Application Note

AN_197

FT232 Emulation with a Vinculum-II

Version 1.0

Issue Date: 2011-12-20

The Vinculum-II (VNC2) is a versatile dual channel USB host/device with many applications. This document describes how to configure a Vinculum-II USB port to emulate the functions of the FTDI FT232 USB-Serial converter.

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

Table of Contents

1 Introduction

The Vinculum-II (VNC2) can be used in a variety of applications. Each of the two USB ports can operate as a Host or Peripheral (aka slave). Layered functions provide specific functions such as HID (keyboards, mice, etc.), BOMS (USB thumb drives) and FT232 emulation. The VNC2 can also be configured to utilize a number of functions available within the devices such as UART, SPI, FIFO and GPIO.

Through the VNC2 Toolchain, it is possible to couple the USB peripheral driver, the FT232 slave layered driver and the UART driver in order to emulate the functionality of a FT232B or FT232R IC. At first glance this may seem like extra effort is needed to create something already done directly in silicon. However, when the power of the VNC2 is coupled with these functions it may be possible to eliminate a discrete microcontroller from a design and incorporate all the required functions within VNC2 itself.

The project files and source code for this application can be downloaded from

http://www.ftdichip.com/Support/SoftwareExamples/VinculumIIProjects/AN_197_Source_Code.zip

The following hardware and software tools are used in this application:

- Vinculum-II Toolchain – the development environment for the VNC2
- V2-EVAL kit with USB A-B cable, USB A-B adapter and external power supply
- An additional USB A-B cable
- USBView – a utility used to display USB device configuration and location information
- Terminal program – any serial terminal program such as CoolTerm or HyperTerminal. The V2-EVAL Terminal Utility can be used for the one port that goes through the V2-EVAL.
- FTDI Device Drivers – These device drivers are used in conjunction with FTDI ICs to provide a host PC with D2XX and/or Virtual COM Ports. Note that FTDI drivers may be used *only* in conjunction with products based on FTDI parts.

Links to the various tools and applications are available in Appendix B.

By following the project outlined within, designers will be able to:

- Program the VNC2 with the basic USB slave to UART functionality of the FT232
- View the USB configuration data on the host PC
- Demonstrate USB to UART functionality
- Be prepared to include additional functionality for their own projects

It is assumed that the reader has installed the Vinculum-II Toolchain and understands how to load and compile projects. Full details on Vinculum-II Toolchain operation are found in AN_151 Vinculum-II User Guide.

1.1 Vinculum-II

The Vinculum-II, or VNC2, is a programmable USB 2.0 host / peripheral controller. It incorporates two USB 2.0 ports that can each operate as a host or slave at full- or low-speed. It is controlled by a 16-bit Harvard MCU core with 256Kbytes of flash program memory and 16Kbytes of RAM. Also connected to the MCU are numerous peripherals: UART, SPI, FIFO, PWM and GPIO. An IO Multiplexor (IO Mux) detects the VNC2 package size and then configures the pin functions at run-time. Six package sizes are available for the VNC2 (32-, 48- and 64-pin in LQFP and QFN). Details on the IOMux are described in AN_139 Vinculum-II IO Mux Explained.

1.2 Vinculum Operating System and Toolchain

Vinculum-II applications revolve around the Vinculum Operating System (VOS). This is a true multi-threaded real-time operating system (RTOS) that underpins all the applications that run on the VNC2. Application programs access the VNC2 resources through an abstraction layer similar in concept to modern desktop operating systems.

The VOS and device drivers for each of the VNC2 functions are provided royalty-free as part of the Vinculum-II Toolchain, the Integrated Development Environment (IDE). Application development, in-circuit programming and debugging capabilities complement the compiler and linker which are designed to enable rapid development. Applications are written in C. Numerous examples and even source code for emulating the previous Vinculum-I are included.

1.3 FT232

The main function of the FTDI FT232R/B devices is to provide a USB slave to UART bridge. It is commonly used to convert existing serial devices to USB as with a common cable such as the US232R-100. Over time developers have removed the RS232 ports on their products and incorporated the FT232 onto their boards.

