Preliminary - Subject to Change

Future Technology
Devices International Ltd.

VINCULUM
BINDING USB TECHNOLOGIES

Vinculum VNC1L

Firmware - VDAP

FTDI USB Slave Device and USB Flash Disk Interface with fixed Monitor Port

The Vinculum VNC1L-1A is the first of F.T.D.I.'s Vinculum family of Embedded USB host controller integrated circuit devices. Not only is it able to handle the USB Host Interface, and data transfer functions but owing to the inbuilt MCU and embedded Flash memory, Vinculum can encapsulate the USB device classes as well. When interfacing to mass storage devices such as USB Flash drives, Vinculum also transparently handles the FAT File structure communicating via UART, SPI or parallel FIFO interfaces via a simple to implement command set. Vinculum provides a new cost effective solution for providing USB Host capability into products that previously did not have the hardware resources available.

The VNC1L-1A is available in Pb-free (RoHS compliant) compact 48-Lead LQFP package.

Version 1.07

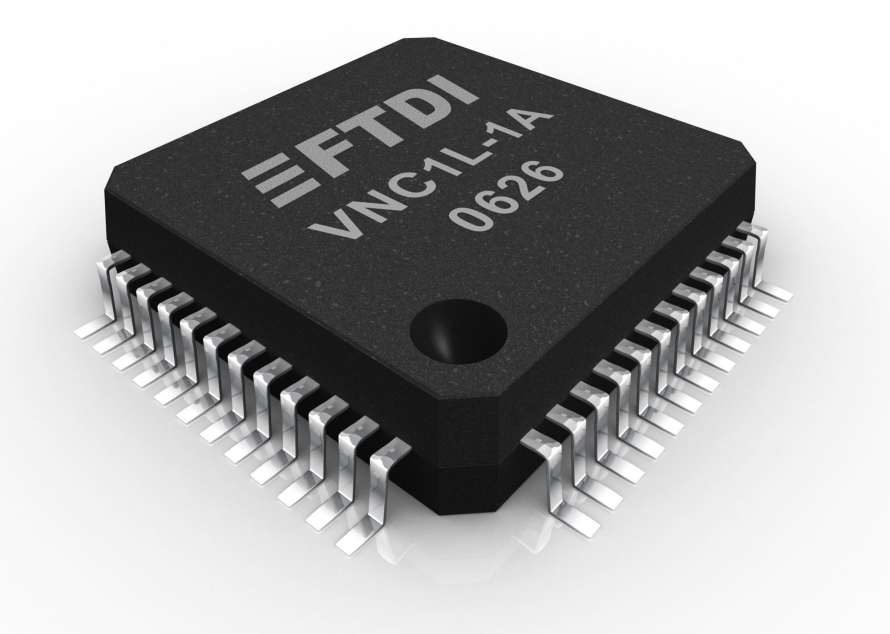
http://www.vinculum.com

1. Overview

1.1 Introduction

This document describes the VDAP version of Vinculum VNC1L USB flash disk / USB device interface and command monitor firmware. VDAP uses the VNC1L's configurable I/O interface fixed as the command monitor port. The I/O interface is configured using a set of jumpers which pull up or pull down two of the VNC1L's pins. These jumpers are used to select between a UART interface, a parallel FIFO interface, or a SPI interface.

The main function of this firmware is to allow an embedded device, based around the VNC1L, to communicate via the VNC1L-1A's UART, parallel FIFO or SPI interface port with USB slave peripheral devices. These include USB Flash disks, FTDI's FT232, FT245 and FT2232 USB slave I.C. as well as devices that are Printer or HID (Human Input Device) compatible. Other devices may also be supported if they have a suitable interface such as mobile phones.



The Vinculum VNC1L I.C. hardware specification is described separately in its own datasheet which is available from the Vinculum website.

2. Firmware Description

2.1 VDAP - Vinculum Disk And Peripheral interface

The VDAP firmware is designed to allow an FTDI Vinculum VNC1L device to act as an interface between a USB flash disk (or other USB mass storage class device) on USB Port 2, and the VNC1L's I/O interface, or a suitable FTDI USB peripheral device on USB Port 1. This firmware allows the VNC1L's I/O interface to be externally configured as a UART, parallel FIFO or SPI, using external jumpers. The device connected to the serial UART / parallel FIFO / SPI interface can issue commands which allow operations to be performed on the USB flash disk and / or USB Slave device using the command set defined herein. The VNC1L port which is configured to receive these commands is know as the **command monitor port**. In this case command monitor port is always the VNC1L's I/O port, i.e. the serial UART / parallel FIFO / SPI interface.

The current firmware supports all FTDI USB slave devices (eg. FT232R, FT245R, FT232B, FT245B and FT2232D) as well as giving support for Printer Class and HID Class. A USB Flash Disk can only be used on port 2.

