



Application Note

AN_323

VNC2 Host to Android Open Accessory and FT232

Version 1.1

Issue Date: 2015-11-18

Android devices are now supplied with support for the Android Open Accessory mode. The mode allows USB peripherals to be connected to the Android device even though the only USB port may be a USB peripheral port. There may be instances when a USB peripheral device, based on a FTDI USB Bridge Chip, needs to connect to the Android Device. This application note describes how to program the FTDI Vinculum II IC as both an Android Open Accessory and a host to a FT232-based device.

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.

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

Web Site: <http://ftdichip.com>

Copyright © 2015 Future Technology Devices International Limited

Table of Contents

1	Introduction	2
1.1	Overview.....	2
1.2	Scope.....	3
2	Required Equipment	4
2.1	Connections	4
3	Firmware Discussion.....	5
3.1	AOAtoFT232_iomux.c	5
3.2	AOAtoFT232.h	5
3.3	AOAtoFT232.c.....	5
3.3.1	main()	5
3.3.2	connect()	5
3.3.3	atof()	6
3.3.4	ftoa()	6
4	Conclusion	7
5	Contact Information	8
Appendix A – References	9	
Document References	9	
Acronyms and Abbreviations	9	
Appendix B – List of Figures.....	10	
List of Figures.....	10	
Appendix C – Revision History.....	11	

1 Introduction

Android devices come in many styles. Often time, the only wired communication port available is a USB peripheral. Since there is no USB host function, there is no means to connect another USB peripheral to the Android device. Current Android devices have a protocol called Android Open Accessory (AOA) to allow connection of a USB host device that follows the protocol.

The USB Host / Peripheral relationship is maintained, however, the logical function of the connection is reversed. The Android device and USB Host use the AOA protocol to allow the Android device to have control over the USB host.

There are also many FTDI-based USB peripherals on the market today and they cannot directly connect to an Android device without a USB Host or OTG port. This application note describes how to program the FTDI Vinculum II (VNC2) USB Host / Slave Controller to provide a connection from a FTDI USB Bridge peripheral to an Android device.

1.1 Overview

The VNC2 has two USB ports that can be configured as Host or Peripheral. In this example, both ports are configured as USB host. The AOA driver is attached to USB1 while the FT232 Host driver is attached to USB2. Once both connections are properly identified and established, a bi-directional data path is established between the two USB ports.

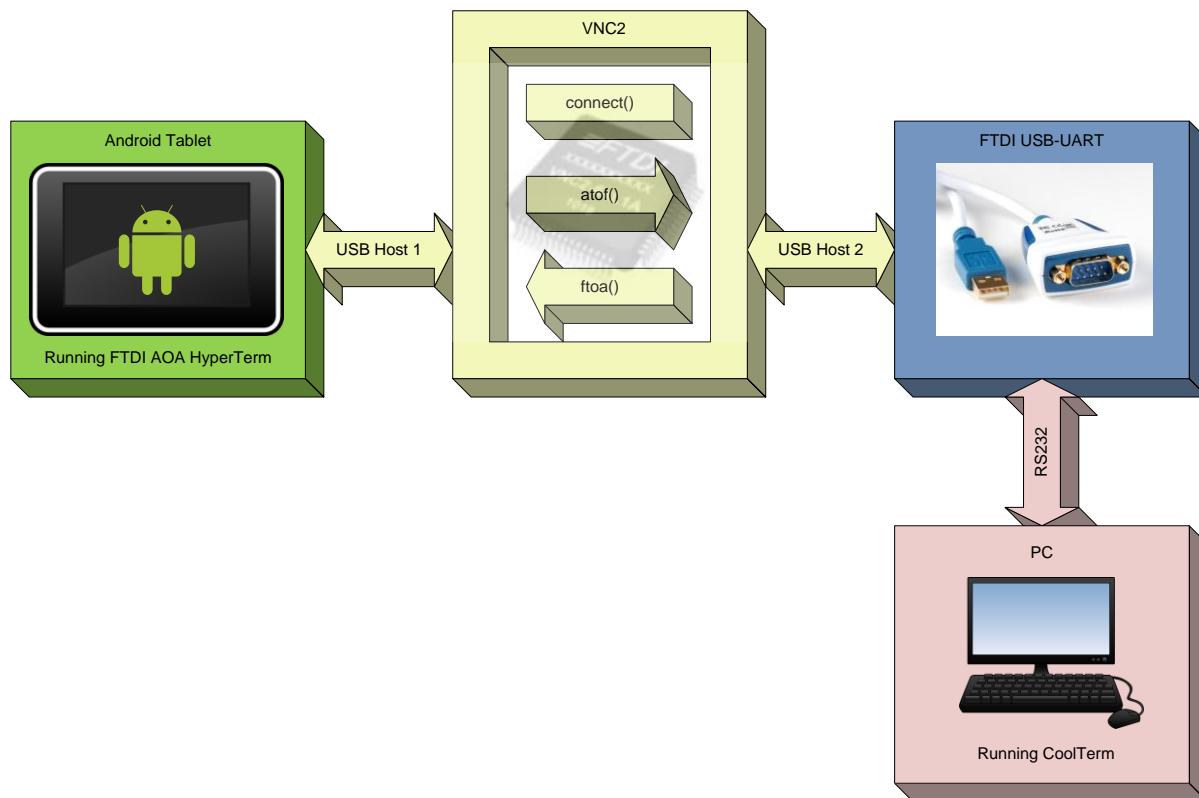


Figure 1.1 - Android to FT232R through VNC2

In order to generate and view data on the Android device, the FTDI AOA HyperTerm application is automatically run upon connection and identification of the VNC2. Similarly, a US232R USB-RS232 cable based on the FT232R is connected to a PC motherboard RS232 COM port and runs a terminal application such as CoolTerm.

