



Future Technology Devices International Ltd.

Technical Note

TN_133

Vinculum II Toolchain Release Notes

Document Reference No.: FT_000501

Version 1.4.2

Issue Date: 2011-08-11

Release Notes for the Vinculum II Toolchain Version 1.4.2

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 © 2011 Future Technology Devices International Limited

Table of Contents

1	Introduction	3
1.1	Overview.....	3
1.2	Scope	3
1.2.1	Features.....	3
1.2.2	Enhancement.....	3
2	Readme File for V1.4.2	4
2.1	Toolchain Release Information	4
2.2	Frequently asked questions	4
2.3	VinAsm Readme	4
2.4	VinC Readme.....	5
2.5	VinCpp Readme	6
2.6	VinL Readme	6
2.7	VinIDE Readme	6
2.8	Firmware Release Information	8
2.8.1	VOS Kernel Readme.....	8
2.8.2	Runtime Libraries Readme.....	8
2.8.3	Drivers Readme	9
2.9	Samples Release Information	10
2.9.1	VNC1L Firmware Readme	10
2.9.2	CDCModem Sample Readme	11
2.9.3	USBHost FT232 UART Echo Sample Readme	11
2.9.4	Firmware Update (AN 159) Readme.....	11
2.9.5	KITT Sample Readme.....	12
2.9.6	Hello World (Library) Sample Application Readme.....	12
2.9.7	Philosophers Sample Readme.....	13
2.9.8	PWMBreathe Sample Readme	13
2.9.9	Runtime Sample Readme	13
2.9.10	SDCard Sample Readme	14
2.9.11	TickTock Sample Readme	14
2.9.12	RTCEXample Sample Readme.....	14
2.9.13	USBHost Android Accessory LED Sample Readme	15
2.9.14	USBHost Still Image Camera Sample Readme	15
2.9.15	USBHost Device Descriptors Sample Readme	15
2.9.16	USBHost FT4232Hub Sample Readme	16
2.9.17	USBHost Generic Device Driver and Application Sample Readme	16
2.9.18	USBHost GPS Logger Sample Readme.....	16

2.9.19	USBHost HID Class Sample Readme	17
2.9.20	USBHost HID Class Sample 2 Readme.....	17
2.9.21	USBHost Printer Sample Readme.....	17
2.9.22	USBHost Microphone Sample Readme	18
2.9.23	Webcam Demo Sample Readme	18
2.9.24	USBSlave FT232 Sample Readme	18
2.9.25	USBSlave HID Keyboard Sample Readme	19
2.10	Revision Information	19
2.10.1	Compiler Tools	19
2.10.2	Development Environment.....	19
2.10.3	Firmware.....	19
3	Contact Information.....	21
4	Appendix A – References.....	23
	Document References	23
	Acronyms and Abbreviations	23
5	Appendix B – List of Tables & Figures	24
6	Appendix C– Revision History.....	25

1 Introduction

This document contains the release notes for the Vinculum II ToolchainV1.4.2.

1.1 Overview

This document is based on the information contained in the ReadMe.txt file installed with the Vinculum-II Toolchain. Additionally recent additions and updates to that file have been included.

Sections added since the publication of the Toolchain are shown with a vertical bar on the left of the text. Any text removed or superseded is shown in ~~strikeout~~.

1.2 Scope

This document applies to the V1.4.2 release.

1.2.1 Features

The V1.4.2 release was made on 19th July 2011.

1.2.2 Enhancement

This release added support for the Android Open Accessory Mode (in addition to sample Android application source code), plus additional Ethernet and RTC drivers.

In addition to performance improvements added to the USBHost, USBSlave and FAT File System, the Application Wizard has also been updated to support Vinco projects.

Also included are samples of pre-compiled VNC2 Romfiles for FT232 to UART and CDC Modem to UART applications, plus an application to allow customisation of the user area of FlashROM. There is an API to permit unused areas of FlashROMto be programmed and read from an application.

2 Readme File for V1.4.2

The following changes have been made from the published ReadMe-V1.4.2.txt file in the toolchain.

Added to VinIDE known bugs.

Add missing function strncpy() to Runtime Libraries known bugs.