2.1.1 VDAP Command Monitor Port Selection

The VDAP firmware always uses the VNC1L's I/O port to act as the command monitor port. USB Port 1 is a USB host port for connecting peripheral devices based on FTDI USB slave I.C. devices. USB Port 2 is available as a USB flash disk interface (or other USB mass storage class device). The jumper configurable UART / Parallel FIFO / SPI interface port always acts as the command monitor port.

The VDAP firmware will use the device interface mode as selected on the ACBUS5 (pin 46) and ACBUS6 (pin 47) jumper pins as the command monitor port, i.e. serial UART, Parallel FIFO, or SPI. The jumper circuit configuration shown in Figure 2, below. This circuit will default to the UART interface if no jumper links are fitted. See Table 1 for the port selection jumper pin configuration. USB Port 1 is a USB host port for connecting peripheral devices based on FTDI USB slave I.C. devices. USB Port 2 is available as a USB flash disk interface only (or other USB mass storage class device).

The VNC1L UART / parallel FIFO / SPI to FT232 / FT245 interface works in command mode or data mode in a similar way to a modem. Command mode is used to communicate with the VNC1L. Data mode is used to communicate with the slave device on USB port 1 or 2. Commands are provided to configure both the VNC1L's UART and for an FTDI slave UART device on USB Port 1 or 2.

Figure 2 - VDAP Firmware model

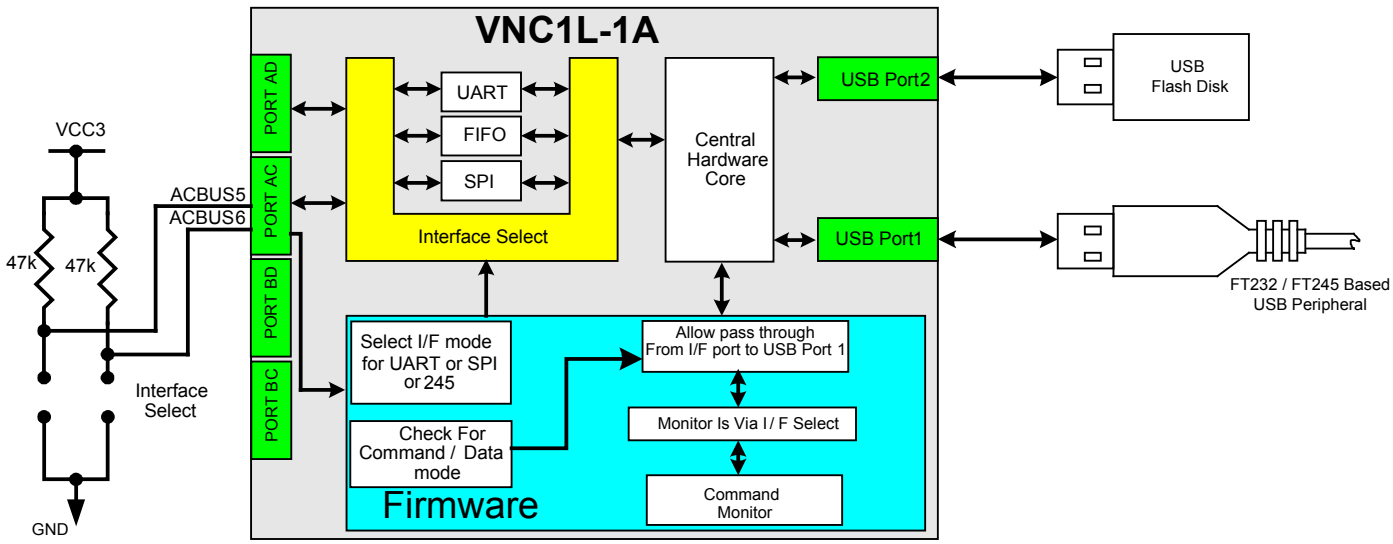


Table 2.1 - Port Selection Jumper Pins

ACBUS6 (pin 47)	ACBUS5 (pin 46)	Mode
Pull-Up	Pull-Up	Serial UART
Pull-Up	Pull-Down	SPI
Pull-Down	Pull-Up	Parallel FIFO
Pull-Down	Pull-Down	Serial UART

Note : Pins ACBUS5 and 6 are only read after reset. They may become output after the interface choice has been selected (e.g. ACBUS5 becomes DATAACK# if FIFO mode is selected). These pins should therefore not be tied directly to GND or VCC but should be pulled using a resistor of around 47 KOhm.

There are three I/O interface options which use ADBUS and ACBUS which are configured using jumpers - serial UART, parallel FIFO and SPI. Two additional signals, DATAACK# and DATAREQ#, allow for switching between command mode and data mode are added. In UART interface mode they are on the DTR# and DSR# interface pins. In parallel FIFO and SPI modes they are on pins 45 and 46. The VNC1L device pin definitions according to I/O mode selected are shown on table 2.1.

