USB3H-IP(USB3.0 Host function IP) demo manual

Rev 1.4 E / 2 May, 2017

This document describes USB3H-IP (USB3.0 host function IP-Core) evaluation procedure using Intel evaluation board (CycloneIV GX board, CycloneVE board, and ArriaV GX starter board) or Xilinx evaluation board (SP605, ML605, AC701, KC705, VC707 and ZC706 board) and USB3.0 adapter board with evaluation sof-file or bit-file

1. Evaluation Environment

This demo design operates under following environment shown at Figure 1 (for Intel) or Figure 2 (for Xilinx)

1-1 Intel Environment

For Intel USB3.0 Host-IP evaluation, user must arrange following environment.

- Intel evaluation board (Cyclone IV GX board in this example)
- USB3.0 adapter board from DesignGateway [Part# AB08-USB3HSMC]
- USB3.0 A to B cable available in the market.
- sof-file download tool (programmer) and NiosII console.
- USB3.0 mass storage class device such as USB3.0-SSD/HDD or USB3.0 storage.

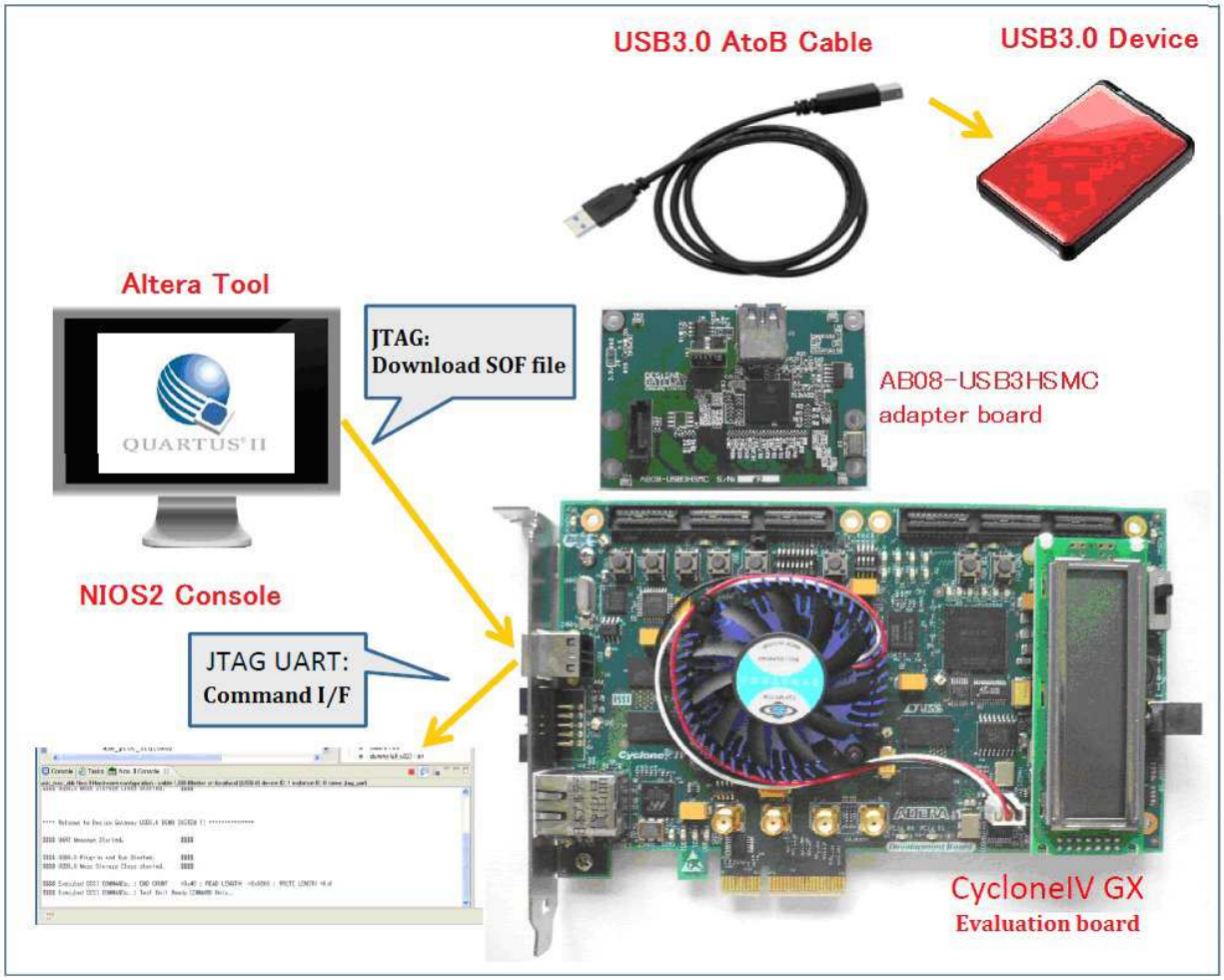


Figure 1: Intel demo environment for USB3H-IP evaluation

(Notes) Evaluation sof-file has 1-hour time limit operation after FPGA configuration.

1-2 Xilinx Environment

For Xilinx USB3.0 Host-IP evaluation, user must arrange following environment.

- Xilinx evaluation board (SP-605 in this example)
- USB3.0 adapter board from DesignGateway [Part# AB07-USB3FMC]
- USB3.0 A to B cable available in the market.
- Xilinx bit-file download tool (iMPACT) and serial console such as Teraterm.
- USB3.0 mass storage class device such as USB3.0-SSD/HDD or USB3.0 storage.

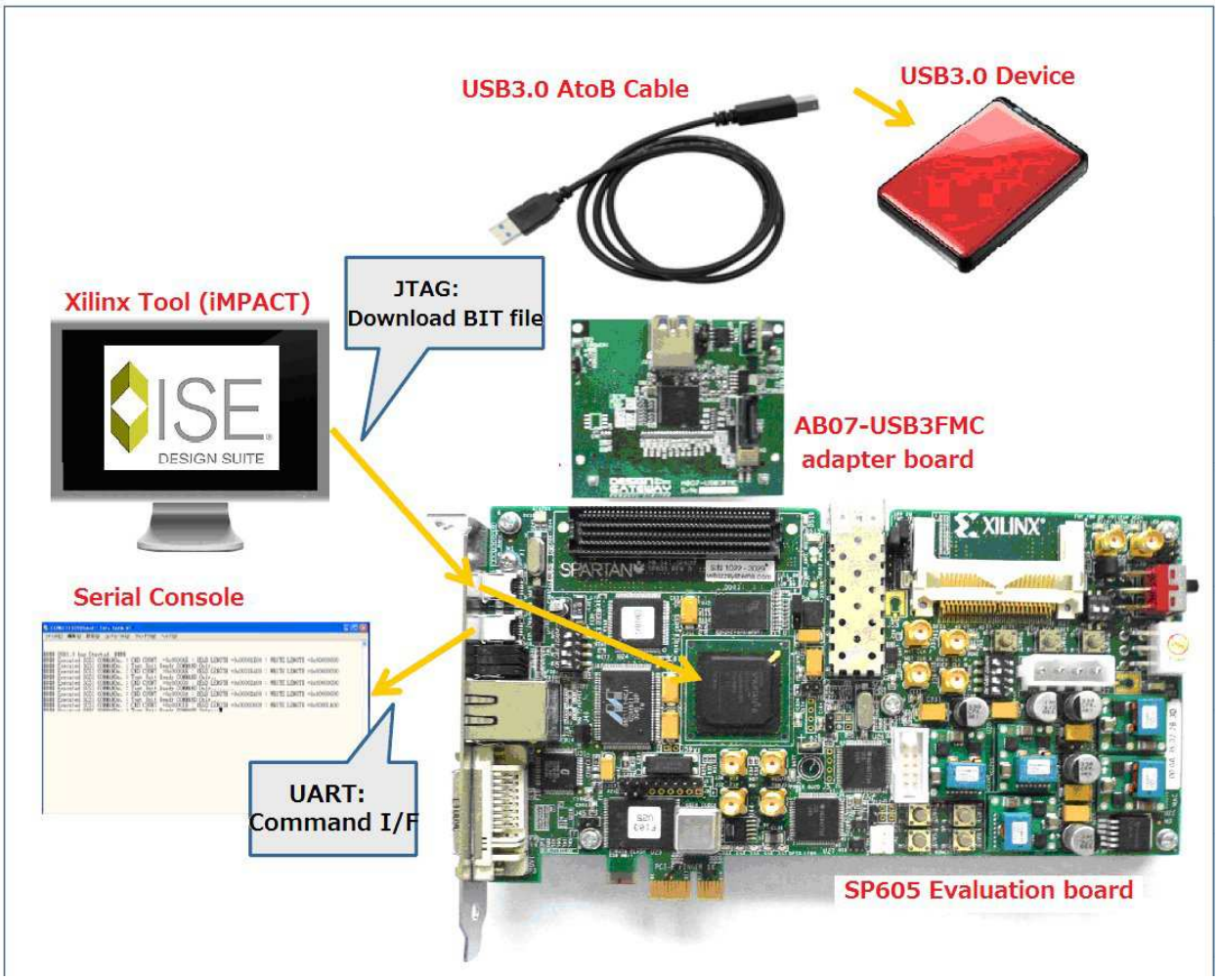


Figure 2: Xilinx demo environment for USB3H-IP evaluation

(Notes) Evaluation bit-file has 1-hour time limit operation after FPGA configuration.

