(Preliminary – Subject to Change)

VNC1 Software For Connecting a USB slave device to UART/245/SPI Port Ver 2.16 (VDAP) and for connecting as a slave to a PC (VDPS)

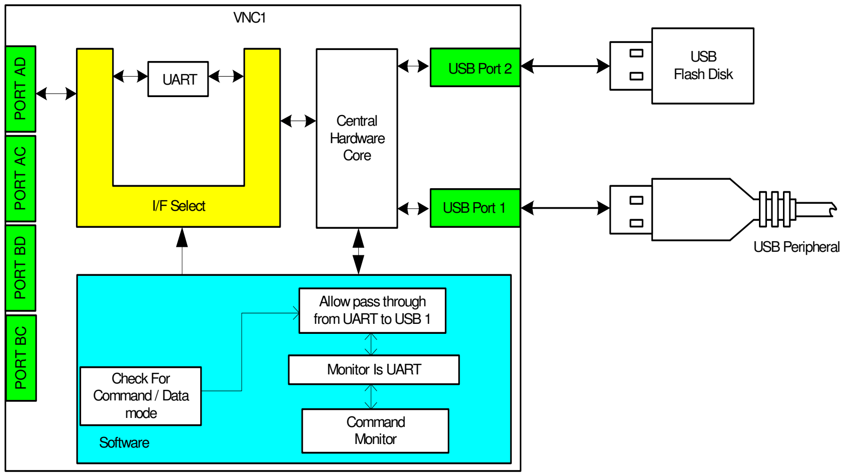
Purpose

This firmware has 3 versions depending on which system configuration you wish to use. One has the UART only as the monitor. The other 2 have a jumper selectable interface that selects between UART, 245 or SPI. These other 2 versions further split down to a version that has 2 host ports and a version that has 1 host and 1 slave for connecting to a PC.

This firmware provides a disk interface in the same manner as the VNC1_Disk_Interface_Software . The difference is that it will always have the UART / 245 / SPI interface as the monitor port. The main function here is that it allows an embedded device to talk via a UART or FT245 style interface or SPI slave interface to a USB slave device. The slave device can have an FT232/245BM/R chip embedded in it. The interface works in command or data mode similar to a modem where in command you are talking to the VNC1 and in data mode you are talking to the slave device on USB. Commands are available to configure the UART on the VNC1 and other commands are available to configure the FT232 device on the USB port. Support has been added for USB printer class (for printers that support direct ASCII printing) and for USB HID class to read data from keyboards / mice / joysticks / barcode readers.

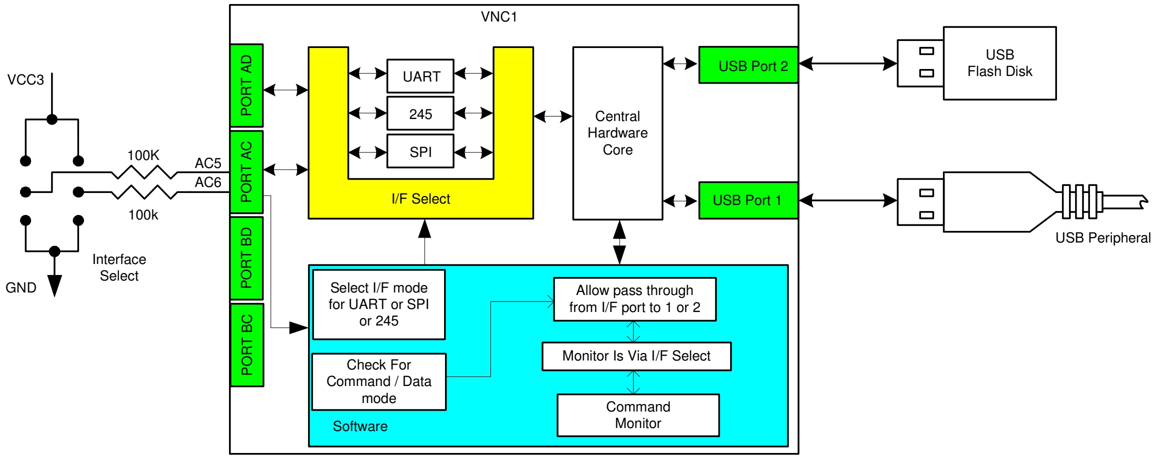
1) Disk Interface with monitor fixed as UART (DIFAU)