Table 2.2 - I/O Interface Options

Pin No.	Name	Type	Description	Interface Mode		
				UART Interface	Parallel FIFO Interface	SPI Slave Interface
31	ADBUS0	I/O	5V safe bidirectional data / control bus, AD bit 0	TXD	D0	SCLK
32	ADBUS1	I/O	5V safe bidirectional data / control bus, AD bit 1	RXD	D1	SDI
33	ADBUS2	I/O	5V safe bidirectional data / control bus, AD bit 2	RTS#	D2	SDO
34	ADBUS3	I/O	5V safe bidirectional data / control bus, AD bit 3	CTS#	D3	CS
35	ADBUS4	I/O	5V safe bidirectional data / control bus, AD bit 4	DTR# / DATAACK#	D4	
36	ADBUS5	I/O	5V safe bidirectional data / control bus, AD bit 5	DSR# / DATAREQ#	D5	
37	ADBUS6	I/O	5V safe bidirectional data / control bus, AD bit 6	DCD#	D6	
38	ADBUS7	I/O	5V safe bidirectional data / control bus, AD bit 7	RI#	D7	
41	ACBUS0	I/O	5V safe bidirectional data / control bus, AC bit 0	TXDEN#	RXF#	
42	ACBUS1	I/O	5V safe bidirectional data / control bus, AC bit 1		TXE#	
43	ACBUS2	I/O	5V safe bidirectional data / control bus, AC bit 2		WR#	
44	ACBUS3	I/O	5V safe bidirectional data / control bus, AC bit 3		RD#	
45	ACBUS4	I/O	5V safe bidirectional data / control bus, AC bit 4		DATAREQ#	DATAREQ#
46	ACBUS5	I/O	5V safe bidirectional data / control bus, AC bit 5		DATAACK#	DATAACK#

2.2 UART Interface Configuration

When using the UART interface as the command monitor port the default baud rate used is 9600 baud, although this can be changed while in command mode. The standard data format is 8 data bits, 1 start bit, 1 stop bit, and no parity with RTS/CTS hardware handshaking enabled. When the firmware is used with the VNC1L I/O configured as a UART the DTR# and DSR# pins also take on the DATAACK# and DATAREQ# functions respectively. See table 2.

2.3 VDAP System Operation - Command Mode

This firmware starts in command mode. In this mode the DATAACK# signal is high. The DATAREQ# line should be held high in order to stay in command mode. While in command mode the firmware will accept commands to modify the UART interface on the VNC1L. If a suitable FTDI USB slave device has been found on USB Port 1 it can also be sent configuration commands. In this mode a USB flask disk on USB Port 2 can also be accessed.

2.4 VDAP System Operation - Data Mode

In order to switch to data mode the DATAREQ# line should be asserted low. Once the DATAACK# line goes low, any data sent to the UART (or parallel FIFO or SPI) port on the VNC1L will be sent to the FTDI slave device connected to USB Port 1. Any data received will be coming from the FTDI slave device connected to USB Port 1. In this mode the VNC1L ignores the data, and simply passes it between USB and the UART / parallel FIFO / SPI interface.

3. Communicating to a USB Slave Device

Connect And Disconnect

When a device is attached to one of the USB ports a message will be sent to monitor port saying :

Device Detected P1

Or

Device Detected P2

When one is removed from a USB port a message will be sent to monitor port saying :

Device Removed P1

Or

Device Removed P2

Query Port 1 and Query Port 2 Commands

To query the device there are new commands available – 'QP1' and 'QP2'. These will tell you all the interfaces types available on port 1 and port 2 respectively.

Table 3.1 - Query Port 1 and Query Port 2 Commands

QP1 and QP2 Commands	
First Byte	
Bit Number	*Meaning*
Bit 0	FTDI 232/245 device attached
Bit 1	Reserved
Bit 2	Printer Class device attached
Bit 3	HID Class device attached
Bit 4	CDC Class device attached
Bit 5	BOMS Class device attached
Bit 6	Unknown Device
Bit 7	Reserved
Second Byte	
Bit Number	*Meaning*
Bit 0-7	Reserved

Query Device Command

Some devices have more than 1 interface (such as the FT2232D chip). To select which device you want to talk to, you can request information on up to 8 interfaces with the Query Device command :

QD n<cr> where n is a number 0 – 7.

This will return a block of 32 bytes to you as shown in Table 3.2 .

This information will give you the device type as well as the USB class codes and USB VID PID and BCD numbers. From this you can decide if you want to talk to this particular device.