Added in missing ReadMe information for firmware sample \Samples\General\RTC Example. See section RTCEXample Sample Readme.

Added known bugs to the VNC1L Firmware.

SPI Master Link to autotoggle in SPIMaster IOCTL page.

2.1 Toolchain Release Information

This is Release 1.4.2 of the Vinculum II Toolchain, comprising the Vinculum II Integrated Development Environment, compiler tools, debugger, kernel, device drivers, libraries and sample code.

New to this release for the Integrated Development Environment are improvements to the Application Wizard and Variable Watch Windows.

The firmware has source code for USB Host CDC class driver, printer class driver and Android Open Accessory drivers. There are more Sample applications including Android Open Accessory samples which include full Java source code for Android devices.

2.2 Frequently asked questions

- Sample applications are written to the folder "FTDI\Firmware\Samples\V1.4.2\" in the user's "My Documents" directory to prevent overwriting of modified existing samples. If any samples will be overwritten the installer will prompt first. Note that the V1.0.0 installer wrote samples to the folder "FTDI\Firmware\Samples".
- The linker will always generate a .map file with the symbols and functions in the application. It will not list symbols and functions in the libraries.
- There is no listing file generated by the linker.
- The FT_PROG utility has been updated to program the VNC2 via the UART interface - Version V1.12.
- Breakpoints are disabled in the following circumstances: The first 3 active breakpoints are used when the debugger is started, remaining breakpoints are disabled. Breakpoints which are on a line with no valid code are disabled. A message will be displayed in the Messages Panel informing you of any breakpoints that are disabled.
- Floating point is not supported.
- If the IDE reports "The System cannot find the specified file" for a library or header file then please check that the VINC_INCLUDE and VINL_INCLUDE environment variables point to the correct directory. There should be four entries in each pointing to the Firmware folder in the toolchain installation directory.
- If the IDE reports "Vinc.exe failed with return code 0" when compiling code please check that the path to the tools in the Program Options dialog box, Environment Options are correct.

We welcome suggestions for new features or improvements to existing features.

Please see the release notes for each tool which will be available in a file called "ReadMe.txt" in the Start Menu folder. The ReadMe files for previous releases are also available in the installation directory.

2.3 VinAsm Readme

Fixed Issues:

- Fix for Lexer not unescaping strings properly

Restrictions:

- N/A

Known bugs:

- N/A

Feature requests:

- N/A

2.4 VinC Readme

Fixed issues:

- Fixed bug where return statements in void functions would return a value if a constant was specified (i.e. `return -1;`). This corrupted data stored in the calling function.
- Removed incorrect warning message for initialisation of signed values with negative values (e.g. `short a = -1500;`)
- Corrected error when an enum type used on right hand side of less than equals or greater than equals in comparison.
- Fixed bug with `sizeof()` function when evaluating constant strings with control characters.
- Fixed the `sizeof()` function to return the size of a dereferenced pointer: `structmyStruct *m; if (sizeof(*m) == 0) {};`
- Corrected Logic operations on pointer values which gave an error: `char *x, *y; if (x || y) {};`
- Efficiency of code generated (both ROM and RAM usage) when optimisation turned off improved.
- More code size optimisations identified and implemented.
- Fixed issue where strings assigned to local arrays of size `char` did not include terminating NULL character.
- Removed compiler error when double de-referenced pointers were incremented or decremented (e.g. `(**p)++;`)
- Corrected implementation of arrays of function pointers and function pointer as first member of structure or union.
- Fix compiler bug where an arithmetic operation of a signed short on the left and a signed integer on the right was treating result as a signed short even if the result was not a short.
- Fix compiler crash when default case was first case in a switch statement.
- Incorrect result from pointer dereference arithmetic operations corrected (e.g. `(*p) += 1;`)
- Remove compiler crash when label name matches another symbol name exactly.
- Corrected issue when symbols which differ only in case were not treated separately.
- Fixed issue with escaped hexadecimal characters in strings not being unescaped correctly.

Restrictions:

- Floating point not supported.
- Structs and unions may not be passed to or returned from a function. Pointers to structs and unions are allowed.
- Vacuous declarations of structures and unions is not supported.
- Structure and union assignments not supported.
- Anonymous aggregates, unnamed members in unions, are not supported.
- Compound literals expressions are not supported.
- Modulus operations treat signed values as unsigned values. Do not use with negative values.