Many designs that contain the FT232 also contain a microcontroller. With the introduction of the VNC2, designers now have the possibility to eliminate this external microcontroller and roll both the FT232 and microcontroller functionality into a single device. This reduces board size requirements, component count and overall cost.

2 Example VNC2 application

This application illustrates how to configure a VNC2 to function similar to the FTDI FT232. The block diagram in Figure 2.1 shows the basic functionality of the application.

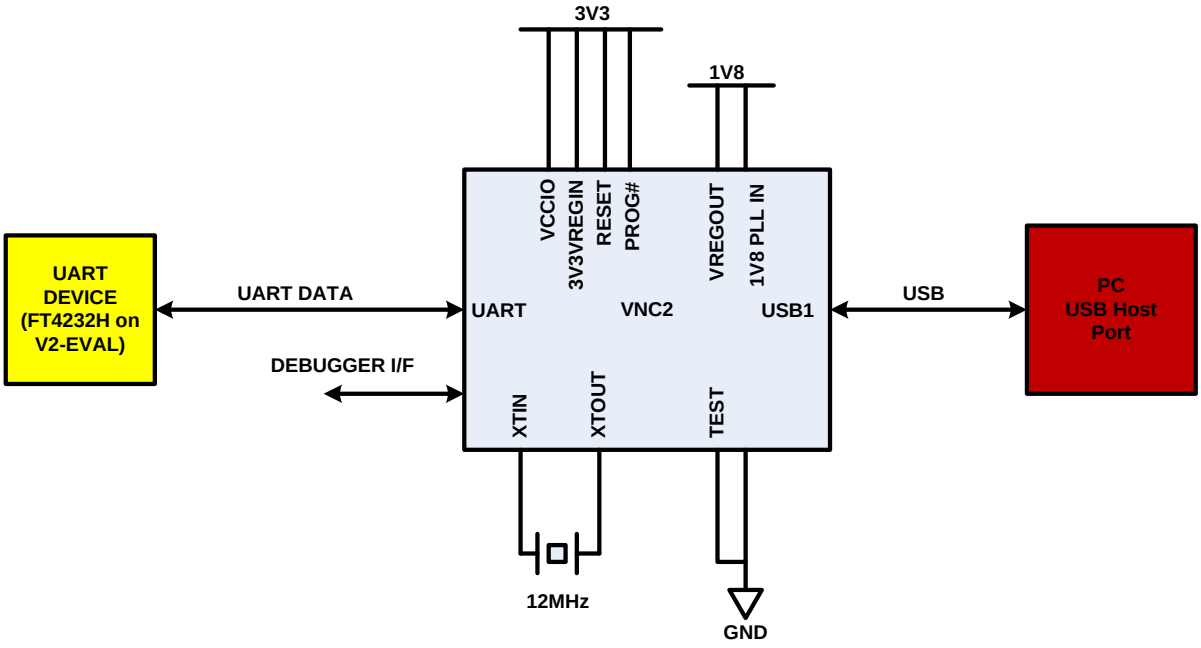

Figure 2.1 Block Diagram of VNC2 Emulating a FT232

The V2-EVAL board is used in this application. On the V2-EVAL board, a FTDI FT4232H provides several functions, including a connection to the VNC2 UART and the debug interface. Another VNC2 circuit could be substituted for the V2-EVAL. The UART interface operates at 3.3V logic levels.

This application began with the "USBSlaveFT232App" example application provided with the Vinculum-II Toolchain download. Throughout the source code, there are several comments that look like:

```
/* FTDI:SDD Driver Declarations */
// UART Driver configuration context
uart_context_t uartContext;
// GPIO Port A configuration context
gpio_context_t gpioContextA;
// USB Slave FT232 configuration context
usbSlaveFt232_init_t usbslaveFT232Context;
/* FTDI:EDD */
```

Notice the "FTDI:SDD" and "FTDI:EDD". These are indications that the block of code contained within the two markers was generated by the Application Wizard. This utility is invoked right through the IDE and provides a quick method of generating an application shell with most of the code required for configuration of the VNC2 at run time. Execution of the Application Wizard is described in AN_151 Vinculum-II User Guide.

The full Vinculum-II Toolchain project is available for free from the FTDI Web Site. A link to the code is provided at the end of this document. Numerous comments describing the code are included throughout the project.