Any data present received by the AOA port is copied to the FT232 device. Any data received by the FT232 device is copied to the AOA port.

Unplugging either USB port will cause the VNC2 to halt. LEDs are used for various status events, such as USB port connection, a "heartbeat" while both ports are active, and error conditions.

1.2 Scope

The reader is expected to be familiar with the VNC2 Toolchain IDE. The Vinculum project utilizes several UART modes, though exhaustive error checking is omitted in order to demonstrate the key elements of the connection between the two USB ports. Familiarity with Android devices, the FT232R, and device driver installation is also assumed. Generation of an Android application is not covered in this project.

2 Required Equipment

- 1) Microsoft Windows PC with the VNC2 Toolchain installed
- 2) V2-EVAL board with 32-, 48- or 64-pin VNC2 daughter-card
- 3) FT232R or FT232B-based USB to RS232 adapter (e.g. US232R-10) for connection to the VNC2*
- 4) PC motherboard COM port or a second USB to RS232 adapter for connection to the PC*
- 5) DE-9S to DE-9S standard null-modem cable*
- 6) CoolTerm or similar serial terminal program to run on the PC
- 7) Android device with Android Open Accessory support – this covers most current devices although the VNC2 AOA mode does not support an embedded hub within the Android device.
- 8) FTDI AOA HyperTerm application from the Google Play Store installed on the Android Device
- 9) The source code for this project may be downloaded [here](#)

* A “[FTDI USB Null Modem Cable](#)” can replace the PC motherboard port, the US232R and null modem cable with one product.

2.1 Connections

- 1) In order to ensure that the V2-EVAL has sufficient power for both USB host ports, be sure to use the external power supply that is provided with the V2-EVAL board.
- 2) Open the AN_323.vproj in the VNC2 Toolchain IDE.
- 3) Compile the project and load the firmware into the V2-EVAL.
- 4) Run the project through the debugger, or press the RESET button on the V2-EVAL board.
- 5) Connect USB1 to the Android device using a USB-A to micro-B cable.
- 6) Connect a FT232-based USB-to-Serial adapter to USB2.
- 7) Connect the RS232-side of the USB-to-Serial adapter to a PC RS232 COM port with the null-modem cable – either motherboard- or USB-to-Serial based.

When first started, LED1 on the V2-EVAL will blink at 10Hz. This is an indication that the firmware is awaiting connection of the USB ports.

Once both USB ports are connected, the LEDs next to both USB ports will come on solid. LED1 will go off, indicating that it is awaiting the serial configuration string (baud rate, number of stop bits, number of data bits/byte, parity and flow control).

If LED1 starts flashing at 20Hz, an error occurred with the Android connection. Disconnect the Android USB port, reset the VNC2 and start again.

At this point, the Android device should recognize the connection to the VNC2 as AOA mode and ask to start the FTDI AOA HyperTerm app. Approve this query.

Once the Android terminal application starts, a message will be displayed to send the serial parameters configuration string. Tap “Settings”. Select the port settings that match the RS232 port that will be enabled next.

At the host PC, start a terminal program, such as CoolTerm. Select the appropriate port number and match all of the settings to those chosen for the Android terminal program. Connect to the port.

Now both ends of the communication path are configured and ready. As a quick check, type some text on each terminal. The typed characters should appear on the opposite end. File transfers can also be checked.

In order to end the program, simply unplug either USB port from the V2-EVAL. If the FT232 is disconnected first, a disconnect notice will be displayed on the Android terminal. Once either port is disconnected, LED1 is turned on solid. In order to run this example again, reset the VNC2 and re-connect the USB ports.

3 Firmware Discussion

For this project, the VNC2 Toolchain New Project Wizard was used to generate the initial files:

- AOAtoFT232_iomux.c – the firmware file used to define the VNC2 pins.
- AOAtoFT232.h – the main header file
- AOAtoFT232.c – the main firmware file

3.1 AOAtoFT232_iomux.c

This file contains pin assignments for the I/O pins used by a project. The New Project Wizard will fill in the default pin assignment as noted in the VNC2 Datasheet. For simplicity, all assignments from the project wizard were deleted, except for the actually required pins.

Only three GPIO pins are needed for LED indication. Since the VNC2 is available in three pin-counts, three sections are used to define the LEDs on each size of VNC2. This way, any VNC2 daughter card for the V2-EVAL can be used to achieve the same output.