Known bugs:

- Function pointer definitions may not be used in function declarations or prototypes.
- Labels may not share the name of a symbol (variable, function etc) within the scope of the function in which the label appears.
- Cannot assign enum identifier expression to enumerator.

Feature requests:

- Full ANSI compliance.

2.5 VinCpp Readme

Fixed Issues:

- Corrected missing line when a C comment spanned 2 lines exactly. This affected debugging only and not the `__LINE__` macro.
- Fixed issue when a comment follows on the same line as a conditional directive with a compare operator, where the condition was being evaluated as false. (e.g. `#if <MACRO>==<VALUE> /* comment */`)
- Fixed issue on exception generated when there's no new-line character after `#endif` directive
- Fixed issue on unable to compile an empty application with all libraries included (Vinco, Hardware and Run-time libraries)

Restrictions:

- N/A

Known bugs:

- N/A

Feature requests:

- N/A

2.6 VinL Readme

Fixed Issues:

- N/A

Restrictions:

- N/A

Known bugs:

- N/A

Feature requests:

- N/A

2.7 VinIDE Readme

Fixed Issues:

- Display ASCII characters for char size variables in watch windows
- Display the correct format of watch values
- Improved display of watch values for structures, unions and arrays
- Maintain collapse/expand state in Watch windows during debugging
- Display of ASCII values in Memory window

-
- Button to view a Map file added
 - Able to remove a shortcut key assignment
 - Add shortcut keys for Close Project, Make, Clean, Close File, Add File & Remove file buttons
 - Current directory will strictly follow the file shown in the editor
 - Memory window and CPU Status window will clear after closing the project or stopping the debugger
 - Added red markers for non-existing archive files belonging to a project
 - Added "Add Watch" Context Menu
 - Fixed issue on Copy & Paste using Goto Line dialog box
 - Corrected issue when executing some shortcut keys
 - Fixed issue where a modified file marker was added to unmodified files in the Project Manager
 - Updated current directory when using recent project list
 - Corrected operation of delete all breakpoints
 - Fixed single step/step in/step out issue in disassembly
 - Added option to deselect a debugger interface.

Restrictions:

- Verify function will only work with Program Loader V1.7 and above. Please contact support for further information on obtaining this update. The IDE will report an error box saying "Cannot verify flash contents! Use latest version of the program loader."
- Cannot add a file with the same name as an existing file in the Project Manager, even if it in a different directory and stored in a virtual folder.
- Conditional code compilation will recompile all sources if header files are modified.
- Verify button will perform slowly unless Prog_Loader V1.7 is programmed onto the VNC2. (Please contact support for details)

Known bugs:

- Compiler may report a "C0011 Preprocessing failed" error if the Windows environment variable is not set to the full path of cmd.exe. It should be set to "%SystemRoot%\system32\cmd.exe" rather than just "cmd.exe".
- Undo limit is currently not implemented by Code Editor.
- Asm files with multiple tab lines take some time to load.
- Busy cursor persists when debugging actions are performed when the Memory Panel has focus. Switching to another panel and performing a debug action will clear the cursor state.
- Certain patterns of comments in source code files may cause syntax colouring to turn off temporarily while scrolling.
- Selecting a line of code which starts with a tab character may not select all characters at the end of the line.
- Flashing a ROM file then immediately verifying may result in an Internal Error message.
- Local Variables Watch Window will always be opened when debugging starts.
- Shortcut keys for Watch Window and adding breakpoints don't work.
- Building a project after modification of a file (but not saving the changes) will prompt a confirmation, choosing 'Yes' will popup an error message, but choosing 'Yes to all' will build the project.
- App Wizard overwrites whole of <solution>_iomux.c file – should only update between markers.
- App Wizard IOMuxsetup output does not make vos_iocell_set_config() calls in output. Drive current, trigger, slew-rate and pull up/down settings must be made manually in the <solution>.c file. Do not add these to the <solution>_iomux.c file as these will be overwritten when the App Wizard is re-run.
- Debugger sometimes appears not to return from a Step or Run operation. If Halt is pressed then the debug cursor appears at the expected location.
- Memory Window, CPU Status Window and Watch Windows may appear blank if they are started after debugging has started. Data will appear when next breakpoint is reached.
- Pasting text multiple times into the IDE editor rapidly may make the program unresponsive.
- If examining pointers to structs in the Watch Windows the incorrect values may be shown.
- Code conditional compilation (Make button in Build tab) will build all files the first time a project is opened.