Note that this code is provided for demonstration purposes only and not guaranteed or supported by FTDI.

Each major section of the code is discussed below.

2.1 Execution Start: main()

As with most C programs, this is where program execution starts. The VNC2 is configured for USB port type, pin functions and thread creation within this function.

In this application the first USB port, the UART and a GPIO output pins are configured through a call to iomux_setup(). This function is defined in the project file USBSlaveFT232Emu_iomux.c, also created by the Application Wizard.

Next, a semaphore is initialized to zero. The application has two threads. One of them completes the device configuration after the scheduler is started. The semaphore prevents the second thread from processing any data until after the first thread is done configuring everything.

The final task of main() is to start the VOS scheduler. The last two lines of the function are a tight loop at the lowest priority. At this point, the operation of the VNC2 is completely defined for this application. No further configuration is possible once the scheduler is started.

2.2 Threads: uartTx() and uartRx()

The uartTx and uartRx threads perform the same function, but in opposite directions. The uartTx thread also takes care of the various setup tasks.

The uartRx thread reads data from the UART and copies that data to the USB port. Before uartRx can start processing data, it must wait on the semaphore. By calling vos_wait_semaphore(), the semaphore count is decremented (i.e. now made less than zero). The thread is then blocked until the uartTx thread completes its setup tasks.

The uartTx thread starts by obtaining handles to all the base devices (USBSlave, UART and GPIO). It then attaches the USBSlaveFT232 layered driver to USBSlave and obtains that handle. Next, it initializes the UART parameters and GPIO direction. It then calls vos_signal_semaphore() which increments the semaphore count back to zero and unblocks the uartRx thread. Finally, it reads data from the USB port and copies that data to the UART port.

At this point, both threads are in their respective while(1) loops and processing data through the VNC2.

2.3 UART Initialization

One of the uartTx initialization tasks is to configure the UART. The function open_drivers() opens the three peripherals used in this application, USBSlave, UART and GPIO. This is followed with a call to initalize_uart() where all of the UART parameters are defined:

- Baud Rate = 9600bps
- Flow Control = RTS/CTS
- Number of Data Bits = 8
- Number of Stop Bits = 1
- Parity = none

In addition, the DMA controller is enabled for the UART port. DMA allows for more efficient data routing between the UART and memory and frees the VNC2 core for other tasks.

Note that all of the UART parameters must match the device to which the UART is connected. In the case of this example, the UART is connected to Port A of the FT4232H on the V2-EVAL board. A common terminal program will be used with matching settings.

2.4 USB Initialization

The USB port is initialized in two steps. As noted for the UART initialization, the function open_drivers() opens the USB port along with the other peripherals. Following the open, a call to attach_drivers() results in another call to ft232_slave_attach(). It is within ft232_slave_attach() that the handle to the FT232Slave is obtained, various USB configuration descriptors are set and finally the FT232Slave driver is layered (attached) to the USBSlave driver. All data through the USB port is then accessed through calls to the FT232Slave driver rather than its underlying USBSlave driver. The table below shows the relationship of driver layer as the FT232 emulation is built from the bottom up.

FT232 Slave Driver Heirarchy
USBSlaveFT232 Driver
USBSlave Driver
VOS Kernel
VNC2 USB Slave Port (hardware)

Table 2.1 USB Slave Driver Heirarchy

Several USB configuration descriptors are fixed, such as the type of transfer, number of endpoints, packet size, etc. Other descriptors can be changed. This application demonstrates setting all of the changeable descriptors:

- Vendor ID = 0x0403 (for FTDI)
- Product ID = 0x6001 (for a FTDI single-port device)
- Manufacturer String = "ACME Ltd"
- Product Description String = "VNC2 as FT232"
- USB Serial Number String = "VNC20001"
- Power Type = Self Powered with remote wakeup disabled
- Power Requirement = 2mA

DMA does not specifically need to be enabled for the USBSlave or FT232Slave devices. This is done by default when these drivers are used.

Full details on the USB configuration descriptors available to the VNC2 are described in AN 168 Vinculum-II USB Slave Customizing an FT232 Device.

2.5 GPIO Initialization

The final configuration task performed by uartTx() is to configure GPIO Port A, bit 7 as output. In this example this bit is connected to a LED on the V2-EVAL board and used to display when data is flowing in either direction.