Table 3.2 - Query Device Command

QD n Command	
Byte Number	***Meaning***
1	USB Address
2	Control EP 0 size
3	Pipe IN Ep no.
4	Pipe In size
5	Pipe Out EP no.
6	Pipe Out size
7	Data Toggles
8	**Device Type** - see Query Port Command
9	Reserved
10	location
11	MI Index
12	**Device Class**
13	**Device Sub Class**
14	**Device Protocol**
15	**VID Low**
16	**VID High**
17	**PID Low**
18	**PID High**
19	**BCD Low**
20	**BCD High**
21	Device Speed
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved
32	Reserved

Example 1 : Talking to an FT2232 Device

As an example, here is what you get if you are in ASCII input mode (see later) when you query an FT2232 dual chip device on port 1.

qd 0<cr>

$01 $08 $81 $40 $02 $40 $00 $01 $01 $01 $00 $FF $FF $FF $03 $04 $10 $60 $00 $05
$01 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00 D:\>

qd 1<cr>
$01 $08 $83 $40 $04 $40 $00 $01 $01 $01 $01 $FF $FF $FF $03 $04 $10 $60 $00 $05
$01 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00 D:\>

This shows they share the same address but have a different interface number at byte 11 and different endpoint numbers. The first one is port A on a FT2232 chip and the second one is port B. It is important that before you attempt to talk to a device you must set it to be the current device with the Set Current 'SC n' command.

sc 0<cr>
D:\>
will select device 0 or port A of the FT2232 chip.

sc 1<cr>
D:\>
will select device 1 or port B of the FT2232 chip.

Important note :

In this example devices 0 and 1 will target the two ports of the FT2232 chip. If another device was already connected to host port 2 of the VNC1 before the FT2232 was connected to port 1, the FT2232 chip will be mapped to different device numbers.

Once you have selected the device you want to talk to you can then issue commands as if it were the only device connected to the VNC1 chip. The 'SC' command opens up channels to use that devices address and endpoint numbers for transferring data.

In this example

sc 0<cr> (select device 0)
D:\>
fbd $384100<cr> (set baud rate to 9600)
D:\>
fmc $0303<cr> (set RTS and DTR active)
D:\> ffc $01<cr> (set RTS / CTS Flow control)
D:\>
You can now enter direct connection mode by setting DATAREQ# active (low) and waiting for DATAACK# (low). Once you have the acknowledge (DATAACK# low) anything you then type will go to port A of the FT2232 chip and anything that comes back from the VNC1 came from the port A of the FT2232 chip.

Hello world!<cr>

This will then be sent out of the UART on port A of the FT2232 chip at 9600 baud.

Example 2 : Talking to an FTDI Device that does not have the default VID and PID

You may have a device that contains an FTDI chip but you have a different VID and PID programmed into it. You can still use the above method to talk to your device but you need to use another command before doing so. The command is 'SF n' where n is a number 0 – 7. The command means Set as FTDI device n.

Here is an example of an FT232BM chip with a non FTDI VID and PID

```
D:\>
qd 0
$01 $08 $81 $40 $02 $40 $00 $40 $01 $02 $00 $FF $FF $FF $34 $12 $78 $56 $00 $04
 $01 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00
D:\>
```

This has VID $1234 and PID $5678 (which are made up for this example). The device type at location 8 is $40, which, from the table, means unknown. In order to use this device you need to tell the chip that it is an FTDI device. To do this type :

```
sf 0<cr>
D:\>
```

If you then issue the query device command again you will get :

```
qd 0
$01 $08 $81 $40 $02 $40 $00 $01 $01 $02 $00 $FF $FF $FF $34 $12 $78 $56 $00 $04
 $01 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00 $00
D:\>
```

Byte 8 is now $01 which means FTDI. You can then use the set baud rate command etc. and go into data mode as you would with a normal FTDI device. You must remember to issue the ‘Set Current’ command after changing the device type

Example 3 : Transferring Data to or from a USB device without using Data Mode

As well as using the DATAREQ# line to enter data mode, it is possible to send or receive data to or from a USB device by issuing a command while still in command mode. The first step is to set the device number using the SC command as mentioned in the previous section. To send data use the DSD command as follows (this assumes the device is device 1):

```
D:\>
ipa<cr>                 ( ensure ASCII input mode )
D:\>
sc 1<cr>                ( set current device to 1 )
D:\>
dsd 12<cr>              ( Device Send Data 12 bytes)
Hello World !
D:\>
```

This will send the string ‘Hello World!’ to the OUT endpoint of device 1.

To check for data available use the command DRD as follows :

```
D:\>
ipa<cr>                 ( ensure ASCII input mode )
D:\>
sc 1<cr>                ( set current device to 1 )
D:\>
drd<cr>                 ( Do a read of device 1)
$05
HelloD:\>
```

Device 1 sent the string ‘Hello’ through its IN endpoint.