2. General Description

This demo design will introduce basic function of USB3.0 Host-IP and can check USB3.0 connectivity. Whole design project files including C-source code of CPU firmware (MicroBlaze/NiosII/ARM on Zynq) are attached to the IP core product, so user can understand core usage by source code reference. To help easy understanding, software includes minimum function for USB3.0 storage class communication, so this demo design has following limitation.

In real product development, it needs to add necessary feature and/or edit software of the reference design.

1. Device connection
   - Direct connection between USB3.0-IP demo board and mass storage class device.
   - Connect via single Hub is also possible.
   - This demo supports both types of bus-power device or self-power device.
   - For bus-power device, power dissipation should be less than 5Watt.
   - For self-power device, it is possible that demo design initialization sequence cannot match.
2. Software Control
   - Demo firmware does not use OS or interrupt function. So user needs to notify device connection change (plug-in / plug-out) event by command line.
   - Some device only can accept connection control within several 10milli-seconds after plug-in.
   - For very slow device case, demo firmware may detect Watch Dog Timeout. (Very slow device such as USB Flash memory.)
3. RAW Access
   - This menu is for basic function and performance evaluation.
   - Minimum software control reveals pure hardware performance.
4. FAT System
   - This menu is for check compatibility in mass storage class.
   - FatFs (freeware) is used for this system implementation.
   - Due to freeware usage, this FAT system operation and FatFs design is not guaranteed. (Technical support is not provided for FAT System in this demo.)
   - Because CPU processor operation clock is slow, performance is not enough.
   - Other File System such as NTFS or exFAT is not supported.
   - User must select proper file system in real product development.

3. Evaluation Procedure

3-1 Intel Evaluation Procedure

For Intel USB3.0 Host-IP evaluation, follow evaluation procedure below.

- Check that all board and USB device are powered off.
- Connect USB Cable to USB connector on Intel board (J4 for CycloneIV GX board/J10 for CycloneVE board/J14 for ArriaV GX starter board) for JTAG programming and JTAG UART operation.
- Set HSMC interface voltage of CycloneIV GX board to 2.5V. (Short J3 header. Refer to CycloneIV GX board manual for more detail.)
- Connect adapter board (AB08-USB3HSMC) to HSMC connector.
- Confirm that jumper socket is set (ON) at JP1 of the adapter board.
- Connect USB3.0 AtoB cable (A connector side) with USB connector on the adapter board.
- Connect opposite (B connector) side of USB3.0 AtoB cable to the USB3.0 device.