Feature requests:

- Code Folding and other editor enhancements.

-
- Regular Expression Searching.
 - Find in all files in project.
 - Context & Keyword help functionality.
 - Code auto completion, parameter hinting, declaration and definition hints.
 - Multiple Memory Panel views and Watch Window views.
 - Application Wizard to configure driver setup values, e.g. number of interfaces for USBHost driver.

2.8 Firmware Release Information

This is version 1.4.2 release of the Vinculum II Firmware. Comprising of the Vinculum Device Drivers, C Runtime Libraries and updated firmware sample code.

Major updates are changes to the USBHost driver to improve handling of device removal and insertion detection especially when devices are over USB hubs; addition of the USBHostAndroidAccessory driver allowing VNC2 to interface to devices using the Android Open Accessory Class; performance improvements for the USBSlaveFT232 driver.

Source code is now installed in the samples directory for USBHostPrinter, USBHostCDC and USBHostAndroidAccessory drivers. This can be modified or used as a basis for other drivers.

Sample code has been rewritten to make full use of the App Wizard in the IDE meaning sample projects can be modified more easily. A sample application for Android Open Accessories, including full Java source code for an Android device, has been added to demonstrate a simple accessory device using VNC2 as a USB Host communicating with an application on an Android device.

2.8.1 VOS Kernel Readme

Fixed Issues:

- `vos_wdt_enable()` now returns a value to indicate success or otherwise.

Restrictions:

- Watchdog operation requires a minimum of Prog_Loader V1.7 to function. `vos_wdt_enable()` will return an error if this is not present on the VNC2.

Known bugs:

- N/A

Features requests:

- N/A

2.8.2 Runtime Libraries Readme

Fixed Issues:

- Corrected output of `sprintf()` which reversed output for decimal values.

Restrictions:

- N/A

Known bugs:

- The `strncpy()` function is missing from the `string.a` library. This will be added in the next release.

Features requests:

- N/A

2.8.3 Drivers Readme

Fixed Issues:

