



VINCULUM

BINDING USB TECHNOLOGIES

Future Technology Devices International Ltd

Accessing Android Open Accessory Mode with Vinco Development Platform

Application Note AN_196

Version Version1.0

Issue Date: 2012-01-20

This application note demonstrates how the Vinco Development Platform can enable the Open Accessory Mode in compatible Android devices and transfer data to and from the Android device over USB.

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 FTDI harmless from any and all damages, claims, suits or expense resulting from such use.

1 Introduction

Android Open Accessory Mode is a new feature added to Android 3.1 (back-ported to 2.3.4). This mode allows a USB host to connect to the Android hardware allowing data transfer between the USB host and the Android hardware over USB.

This application note demonstrates how the FTDI Vinculum-II (VNC2) device can enable the Open Accessory Mode in compatible Android devices and transfer data to and from the Android device over USB. It illustrates how the VNC2 drivers for Open Accessory Mode (available as of IDE version 1.4.2) are loaded onto the VNC2 and how a simple application accesses them. It also provides a small Android Platform application (app) to complete the demonstration.

For development purposes an FTDI [Vinco Development Platform](#) and an FTDI Touch-Key Shield are used.

Vinco is a development module inspired by the Arduino concept and uses the Vinculum-II (VNC2) dual port USB Host/device IC. Vinco uses a VNC2-64Q package to facilitate 38 GPIO options on 0.1" pitch sockets. Vinco is designed as a prototyping platform for VNC2 based designs and applications.

The Vinco Touch-Key is a shield that mates with the Vinco module to allow touch pad operation. Onboard the shield is a [STMPE821](#) device, this is a 8-channel GPIO capacitive touch-key sensor.

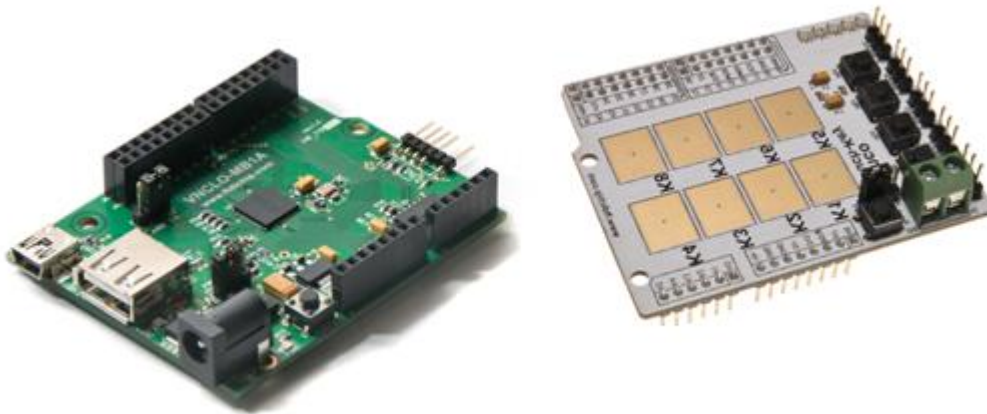


Figure 1.1 – Vinco Development Module and the Touch-Key Shield

A Motorola Xoom running Android OS version 3.1 is used as the Android target hardware.

The demonstration shows the ability of the VNC2 to enumerate the Xoom device, interrogate the device to determine if it supports Open Accessory Mode, then enable the Open Accessory Mode and re-enumerate the Xoom.

When the USB link has been established an application running on the Xoom tablet is able to control the LEDs on the Vinco Touch-Key Shield, while the touch-key buttons can control "LEDS" on the Android application GUI.

Full source code and precompiled ROM files for VNC2 are provided on an as is basis.

1.1 VNC2 Devices

Vinculum-II (VNC2) is the second of FTDI's Vinculum family of embedded dual USB host controller devices. The VNC2 device provides USB Host interfacing capability for a variety of different USB device classes including support for BOMS (bulk only mass storage), Printer and HID (human interface devices). For mass storage devices such as USB Flash drives, VNC2 transparently handles the FAT file structure.