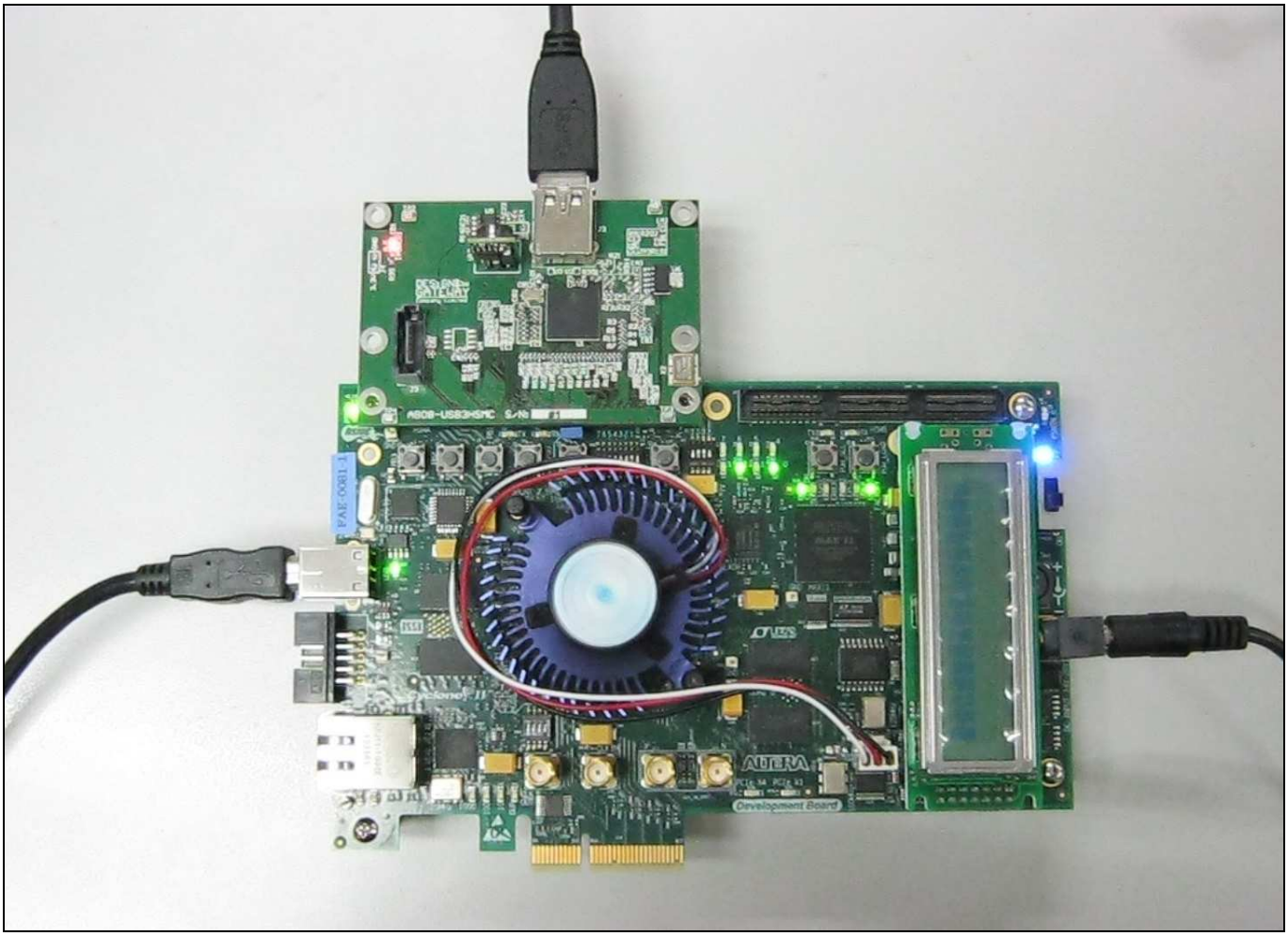


Figure 3: Adapter board connection to CycloneIV GX board

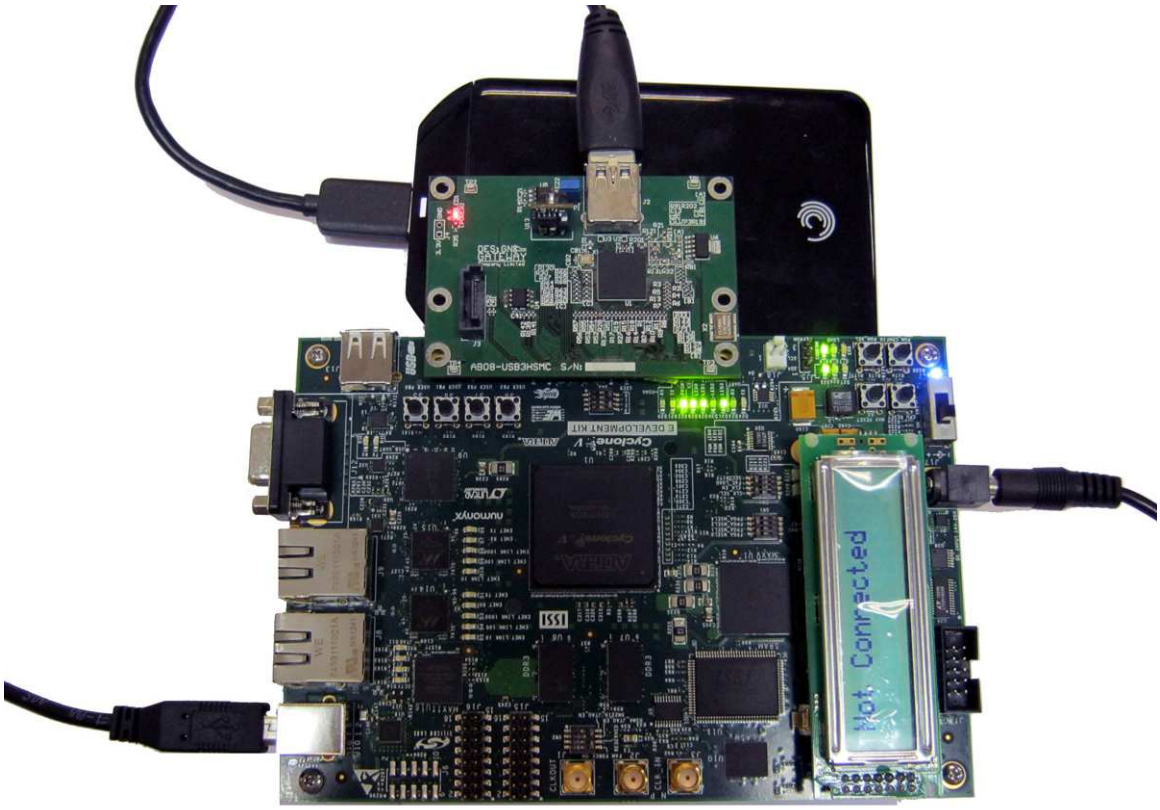


Figure 4: Adapter board connection to CycloneVE board

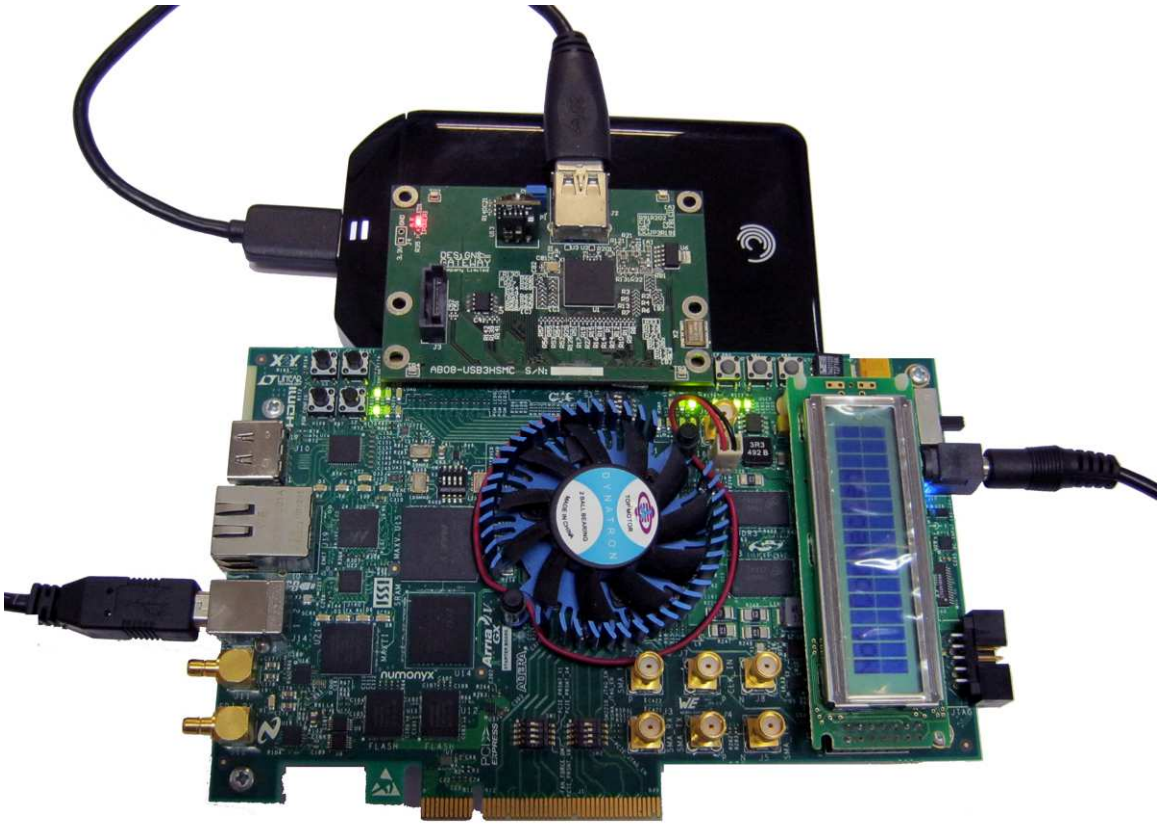


Figure 5: Adapter board connection to ArriaV GX starter board

- Power up all demo device, run Quartus Programmer on the PC, and download evaluation sof-file to the FPGA. After download finish, close Programmer software. (FPGA operation is already running and FPGA is waiting JTAG UART output.)
- Run nios2-terminal from "NIOS2 Command Shell" as below Figure 6.

Figure 6: start nios2-terminal

- When JTAG UART starts its operation, it shows message as Figure 7. If nios2-terminal cannot start or this message is not appeared, check USB cable or download settings of Programmer.

Figure 7: Host demo operation start message

- After sof-file download finish and JTAG UART operation start, check USER_LED0-3 status near by HSMC connector on Intel board. See Table 1 for LED definition. Note that LED0 blinks and other LEDs are OFF when USB cable is unplugged. When USB cable is connected again, LED will change its state as shown in Figure 8.

LED	status	Description
LED0	OFF	FPGA configuration is not completed.
	Blink	USB Cable is not connected.
	ON	USB Cable is properly connected.
LED1	OFF	USB3.0 host operation is not started. FPGA board may have some trouble such as memory initialization fail.
	ON	USB3.0 host operation started successfully.
LED2	OFF	Cannot complete USB3.0 Link initialization. Check HSMC connection between FPGA board and adapter board. Check that connected USB cable is compliant with USB3.0 standard.
	ON	USB3.0 Link initialization is completed successfully.
LED3	OFF	Cannot start USB3.0 host operation. USB cable quality or connected device may have some problem.。
	ON	USB3.0 host operation started and detected USB3.0 device successfully.

Table 1: LED definition of Intel board

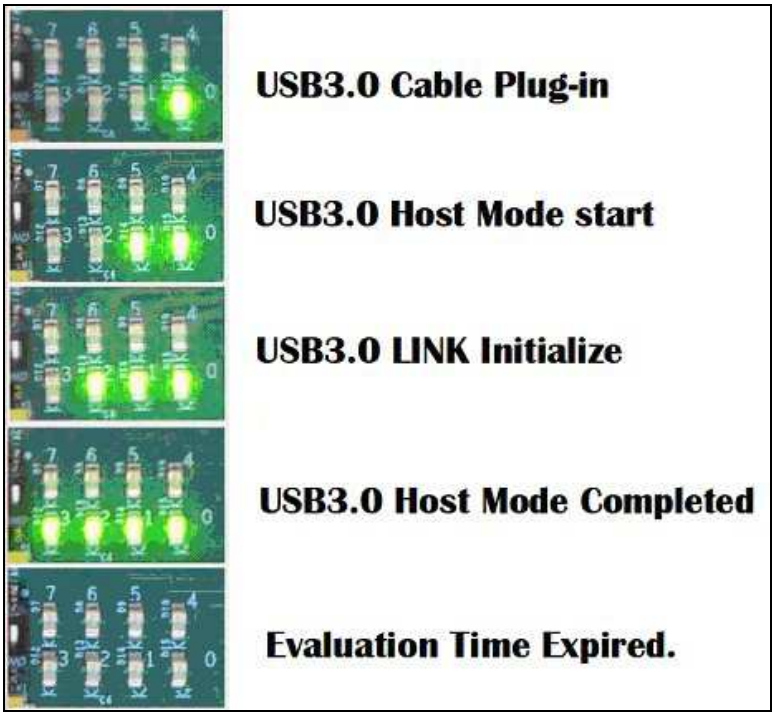


Figure 8: USB3.0 host operation LED status on Intel board

※ All LED will be OFF when 1-hour time limitation is expired.

3-2 Xilinx Evaluation Procedure

For Xilinx USB3.0 Host-IP evaluation, follow evaluation procedure below.

- Check that all board and USB device are powered off.
- a) SP605/ML605: Connect USB mini-cable#1 to USB mini-connector on Xilinx board (J4 for SP605/J22 for ML605) for JTAG programming.
- b) AC701/KC705/VC707/ZC706: Connect USB micro cable to USB micro connector on Xilinx board (U26 for AC701/U29 for KC705/U26 for VC707/U30 for ZC706) for JTAG programming.
- Connect USB mini-cable#2 to USB mini-connector on Xilinx board (J23 for SP605/J21 for ML605/J17 for AC701/J6 for KC705/J17 for VC707/J21 for ZC706) for serial console I/F on HostPC.
- Confirm that FMC interface voltage is 2.5V, and then connect adapter board (AB07-USB3FMC) to the FMC-LPC connector on Xilinx board.
- Confirm that jumper socket is set (ON) at JP1 of the adapter board.
- Connect USB3.0 AtoB cable (A connector side) with USB connector on the adapter board.
- Connect opposite (B connector) side of USB3.0 AtoB cable to the USB3.0 device.