Disk Interface Software with UART as the monitor port



2) Disk Interface with monitor fixed as UART or 245 or SPI (VDAP)

Disk Interface Software with UART or 245 or SPI as the monitor port



System Operation For VDAP firmware

Command Mode

This firmware works by starting in command mode. In this mode DATAACK# is high. You should keep the DATAREQ# line high to stay in command mode. The default baud rate on the UART port is 9600, 8 data bits, no parity, 1 stop bit. While in command mode the firmware will accept commands to modify the UART interface in the VNC1. If a suitable FTDI device has been found on USB port1 or USB port2 it will also allow you to send commands to configure the FTDI device. In this mode you can also have access to a USB Flash Disk if it is connected to USB port 2.

Data Mode

To switch to data mode you should assert the DATAREQ# line low and wait for the DATAACK# line to go low. Once the DATAACK# line has been asserted low any data you send will be sent to the FTDI or other USB slave device and any data you receive will come from the FTDI or other USB slave device. In this mode the VNC1 will ignore the data except to pass it between the chosen USB port and the UART/245/SPI interface.

Communicating To a USB slave device

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

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.

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 7 – 0	Reserved

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 command :

QD n<cr>   where n is a number 0 – 7.

This will return a block of 32 bytes to you :

QD n - command 32 byte output	
Byte Number	**Description**
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 Command Query Device

	Port Status
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

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.