3 Compiling and Running the Application

3.1 V2-EVAL configuration

The V2-EVAL requires the following configuration. These connections and settings are also shown in Figure 3.2 below.

1. Enable the "USB Terminal" function by setting JP11 to the ON position. The red Terminal Active LED will be illuminated while the V2-EVAL is powered-up and JP11 is ON.
2. Disable the "VBUS Enable" connection for USB1 by removing JP3. It can be stored on a single pin to avoid loss.
   NOTE: *Failure to remove this jumper may result in damage to the attached Host PC or hub, the V2-EVAL or both.*
3. Connect JP12 to the "P.S." position. Apply 5VDC from the supplied power supply to the V2-EVAL through the power port, CN13, and turn SW7 to the ON position.
4. Connect a USB A-B cable to CN12 and the host PC or hub running the Vinculum-II Toolchain.
   a. Install the device drivers. Windows Vista and Windows 7 with default settings will obtain the drivers directly from Microsoft Windows Update. See the appropriate installation guide if the operating system is running Windows XP or before, or has settings to prevent Windows Update driver downloads.
   b. Two COM ports will be assigned. Note the COM port number with the smaller number. This will be used later with the terminal program. The COM port number can be found through the Windows Device Manager.

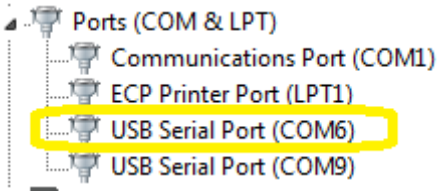

Figure 3.1 V2-EVAL UART (Port A)

5. Connect a second USB A-B cable to the host PC or hub. Double-check to ensure JP3 is removed. Connect the B end to the A-B adapter and then plug the adapter into USB Port 1 on the V2-EVAL, CN1.
   NOTE: *The A-B adapter is only for use for development or test purposes. It is not intended for use in normal end-user applications. Hardware designs that use the VNC2 as a USB slave device should use one of the standard B-type peripheral connectors (regular, mini or micro).*
6. LED1 is used to indicate data traffic in both directions. A single LED is used due to the limited number of pins available for the 32-pin VNC2 packages. Any size VNC2 may be used, and each direction can have its own traffic LED.