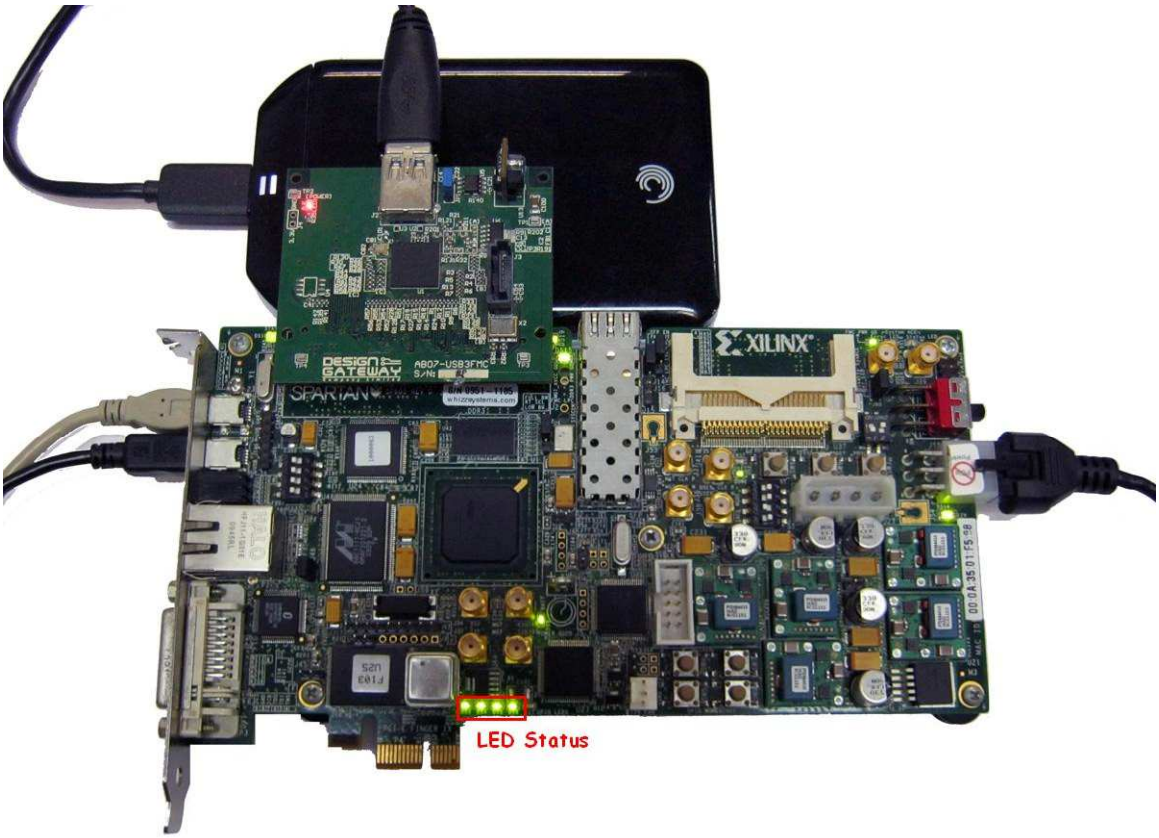


Figure 9: Adapter board connection to SP605 board

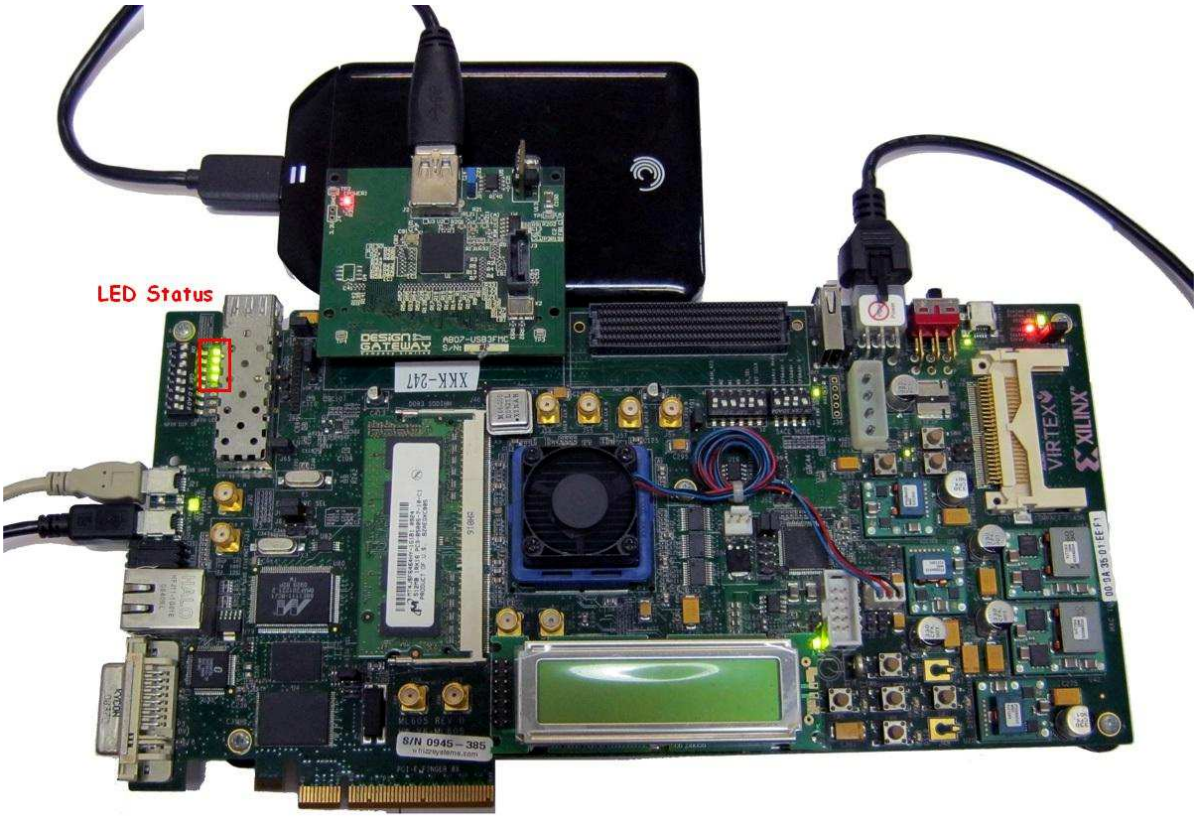


Figure 10: Adapter board connection to ML605

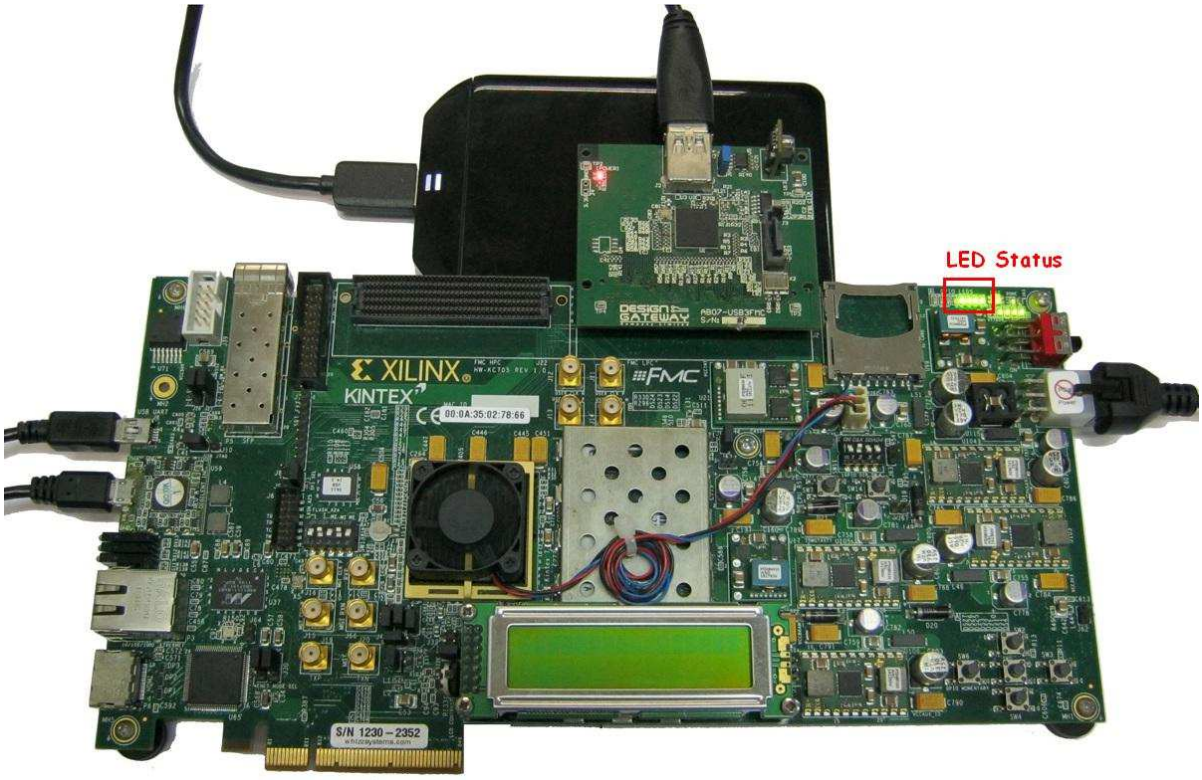


Figure 11: Adapter board connection to KC705

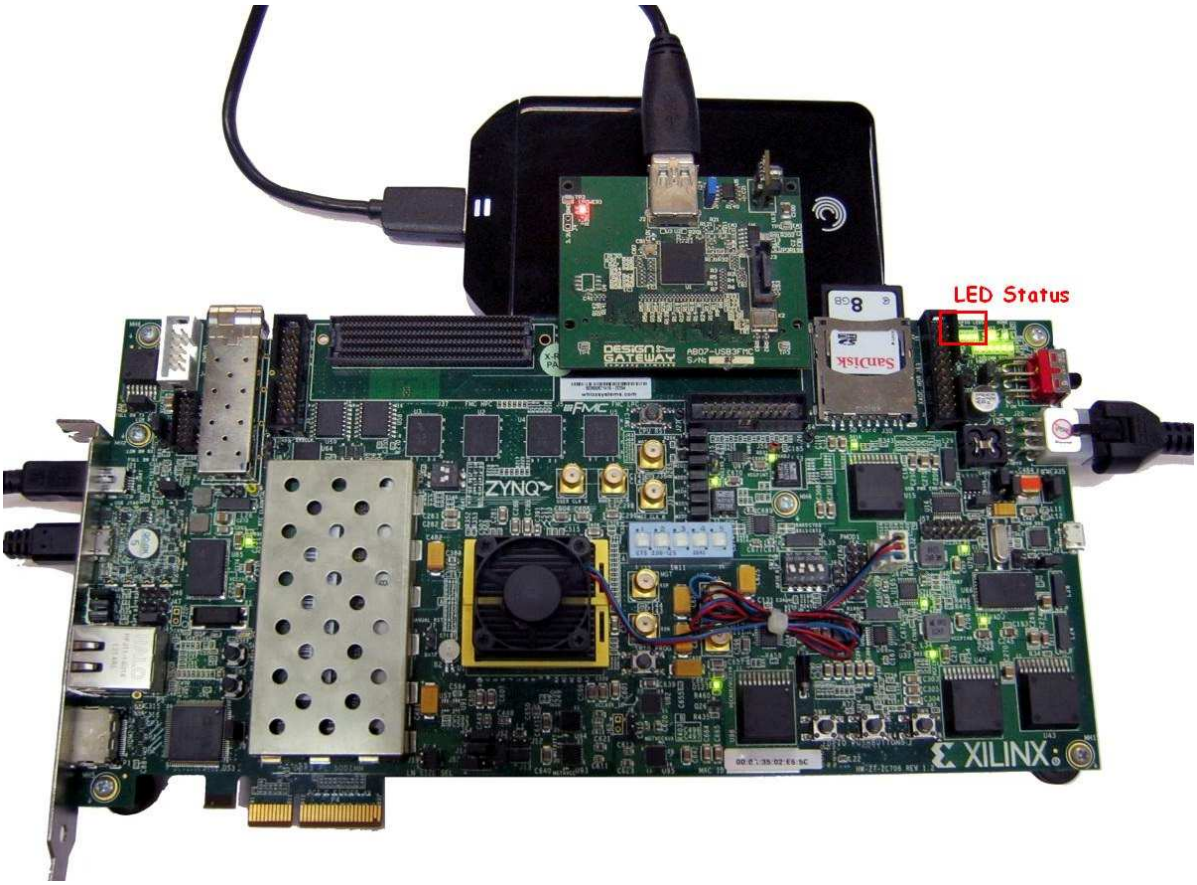


Figure 12: Adapter board connection to ZC706

Figure 13: Adapter board connection to AC701

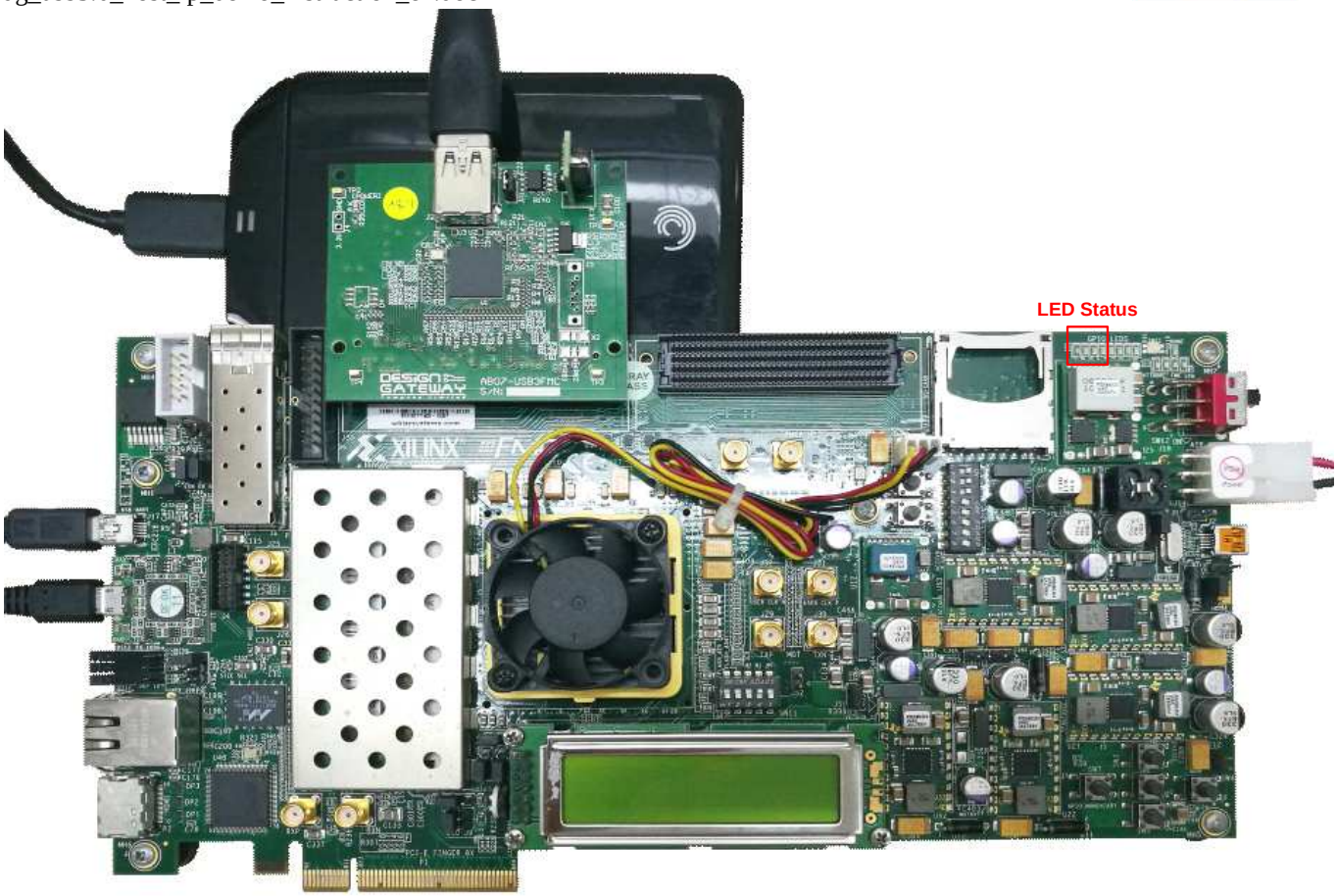


Figure 14 Adapter board connection to VC707

- Power up all demo device, run serial terminal (TeraTerm for example) and set following communication parameter
  Baud Rate = 115,200 Data = 8bit Stop Bit = 1 Parity = None
- a) Start iMPACT and download evaluation bit-file to the FPGA.
  b) For ZC706 board only, set SW11="00000" to configure PS from JTAG, and set SW4="01" to connect JTAG to USB-to-JTAG interface, as shown in Figure 15 and Figure 16. After that, open ISE command prompt and run bat file to download bit-file and elf-file to FPGA, as shown in Figure 17.

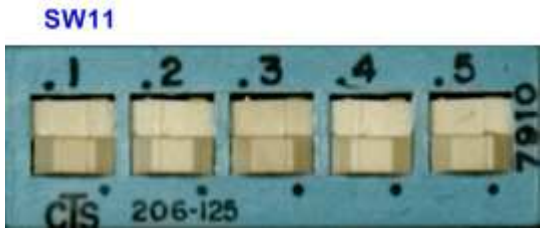


Figure 15: SW11 setting on ZC706 board

Figure 16: SW4 setting on ZC706 board

Figure 17: Run bat file to program bit and elf file on ZC706 board

