FAT32-IP for NVMe Demo Instruction

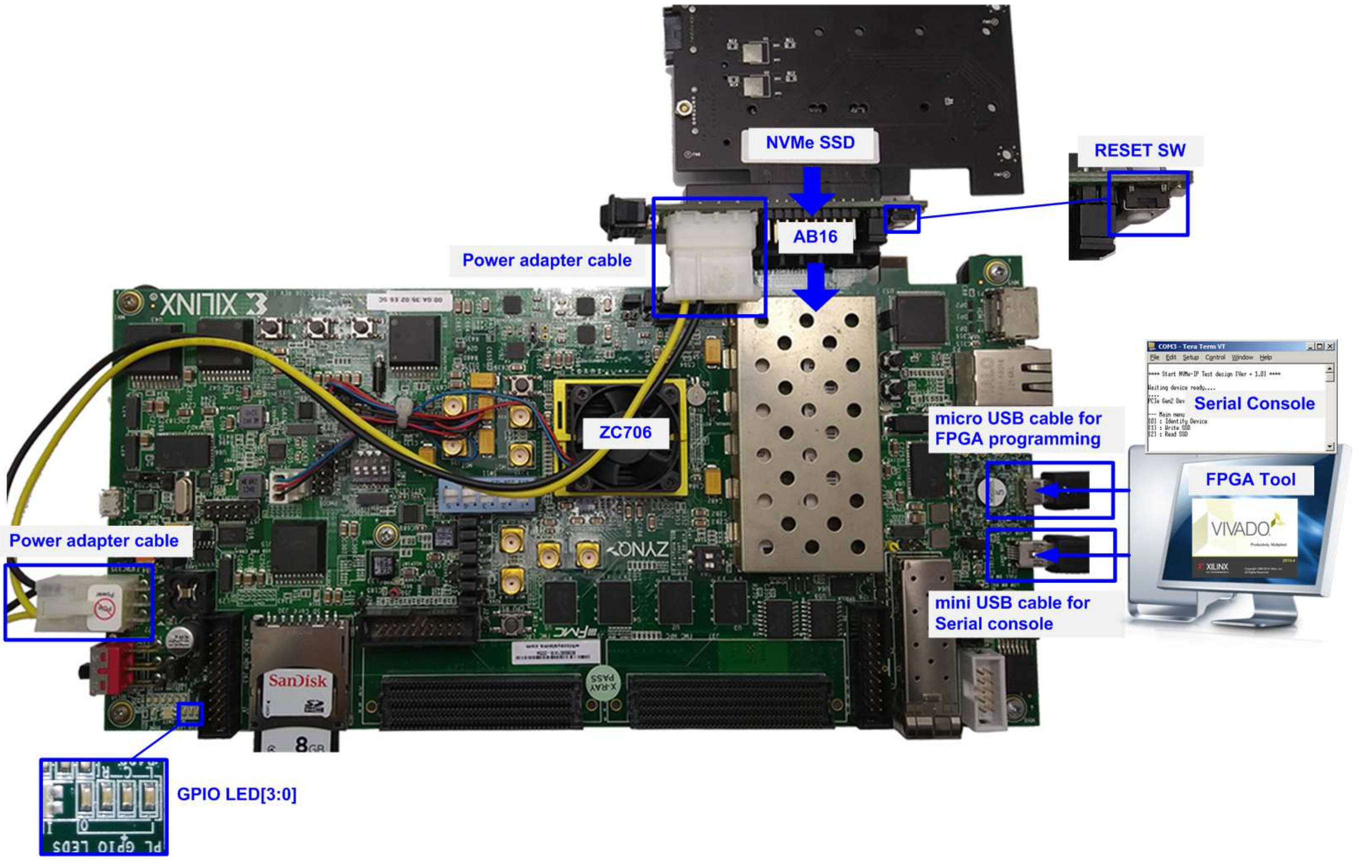
Rev1.1 25-May-18

This document describes the instruction to run FAT32-IP for NVMe demo on FPGA development board by using AB16-PCIeXOVR board. The demo is designed to write and verify data with NVMe SSD. User controls test operation through Serial console.