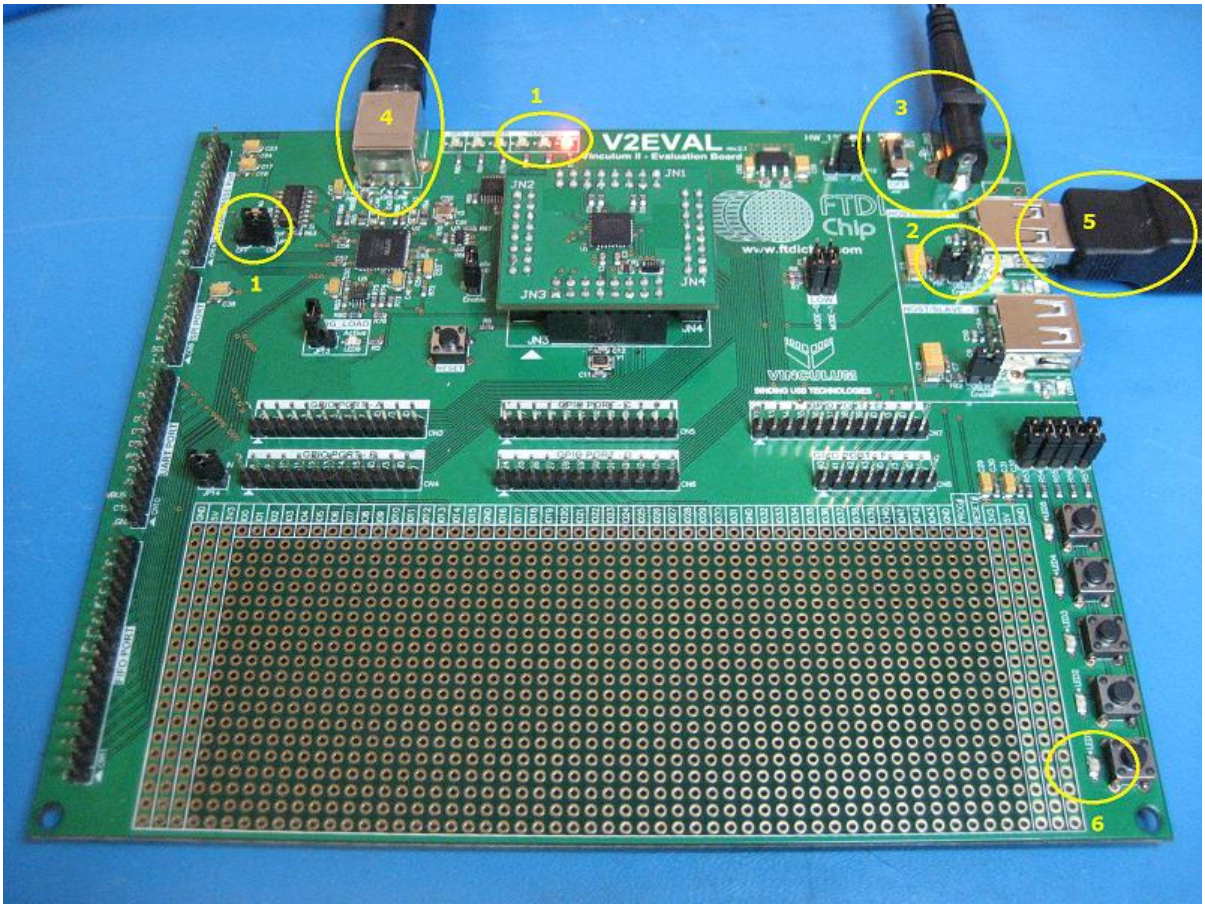

Figure 3.2 V2-EVAL Setup

3.2 Vinculum-II Toolchain Configuration and Project Build

1. Open the Vinculum-II Toolchain IDE (hereafter referred to as the IDE).
2. Open the USBSlaveFT232Emu.vproj project.

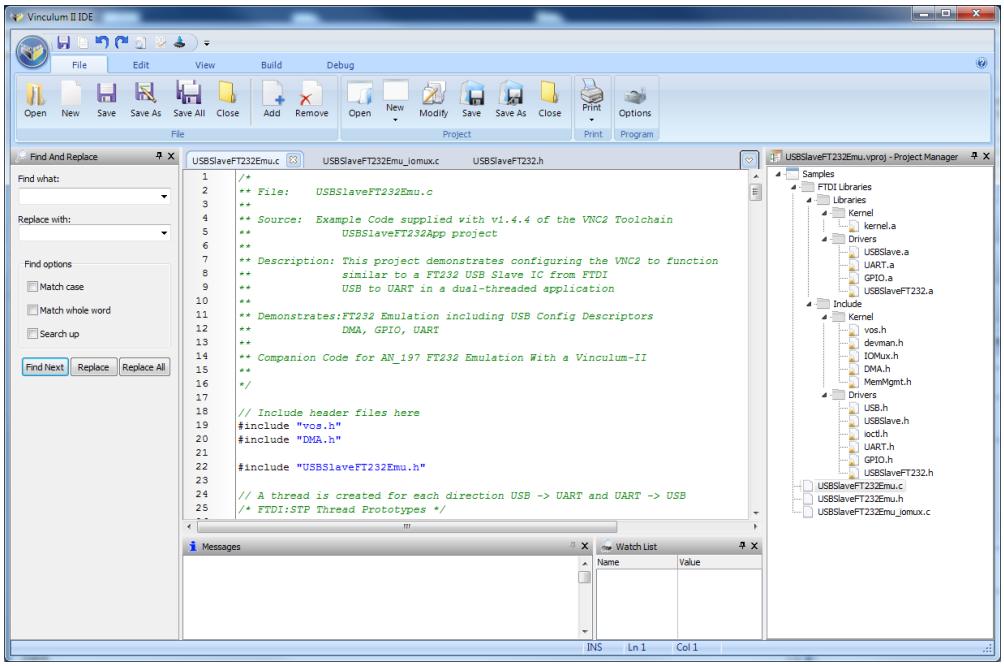

Figure 3.3 Vinculum-II Toolchain Showing Source Code

3. Display and review the source code, USBSlaveFT232Emu.c and USBSlaveFT232Emu_iomux.c.
    a. Note the various functions outlined in Section 2.
4. Click on the Build tab and build the ROM file. (The warnings during the build can be ingored.)
    a. If a debug build configuration is selected, interactive debugging such as single-step and breakpoints can be used.
    b. If a release build configuration is selected, debug functions are disabled.