Communication with non USB devices, such as a low cost microcontroller, is accomplished via either UART, SPI or parallel FIFO interfaces. VNC2 provides a new, cost effective solution for providing USB Host

capability into products that previously did not have the hardware resources available.

VNC2 allows customers to develop their own firmware using the Vinculum II software development tool suite. These development tools provide compiler, assembler, linker and debugger tools complete within an integrated development environment (IDE).

The Vinculum-II VNC2 family of devices are available in Pb-free (RoHS compliant) 32-lead LQFP, 32-lead QFN, 48-lead LQFP, 48-lead QFN, 64-Lead LQFP and 64-lead QFN packages For more information on the ICs refer to [VNC2 datasheet](#).

1.2 Motorola Xoom

This application example uses a Motorola Xoom as the Android target. It was chosen since it was readily available and had Android OS 3.1 installed (which is required for the Open Accessory Mode). Other Android Platforms that support the Open Accessory Mode, could be used instead. Click [here](#) for more information on the Motorola Xoom.



Figure 1.2 – Motorola Xoom Tablet

Table of Contents

1	Introduction	2
1.1	VNC2 Devices.....	2
1.2	Motorola Xoom	3
2	Block diagram	5
3	Android Application Demo	6
3.1	Requirements	6
3.2	Demo Setup	6
3.3	Demo Description	6
4	Source code for the VNC2 Application	7
4.1	Android_ACC.C.....	7
4.1.1	main().....	7
4.1.2	Iomux_setup()	7
4.1.3	Open drivers/close drivers	8
4.1.4	USB_Host_Connect_state	8
4.1.5	Android_attach/Android_detach.....	8
4.1.6	Firmware	8
4.1.7	Usb_host_processing	8
4.1.8	PWM_Processing.....	8
4.2	Building and Loading the VNC2 Application into the VNC2 Device.....	8
5	The Android Application	10
5.1	Building and Loading the Android Application into the Xoom Tablet .	10
6	Running the Android Application Demo	11
7	Contact Information	12
Appendix A – VNC2 Application Code		13
Appendix B – Android Java Application Code		14
Appendix C – References.....		15
Acronyms and Abbreviations		16
Appendix D – List of Figures and Tables.....		17
Appendix E – Revision History.....		18

2 Block diagram

The following block diagram, Figure 2.1, illustrates how the demonstration hardware is connected.

The development PC (red block) is not required when the ROM file has been programmed into the VNC2 on the [Vinco Development Platform](#)

