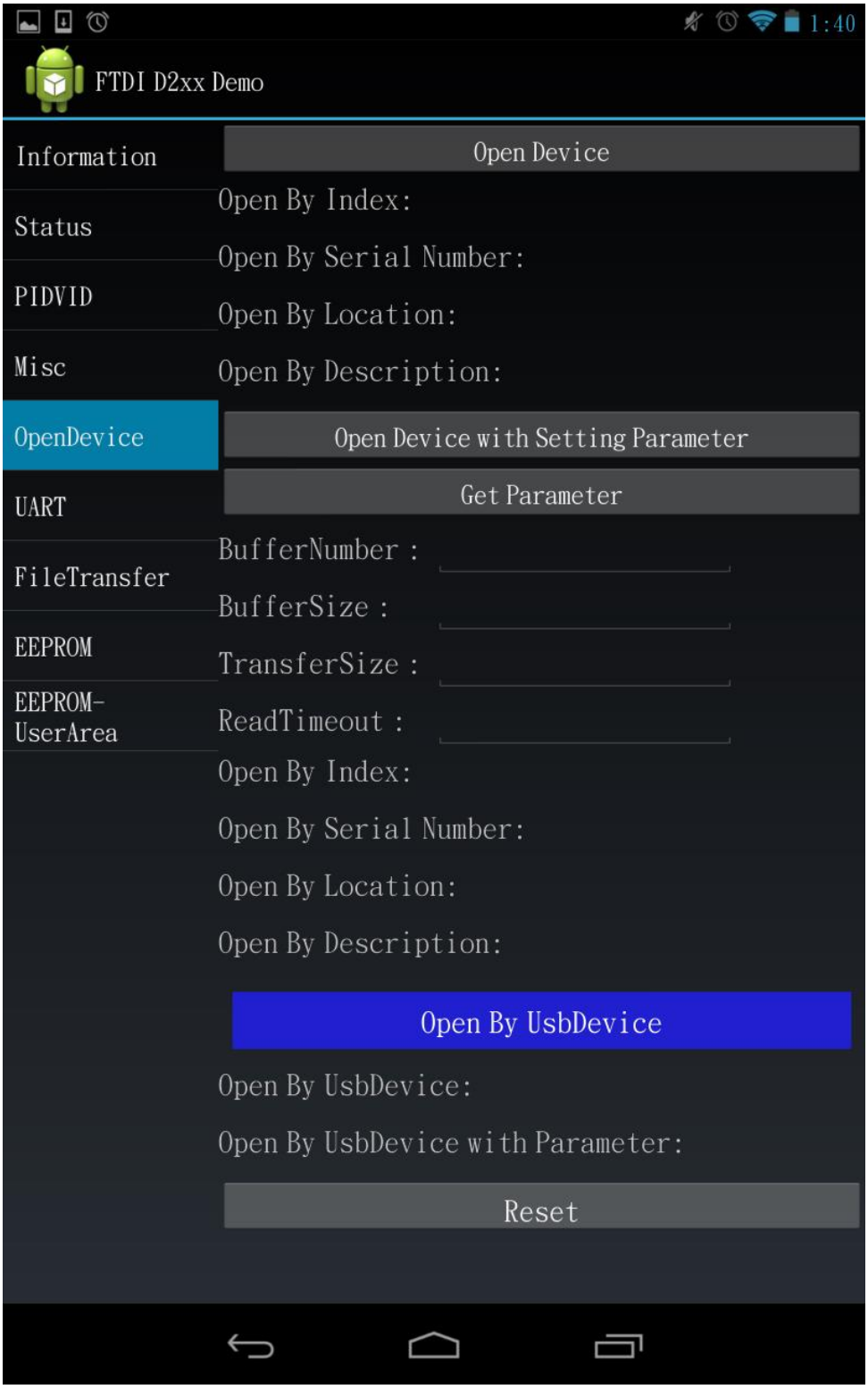

Figure 9: Demo Application Screenshot – OpenDevice

This demo application shows different ways to open the device.

Select **Open Device** and **Open Device with Setting Parameter** to test the open-related APIs. When **BufferNumber**, **BufferSize**, **TransferSize** or **ReadTimeout** is not set, the device uses the default value to open the device.

Get Parameter: gets current parameters.

Copyright © **2013**Future Technology Devices International Limited

About **Open By USBDevice**, when Android detects an USB device is plugged-in, it will send out a broadcast containing a UsbDevice object corresponding to the just attached device to notify interested party; when this application receive the notification, it will automatically call openByUsbDevice()to open the just attached device with the UsbDevice object provided by that notification.