Command Entry Format – ASCII input mode

There are 2 commands to switch the way numbers are sent or received by the chip. These are IPA (InPut ASCII) and IPH (InPut Hex). The default starting condition is HEX. If the command IPA is used then numbers can be entered from an ASCII terminal as ASCII characters.

For example to read twelve bytes from an open file in IPH (HEX default mode) :

‘rdf’ <sp> $00 $00 $00 $0C <cr>

in terms of actual bytes sent to the chip this would be

$72 $64 $66 $20 $00 $00 $00 $0C $0D

To enter the same command in IPA (ASCII mode) :

Decimal

‘rdf’ <sp> 12 <cr>

in terms of actual bytes sent to the chip this would be

$72 $64 $66 $20 $31 $32 $0D

or

Hex $ format

‘rdf’ <sp> $C <cr>

in terms of actual bytes sent to the chip this would be

$72 $64 $66 $20 $24 $43 $0D

or

Hex 0x0 format

‘rdf’ <sp> 0xC <cr>

in terms of actual bytes sent to the chip this would be

$72 $64 $66 $20 $30 $78 $43 $0D

In HEX mode the exact number of bytes specified for the command must be entered. In ASCII mode leading zeros are removed and the number is terminated by the carriage return (<cr> = $0D)

In this way for ASCII mode :

12 == $C == $000000C == 0xC == 0Xc == 0x000C

Firmware Defaults after RESET

The default condition for the firmware after a reset is HEX mode and Extended Command Set.

4. Firmware Command Set

4.1 Monitor Port Commands

This VNC1L firmware command monitor system uses the following command set. There is an extended ASCII command set which is designed for use with a terminal, and there is a shortened hexadecimal command set designed for use with a microprocessor.

Table 4.1 - Monitor Port Commands

Extended ASCII Command for Terminal mode	*Shortened Hexadecimal Command for microprocessor mode*	*Command function*	*Response*
Switching between Shortened and Extended Command sets			
'SCS'<cr>	$10,$0D	Switches to the shortened command set	This will return the prompt '>',$0D to indicate that the device is in shortened command set mode.
'ECS'<cr>	$11,$0D	Switches to the extended command set	This will return the prompt 'D:\>',$0D to indicate that the device is in extended command set mode.
'E'<cr>	'E'<cr>	Echo	This will return 'E',$0D for synchronisation purposes
'e'<cr>	'e'<cr>	Echo	This will return 'e',$0D for synchronisation purposes
'IPA'<cr>	$90,$0D	Input numbers in ASCII	<prompt>$0D
'IPH'<cr>	$91,$0D	Input numbers in HEX	<prompt>$0D
Responses to indicate if disk is online			
<cr>	$0D	Check if online	This will return the appropriate prompt or 'no disk' message for the current command set.
Response to Check if online for Extended Command Mode		If no valid disk is found	'No Disk',$0D
		If a valid disk is found	'D:\>',$0D
Response to Check if online for Short Command Mode		If no valid disk is found	'ND',$0D
		If a valid disk is found	'>',$0D
Directory operations			
'DIR'<cr>	$01,$0D	Lists the current directory	A list of file names and directory names are returned. Each entry is terminated by $0D. A directory entry has <sp>'DIR' after the name and before the $0D.
'DIR'<sp> <name><cr>	$01,$20, <name>,$0D	Lists the file name followed by the size. Use this before doing a file read to know how many bytes to expect.	$0D,<name><sp><size in hex(4 bytes) LSB first> $0D
'DLD'<sp> <name><cr>	$05,$20,<name>, $0D	Delete directory	Deletes the directory <name> from the current directory. <prompt>$0D
'MKD'<sp> <name><cr>	$06,$20, <name>,$0D	Make directory	Creates a new directory <name> in the current directory. <prompt>$0D
'CD'<sp> <name><cr>	$02,$20,<name> $0D	The current directory is changed to the new directory <name>	<prompt>$0D
'CD'<sp>'..'<cr>	$02,$20,$2E,$2E,$0D	Move up one directory level.	<prompt>$0D

Table 4.1 - Monitor Port Commands (continued)