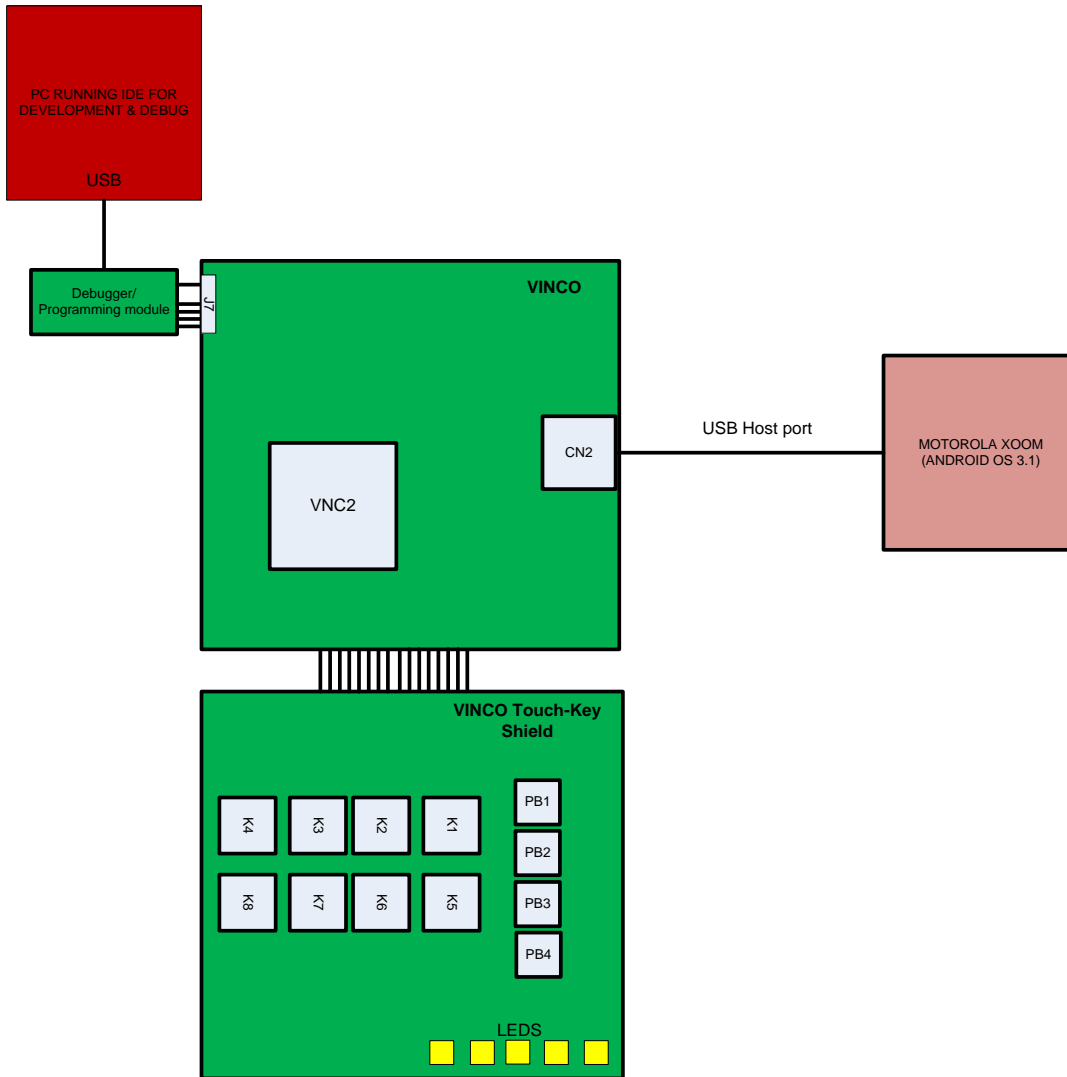


Figure 2.1 – VNC2 Open Accessory Mode Demo Block Diagram

The ROM file (and source code) for this project can be viewed in appendix A and is available to download from [FTDI website](#). The .ROM file can be found in the “debug” directory of the .zip file

3 Android Application Demo

3.1 Requirements

- The [Vinco Development Platform](#) (other VNC2 boards with a USB host connector exposed will also work but a different IOMUX setup may be required)
- FTDI Touch-Key Shield
- Motorola Xoom (other device running either v2.3.4 or v3.1 onwards, of the Android operating system with support for accessory mode will also work).
- PC with the [VNC2 IDE](#) Version 1.4.4 or later
- USB cable
- VNC2 [debugger/programmer module](#) (also available from [FTDI](#))

3.2 Demo Setup

1. Connect the [Vinco Development Platform](#) to the PC with the VNC2 [debugger module](#) via J7 connector to load the .ROM file for the VNC2 application into the VNC2 chip. The [debugger module](#) allows the [VNC2 IDE](#) on the development PC to communicate with the VNC2 debugger port on the [Vinco Development Platform](#). The precompiled .ROM file for the VNC2 application can be downloaded from the [FTDI website](#) and can be found inside the "debug" directory of the .zip file. Please refer to section 4.2 on how to program firmware into the VNC2 device.
2. Mount the Touch-Key Shield on the [Vinco Development Platform](#) with the VNC2 application .ROM file loaded into the VNC2 chip. When mounting the boards, the connectors should mate perfectly.
3. Connect the Android target (loaded with the Android application) to the [Vinco Development Platform](#) with the USB cable via CN2 connector (VNC2 USB port 2). The source code for the Android application can be downloaded from the [FTDI website](#). Please refer to section 5.1 on how to load Android application on the Android Platform

3.3 Demo Description

This demo uses LEDs, LED1:LED5 and push buttons, PB1:PB4 on Vinco Touch-Key Shield. Java Android GUI application was developed to create 4 LEDs and 4 push buttons on the Android Tablet to duplicate the setup.