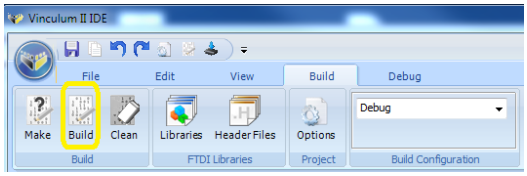

Figure 3.4 Building the Application

5. Click on the Debug tab.
    a. Select "V2EVAL Board C" as the debugger interface. There may be additional digits following the debug interface name.
    b. Click on the Flash button to download the newly compiled file.

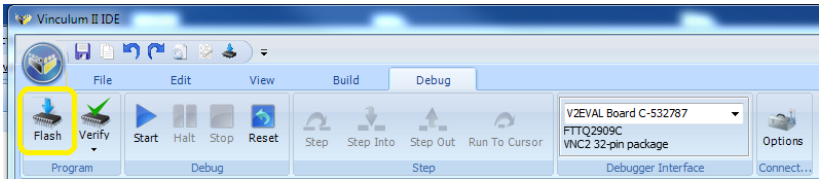

Figure 3.5 Programming the Application into the VNC2 Flash

Full details on Vinculum-II Toolchain operation are found in AN_151 Vinculum-II User Guide.

3.3 Run the Application

1. If a debug build configuration was selected, click the Start button in the IDE. If a release build configuration was selected, either click the Start button in the IDE or press the hardware reset button (SW6) on the V2-EVAL board.
2. As the VNC2 is configured through the application firmware, the USB Slave device and FT232 layered drivers are initialized and recognized by the host operating system. To view the new device, open the Device Manager and make note of the newly added COM port number.

Figure 3.6 New VNC2 COM Port

3. Open two instances of a COM port terminal program. The V2-EVAL Terminal Utility can be used for the UART that is connected through the FT4232H on the V2-EVAL.
4. If using the V2-EVAL Terminal Utility, select "V2EVAL Board A". If using another terminal program, select the COM port number noted in Section 3.1, item 4b. The port will need to be set for 9600bps, no parity, 8 data bits and one stop bit with RTS/CTS flow control. Figure 3.7 shows the V2-EVAL Terminal Utility connected to V2EVAL Board:

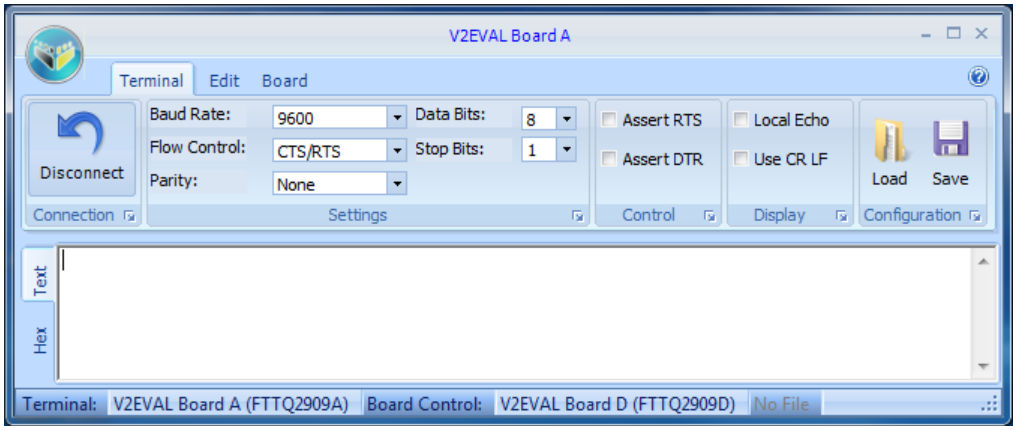

Figure 3.7 V2-EVAL Terminal on Port A

Note that throughout this example, the "V2-EVAL Board A" has been assigned to COM6.