An additional pin for each VNC2 size is defined as pin 199 for the debug interface. The debug interface is physically on pin 11 on the three available sizes. The Toolchain interprets the pin 199 assignment as pin 11.

The IOMUX setup is called only once while the main() function is executing.

3.2 AOAtoFT232.h

This file contains the required header references for the chosen VNC2 drivers and defines peripheral numbers for each primary and layered driver. The New Project Wizard created this file and no modifications were required.

3.3 AOAtoFT232.c

This file is the primary source of the routines needed to accomplish the desired task. There are four defined functions:

3.3.1 main()

The firmware execution starts here. While main() is running, no other threads have started. The job of main() is to

- configure all of the pins
- setup the threads
- initialize the peripherals
- start the scheduler

Once the scheduler is started, the underlying Vinculum Operating System (VOS) takes control and main() spins in an idle loop until the VNC2 is reset.

3.3.2 connect()

By virtue of locking two mutexes just before starting the scheduler in main(), the connect() thread is the only one that starts when the scheduler is started. This thread assigns the AOA identification strings, initializes the USB ports and the FT232.

The Android Open Accessory protocol uses an “intent filter” – a mechanism of identifying the attached accessory and automatically starting an application that corresponds to the accessory. For this project, the Android device uses the “FTDI AOA HyperTerm” application that is available from the Google Play Store. Since the FTDI AOA HyperTerm app is written to recognize the FTDI FT311D and FT312D dedicated AOA chips, this project programs the VNC2 to have the same strings to allow direct interaction of the Android application with the VNC2 AOA libraries.

The strings consist of the following parameters:

```
char *manufacturer      = "FTDI\0";      // Android Open Accessory ID strings
char *model             = "FTDIUARTDemo\0";
char *description       = "Vinculum Accessory Test\0";
char *version            = "1.0\0";
char *uri               = "http://www.ftdichip.com\0";

char *serial           = "VinculumAccessory1\0"; The connect() thread waits until both ports are
connected with the correct devices (USB1 to the Android device and USB2 to the FT232). Once
both USB ports are occupied, a message prompting the user to send the AOA terminal port
configuration is displayed on the AOA terminal. Once the configuration is sent, the connect()
thread interprets the configuration data into the required settings for the FT232 drivers and passes
those settings onto the FT232.
```

The connect() thread then unlocks both mutexes and spins in a loop while checking the connection status of both USB ports. If one of the USB ports is unplugged, it will drop out of the loop, make all the necessary disconnections and halt the VNC2.

3.3.3 atof()

The atof() thread performs the sole function of reading data entered at the Android terminal program and copies it to the FT232 for transmission out of the UART TXD pin.

It is important to note that the Android Open Accessory driver for the VNC2 does not have a queue status IOCTL call, so no checks for available data are necessary. In order to receive data from the Android device, a vos_dev_read() is executed. This read will block until data is received, up to the size of the buffer used in the call.

Once data is received, it is then transmitted to the FT232 through a vos_dev_write().

The read/write sequence is repeated until the connect() thread senses a USB unplug event and halts the VNC2.

3.3.4 ftoa()

The ftoa() thread sends data in the opposite direction from the atof() function, reading from the data received by the FT232 on the RXD pin and sending it to the Android device.

Unlike the AOA read, the FT232 read *does* have a data available IOCTL call. Since the vos_dev_read() is a blocking call, only the number of bytes available from the IOCTL call should be requested.

This loop is slightly longer than atof() since the presence of data from the FT232 is checked. If data is present, it is then read with vos_dev_read() and copied to the Android device with vos_dev_write().

4 Conclusion

There may be instances when a FTDI-based USB peripheral needs to connect to an Android device.

Possible applications would be an external GPS, card scanner, fingerprint reader, or simply a serial port as shown here. This application note shows how such a connection can be made with any size Vinculum II (VNC2) chip without any special activity, such as rooting, on the Android device. Standard Google Play applications can also be used.

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

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

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 1103, No. 666 West Huaihai Road,
Shanghai, 200052
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.

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

Appendix A – References

Document References

[FTDI AOA HyperTerm application](#)
[FT232R Datasheet](#) (used in US232R cable)
[Vinculum II Datasheet](#)
[Vinculum II User Guide \(AN_151\)](#)
[Vinculum II Toolchain](#)
[Android Open Accessory](#)
[Android Developer Site](#)
[CoolTerm serial terminal program](#)
[AN_323 Source Code](#)

Acronyms and Abbreviations

Terms	Description
AOA	Android Open Accessory
LED	Light Emitting Diode
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USB-IF	USB Implementers Forum

Appendix B – List of Figures

List of Figures

Figure 1.1 - Android to FT232R through VNC2	2
---	---



Appendix C – Revision History

Document Title: AN_323 VNC2 Host to Android Open Accessory and FT232

Document Reference No.: FT_001038

Clearance No.: FTDI# 397

Product Page: <http://www.ftdichip.com/Android.htm>

Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2014-07-01
1.1	Updated the following hyperlinks – Section 2 Required Equipment (on page 4)- step 9 of AN_323 source code Appendix A – References – Document References - AN_323 Source Code.	2015-11-18