LED1:LED4 on the Touch-Key shield and 4 LEDs on Android target hardware are controlled by PB1:PB4 on the shield as well as 4 buttons placed on the Android application.

When a push button is pressed on the Vinco Touch-Key Shield or Android Tablet application, toggles the state of corresponding LED on the Vinco Touch-Key board and on the Android application.

PWM is implemented on LED5 on the shield. This LED is controlled by Volume control bar on the Android Tablet and demonstrates volume up/down characteristic with varying brightness.

All LEDs, LED1:LED4 and push buttons PB1:PB4 on the Vinco Touch-Key Shield are connected to the VNC2 GPIOs on the [Vinco Development Platform](#) via pin headers.

As all the GPIO connectivity is part of the [Vinco Development Platform](#), the user only needs to provide a USB cable to connect to the development PC and the XOOM tablet.

4 Source code for the VNC2 Application

All VNC2 application firmware follows a similar format and most of the code can be “written” using the [VNC2 IDE](#) application wizard. The basic steps are: Initialise device drivers

1. Define pinouts
2. Open ports to be used
3. Configure ports to be used
4. Read/write data
5. Close ports

The VNC2 application source code for this project can be viewed in appendix A and is available to download on an as is basis from the [FTDI website](#):

NOTE: A Precompiled ROM file is also downloadable at the same address for users not wishing to rebuild the project. The . ROM file can be found inside the “debug” directory of the .zip file.

4.1 Android_ACC.C

Android_ACC.c is the main firmware file. This file is split into multiple functions.

```
void main();
void iomux_setup(void);
void open_drivers(void);
void close_drivers(void);
unsigned char usbhost_connect_state(VOS_HANDLE hUSB)
VOS_HANDLE android_attach(VOS_HANDLE hUSB, unsigned char devANDACC,
    char *manufacturer, char *model, char *description,
    char *version, char *uri, char *serial)
void android_detach(VOS_HANDLE hANDACC)
void usb_host_processing();
void firmware();
void pwm_processing();
```

4.1.1 main()

Main is where the application starts. It defines the VNC2 core clock speed, loads the drivers to be used and creates the threads to be used in the application. At the very end of main is the call

```
vos_start_scheduler();
```

After this call there can be no further configuration of the device.

4.1.2 Iomux_setup()

Iomux_setup actually refers to the other file in the project, andropid_acc_iomux.c and is used to define the VNC2 pinout. Most functions can be programmed to appear on different pins. The notable exceptions are power, GND and the USB ports.

4.1.3 Open drivers/close drivers

The open drivers function call will provide a handle to each hardware block used in the project and this handle can be used by subsequent commands to control the hardware. Close drivers closes the handle at the end of the project.

4.1.4 USB_Host_Connect_state

USB_Host_Connect_State is a function to check if anything is connected to the USB host.

4.1.5 Android_attach/Android_detach

As the VNC2 uses a layered architecture to control the drivers it is important to attach, detach the correct USB class driver to the USB host port. In this case the android class driver is being attached and detached.

4.1.6 Firmware

The firmware function is monitoring/controlling the VNC2 GPIO.

4.1.7 Usb_host_processing

USB_host_Processing is the section that interrogates the Android device as to whether it is capable of supporting open Accessory Mode and if it is, will enable it.

4.1.8 PWM_Processing

As an extra feature the 5th LED can be made to show variable brightness by using the PWM interface of the VNC2.

