Future Technology Devices International Ltd.

Technical Note

TN_134 FTDI Android D2XX Driver

Document Reference No.: FT_000522

Version 1.0

Issue Date: 2011-09-29

This document describes the installation and use of the FTDI D2XX 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

Table of Contents

1 Introduction

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

FTDI has now developed a port of the D2XX driver for the Android OS. This includes the native D2XX library, but in the case of Android also includes Java native interface code and a Java class to allow the driver to be easily accessed from an Android application.

The library provided is called libftd2xx-jni.so. This includes the Java native interface code and is statically linked to the D2XX native driver library.

At the time of writing, the Android D2XX driver is available as a beta release. Customers are encouraged to provide feedback on the release to FTDI support.

The package provided by FTDI includes a compiled native library (libftd2xx-jni.so), a D2XX Java class file (D2xx.java) and a sample application project for Eclipse (D2XX_Sample). The source to the JNI portion of the native library is also available and is statically linked with the native libftd2xx.a library to produce the Java compatible libftd2xx-jni.so file.

The driver and associated files can be downloaded from the FTDI website at
http://www.ftdichip.com/Drivers/D2XX/D2XXSample/D2XXSample.zip

1.1 Prerequisites

In order to install the FTDI D2XX driver and use it successfully, the following are required:

- A hardware platform including a USB host device supported by the Android/Linux kernel.
  - FTDI testing was conducted using a BeagleBoard-xM Rev C.
- An FTDI based device for testing with
  - FTDI testing was conducted with an FT232R based US232R cable.

In addition, to develop an application using the FTDI D2XX driver for Android a development machine must have the Eclipse IDE, Android SDK including the ADB program and Android ADT Plugin installed. The installation and configuration of these tools is not within the scope of this document and is outlined on the Android developer web site (http://developer.android.com/sdk/index.html).

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

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

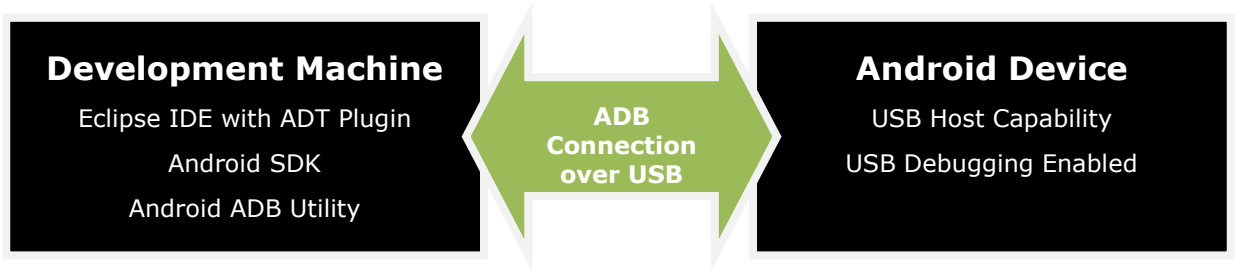

Figure 1: Development Configuration

2 Installing the FTDI D2XX JNI Library

The driver library can be installed on the Android device via the Android debugger. Connect the device to the development machine and open a command terminal. If the Android development tools are correctly installed, typing

```
adb devices
```

should result in the Android device being found and its serial number being displayed in a manner similar to this:

```
List of devices attached
20100720        device
```

Accessing the ADB interface may fail for a number of reasons, including device permissions on Linux systems. Troubleshooting the ADB connection is not within the scope of this document.

The following ADB command assumes that the library is located in a local directory called D2XX and it is to be installed in the /sdcard/Android/data/com.ftdi.d2xx directory on the Android device:

```
adb push D2XX/libftd2xx-jni.so
/sdcard/Android/data/com.ftdi.d2xx/libftd2xx-jni.so
```

Permissions may pose problems when installing the library**; libraries are normally located in /system/lib but if root access is not available then this location will not be accessible**. The D2XX Java class allows loading of the library from any specified location, so any location on the device is permitted provided that the D2XX Java class is modified to load the library from the correct place.

3 Using the FTDI D2XX JNI Library

To accompany the JNI library, FTDI have provided a Java class which can be easily included in an application. The class provides access to all of the classic D2XX functions including EEPROM functions. The D2xx Java class can readily be included in an Android application project in Eclipse.

The D2xx class (not to be confused with the D2XX native API) provides some static methods that allow access to driver-wide information such as the VID and PID combinations to match with and the device information list.

All other methods require a D2xx object to be created and subsequently opened using one of the four open methods (openByIndex, openBySerialNumber, openByDescription or openByLocation). Executing an open method will, if successful, cause the instance of the D2xx class to internally maintain a native handle value; this is used for all subsequent communication with the device. When the device is no longer required, the native handle can be closed with the close method.

Since the JNI calls ultimately call native D2XX functions, exception generation is included in the JNI layer. An exception of type D2xxException (extended from IOException) is thrown in the case of a native D2XX call returning an FT_STATUS code other than FT_OK. The exception also generates a message indicating the native status code and the native function that the exception occurred in.

The D2xx class is fully documented using the Javadoc standard. For information on the D2xx class methods, constants and sub-classes, please consult the Javadoc entry for the item of interest.