- After bit-file download finish, check check GPIO_LED0-3 status on Xilinx board. See Table 2 for LED definition. Note that LED0 blinks and other LEDs are OFF when USB cable is unplugged. When USB cable is connected again, LED will change its state as shown in Figure 18.

LED	Light	Description
DS3	OFF	SP-605 FPGA configuration is not completed.
	Blink	USB Cable is not connected.
	ON	USB Cable is properly connected.
DS4	OFF	USB3.0 host operation is not started. FPGA board may have some trouble such as memory initialization fail.
	ON	USB3.0 host operation started successfully.
DS5	OFF	Cannot complete USB3.0 Link initialization. Check HSMC connection between FPGA board and adapter board. Check that connected USB cable is compliant with USB3.0 standard.
	ON	USB3.0 Link initialization is completed successfully.
DS6	OFF	Cannot start USB3.0 host operation. USB cable quality or connected device may have some problem.。
	ON	USB3.0 host operation started and detected USB3.0 device successfully.

Table 2: LED definition of Xilinx board

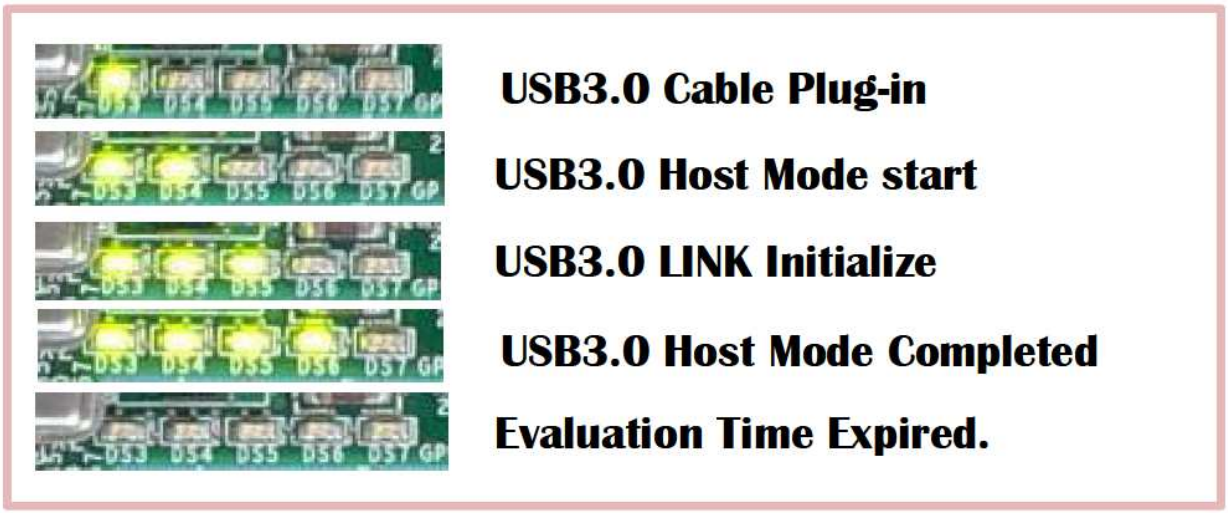


Figure 18: USB3.0 host operation LED status on SP605 board

※ All LED will be OFF when 1-hour time limitation is expired.