4.2 Building and Loading the VNC2 Application into the VNC2 Device

The VNC2 applications codes must be compiled with the [VNC2 IDE](#) running on the PC before being programmed into the VNC2 on the [Vinco Development Platform](#). The [debugger module](#) gives access to the VNC2 debugger port on the [Vinco Development Platform](#) through a USB connection to a PC. To build and load the application simply follow below steps:

1. Unzip the AndroidAccessoryLEDDemo.zip project to a directory on the development machine
2. Open the AndroidAccessoryLEDDemo project in the VNC2 IDE
3. Press the Build button on the IDE ribbon bar under the build tab

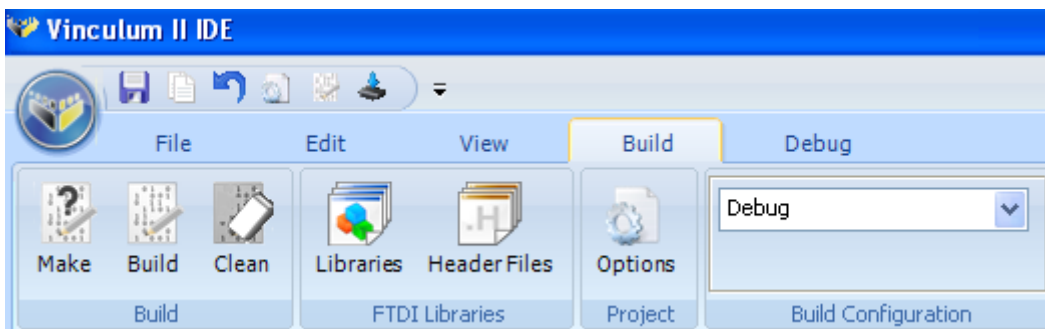


Figure 4.1 – Vinculum II IDE Build Button

4. Click on the "Flash" button on the ribbon bar under the debug tab.

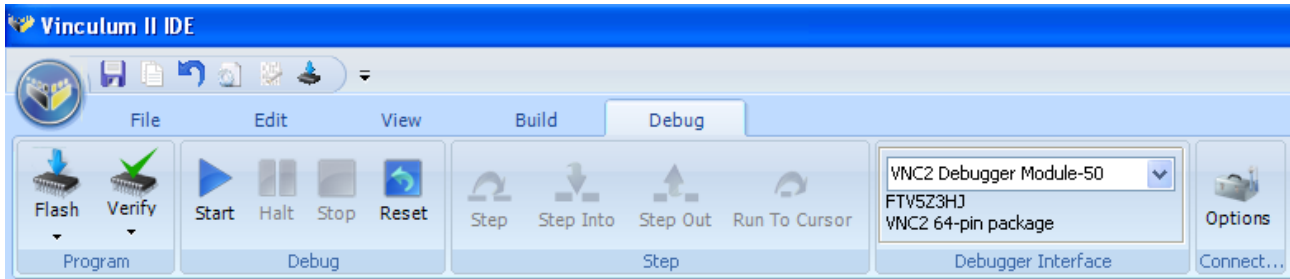


Figure 4.2 – Vinculum II IDE Flash Button

In the IDE Debug tab, the VNC2 that is connected will be shown in the Debugger Interface area as "VNC2Debugger Module-50" (see Figure 5.2). The Debugger Interface drop down box might have a different label depending on which debug interface type is in use.

NOTE: Ensure that the drop down box of the Debugger Interface area shows a device is connected before attempting to flash a device.

For further details please see [AN_159 Vinculum-II Firmware Flash Programming](#) and [AN_142 Vinculum-II Tool Chain Getting Started Guide](#) or [AN_138 Vinculum-II Debug Interface](#)

5 The Android Application

The Android application source code is developed using JAVA as this is the standard development language for Android GUI applications. The source code for this application can be downloaded from the [FTDI website](#).

5.1 Building and Loading the Android Application into the Xoom Tablet

The Android application can be compiled and built using Eclipse tools which is a free compiler available on the [Eclipse website](#). The resultant ledactivity.apk is then be loaded onto the Android Platform (XOOM). If a pre-compiled ledactivity.apk file is used, the Eclipse tools are not required. Simply connect the Android Platform (XOOM) to the PC with a USB cable and load the ledactivity.apk file onto the XOOM tablet.