5. For the second terminal program, select the COM port number noted in step 2 in this section. Baud rate and port parameters are not passed from the host PC to the VNC2.

They are configured as noted in Section2.3 Also see section 4.1, below, regarding this limitation.

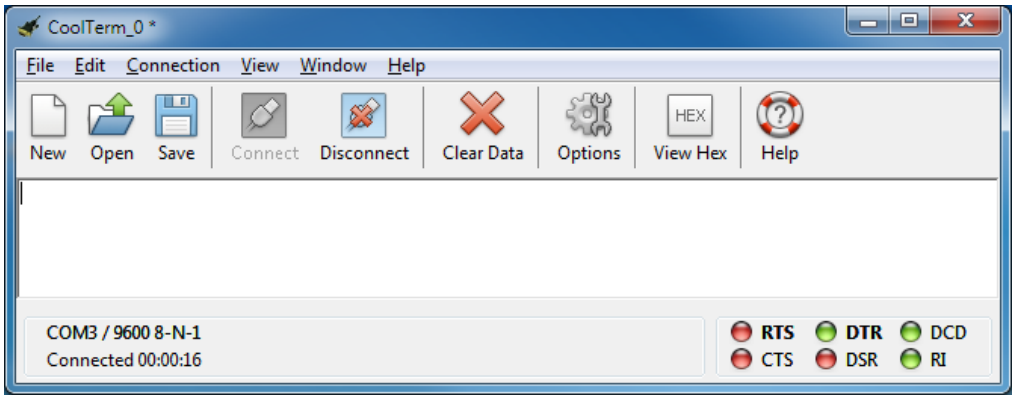

Figure 3.8 CoolTerm on COM3 (VNC2 as FT232)

6. Finally, make sure both terminal connections are active.
7. Data entered in one terminal will appear in the other and vice-versa. In Figure 3.9, "Hello World" was typed into the V2EVAL Board A window and "VNC as FT232" was typed in the CoolTerm window. If data appears in both windows when typing, the "Local Echo" function of the terminal is likely enabled.