- After FPGA operation starts, nios2-terminal (for Intel) or serial console (for Xilinx) shows Figure 19 message (same definition of LED1 for Intel and Xilinx board), if this message does not appear, check JTAG UART or USB-UART connection.

```
==== Welcome to Design Gateway USB3.0 DEMO SYSTEM !! ==============
```

Figure 19: Host operation start message

- And when demo operation starts successfully, nios2-terminal/serial console shows Figure 20 message that lists connection status (this message should appear after LED3 for both Intel and Xilinx board).

```
==== Welcome to Design Gateway USB3.0 DEMO SYSTEM !! ==============

---> Bus is Ready.
---> Port(1) is prepared.
---> Port(2) is prepared.
---> Port(3) is prepared.
---> Port(1) is Ready.
---> Port(2) is Ready.
---> Port(3) is Ready.

....[Type]:[Name                    ].........................
>>>> Hub  : VIA Labs, Inc. 4-Port USB 3.0 Hub
>>>> MSC1 : INTEL SS ASMedia (78 GB)
>>>> MSC2 : WD       My Passport 0740 (488 GB)
>>>> MSC3 : JetFlash Mass Storage Device (7 GB)
>>>> MSC5 : Design Gateway RAM Disk (64 MB)
======<<< USB Bus Status Reported >>>=========( 0 )======

Current Date ---> 2011/1/23 12:34:56
                  Use "DATE" Command to change DATE.
$$$$ Please Type Your Command OR "HELP" to show command lists.

5:>
```

Figure 20: Successful operation message

4. Console Command

This demo design includes following four command category.
1. System and evaluation support command
2. RAW access command
3. FAT access command
4. Component control command

This demo design prepares following two work spaces in DDR memory
1. RAM Disk storage area (Mass storage5) : Capacity=63MB.
2. System work area: Capacity=64MB.

Because above work spaces are mapped on DDR memory, data in these area are disappeared by power cycle or FPGA configuration.
RAM Disk is for command exercise of this demo system, and useful for command operation training, but small capacity as 63Mbytes available space and rather low performance.
System work area is test data space for write and read operation to/from connected USB3 device.

Command string does not distinguish large or small letter. All parameters of address and length are expressed as Hexadecimal ('0x' prefix is not necessary), and sector size is 512bytes. Command string can accept BS (BackSpace) on Serial console, but cannot accept BS on NiosII command shell. Note that this demo does not include perfect command string check.

FAT system use "0:" or "1:" as a drive number. Directory name and file name should be Short Name (i.e. 8 character + "." + 3 character file extension). Directory separator use "\" or "/". This specification is determined by FatFs system.

Refer to the following description for more detailed command usage.

1. System and evaluation support command

1.1. HELP command

1:> HELP

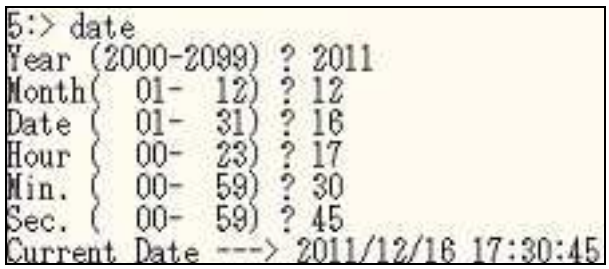
List support command on the console, this list may change by current access mode setting.

```
Supervisor: 1: > help

++<COMMAND>++++++++++++++++++<description>++++++++++++++++++++++++++++++++++++++
  help                      :Display Command List.      --> (1.1)
  config [R]                :Display Device Configuration.(R:Reconnect) --> (4.1)
  date                      :Set Data and Time.         --> (1.2)
  discon [n:]               :Disconnect Device.         --> (4.2)
  drv    [n:]               :Select Current Device.     --> (1.3)
  mode   [S/N]              :Select Mode.(S:Supervisor,N:Normal) --> (1.4)
  gen    [ADDR] [LEN] [Num/1] :Generate Data on Buffer. --> (1.5)
  dump   [ADDR] [LEN]       :Dump     Data on Buffer.   --> (1.6)
.... (FAT system Command) ..................................................
  dir    [DIR]              :Display a list of files in current directory. --> (3.2)
  cd     [DIR]              :Change current directory   --> (3.3)
  md     [DIR]              :Make(create) new directory --> (3.4)
  del    [FN/DIR]           :Delete file.               --> (3.5)
  edit   [FN]               :Edit text file from Key Board. --> (3.6)
  type   [FN]               :Type text file.            --> (3.7)
  cbin   [ADDR] [LEN] [FN]  :Create binary file from Buffer --> (3.8)
  rbin   [ADDR] [LEN] [FN]  :Read   binary file to   Buffer --> (3.9)
  copy   [SRC] [DST]        :Copy   file          from SRC to DST --> (3.10)
  ren    [OLD] [NEW]        :Rename file/directoy from OLD to NEW --> (3.11)
.... (Supervisor Command) ..................................................
  format [n:]               :Format Device.             --> (3.1)
  read   [ADDR] [LEN] [SEC] :Read  Data to Buffer from Device. --> (2.1)
  write  [ADDR] [LEN] [SEC] :Write Data to Device from Buffer. --> (2.2)
---- (Paramerter) ------------------------------------------------------------
         [ADDR/LEN]         :Buffer address or Length by hex.[w/o 0x]
         [SEC]              :Start Sector by hex.[w/o 0x]
         [DIR]              :Absolute/Relative path.[1:/one/two]
         [FN]               :Short name only.       [1:/one/two/filename.bin]
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

1.2. DATE command

1:> DATE

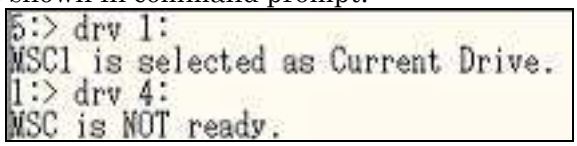
This command is to set Year, Month, Date, Hour, Minute, Second parameter used in the FAT system. This Date parameters are always same value after system start up, so that manual update by this command is necessary if user wants to specify parameter. Note that system will not update these parameters automatically. This command will be failed and date settings are not changed when user input invalid value.



1.3. DRV command

1:> DRV [n:]

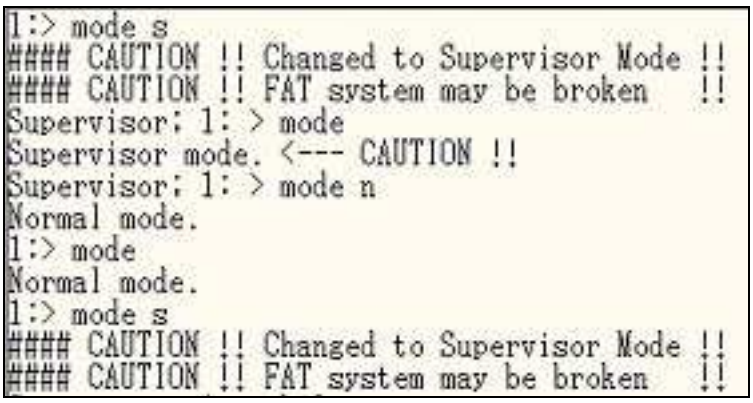
Change current drive number.
This number is also used for target drive of RAW access command.
In FAT system usage, when user do not specify drive number (such as “1”), this number is the target drive.
Current drive number is shown in command prompt.



1.4. MODE command

1:> MODE {S/N}

This command changes access mode. If argument is not specified, it shows current access mode.
'S' argument means 'Supervisor' mode that enables both RAW command and FORMAT command.
'N' argument means 'Normal' mode that cannot execute RAW and FORMAT command.
After start up, default access mode is Normal mode.
When Supervisor mode is specified, "Supervisor" string appears on command prompt.



1.5. GEN command

1:> GEN [ADDR] [LEN] [Num/I]

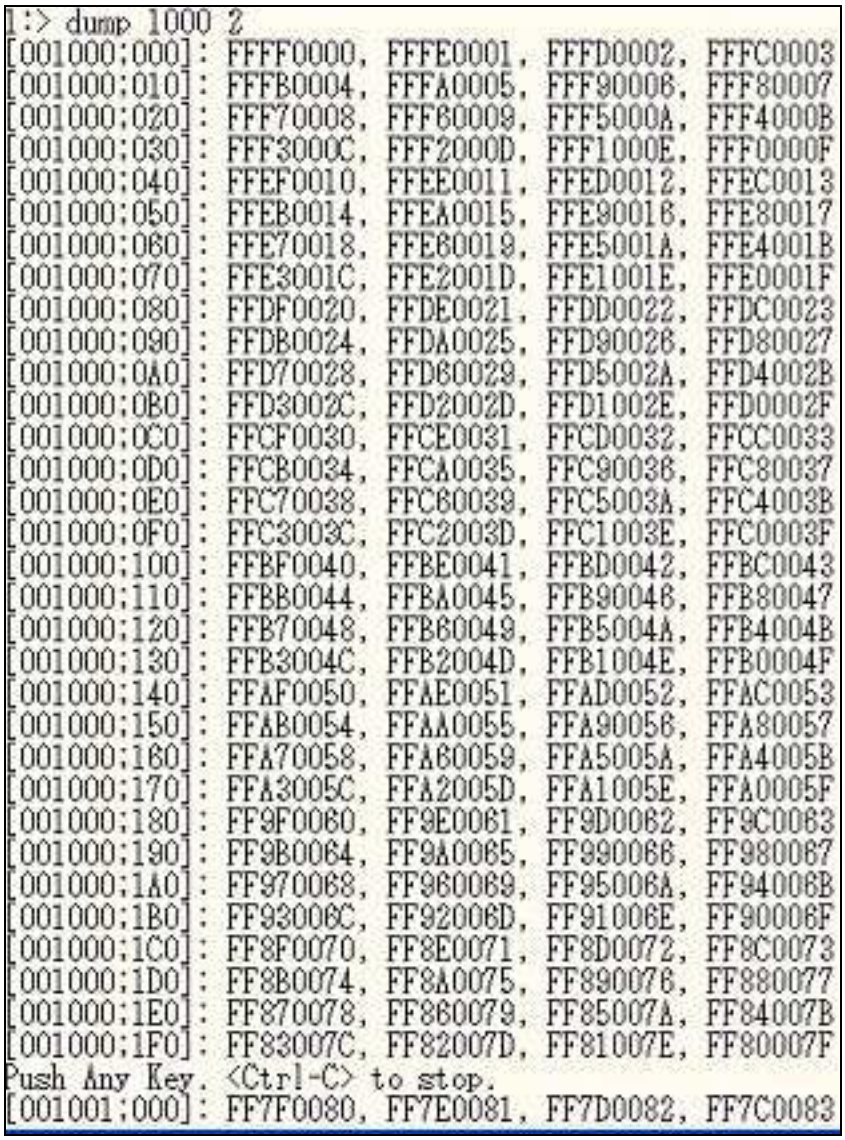
This command generates test data pattern on the specified address of system work space.
User can specify start sector address as [ADDR], fill length (sector unit in Hexadecimal) as [LEN], and test pattern as [Num/I].
If [Num/I] test pattern is specified by Hexadecimal value, this command fills specified area with this value. If [Num/I] is specified 'I' (or 'i'), the command fills incremental + decremental pattern data.
Because system work space is 64MBytes (= 128Ksector = 0x20000), user must specify [ADDR] and [LEN] within this address range of [0x20000].