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:\>
fsd $0800<cr>        ( set Data characteristics 8 Data no parity 1Stop)
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

```
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*

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.

Interface Selection

There are three choices of interface on the AD/ACBUS pins

VNC1 Interface Modes

Signal Name	UART Mode	Parallel 245 Mode	SPI Slave
ADBUS0	TXD	D0	SCLK
ADBUS1	RXD	D1	SDI
ADBUS2	RTS#	D2	SDO
ADBUS3	CTS#	D3	CS
ADBUS4	DTR#/ DATAACK#	D4	IRQ
ADBUS5	DSR#/ DATAREQ#	D5	
ADBUS6	DCD#	D6	
ADBUS7	RI#	D7	
ACBUS0	TXACTIVE	RXF#	
ACBUS1		TXE#	
ACBUS2		RD#	
ACBUS3		WR#	
ACBUS4		DATAREQ#	DATAREQ#
ACBUS5		DATAACK#	DATAACK#

The selection of which is used is performed at power on pulling 2 lines high or low with resistors.

ACBUS6	ACBUS5	Mode
Pull-Up	Pull-Up	UART
Pull-Up	Pull-Down	SPI
Pull-Down	Pull-Up	245
Pull-Down	Pull-Down	UART

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 $31 $78 $43 $0D

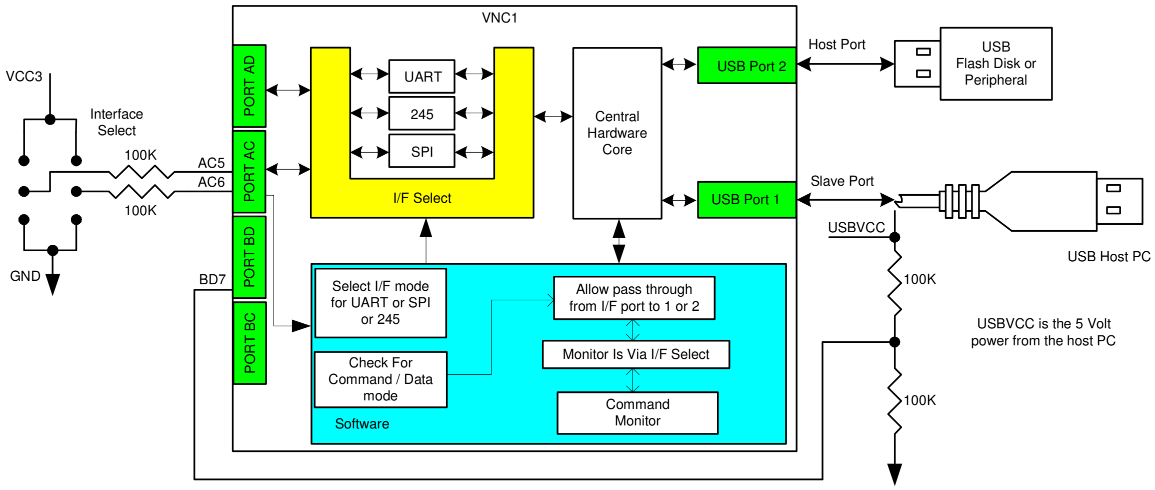
In HEX mode the exact number of bytes specified for the command must be entered. In ASCII mode leading zeros are removed and the number is terminated by the carriage return (<cr> = $0D)

In this way for ASCII mode :

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

VDPS Host Disk or Peripheral or Slave mode

Host Disk and Peripheral Interface with Slave connection Software with UART or 245 or SPI as the monitor port


VNC1
VCC3
Interface Select
100K
AC5
AC6
100K
GND
BD7
PORT AD
PORT AC
PORT BD
PORT BC
UART
245
SPI
I/F Select
Central Hardware Core
USB Port 2
USB Port 1
Host Port
USB Flash Disk or Peripheral
Slave Port
USBVCC
100K
100K
USB Host PC
USBVCC is the 5 Volt power from the host PC
Select I/F mode for UART or SPI or 245
Check For Command / Data mode
Software
Allow pass through from I/F port to 1 or 2
Monitor Is Via I/F Select
Command Monitor

System Operation For VDPS firmware

This is the same as if using host port 2 with the VDAP firmware. You can connect USB disks or peripherals to port 2 and operate in the same manner. The difference here is host port 1 has been set to be in slave only mode. This allows you to have host and peripheral capability on your device. When it is connected to a host pc it will enumerate as if it were an FT232BM device and load the FTDI drivers accordingly. You can select the slave port to be the current device and go into data mode using DATAREQ# and DATAACK#.

When the device is connected to a PC and the drivers have been loaded you will receive a message :

Slave Enabled

When the device is disconnected you will receive a message :

```
Slave Disabled
```

You can check the status of the slave connection at any time with the command QSS (Query Slave Status). This will return 3 bytes of data for example :

```
Slave Enabled

IPA<cr>

D:\>

QSS<cr>

$01 $00 $00

D:\>
```

To access the slave port you have to use the data mode via DATAREQ# and DATAACK# lines. First you should slect the slave device with the SC command. For example:

```
SC S<cr>

D:\>
```

(Now activate DATAREQ# and wait for DATAACK#)

Now when you send anything to the VNC1 device it will pass it on to the host PC and anything the host PC sends back will be passed on to you without the VNC1 interpreting the data. To come out of data mode drop the DATAREQ# line.

Command Set

Extended ASCII Command for Terminal mode	Shortened Hexadecimal Command for microprocessor mode	Command function	Response

Switching between Short and Extended Command sets			
‘SCS’<cr>	$10,$0D	Switches to the short command set	This will return the prompt ‘>’,$0D to indicate it is in Short command mode.
‘ECS’<cr>	$11,$0D	Switches to the extended command set	This will return the prompt ‘D:\>’,$0D to indicate it is in Extended command 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	Input numbers in ASCII	<prompt>$0D
‘IPH’<cr>	$91	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 so that you 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
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 you will know how many bytes to expect. <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	$08,$20,size in hex(4 bytes) , $0D	Writes the data of <size in hex(4	<prompt>$0D

first) ><cr> <data bytes of size><cr>	$data,$0D	bytes) > to the end of the current open file.	
‘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	Returns free space in bytes on disk	<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 when 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 FT232BM/R/FT2232C/D device on USB host 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)	<prompt>$0D
‘FMC’<sp><value (2 bytes) ><cr>	$19, $20,<value (2 bytes)>,$0D	Set Modem Control for RTS/DTR(see Table)	<prompt>$0D
‘FSD’<sp><value (2 bytes) LSB first ><cr>	$1A, $20,value (2 bytes)LSB first >,$0D	Set Data Characteristics (See Table)	<prompt>$0D
‘FFC’<sp><value (1 bytes)><cr>	$1B, $20,value (1 bytes),$0D	Set Flow Control (See Table)	<prompt>$0D
‘FGM’<cr>	$1C, $0D	Get Modem Status (See Table)	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
‘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
EEPROM commands to an FT232BM/R/FT2232C/D device on USB host port 1 or 2			
‘FEE’<cr>	$95, $0D	Erase EEPROM	<prompt>$0D
‘FEW’<sp>Address, dataword (MSB first)(3 bytes)<cr>	$96,$20, Address, dataword (3 bytes) $0D	Write EEPROM	<prompt>$0D
‘FER’<sp>Address, (1 byte)<cr>	$97,$20, Address, (1 bytes) $0D	Read EEPROM	Sends back 2 bytes (MSB first) <prompt>$0D
Commands to Unused I/O pins			
‘IOR’<sp><port (1 byte) <cr>	$29,$20,port number	Read I/O port	Reads the I/O port and returns the data

	⇨ AD = $00 ⇨ AC = $01 ⇨ BD = $02 ⇨ BC = $03, $0D		
‘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			
‘PGI’ <cr>	$80, $0D	Get Printer ID	<prompt>$0D
‘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 for VDAP firmware or ‘S’ for slave with VDPS firmware)	$86,$20,n, $0D (where n is a number in hex of 0 to 7 for VDAP firmware or ‘S’ for slave with VDPS firmware)	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
USB Port 1 Slave Commands for VDPS			
‘QSS’<cr>	$98,$0D	Query Slave Status on port 1	Sends back 3 bytes showing the state of the slave port (see table) <prompt>$0D

‘SC’ <sp>’S’ <cr>	$86,$20,’S’,$0D	Set Current Device to slave.	<prompt>$0D
VMUSIC Commands			
‘VPF’<sp><name> <cr>	$1D, $20,<name>,$0D	Play an MP3 file	Sends file to SPI interface 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			
‘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
‘SD’<sp> <sector number 4 bytes MSB first><cr>	$03,$20,…$0D	Sector Dump. This has been modified to use the standard input format for a number. It can be in binary or the command ‘IPA’ can be used and	Sends back 512 bytes from the sector specified.

		the number input in ASCII as decimal or hex.	
‘IDD’<cr>	$0F,$0D	Identify Disk Drive. This will display information about the attached disk up to 4 GigaBytes.	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 versions of the main firmware and the reprogramming firmware ‘MAIN x.xx’$0D ‘RPRG x.xx’$0D then <prompt>$0D

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

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

Baud Rate Table When Using 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 = 9600

FTDI 232BM/R Device on USB Port1

Baud Rate Table FBD command

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

Set Modem Control FMC command

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’

Set Data Characteristics FSD command

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 1 – 1 2 – 2
Bit 6	1 = Send Break 0 = Stop Break
Bit 7	Reserved ‘0’

Set Flow Control FFC command

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

Query Device Port Status

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 7 – 0	Reserved

Query Device Command

QD n - command 32 byte output	
Byte Number	**Description**
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 Command Query Device Port Status
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

QSS – Query Slave Status

QSS – Query Slave Status – only available with VDPS firmware	
First Byte	
Bit Number	**Meaning**
Bit 0	Slave connected to host PC
Bit 1	Slave Port Suspended
Bit 2	Data avail from slave (RX)
Bit 3	Data waiting for slave to send (TX)
Bit 4	Reserved
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved

Second Byte	
Bit Number	**Meaning**
Bit 7 - 0	Byte count for Slave to PC (TX)

Third Byte	
Bit Number	**Meaning**
Bit 7 - 0	Byte count for Slave from PC (RX)