Use "Reset" to clear the memory after the demo.

In this demo, these APIs are used:

Class	API Name
D2xxManager	getDeviceInfoListDetail
D2xxManager	isFtDevice
D2xxManager	openByDescription
D2xxManager	openByDescription(*)
D2xxManager	openByIndex
D2xxManager	openByIndex(*)
D2xxManager	openByLocation
D2xxManager	openByLocation(*)
D2xxManager	openBySerialNumber
D2xxManager	openBySerialNumber(*)
D2xxManager	openByUsbDevice
D2xxManager	openByUsbDevice(*)
D2xxManager.DriverParameters	getBufferNumber
D2xxManager.DriverParameters	getMaxBufferSize
D2xxManager.DriverParameters	getMaxTransferSize
D2xxManager.DriverParameters	getReadTimeout
FT_Device	resetDevice
D2xxManager.DriverParameters	setBufferNumber
D2xxManager.DriverParameters	setMaxBufferSize
D2xxManager.DriverParameters	setMaxTransferSize
D2xxManager.DriverParameters	setReadTimeout

Note: * : this function open device with D2xxManager.DriverParameters parameter

Copyright © **2013**Future Technology Devices International Limited

2.2.6 UART

Figure 10: Demo Application Screenshot – UART

This demo application displays the functionality of UART.

The first row contains an **Open** button to open selected **Port** numbers and a **Config** button to set UART configuration with several selectable setting items.

The UART configuration settings allow the baud rate to be set at standard values between 300 and 921600 baud.

Stop bits may be set for 1 or 2.

Data bits may be set for 7 or 8.

Parity may be set for ODD, EVEN, Mark, Space, None.

Flow allows for no flow control, RTS/CTS, DTR/DSR and XOFF/XON flow control.

The **Read Bytes** box displays the received data as ASCII values.

The **Read Enabled** button is used to test stop/restart IN task APIs.

Read Enabled will switch to **Read Disabled** when tapped, causing the application to stop receiving data.

Tap it again to enable the receive functionality.

Write Bytes allows a user to type in plain text and send it by tapping **Write**.

This application receives and sends data from and to the PC; or the Android device itself if a 'loop-back' device is connected.

Copyright © **2013**Future Technology Devices International Limited

In this demo, these APIs are used:

Class	API Name
FT_Device	close
FT_Device	isOpen
FT_Device	purge
FT_Device	read(*1)
FT_Device	restartInTask
FT_Device	setBaudRate
FT_Device	setDataCharacteristics
FT_Device	setFlowControl
FT_Device	setLatencyTimer
FT_Device	stopInTask
FT_Device	write(*2)

Note:

*1: read with length parameter

*2: write with length parameter

2.2.7 FileTransfer

Figure 11: Demo Application Screenshot – FileTransfer

This demo application displays the functionality of UART, but it could receive or send a file.

It also opens the device of index 0, the UART configuration is the same as UART demo introduced in previous section.

To use this application, please follow the steps below:

Step 1: Setup the connection between device and PC.

Step 2: Configure desired UART parameters.

Step 3: Tap the **Start Getting Data** button on the Android, then use a PC terminal to emulate sending a text file to the Android in raw mode, i.e., send all text data as it is, without extra encoding by X/Y/Z-modem protocol.

Step 4: After the file is sent, tap **Start Getting Data** again to stop receiving the file.

Step 5: Now on the PC terminal, prepare to receive a text file from Android.

Step 6: Tap the **Send Saved File** button on the Android. The data stored during Steps 3~4 will now be received by the PC.

Step 7: Now compare whether the sent and received files are the same.

The **Status** area will show some information during the get data and send file process.