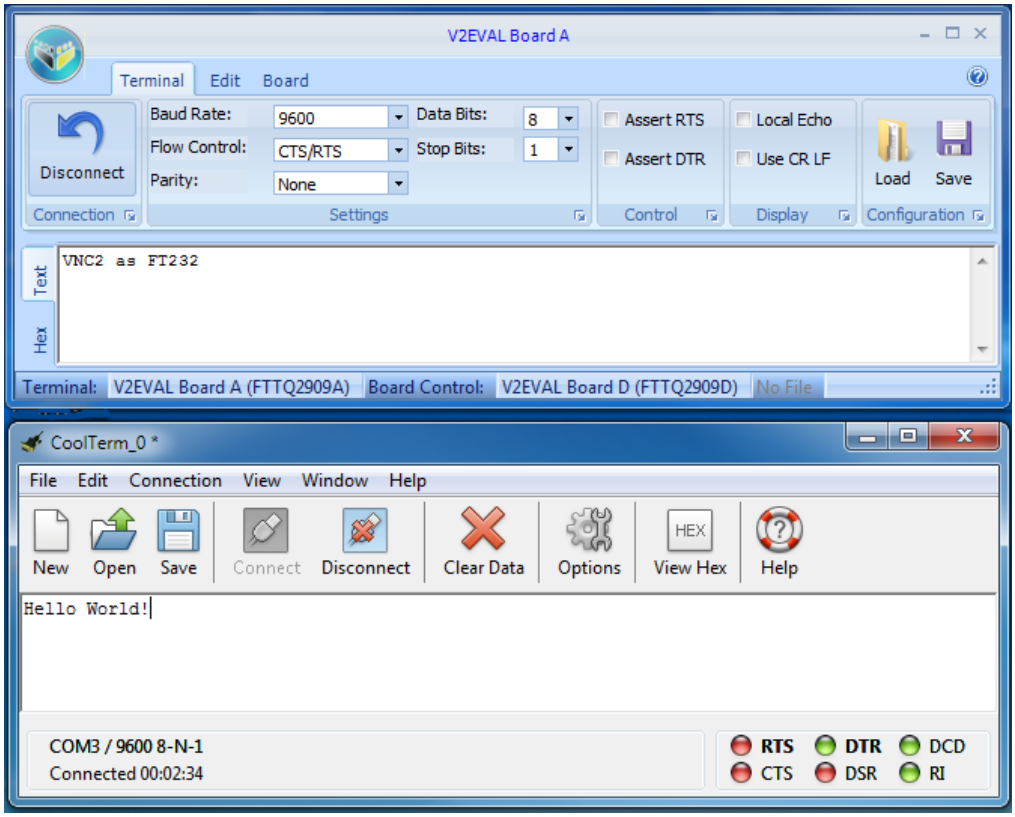

Figure 3.9 Data Transfer Between Ports

4 Program Notes

4.1 Limitations

This application note shows how to use the VNC2 as if it were a FT232 device. There are a few limitations that should be noted:

- The USB configuration descriptors are defined within the VNC2 source code and resulting ROM file. Utilities such as FT_Prog are not designed to be used with the VNC2 to modify these descriptors.
- The UART parameters (baud rate, flow control, etc.) are also defined in the VNC2 code. COM port settings configured in the host PC terminal program, or other application, are not passed through and have no effect on the VNC2 UART.
- The VNC2 I/O signals operate at 3.3V and are 5V tolerant where the FT232R and FT232B can operate over a wider voltage range as noted in the respective datasheets.

4.2 Further development

This application demonstrates a simple USB peripheral to UART bridge. While this is readily available with the FT232-series parts, this function is only tapping a small portion of the data processing capabilities and other peripherals available inside the VNC2. Other tasks can be performed, such as:

- Data processing – Read data from either from the PC host or UART and then act on or modify it before sending it to the other port.
- Other peripherals – Use the SPI, PWM, USB Host. For example a VNC2 application could use USB1 to connect to a PC as shown here while also making the USB Host port available for a thumb drive. In the field the thumb drive is used for data logging. When the device is brought back to a host PC, the FT232Slave can be used to retrieve the data from the thumb drive.
- USB Slave Connect/Disconnect – There may be applications that need to dynamically attach and detach the USB Slave drivers based on whether a USB cable is connected. This is explained in AN_163 Vinculum-II USB Slave Detecting Disconnect and not covered here. Designs destined for USB-IF certification will need this detection mechanism added to the VNC2 application.

4.3 Use your imagination!

The VNC2 can be used to replace not only a FT232 but also utilize the MCU functionality to perform multiple tasks and eliminate the need for a discrete MCU chip. This saves board space, bill-of-materials count and money.

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 – 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

Appendix A – References

Document References

http://www.ftdichip.com/Support/SoftwareExamples/VinculumIIProjects/AN_197_Source_Code.zip

Vinculum-II Toolchain

AN_139 Vinculum-II IO Mux Explained

AN_151 Vinculum-II User Guide

AN_163 Vinculum-II USB Slave Detecting Disconnect

Vinculum-II Datasheet

V2-EVAL Evaluation and Prototyping Platform for VNC2

FT232B Datasheet

FT232R Datasheet

FTDI Device Drivers

CoolTerm – A freely available third-party terminal program

Acronyms and Abbreviations

Terms	Description
BOMS	Bulk-Only Mass Storage
FIFO	First-In First-Out
GPIO	General Purpose Input/Output
HID	Human Interface Device
IC	Integrated Circuit
IDE	Integrated Development Environment
IOMux	Input/Output Multiplexor
MCU	Microcontroller
PWM	Pulse-width Modulation
RTS/CTS	Request To Send / Clear To Send Hardware Flow Control
SPI	Serial Peripheral Interconnect
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USB-IF	USB Implementers Forum

Appendix B – List of Tables and Figures

List of Tables

List of Figures

Appendix C – Revision History

Document Title: AN_197 FT232 Emulation with a Vinculum-II
Document Reference No.: FT_000548
Clearance No.: FTDI# 238
Product Page: http://www.ftdichip.com/Products/ICs/VNC2.htm
Document Feedback: Send Feedback

Revision	Changes	Date
1.0	Initial Release	2011-12-20