```
1:> gen 0 1000 0
1:> dump 0
[000000:000]: 00000000, 00000000, 00000000, 00000000
[000000:010]: 00000000, 00000000, 00000000, 00000000
[000000:020]: 00000000, 00000000, 00000000, 00000000
[000000:030]: 00000000, 00000000, 00000000, 00000000
1:> gen 1000 1000 i
1:> dump 1000
[001000:000]: FFFF0000, FFFE0001, FFFD0002, FFFC0003
[001000:010]: FFFB0004, FFFA0005, FFF90006, FFF80007
[001000:020]: FFF70008, FFF60009, FFF5000A, FFF4000B
[001000:030]: FFF3000C, FFF2000D, FFF1000E, FFF0000F
1:> dump 1fff
[001FFF:000]: 007FFF80, 007EFF81, 007DFF82, 007CFF83
[001FFF:010]: 007BFF84, 007AFF85, 0079FF86, 0078FF87
[001FFF:020]: 0077FF88, 0076FF89, 0075FF8A, 0074FF8B
[001FFF:030]: 0073FF8C, 0072FF8D, 0071FF8E, 0070FF8F
1:> dump 2000
[002000:000]: 323B7017, F047F817, B006F037, D014C857
[002000:010]: F3177213, F0133017, F416D016, E017F855
[002000:020]: 30131013, 9007F117, F817F057, D814C016
[002000:030]: F017F017, F027B032, E814D017, F01FFC17
1:> dump 1000 2
[001000:000]: FFFF0000, FFFE0001, FFFD0002, FFFC0003
[001000:010]: FFFB0004, FFFA0005, FFF90006, FFF80007
[001000:020]: FFF70008, FFF60009, FFF5000A, FFF4000B
[001000:030]: FFF3000C, FFF2000D, FFF1000E, FFF0000F
[001000:040]: FFEF0010, FFEE0011, FFED0012, FFEC0013
[001000:050]: FFEB0014, FFEA0015, FFE90016, FFE80017
[001000:060]: FFE70018, FFE60019, FFE5001A, FFE4001B
[001000:070]: FFE3001C, FFE2001D, FFE1001E, FFE0001F
[001000:080]: FFDF0020, FFDE0021, FFDD0022, FFDC0023
[001000:090]: FFDB0024, FFDA0025, FFD90026, FFD80027
[001000:0A0]: FFD70028, FFD60029, FFD5002A, FFD4002B
[001000:0B0]: FFD3002C, FFD2002D, FFD1002E, FFD0002F
```

1.6. DUMP command

1:> DUMP [ADDR] {LEN}

This command displays data in the system work area.
User can specify start address as [ADDR] and length as [LEN] by sector unit. If [LEN] is not specified, this command shows first 16 words data.
Because system work space is 64MBytes (= 128Ksector = 0x20000), so that user must specify [ADDR] and [LEN] within this address range of [0x20000].



2. RAW access command

2.1. READ command

1:> READ [ADDR] [LEN] [SEC]

This command reads from the [SEC] sector address of the current drive with [LEN] sector length, and store to the [ADDR] sector address of the system work area. After command execution, it displays performance result on the console.

When read data excesses end of system work space range (0x20000), command will store read data with overwrite operation from [ADDR] to the possible store range that is power of two within [ADDR]+[LEN] of the system work space. So in this case, last read data will remain in the system work space. For basic operation, set small read size if user wants to check or verify read data in detail, while set large read size if user wants to check read performance.

If final sector address in current drive does not exists (i.e. exceeds final sector), then command fails with error result.

If read length is more than 100Mbytes, then "." mark is shown in the console to display read progress.

User can execute this command only under Supervisor mode.

```
Supervisor; 1: > read 100 100000 200
.....
Total= 524[MB] , Time= 1990[mS] , Transfer speed = 269[MB/s]
Supervisor; 1: > read 100 800000 200
........................................
Total= 4[GB] , Time= 15[S] , Transfer speed = 269[MB/s]
Supervisor; 1: > read 100 1000000 400
................................................................................
Total= 8[GB] , Time= 31[S] , Transfer speed = 270[MB/s]
Supervisor; 1: > write 100 100 500
Do you want to write[Y/N] ? y
Total= 128[KB] , Time= 648[uS] , Transfer speed = 202[MB/s]
Supervisor; 1: > write 100 100000 600
Do you want to write[Y/N] ? y
.....
Total= 524[MB] , Time= 6685[mS] , Transfer speed = 80[MB/s]
Supervisor; 1: > write 100 800000 700
Do you want to write[Y/N] ? y
........................................
Total= 4[GB] , Time= 53[S] , Transfer speed = 79[MB/s]
```

2.2. WRITE command

1:> WRITE [ADDR] [LEN] [SEC]

This command writes data from [ADDR] sector address of the system work area to the [SEC] sector address of current drive with [LEN] sector length. After command execution, it displays performance result on the console.

When [ADDR]+[LEN] address excesses system work space range (0x20000), command will send data from [ADDR] to the possible store range that is power of two within [ADDR]+[LEN] of the system work space, and write data with repetitive operation. For basic operation, set small write size if user wants to simply check write operation itself, while set large write size if user wants to check write performance.

If final sector address in current drive does not exists (i.e. exceeds final sector), then command fails with error result. If write length is more than 100Mbytes, then "." mark is shown in the console to display write progress.

Take care that this command will destroy existing file system on the current drive.

User can execute this command only under Supervisor mode.

3. FAT access command

3.1. FORMAT command

1:> FORMAT [n:]

This command executes format to the specified drive. FAT32 supports up to 2T Bytes capacity.
Format command may take more than several minutes.
All files and format information are removed by this command.
User can execute this command only under Supervisor mode.

```
Supervisor; 1: > format 1:
Do you want to format[Y/N] ? y
Supervisor; 1: > mode s
#### CAUTION !! Changed to Supervisor Mode !!
#### CAUTION !! FAT system may be broken   !!
Supervisor; 1: > mode n
Normal mode.
```

3.2. DIR command

1:> DIR {DIR}

This command lists current or specified directory.

```
1:> md dir1
1:> md dir2
1:> md dir1/dir2
1:> md dir1/dir2/dir3
1:> cd dir1/dir2/dir3
1:> dir
<dir>  .
<dir>  ..
1:> cd ../../..
1:> del dir1/dir2/dir3
1:> dir dir1/dir2
<dir>  .
<dir>  ..
1:> dir
<dir>  DIR1
<dir>  DIR2
1:> dir dir1
<dir>  .
<dir>  ..
<dir>  DIR2
```

3.3. CD command

1:> CD [DIR]

Change current directory.

3.4. MD command

1:> MD [DIR]

Make directory

3.5. DEL command

1:> DEL [FN/DIR]

Delete file or directory. Specified directory must be blank when user delete directory.

3.6. EDIT command

1:> EDIT [FN]

Generate file by text style. After command type, command will change to edit mode and input type text into the file until user specify edit finish by <Ctrl + C> key.