2.2.8 EEPROM

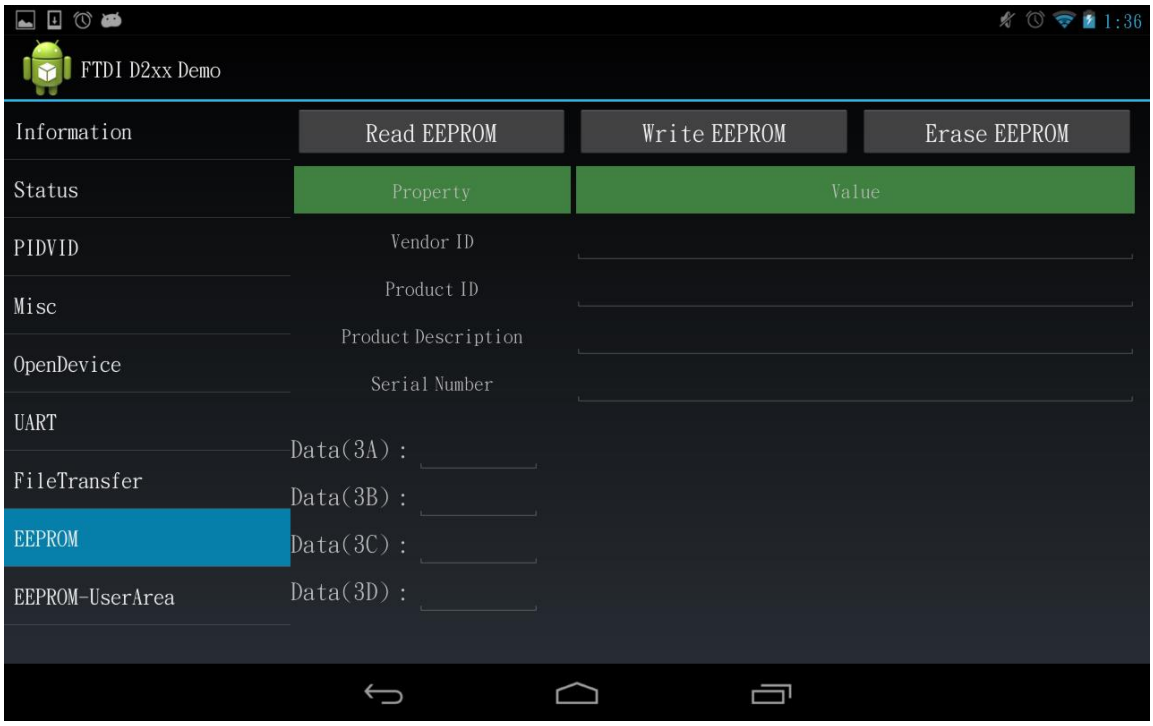

Figure 12: Demo Application Screenshot – EEPROM

WARNING: Backup the test device's EEPROM data(using FT_Prog) before performing the Write or Erase functions. Erasing the EEPROM might cause some devices to stop functioning.

(FT_Prog: http://www.ftdichip.com/Support/Utilities.htm#FT_Prog)

This demo application contains **Read EEPROM, Write EEPROM** and **Erase EEPROM** buttons.

Tap **Read EEPROM** to get and parse some data of the connected device.

Click **Write EEPROM** to write data back to EEPROM. This demo is limited to modifying the content of **Product Description**, **Serial Number** and **Data(3A)~Data(3D)**.

Erase EEPROM erases the entire EEPROM.

Copyright © **2013**Future Technology Devices International Limited

In this demo, these APIs are used:

Class	API Name
FT_Device	eepromErase
FT_Device	eepromRead
FT_Device	eepromReadWord
FT_Device	eepromWrite
FT_Device	eepromWriteWord