A sample application demonstrating how to use several of the methods in the D2xx class is also provided to assist with customer application development. The sample application is shown below:

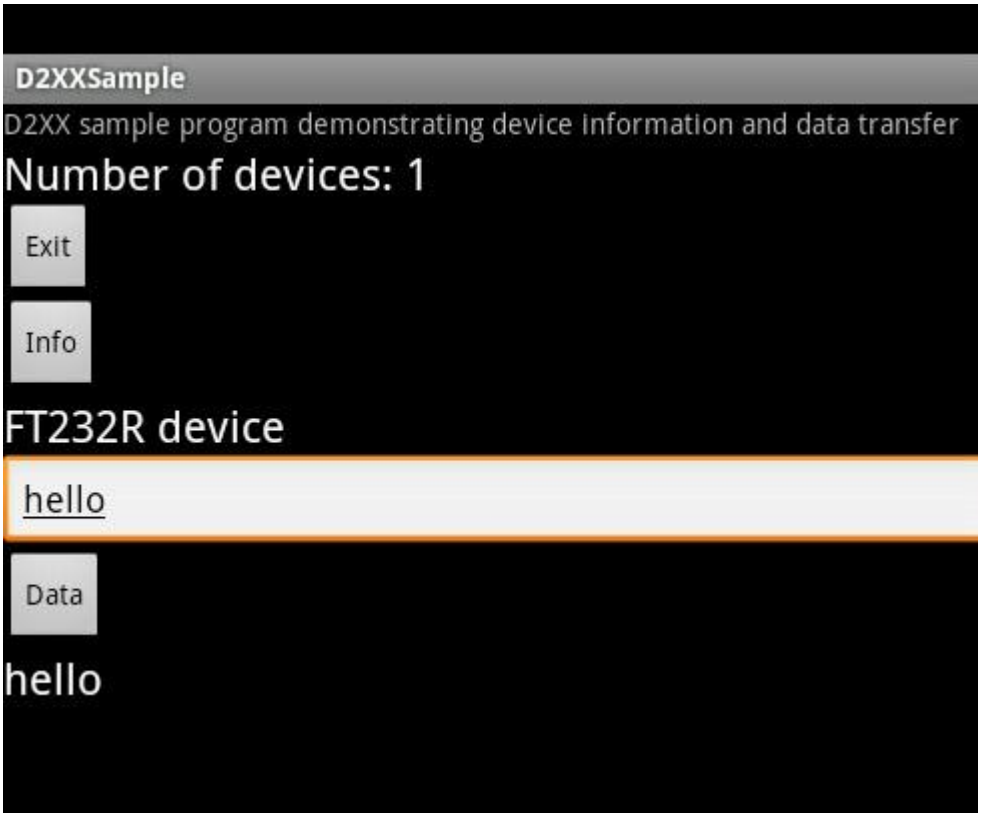

Figure 2: D2XXSample Application

Clicking the Info button in the sample application displays the number of devices available and the chip type of the first device in the device list.

Clicking the Data button writes the text from the edit box to the device, waits for 1 second then reads any available data back for displaying below the Data button. In the above screen shot, a loopback connector was fitted so the data received is the same as the data sent.

4 Limitations and Restrictions

4.1 Non-Default VID and PID Combinations

At the time of writing, the Android D2XX driver will support all default FTDI VID and PID combinations and can also support a single custom VID and PID combination via the setVIDPID Java method (FT_SetVIDPID function).

However, it is currently not possible to match several non-default VID and PID combinations simultaneously. This is due to the Android OS hanging on a call to dlopen which precludes the use of an external libtable library at this time.

4.2 USB Device Permissions

Many Android systems with USB host capability enumerate devices with application incompatible permissions (0660). These permissions are specified in the ueventd.rc file as follows:

```
/dev/bus/usb/*     0660   root   usb
```

In order to modify the default permissions for USB devices in the ueventd.rc file a user must have root access. This may render the D2XX library unusable on devices that have default permissions of 0660 and do not allow root access.

To make the device accessible, the ueventd.rc entry listed above should be changed to make the device world readable and world writeable as follows:

```
/dev/bus/usb/*     0666   root   usb
```

NOTE: FTDI does not accept any responsibility for customers who choose to enable root access on their Android platform and subsequently damage the unit or void the warranty. Enabling root access on such devices is entirely at the user's own risk.

5 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
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 8791 3570
Fax: +886 (0) 2 8791 3576

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 – Hillsboro, Oregon, USA

Future Technology Devices International Limited (USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
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 408, 317 Xianxia Road,
Shanghai, 200051
China
Tel: +86 21 62351596
Fax: +86 21 62351595

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.

6 Appendix A – References

Document References

http://developer.android.com

http://beagleboard.org/hardware-xM

http://processors.wiki.ti.com/index.php/TI-Android-GingerBread-2.3-DevKit-1.0_ReleaseNotes

http://www.ubuntu.com/

Acronyms and Abbreviations

Terms	Description
OS	Operating System
USB	Universal Serial Bus
JNI	Java Native Interface
SDK	Software Development Kit
ADB	Android Debug Bridge
NDK	Native Development Kit

7 Appendix B – List of Tables & Figures

List of Figures

8 Appendix C – Revision History

Revision	Changes	Date
1.0	Initial Release for beta test	2011-09-29