File operations			
'RD'<sp> <name><cr>	$04,$20,<name> $0D	Read file <name>	This will send back the entire file in binary to the monitor. The size should first be found by using the 'DIR' <sp> <name> <cr> command so that the expected number of bytes is known. <prompt>$0D
'RDF'<sp> <size in hex(4 bytes MSB first) ><cr>	$0B,$20,size in hex(4 bytes) , $0D	Reads the data of <size in hex(4 bytes) > from the current open file.	This will send back the requested amount of data to the monitor. <prompt>$0D
'DLF'<sp> <name><cr>	$07,$20,<name> $0D	Delete file <name>	This will delete the file from the current directory and free up the FAT sectors. <prompt>$0D
'WRF'<sp> <size in hex(4 bytes MSB first) ><cr> <data bytes of size><cr>	$08,$20,size in hex(4 bytes) , $0D $data,$0D	Writes the data of <size in hex(4 bytes) > to the end of the current open file.	<prompt>$0D
'OPW'<sp> <name><cr>	$09,$20, <name>,$0D	Opens a file for writing to with 'WRF'	<prompt>$0D
'OPR'<sp> <name><cr>	$0E,$20, <name>,$0D	Opens a file for reading to with 'RDF'	<prompt>$0D
'CLF'<sp> <name><cr>	$0A,$20, <name>,$0D	Closes a file for writing.	<prompt>$0D
'REN'<sp> <orig name> <sp> <new name><cr>	$0C,$20, <orig name>,$20, <new name> <cr>	Rename a file or directory	<prompt>$0D
'FS'<cr>	$12,$0D	Returns free space in bytes on disk. For disks of over 4 GBytes in size this will return $FFFFFFFF if more than 4 GByte available. Otherwise use 'FSE' command	<free space in hex(4 bytes) LSB first> $0D
'FSE'<cr>	$93,$0D	<free space in hex(6 bytes) LSB first> $0D	<free space in hex(6 bytes) LSB first> $0D
'SEK'<sp><offset in hex(4 bytes MSB first) ><cr>	$28,$0D	Seek to an offset within the file	<prompt>$0D
Commands for UART monitor mode only			
'SBD'<sp><divisor (3 bytes) LSB first ><cr>	$14, $20,divisor (3 bytes) LSB first >,$0D	Set Baud Rate (See Baud Rate Table)	<prompt>$0D
Power Management Commands			
'SUD'<cr>	$15,$0D	Suspend the disk when not in use to conserve power. The disk will be woken up automatically the next time a disk command is sent to it.	<prompt>$0D
'WKD'<cr>	$16,$0D	Wake Disk and do not put it into suspend when not in use.	<prompt>$0D
'SUM'<cr>	$17,$0D	Suspend Monitor and stop clocks	<prompt>$0D
Commands to FT232 / FT245 / FT2232 on USB Port 1 or 2			
'FBD'<sp><divisor(3 bytes)LSB first><cr>	$18,$20,<divisor (3 bytes) LSB first>$0D	Set baud rate (See baud rate table 7)	<prompt>$0D
'FMC'<sp><value (2 bytes) ><cr>	$19, $20,<value (2 bytes)>,$0D	Set modem control for RTS/DTR (See table 8)	<prompt>$0D
'FSD'<sp><value (2 bytes) LSB first ><cr>	$1A, $20,value (2 bytes)LSB first >,$0D	Set data characteristics (See table 9)	<prompt>$0D
'FFC'<sp><value (1 byte)><cr>	$1B, $20,value (1 byte),$0D	Set flow control (See table 10)	<prompt>$0D
'FGM'<cr>	$1C,$0D	Get modem status (See table 8)	Returns the modem and line status (2 bytes),$0D
'FSL<sp><value (1 bytes)><cr>	$22, $20,value (1 bytes),$0D	Set Latency Timer	Set the latency timer in milliseconds. The default value is 16 mS <prompt>$0D

Table 4.1 - Monitor Port Commands (continued)

'FSB'<sp><BitMask 1 byte><Enable 1 byte><cr>	$23,$20,$BitMask, $Enable, $0D	Set Bit Mode	Sends the SetBitMode command <prompt>$0D
'FGB'<cr>	$24, $0D	Get Bit Mode	Returns the state of the pins (1 byte),$0D
Commands to Unused I/O pins			
'IOR'<sp><port (1 byte) <cr>	$29,$20,port number AD = $00 AC = $01 BD = $02 BC = $03, $0D	Read I/O port	Reads the I/O port and returns the data
'IOW'<sp><port, direction,value (3 bytes) <cr>	$2A,$20,port number AD = $00 AC = $01 BD = $02 BC = $03, $Direction (1=output), $value $0D	Write I/O port	Writes to the I/O port if it is not being used (i.e. ADBUS 0 -7 will all be used when UART or FIFO interface is active. ACBUS0 will be used in UART mode but ACBUS1-7 are available)
Printer Class Commands			
'PGS' <cr>	$81, $0D	Get Printer Status	Returns the status byte of the printer (1 byte), $0D, (Bit 5 – Paper Empty Bit 4 – Selected Bit 3 – Not Error The rest of the bits are 0) <prompt>$0D
'PSR' <cr>	$82, $0D	Printer Soft Reset	<prompt>$0D
USB Device Commands			
'DSD' <sp> <size in hex(1 bytes)><cr> <data bytes of size><cr>	$83,$20, size (1byte) ,$0D, data of size ,$0D	Send data to USB Device	<prompt>$0D
'DRD' <cr>	$84	Read Data from USB Device	Sends back a byte with the number of bytes n available then <cr> then sends n data bytes then <prompt>$0D
'QP1'<cr>	$2B,$0D	Query Device Port 1 Status	Sends back 2 bytes showing the device types connected to port 1 (see table) <prompt>$0D
'QP2'<cr>	$2C,$0D	Query Device Port 2 Status	Sends back 2 bytes showing the device types connected to port 2 (see table) <prompt>$0D
'QD' <sp> n <cr> (where n is a number in hex of 0 to 7)	$85,$20,n, $0D (where n is a number in hex of 0 to 7)	Query device n	This returns the device data for device n ,$0D<prompt>$0D See table
'SC' <sp> n <cr> (where n is a number in hex of 0 to 7)	$86,$20,n, $0D (where n is a number in hex of 0 to 7)	Set Current Device to n so if the DATAREQ# DATAACK mode is entered, then this device interface will be used for it. Useful if a FT2232C chip with 2 interfaces is connected for example	<prompt>$0D