2.2.9 EEPROM-UserArea

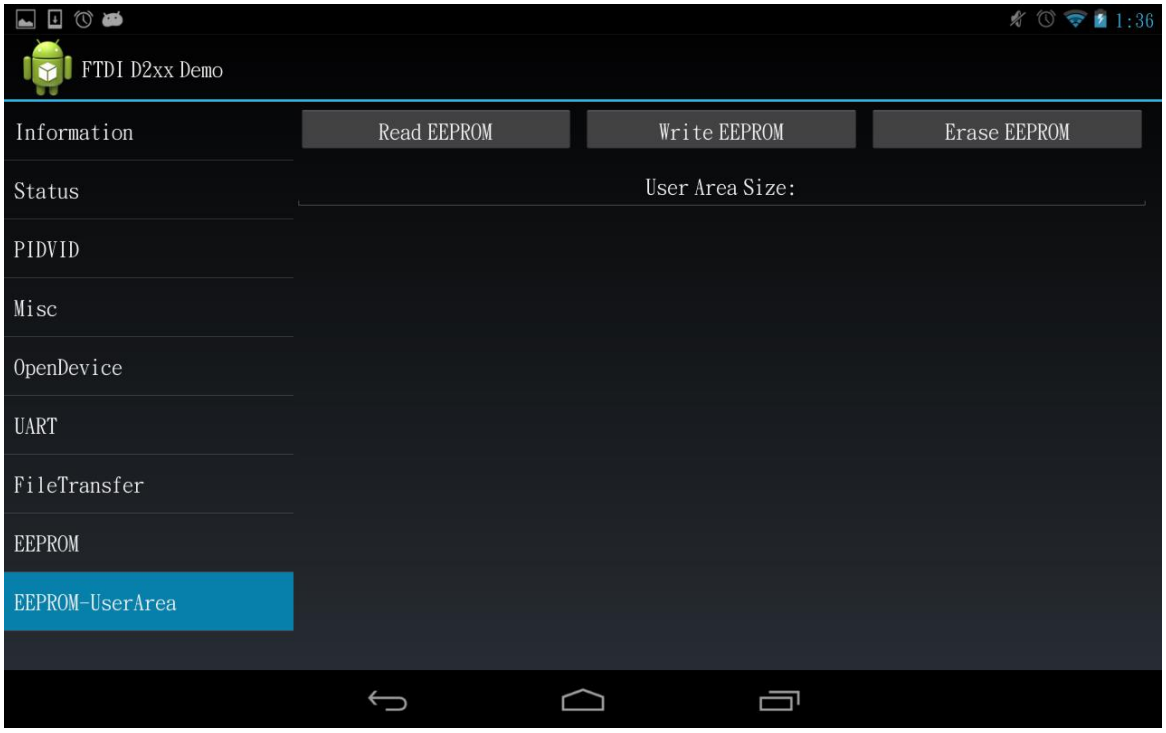

Figure 13: Demo Application Screenshot – EEPROM-UserArea

This demo application accesses the EEPROM user area.

Click **Read EEPROM** to get **User Area Size** and list the contents of user area.

Write EEPROM writes fixed data: 0, 1, 2, 3, 4, etc. to the entire user area.

Erase EEPROM writes fixed data: 0 to the entire user area.

NOTE: Please remember to backup EEPROM data(using FT_Prog) before writing or erasing.

In this demo, these APIs are used:

Class	API Name
FT_Device	eepromGetUserAreaSize
FT_Device	eepromReadUserArea
FT_Device	eepromWriteUserArea

Copyright © 2013Future Technology Devices International Limited

3 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1,2 Seaward Place, Centurion Business Park
Glasgow G411HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales)	sales1@ftdichip.com
E-mail (Support)	support1@ftdichip.com
E-mail (General Enquiries)	admin1@ftdichip.com
Web Site URL	http://www.ftdichip.com
Web Shop URL	http://www.ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8797 1330
Fax: +886 (0) 2 8751 9737

E-mail (Sales)	tw.sales1@ftdichip.com
E-mail (Support)	tw.support1@ftdichip.com
E-mail (General Enquiries)	tw.admin1@ftdichip.com
Web Site URL	http://www.ftdichip.com

Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)
7130 SW Fir Loop
Tigard, OR 97223-8160
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales)	us.sales@ftdichip.com
E-Mail (Support)	us.support@ftdichip.com
E-Mail (General Enquiries)	us.admin@ftdichip.com
Web Site URL	http://www.ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited (China)
Room 1103, No. 666 West Huaihai Road,
Shanghai, 200052
China
Tel: +86 2162351596
Fax: +86 2162351595

E-mail (Sales)	cn.sales@ftdichip.com
E-mail (Support)	cn.support@ftdichip.com
E-mail (General Enquiries)	cn.admin@ftdichip.com
Web Site URL	http://www.ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

4 Appendix A – References

http://developer.android.com

http://code.google.com/p/rowboat/wiki/JellybeanOnBeagleboard_WithSGX

http://beagleboard.org/hardware-xM

http://www.ubuntu.com/

Acronyms and Abbreviations

Terms	Description
ADB	Android Debug Bridge
ADT	Android Development Tools
AOSP	Android Open Source Project
API	Application Programming Interface
BSP	Board Support Package
CTS	Clear To Send
DSR	Data Set Ready
DTR	Data Terminal Ready
EEPROM	Electrically Erasable Programmable Read-Only Memory
IDE	Integrated Development Environment
JNI	Java Native Interface
NDK	Native Development Kit
OS	Operating System
PID	Product ID
RTS	Request To Send
SDK	Software Development Kit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VID	Vendor ID

Copyright © 2013Future Technology Devices International Limited

5 Appendix B – List of Tables & Figures

List of Figures

Copyright © 2013Future Technology Devices International Limited

6 Appendix C– Revision History

Revision	Changes	Date
1.0	Initial Release for beta test	2013-01-24
1.1	First Public release	2013-02-05
1.2	Update to section 2.2.2	2013-09-16

Copyright © 2013Future Technology Devices International Limited