The followings are required to build and load Android application source code:

- A device running either v2.3.4 or v3.1 of the Android operating system with support for accessory mode. The debugger should be enabled on this unit.
- A PC with an Android application development environment installed which can be found from [Android Developer website](#). Note that the Google API library including support for Android Accessories must also be present on the development machine if building for 2.3.4. Full details are available [here](#).
- The Android application ([android_acc_appl.zip](#))

To build and load the Android application onto the Xoom tablet follow below steps:

1. Extract the VNC2Accessory demo Android application to a directory on the development PC.
2. Build the VNC2Accessory Java application if desired. If not, the application is supplied pre-compiled (ledactivity.apk)
3. Enable non-market apps to be installed on the Android device (on our Nexus S this is under Settings > Applications > Unknown sources)
4. Ensure these settings are applied on the XOOM tablet: Settings-> Applications ->Development -> USB Debug is selected to allow the application to be loaded over USB.
5. Connect the Android device to the development PC.
6. Use the Android debugger interface to install the demo application on the Android device (from a command prompt: adb install {path to application}/ ledactivity.apk)

Further help on using Eclipse to build Android applications can be found [here](#).

6 Running the Android Application Demo

To run the demonstration perform the following steps:

1. Power up the Vinco Touch-Key shield which is mated with the [Vinco Development Platform](#) with the VNC2 application code loaded into the VNC2 chip.
2. Power up the XOOM tablet with the Android application loaded which is connected to the [Vinco Development Platform](#) via CN2 (USB port 2 of the VNC2).
3. The application on the tablet will start automatically.
4. Pressing PB1 on the Vinco Touch-Key Shield will toggle the state of the left hand LED on the XOOM application.
5. PB2, 3, 4 correspond to the other LEDs on the ZOOM.
6. Pressing the left most button on the XOOM application will toggle the state of the LED1 on the Vinco Touch-Key shield.
7. The other buttons on the XOOM application correspond to LED2, 3, 4 on the Vinco Touch-Key shield.
8. The volume control slider on the ZOOM controls the brightness of LED 5 on the Vinco Touch-Key Shield

NOTE: The PC is not required if the .ROM files are loaded into to the XOOM tablet and VNC2 on the [Vinco Development Platform](#)

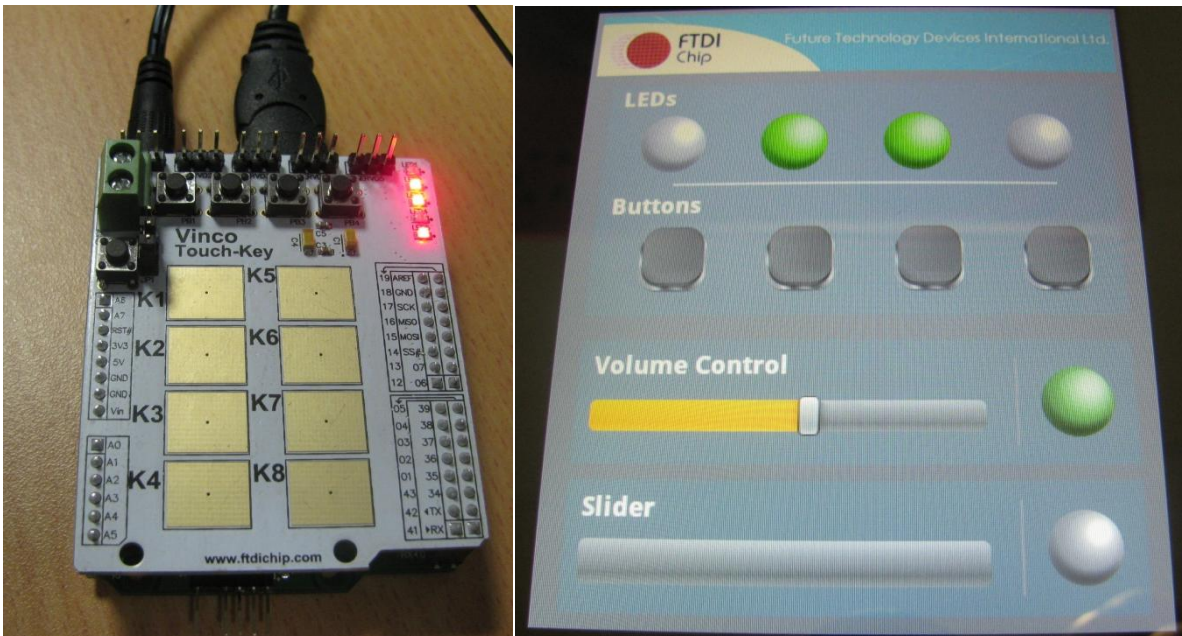


Figure 6.1 – Running the demo

NOTE: At the time of release, the application source does not provide control of the slider.