'SF' <sp> n <cr> (where n is a number in hex of 0 to 7)	$87,$20,n, $0D (where n is a number in hex of 0 to 7)	Set Device to be an FTDI device. This is useful if the VID of a FT232R/FT245R (or BM etc) has been changed from the FTDI default. Use 'QD n' to find your device. 'SF n' to set the device to FTDI and then 'SC n' to set as the current device.	<prompt>$0D
VMUSIC commands - only with VMSC firmware not standard VDAP			
'VPF'<sp><name> <cr>	$1D, $20,<name>,$0D	Play an MP3 file	Sends file to SPI interface (I/O pins between VNC1 and VLSI1003) then returns <prompt>$0D
'VWR'<sp> <Address>(1 byte) <value (2 bytes) LSB first ><cr>	$1E, $20, <Address>(1 byte) <value (2 bytes) LSB first >,$0D	Write to command register of VS1003	<prompt>$0D
'VRD'<sp> <Address>(1 byte) <cr>	$1F, $20, <Address>(1 byte),$0D	Read from command register of VS1003	Returns 2 bytes followed by <prompt>$0D
'VST'<cr>	$20,$0D	Stop playing current track	<prompt>$0D
'V3A'<cr>	$21,$0D	Play all tracks with MP3 as the extention.	Sends all MP3 files in all sub directories to SPI interface then returns <prompt>$0D
'VSF'<cr>	$25,$0D	Skip to next track.	<prompt>$0D
'VSB'<cr>	$26,$0D	Skip to beginning of current track. If pressed twice within 1 second it will go to the beginning of the previous track..	<prompt>$0D
Debug commands			
'SD'<sp> <sector number in ASCII hex><cr>	$03,$20,…$0D	Sector Dump. This is used for debug purposes and may be removed. e.g. 'SD 0000>cr>' will dump sector 0000. 'SD 0010'<cr> will dump sector 16 decimal.	Sends back 512 bytes from the sector specified in HEX converted to ASCII. Every 16 bytes is followed by a $0D. <prompt>$0D
'SW'<sp> <sector number 4 bytes MSB first><cr> <data bytes of size>	$92,$20,…$0D,< 512 bytes of data>	Sector Write. This writes a block of data to the sector specified. Misuse of this command may destroy the disk contents.	<prompt>$0D
'IDD'<cr>	$0F,$0D	Identify Disk Drive. This will display information about the attached disk.	Sends IDD data block and then <prompt>$0D
'IDDE'<cr>	$94,$0D	Identify Disk Drive Extended. This will display information about the attached disk allowing for a disk capacity up to 2 Terra bytes.	Sends IDDE data block and then <prompt>$0D
'FWV'<cr>	$13,$0D	Get Firmware Versions	Displays the version numbers of the main firmware and the reprogramming firmware in the VNC1L-1A 'MAIN x.xx'$0D 'RPRG x.xx'$0D then <prompt>$0D

Table 4.2 - Error Reporting

Error	Command Mode	Result
If command is unrecognised	Extended Command set	'Bad Command',$0D
	Shortened Command Set	'BC',$0D
If command fails	Extended Command set	'Command Failed',$0D
	Shortened Command Set	'CF',$0D

Table 4.3 - IDD Command Results Format

IDD / IDDE - Identify Disk Drive Results	
'USB VID = $', 2 bytes in ASCII, $0D	
'USB PID = $', 2 bytes in ASCII, $0D	
'Vendor Id = ', 8 bytes in ASCII, $0D	
'Product Id = ', 16 bytes in ASCII, $0D	
'Revision Level = ', 4 bytes in ASCII, $0D	
'I/F = ','SCSI' or 'ATAPI' in ASCII, $0D	
'FAT12' or 'FAT16' or 'FAT32' in ASCII, $0D	
'Bytes/Sector = $', 2 bytes in ASCII, $0D	
'Bytes/Cluster = $', 3 bytes in ASCII, $0D	
IDD	'Capacity = $', 4 bytes in ASCII, $0D
	'Free Space = $', 4 bytes in ASCII, $0D
IDDE	'Capacity = $', 6 bytes in ASCII, $0D
	'Free Space = $', 6 bytes in ASCII, $0D