- USBHostCDC driver missing first 2 characters returned by device.
- USBHost corrected behaviour of `VOS_IOCTL_USBHOST_DEVICE_GET_NEXT_ENDPOINT_HANDLE` to find next endpoint of same type. Previously matched type (bulk, interrupt, isochronous) but did not match IN and OUT direction.
- Android OpenAccessory driver for USBHost added.
- RTC driver for Vincoshield added.
- All drivers changed to use `usbhost_ep_handle_ex` and `usbhost_device_handle_ex` (backward compatible)
- FAT Driver improved device removal checks when performing read or write operations.
- BOMS driver fixed issue where it would not set LUN To last LUN on device.
- FirmwareUpdateFATFile/Reflash can show progress of UART dot trail, GPIO port A bit toggle or both.
- FirmwareUpdateFATFile/Reflash did not work when UART was in DMA mode.
- FirmwareUpdateFATFile/Reflash will checksum file before allowing programming.
- FirmwareUpdateFATFile/Reflash will restart when complete and run installed code.
- USBHost hub device detection of insertion and removal improved.
- USBHost configuration for devices on Root hub can be changed using `VOS_IOCTL_USBHOST_DEVICE_SET_PORT_CONFIGURATION` call.
- USBHost enumeration carried out in separate thread resulting an overall reduction in stack space used by the USBHost driver.
- USBHost extended endpoint and device handles available for dynamic checking of device disconnect status without requiring to poll for bus disconnection. Enabled via `VOS_IOCTL_USBHOST_SET_HANDLE_MODE_EXTENDED` call.
- UART, SPIMaster, SPISlave, FIFO improved performance and reduced memory usage when sending data in non-DMA mode.
- SPIMaster added new IOCTL (`VOS_IOCTL_SPI_MASTER_AUTO_TOGGLE_SS`) to automatically toggle the specified SS lines during an SPI master write to improve performance.
- USBHost correct periodic polling intervals of interrupt endpoints. Endpoints will be polled at least as often as specified in the device descriptors. Polling interval can be 1, 2, 4, 8, 16 or 32 msecs.
- USBSlaveFT232 fix synchronisation issue with IN endpoint.
- FAT `fat_fileWrite()` EOF marker could be placed at wrong location in file when overwriting and truncating a file.
- FAT `fat_freeSpace()` (and hence `FAT_IOCTL_FS_INFO`) would count beyond the end of the FAT table on some disk geometries.
- FAT performing a `fat_freeSpace()` operation when a file larger than one cluster was open for writing would result in the last cluster of the file being lost.
- FAT incorrect cluster read on a `fat_fileRead()` following a single read of more than 2 sectors (>1024 bytes) which spans a cluster boundary.
- FAT could read past end of file if current position was within first sector of a cluster and length to read would not breach current sector.
- SPISlave use bitmask to ensure that the slave address passed by the application was within bounds.
- UART, SPISlave, SPIMaster, FIFO when using `VOS_IOCTL_COMMON_ENABLE_DMA` parameter in IOCTL structure was not bounds checked. Can be only `DMA_ACQUIRE_AS_REQUIRED` or `DMA_ACQUIRE_AND_RETAIN`.
- Support multiple instances of the FAT file system. Previously only one was allowed. Be aware that this library requires over 1.5kB of RAM per instance.
- USBHost will configure the first available (default) configuration for a device during enumeration. Previously it would only attempt to use configuration value 1.
- FAT library issue with cluster allocation on large disks fixed.
- USBHost fixed failure to free storage allocated to devices when removed. This caused the enumeration state to stay at "PORT_STATE_CONNECTED" after several connect/disconnect sequences.
- FAT library should not create a new file when a non-existent file is opened for "FILE_MODE_HANDLE".
- USBSlave supports not sending an IN with no data after a get descriptor request (if required).

-
- USBSlaveinitialisation state processing improved.
 - USBSlaveFT232 prevent possible buffer overrun when writing to IN endpoint.
 - USBHost when used with the debugger, starting with a device connected will now detect and enumerate the device.
 - USBHost corrected behaviour with rapid removal/insertion of devices.
 - USBHost improved response time for enumeration to begin after device insertion.
 - USBHost prevented remote possibility of 2 devices being allocated the same bus address when used with a hub.
 - USBHost corrected delays before and after SetAddress to allow devices maximum allowable time as per USB specification.

Restrictions:

- FAT library will not support ATAPI disks.
- FAT library does not support FAT12 format disks. Please use FAT16 or FAT32 format.

Known bugs:

- "SPI Master IOCTL Calls" page in User Guide Help file does not have link for VOS_IOCTL_SPI_MASTER_AUTO_TOGGLE_SS page. This new feature improves performance of the SPI Master by automatically toggling SS lines.

Features requests:

- N/A

2.9 Samples Release Information

2.9.1 VNC1L Firmware Readme

Fixed Issues:

- Added V2F2F firmware for disk copying.
- V2DPS firmware can modify configuration descriptors for USB Slave device.
- Changed "No disk" error message to "No Disk" to match VNC1L response.
- Can change to SCSI mode using Short Command Set code 0x10 from ECS mode.
- Time and date information now available in OPW and MKD commands.
- DIRT command output corrected.
- V2DPS issue with long transfers corrected.
- Add support for 32 pin devices - note that all IO pins are not available on 32 pin devices.
- Handling of removal and insertion of disks improved.
- Initial mode to IPH and ECS.
- Added second bytes (always zero) to output of QP1 and QP2 commands.
- Added space after "MAIN" in output of FWV command.

Restrictions:

- When using with a V2EVAL board Rev 1 always REMOVE jumper JP9.
- V2DAP CDC driver not implemented therefore data-mode use of CDC devices not supported.
- ATAPI flash disks are not supported. SCSI flash disks with a sector size of 512 bytes are supported.
- LEDs do not flash during enumeration.
- No Firmware Update class built into V2DAP or V2DPS firmware. Application Note on modifying firmware to do this is AN 159.
- EEPROM commands for FT232 devices connected USB Host are not supported in V2DAP or V2DPS. USBHostFT232 Driver allows the EEPROM data to be changed.
- V2DPS and V2DAP firmware must be compiled at optimisation level 1 or above due to space restriction in the ROM. This affects some watch variables while debugging. A solution to allow full debugging will be provided at a later date.