7 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

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

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

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

<http://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.

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied. Future Technology Devices International Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. This document provides preliminary information that may be subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

Appendix A – VNC2 Application Code

Source code for this project is provided on an “as is” basis and functionality is neither guaranteed or supported. The code has been verified as running on a Motorola Xoom Platform, with Android OS 3.1.

The source code for VNC2 Application can be downloaded from [FTDI website](#)

Appendix B – Android Java Application Code

The source code for Android Java Application can be downloaded from [FTDI website](#)

Appendix C – References

The following VNC2 documents and the full Vinculum-II Toolchain software suite can be downloaded by clicking on the appropriate links below:

Application note AN_181	Accessing Android Open Accessory with Vinculum-II
Application note AN_159	Vinculum-II Firmware Flash Programming
Technical note TN_108	Vinculum Chipset Feature Comparison
Technical note TN_118	Vinculum-II Errata Technical Note
Application note AN_118	Migrating Vinculum Designs From VNC1L to VNC2-48L1A
Application note AN_137	Vinculum-II IO Cell Description
Application note AN_138	Vinculum-II Debug Interface Description
Application note AN_139	Vinculum-II IO Mux Explained
Application note AN_140	Vinculum-II PWM Example
Application note AN_142	Vinculum-II Toolchain Getting Started Guide
Application note AN_144	Vinculum-II IO Mux Configuration Utility User Guide
Application note AN_145	Vinculum-II Toolchain Installation Guide
Application note AN_151	Vinculum-II User Guide
VNC2 FTDI Web Page	Vinculum-II Web Page
STMPE821 datasheet	DATASHEET CD00186453
Vinco Touch-Key datasheet	Vinco Touch-Key datasheet
Vinco datasheet	Vinco Datasheet
VNC2 datasheet	Vinculum-II Datasheet
IDE Toolchain	Vinculum-II Toolchain

Acronyms and Abbreviations

Terms	Description
USB	Universal Serial Bus
FIFO	First In First Out
SPI	Serial Peripheral Interface
PWM	Pulse Width Modulation
GPIO	General Purpose Input Output
I/O	Input / Output
VNC1L	Vinculum-I
VNC2	Vinculum-II
DMA	Direct Memory Access
IDE	Integrated Development Environment
BOMS	Bulk Only Mass Storage
UART	Universal Asynchronous Receiver/Transmitter
SIE	Serial Interface Engine
CPU	Central Processing Unit
SoC	System-on-a-chip
FAT	File Allocation Table
RTOS	Real Time Operating System
VOS	Vinculum Operating System
OSI	Open System Interconnection
MOSI	Master Out Slave In
MISO	Master In Slave Out
SE0	Single Ended Zero
EMCU	Embedded Micro Central Processing Unit
FPGA	Field Programmable Gate Array

Appendix D – List of Figures and Tables

List Figures

Figure 1.1 – Vinco Development Module and the Touch-Key Shield	2
Figure 1.2 – Motorola Xoom Tablet	3
Figure 2.1 – VNC2 Open Accessory Mode Demo Block Diagram	5
Figure 4.1 – Vinculum II IDE Build Button.....	8
Figure 4.2 – Vinculum II IDE Flash Button	9
Figure 6.1 – Running the demo	11

Appendix E – Revision History

Document Title: Accessing Android Open Accessory Mode with [Vinco Development Platform](#)

Document Reference No.: FT_000547

Clearance No.: FTDI #257

Document Folder: [Vinculum-II](#)

Document Feedback: [Send Feedback](#)

Revision	Changes	Date
draft	First draft	2011-10-11
1.0	First Release	2012-01-20