Table 4.4 - Baud Rate Table for VNC1L UART Interface

Baud Rate	1st Byte	2nd Byte	3rd Byte
300	$10	$27	$00
600	$88	$13	$00
1200	$C4	$09	$00
2400	$E2	$04	$00
4800	$71	$02	$00
9600*	$38	$41	$00
19200	$9C	$80	$00
38400	$4E	$C0	$00
57600	$34	$C0	$00
115200	$1A	$00	$00
230400	$0D	$00	$00
460800	$06	$40	$00
921600	$03	$80	$00
1000000	$03	$00	$00
1500000	$02	$00	$00
2000000	$01	$00	$00
3000000	$00	$00	$00

* default baud rate after reset is 9600 baud.

Table 4.5 - Baud Rate Table for FT232B, FT232R or FT2232 Device on USB Port 1 or 2

Baud Rate	1st Byte	2nd Byte	3rd Byte
300	$10	$27	$00
600	$88	$13	$00
1200	$C4	$09	$00
2400	$E2	$04	$00
4800	$71	$02	$00
9600*	$38	$41	$00
19200	$9C	$80	$00
38400	$4E	$C0	$00
57600	$34	$C0	$00
115200	$1A	$00	$00
230400	$0D	$00	$00
460800	$06	$40	$00
921600	$03	$80	$00
1000000	$03	$00	$00
1500000	$02	$00	$00
2000000	$01	$00	$00
3000000	$00	$00	$00

* default baud rate after reset is 9600 baud.

Table 4.6 - Set Modem Control FMC Command Table for FT232B, FT232R or FT2232 Device on USB Port 1 or 2

1st Byte	Operation
Bit 0	DTR# State 0 = off, 1 = on
Bit 1	RTS# State 0 = off, 1 = on
Bits 7 - 2	Reserved ‘0’
2nd Byte	**Operation**
Bit 0	1 = change DTR, 0 = leave DTR alone
Bit 1	1 = change RTS, 0 = leave RTS alone
Bits 7 - 2	Reserved ‘0’

Table 4.7 - Set Data Characteristics FSD Command for FT232B, FT232R or FT2232 Device on USB Port 1 or 2

1st Byte	Operation
Bit 7-0	Number of Data bits - 7 or 8
2nd Byte	**Operation**
Bit 2-0	Parity : 0 - none 1 - odd 2 - even 3 - mark 4 - space
Bit 5-3	Number of Stop bits : 0 - 1 stop bit 1 - 1 stop bit 2 - 2 stop bits
Bits 6	1 = Send break, 0 = Stop break

Bit 7	Reserved ‘0’

Table 4.8 - Set Flow Control FFC Command Table for FT232B, FT232R or FT2232 Device on USB Port 1 or 2

1st Byte	*Operation*
Bit 0	Hardware handshake RTS/CTS
Bit 1	Hardware handshake DTR/DSR
Bits 2	Software handshake XOFF / XOFF
Bits 7 - 3	Reserved ‘0’

Disclaimer

Copyright © Future Technology Devices International Limited, 2006.

Version 1.07 - created December 2006 - Added information for firmware 2.08 and later
Version 1.06 - Initial firmware datasheet created August 2006

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.

Contact FTDI

Head Office -
Future Technology Devices International Ltd.
373 Scotland Street,
Glasgow G5 8QB,
United Kingdom

Tel. : +(44) 141 429 2777
Fax. : +(44) 141 429 2758

E-Mail (Sales) : **sales1@ftdichip.com**
E-Mail (Support) : **support1@ftdichip.com**
E-Mail (General Enquiries) : **admin1@ftdichip.com**

Regional Sales Offices -

Future Technology Devices International Ltd. (Taiwan)
4F, No 16-1,
Sec. 6 Mincyuan East Road,
Neihu District,
Taipei 114,
Taiwan, R.o.C.

Tel.: +886 2 8791 3570
Fax: +886 2 8791 3576

E-Mail (Sales): **tw.sales1@ftdichip.com**
E-Mail (Support): **tw.support@ftdichip.com**
E-Mail (General Enquiries): **tw.admin@ftdichip.com**

Future Technology Devices International Ltd. (USA)
5285 NE Elam Young
Parkway, Suite B800
Hillsboro,
OR 97124-6499
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**

Website URL : **http://www.ftdichip.com**