-
- V2F2F does not wait for GPIO input before starting. This was an input switch on the VNC1L version to start copying. Once powered up and the USB buses are enumerated then the firmware will start a copy.

Known bugs:

- On V2DPS datamode only supports USB Slave port (USB port 1). Datamode for USB devices on the USBHost port are not supported.
- The OPW command can open a file naming it (OPW<space><cr>). This will appear as a blank filename on Windows and other operating systems. With VNC2 FAT File System Driver, files which are after this blank file in the directory table will not be listed or found.

Features requests:

- N/A

2.9.2 CDCModem Sample Readme

This will connect to a CDC device present on USB Host 1. It will establish a connection between the data interface on the CDC device and the UART on the VNC2.

This project was previously included as sample USBHostCDC.

Restrictions:

- This has only been tested on a Samsung SGH-E900 mobile phone.

Known bugs:

- N/A

Features requests:

- N/A

2.9.3 USBHost FT232 UART Echo Sample Readme

This demonstration application reads data from an FT232 device on USB port 1 and relays it to the UART interface. The FT232 and the UART are set to 9600 baud.

This sample was previously called USBHostFT232Echo.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.4 Firmware Update (AN 159) Readme

First release to be used with Application Note 159. See document named "AN_159 Vinculum-II Firmware Programming".

Fixed Issues:

- N/A

Restrictions:

- Fragmented ROM files not supported. Do "CHKDSK <filename>" from command line to check.

Known bugs:

- N/A

Features requests:

- N/A

2.9.5 KITT Sample Readme

This demonstrates the use of the GPIO driver to flash LEDs in sequence.

Restrictions:

- This sample has been designed for the V2Eval Board Rev 2.0 and later. V2Eval Boards have the version printed on the silk screen next to the V2EVAL label.

The V2Eval boards Rev 1 require a different IOMuxsetting +for this sample to function.

The IOMux routing can be customised by the user to suit a custom PCB or other FTDI supplied evaluation module.

Known bugs:

- N/A

Features requests:

- N/A

2.9.6 Hello World (Library) Sample Application Readme

This demonstrates the use of the FAT file system to write a message to a test file. The file is called TEST.TXT and will be appended with the phrase "Hello World!" every 5 seconds.

This version uses stdio calls to write to the file on the disk.

Restrictions:

- This sample has been designed for the V2Eval Board Rev 2.0 and later. V2Eval Boards have the version printed on the silk screen next to the V2EVAL label.

The V2Eval boards Rev 1 require a different IOMux setting for this sample to function.

The IOMux routing can be customised by the user to suit a custom PCB or other FTDI supplied evaluation module.

Known bugs:

- N/A

Features requests:

- N/A

2.9.7 Philosophers Sample Readme

This demonstrates message passing and resource management within the VOS kernel using the well known dining philosophers problem.

Restrictions:

- This sample has been designed for the V2Eval Board Rev 2.0 and later. V2Eval Boards have the version printed on the silk screen next to the V2EVAL label.

The V2Eval boards Rev 1 require a different IOMUX setting for this sample to function.

The IOMUX routing can be customised by the user to suit a custom PCB or other FTDI supplied evaluation module.

Known bugs:

- N/A

Features requests:

- N/A

2.9.8 PWMBreathe Sample Readme

This demonstrates the use of PWM to vary the brightness of LEDs with a “breathing” effect.

Restrictions:

- This sample has been designed for the V2Eval Board Rev 2.0 and later. V2Eval Boards have the version printed on the silk screen next to the V2EVAL label.

The V2Eval boards Rev 1 require a different IOMUX setting for this sample to function.

The IOMUX routing can be customised by the user to suit a custom PCB or other FTDI supplied evaluation module.

Known bugs:

- N/A

Features requests:

- N/A

2.9.9 Runtime Sample Readme

This demonstrates the use of the stdio library for writing formatted data to the UART.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.10SDCard Sample Readme

This project initialises an SD card in SPI mode and then attempts to create a file called TEST.TXT on the SD card.