```
1:> edit sample.txt
Please Edit as you like. Press <Ctrl-C> to Exit.
This is Design Gateway USB3.0 host demo system.
In Edit command, you can type any text as you like.
This text can look by Type Command.

1:> type sample.txt
This is Design Gateway USB3.0 host demo system.
In Edit command, you can type any text as you like.
This text can look by Type Command.
```

3.7. TYPE command

1:> TYPE [FN]

This command shows text data on the console.
Because there is no way to stop or pause display operation in this command, user must avoid long text display by this command. The demo system does not guarantee any result if this command is used for non-text file.

3.8. CBIN command

1:> CBIN [ADDR] [LEN] [FN]

This command writes data from [ADDR] sector address of the system work space with [LEN] sector length to the [FN] file of the current drive. After command execution, it displays performance result on the console.

When [ADDR]+[LEN] address excesses system work space range (0x20000), command will send data from [ADDR] to the possible store range that is power of two within [ADDR]+[LEN] of the system work space, and write data with repetitive operation. For basic operation, set small write size if user wants to simply check write operation itself, while set large write size if user wants to check write performance.

If final sector address in current drive does not exists (i.e. exceeds final sector), then command fails with error result. Take care that maximum file size in FAT system is 4GByte (= 8Mega sectors = 0x800000-1). If write length is more than 100Mbytes, then "." mark is shown in the console to display write progress.

```
1:> cbin 1000 7fffff len4g.bin
........................................
Total= 4[GB] , Time= 62[S] , Transfer speed = 69[MB/s]
1:> cbin 2000 7ffff  len256m.bin

Total= 262[MB] , Time= 3815[mS] , Transfer speed = 70[MB/s]
1:> cbin 3000 800000 error.bin
LEN is too large for FAT32.
1:> rbin 4000 7fffff len4g.bin
........................................
Total= 4[GB] , Time= 26[S] , Transfer speed = 160[MB/s]
1:> copy len256m.bin cpy256m.bin

1:> ren cpy256m.bin ren256m.bin
1:> dir
<dir>  DIR1
<dir>  DIR2
<file> SAMPLE.TXT:       139 [ B] ; 0x000000 [Sector] + 0x08B [Byte  ]
<file> LEN4G.BIN:          4 [GB] ; 0x7FFFFF [Sector]
<file> LEN256M.BIN:      262 [MB] ; 0x07FFFF [Sector]
<file> REN256M.BIN:      262 [MB] ; 0x07FFFF [Sector]
```

3.9. RBIN command

1:> RBIN [ADDR] [LEN] [FN]

This command reads from the [FN] file of the current drive with [LEN] sector length, and store to the [ADDR] sector address of the system work area. After command execution, it displays performance result on the console.
When read data excesses end of system work space range (0x20000), command will store read data with overwrite operation from [ADDR] to the possible store range that is power of two within [ADDR]+[LEN] of the system work space. So in this case, last read data will remain in the system work space. For basic operation, set small read size if user wants to check or verify read data in detail, while set large read size if user wants to check read performance.
If final sector address in current drive does not exists (i.e. exceeds final sector), then command fails with error result.
If read length is more than 100Mbytes, then "." mark is shown in the console to display read progress.

3.10. COPY command

1:> COPY [SRC] [DST]

This command executes copy operation from the [SRC] file to the [DST] file.
If available drive capacity is not enough, this command fails with error result.
If copy length is more than 100Mbytes, then "." mark is shown in the console to display copy progress.
Note that this command operation is very slow due to the software process time.

3.11. REN command

1:> REN [OLD] [NEW]

This command change [OLD] file or directory name to [NEW] name.

4. Component Control command
   To change device component such as plug-in or plug-out, following procedure is required.

   - Plug-in
     - Execute CONFIG command and recognize component information.
     - This recognition sometimes fails by device or timing reason, in such case, retry from plug-out.
     - If plug-out retry even cannot recognized, then execute CONFIG R command.
     - When recognition component is completed, then user can execute access command.
   - Plug-out
     - In normal condition, user can execute plug-out without any special process when device is idle (when console shows prompt).
     - Some device requires DISCON command before plug-out.
     - Execute CONFIG command and recognize new component information of disconnect.
     - When plug-out device is default device, then user must change default device.
     - User can execute plug-in again, in this case, CONFIG command of disconnection is not necessary.
     - In such plug-in case, user must follow plug-in procedure described above.

4.1. CONFIG command

1: > CONFIG {R}

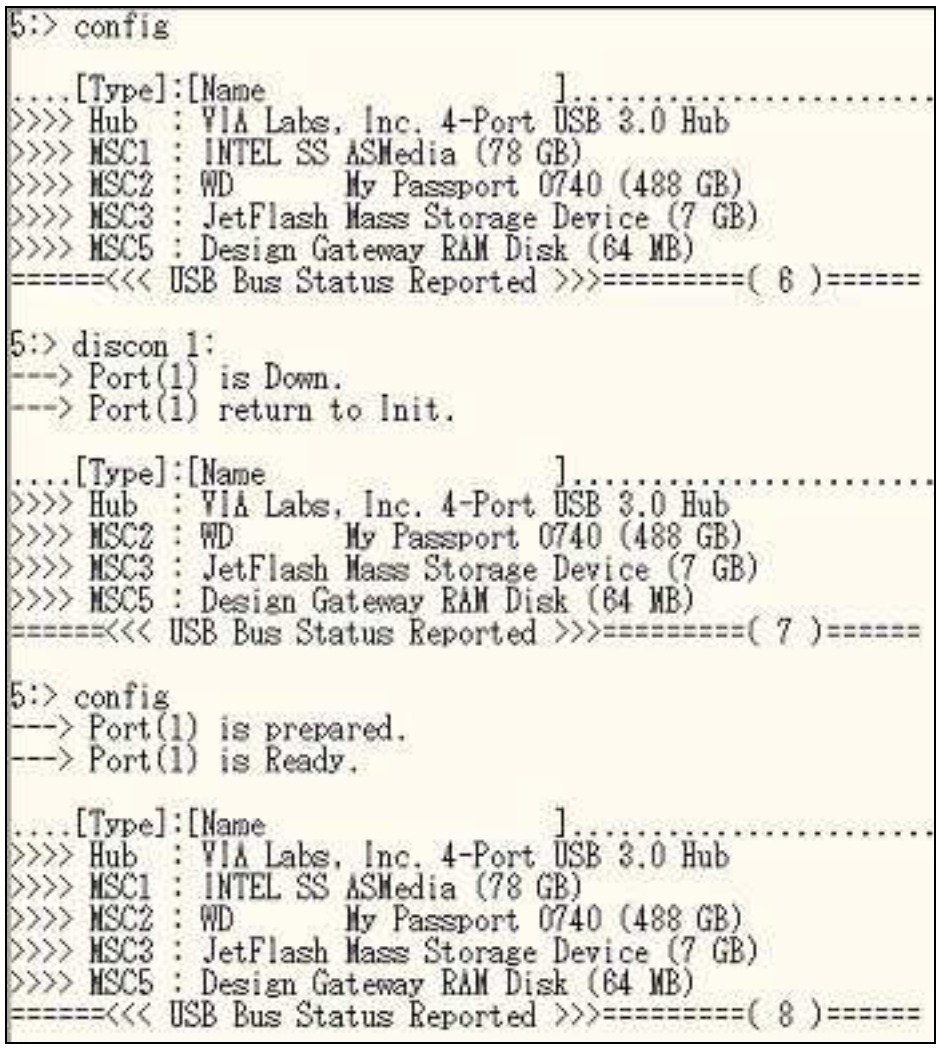
This command scans device component (connection) status and display.
Because this design does not use interrupt feature, user must manually execute this command to recognize device component change by plug-in or plug-out.
If device recognition is not successful, the user must retry plug-in/plug-out or type
1:> CONFIG R
… to execute reconnection.

```
5:> config

....[Type]:[Name                    ].........................
>>>> Hub  : VIA Labs, Inc. 4-Port USB 3.0 Hub
>>>> MSC1 : INTEL SS ASMedia (78 GB)
>>>> MSC2 : WD       My Passport 0740 (488 GB)
>>>> MSC3 : JetFlash Mass Storage Device (7 GB)
>>>> MSC5 : Design Gateway RAM Disk (64 MB)
======<<< USB Bus Status Reported >>>=========( 1 )======

5:> config
---> Port(1) return to Init.
---> Port(1) is prepared.
---> Port(1) is Ready.

....[Type]:[Name                    ].........................
>>>> Hub  : VIA Labs, Inc. 4-Port USB 3.0 Hub
>>>> MSC1 : INTEL SS ASMedia (78 GB)
>>>> MSC2 : WD       My Passport 0740 (488 GB)
>>>> MSC3 : JetFlash Mass Storage Device (7 GB)
>>>> MSC5 : Design Gateway RAM Disk (64 MB)
======<<< USB Bus Status Reported >>>=========( 2 )======
```

4.2. DISCON command

1:> DISCON [n:]

This command execute disconnect process before mass-storage device plug-out.
In normal operation, this command is not necessary.



5. Revision History

Revision	Date	Description
1.0	2012/01/06	1st Japanese version release
1.1E	04- Mar-2015	1st English version release for both Altera and Xilinx
1.2E	18-May-2015	Add board support
1.3E	11-Mar-2016	Support AC701
1.4E	02-May-2017	Support VC707