The file is then appended with the string "Hello World!".

Fixed Issues:

- N/A

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.11TickTock Sample Readme

This demonstrates the use of the timer driver to flash LEDs in at different intervals corresponding to the specified timer period.

Restrictions:

- This sample has been designed for the V2Eval Board Rev 2.0 and later. V2Eval Boards have the version printed on the silk screen next to the V2EVAL label.

The V2Eval boards Rev 1 require a different IOMUX setting for this sample to function.

The IOMUX routing can be customised by the user to suit a custom PCB or other FTDI supplied evaluation module.

Known bugs:

- N/A

Features requests:

- N/A

2.9.12RTCExample Sample Readme

This demonstrates the use of the NXP PCF2123 Real Time Clock to set/read the time and start a countdown timer.

Restrictions:

- The RTC driver will only work with the RTC device listed above.
- This sample has been written for the Ethernet/MP3/RTC shield designed by FTDI for the Vinco development platform.
- The driver must use GPIO Port B to receive interrupts from the RTC device.

Known bugs:

- N/A

Features requests:

- N/A

2.9.13 USBHost Android Accessory LED Sample Readme

This will attempt to find a USB device that supports the Android accessory mode and enable it.

Once enabled, the application will wait to receive a toggle command from the Android device and will respond with a status byte to indicate the current state of the LED being toggled.

This has been tested with a Samsung Nexus S phone (Android 2.3.4) and a Motorola Xoom tablet (Android 3.1).

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.14 USBHost Still Image Camera Sample Readme

This demonstration application will take a photo on a compatible (PMP compliant) camera and copy the image taken onto a Flash disk.

The camera is connected to USB Host 1 and the Flash disk to USB Host 2.

Restrictions:

- Only tested on Canon Powershot SX 110 IS

Known bugs:

- N/A

Features requests:

- N/A

2.9.15 USBHost Device Descriptors Sample Readme

This will decode and display standard device descriptors for devices inserted into the USBHost ports. The UART interface is used to both display the descriptors and ask the user which port (1 or 2) to query.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.16 USBHost FT4232Hub Sample Readme

This demonstration application connects to ports A and B on an FT4232 device connected to USB Port 1. An FT4232Hub module is typically used for this demo.

The program will receive data from PORT A and relay it to PORT B on FT4232H using a packet size of 16 bytes.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.17 USBHost Generic Device Driver and Application Sample Readme

Example layered driver for a USB device. Provides a read, write and IOCTL interface to USB device.

Connect to the device using the attach IOCTL function then read and write using the vos_dev_read and vos_dev_write functions.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.18 USBHost GPS Logger Sample Readme

This demonstration application connects to a GPS receiver via an FT232 device at 4800 baud and saves the output of the receiver in a file on a flash disk. The GPS receiver is connected to USB Host 1 and the Flash disk to USB Host 2.

Restrictions:

- GPS receiver must use an FT232 device.

Known bugs:

- N/A

Features requests:

- N/A

2.9.19 USBHost HID Class Sample Readme

This demo will read in data from an interrupt endpoint on a HID class device and output it to the UART interface. It is hardcoded to use a Logitech Wingman joypad although this can be changed to any device VID and PID or modified to allow any HID class device.

The device to test is connected to USB Port 1.

Restrictions:

- Reports are set to be 5 bytes. For a USB keyboard or scanner this is typically 8 bytes.

Known bugs:

- N/A

Features requests:

- N/A

2.9.20 USBHost HID Class Sample 2 Readme

This demo will read in data from interrupt endpoints on two HID class devices and output the received data to the UART interface. It will not check the device type connected although this can be changed to check for specific HID class devices. It the program will wait until devices are plugged into both USB ports before starting.

It has been tested with various keyboard, barcode scanners and games controllers.

Retrictions:

- The program will make no attempt to recover after either device is removed.
- Reports are assumed to be 8 bytes.

Known bugs:

- N/A

Features requests:

- N/A

2.9.21 USBHost Printer Sample Readme

This will connect to a printer class device present on USB Host 1 and will print the device ID string on the printer.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.22 USB Host Microphone Sample Readme

This demonstration application reads isochronous data from a USB microphone on USB Port 1 and sends the raw data to the UART port. The UART is set to 9600 baud. A USB Streaming Audio interface is searched for on the USB Host to use as the source of the data.

Restrictions:

- The constant ISO_TRANSFER_SIZE must be set to the size of data sent by the endpoint of the device.
- Only tested on a small selection of USB microphones available on the market.

Known bugs:

- N/A

Features requests:

- N/A

2.9.23 Webcam Demo Sample Readme

Please refer to Application Note AN_158 Webcam Application Using OLED Display. This is available from the FTDI website.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.24 USB Slave FT232 Sample Readme

This demonstration application emulates an FT232 device in the VNC2 and simply echoes data received from the host. It demonstrates how to use the USBSlaveFT232 driver.

The VNC2 is connected to the host (PC) via USB Port 1.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

2.9.25 USB Slave HID Keyboard Sample Readme

This demonstration application emulates a HID keyboard on the VNC2. It will simulate pressing keys 'a' through to 'z' repeatedly. The key presses will be interpreted by the host computer as real key presses and the application with the focus will act as if the keys were actually being pressed.

WARNING: This may cause applications to misbehave or data to be overwritten on the host computer.

The VNC2 must be connected to the host via USB Port 1.

Restrictions:

- N/A

Known bugs:

- N/A

Features requests:

- N/A

Bug fixes:

- Resolved an issue where casting the USB slave handle when attaching the HID keyboard driver resulted in slow performance.
- Cleaned up code.

2.10 Revision Information

Installer: 1.4.2 (8975)

2.10.1 Compiler Tools

VinAsm: 1.4.2 (8966)

VinC: 1.4.2 (8975)

VinCpp: 1.4.1 (8975)

VinL: 1.4.2 (8922)

VinDbg: 1.4.2 (8922)

2.10.2 Development Environment

VinIDE: 1.4.2 (8966)

2.10.3 Firmware

Kernel: 1.4.2 (3004)

ADC_MCP3008: 1.4.2 (2926)

BOMS: 1.4.2 (2917)

FAT: 1.4.2 (2917)

FIFO: 1.4.2 (2917)

GPIO: 1.4.2 (2917)

PWM: 1.4.2 (2917)

SDCard: 1.4.2 (2917)
SPIMaster: 1.4.2 (2917)
SPISlave: 1.4.2 (2917)
StillImage: 1.4.2 (2917)
Timers: 1.4.2 (2917)
UART: 1.4.2 (2917)
USBHost: 1.4.2 (3003)
USBHostCDC: 1.4.2 (3019)
USBHostFT232: 1.4.2 (3003)
USBHostPrinter: 1.4.2 (3019)
USBSlave: 1.4.2 (3003)
USBSlaveFT232: 1.4.2 (3003)
USBHostAndroidAccessory: 1.4.2 (3019)
RTC: 1.4.2 (3003)
ethernet_w5100: 1.4.2 (2927)
Flash: 1.4.2 (3003)

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 87913570
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, OR97123-5803
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 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

Document References

- 1) AN_151 Vinculum II User Guide
- 2) AN_145 Vinculum II Toolchain Installation Guide
- 3) AN_158 Vinculum II Webcam Application Using OLED Display
- 4) AN_155 Vinculum II SD Card Example
- 5) AN_159 Vinculum II Firmware Flash Programming
- 6) AN_172 Vinculum II Using the USB Slave Driver
- 7) AN_171 Vinculum II USB Host Using the CDC Driver
- 8) AN_163 Vinculum II USB Slave Detecting Disconnect
- 9) AN_164 Vinculum II USB Slave Writing a Function Driver
- 10) AN_147 Vinculum II Using Bulk Only Mass Storage (BOMS) Class Devices Example
- 11) AN_148 Vinculum II Using USB Still Image Capture Devices
- 12) AN_144 Vinculum II IO_Mux Configuration Utility User Guide

Acronyms and Abbreviations

Terms	Description

5 Appendix B – List of Tables & Figures

List of Tables

No table of figures entries found.

List of Figures

No table of figures entries found.

6 Appendix C– Revision History

Revision	Changes	Date
1.4.2	Initial Release	2011-08-11