1 Environment Requirement

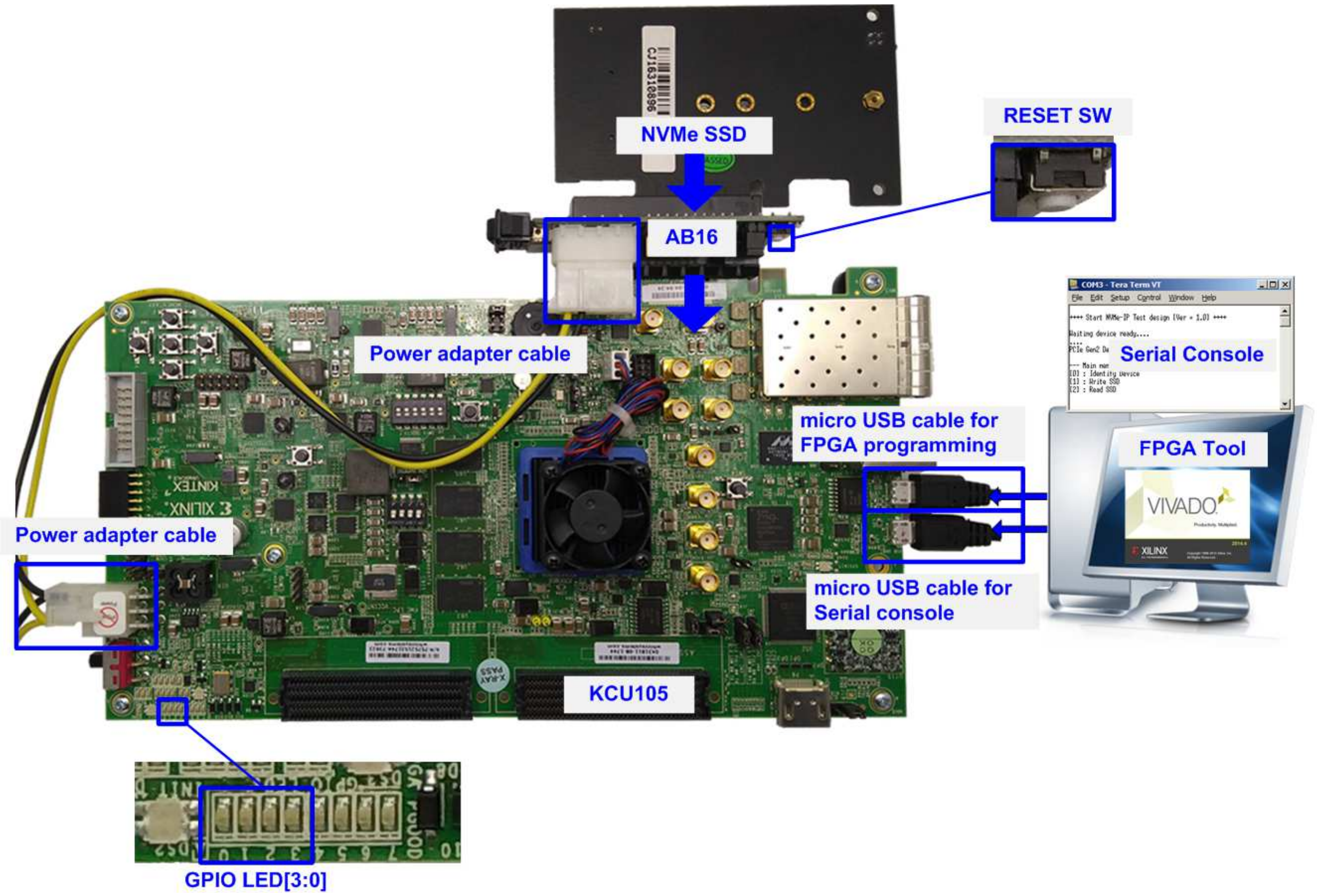
To run the demo on FPGA development board, please prepare following environment.

1) FPGA Development board: ZC706/KCU105
2) PC installing Xilinx programmer software (Vivado) and Serial console software such as HyperTerminal
3) AB16-PCIeXOVR board + power adapter cable from AB16 delivery set, provided by Design Gateway
4) Xilinx Power adapter for FPGA board
5) NVMe SSD connecting to PCIe Female connector on AB16 board
6) micro USB cable for programming FPGA, connecting between FPGA board and PC
7) mini/micro USB cable for Serial console, connecting between FPGA board and PC



```
**** Start NVMe-IP Test design [Ver = 1.0] ****

Waiting device ready....
....
PCIe Gen2 Dev

--- Main menu
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Figure 1-1 FAT32-IP for NVMe demo setup on ZC706



```
++++ Start NVMe-IP Test design [Ver = 1.0] ++++

Waiting device ready....
....
PCIe Gen2 De

--- Main men
[0] : Identity Device
[1] : Write SSD
[2] : Read SSD
```

Figure 1-2 FAT32-IP for NVMe demo setup on KCU105

2 Demo setup

1) Power off system.
2) For ZC706 board only,
   - i. Set SW11=”00000” to configure PS from JTAG, as shown in Figure 2-1.
   - ii. Set SW4=”01” to connect JTAG with USB-to-JTAG interface, as shown in Figure 2-2.

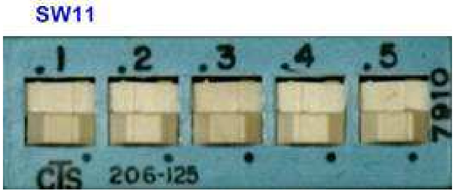


Figure 2-1 SW11 setting to configure PS from JTAG on ZC706 board

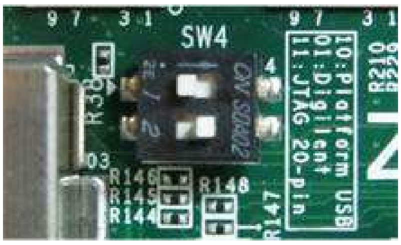


Figure 2-2 SW4 setting to use USB-to-JTAG on ZC706 board

3) Connect three connectors on power adapter cable from AB16-PCIeXOVR delivery set to Xilinx power adapter, AB16-PCIeXOVR board, and FPGA board as shown in Figure 2-3.

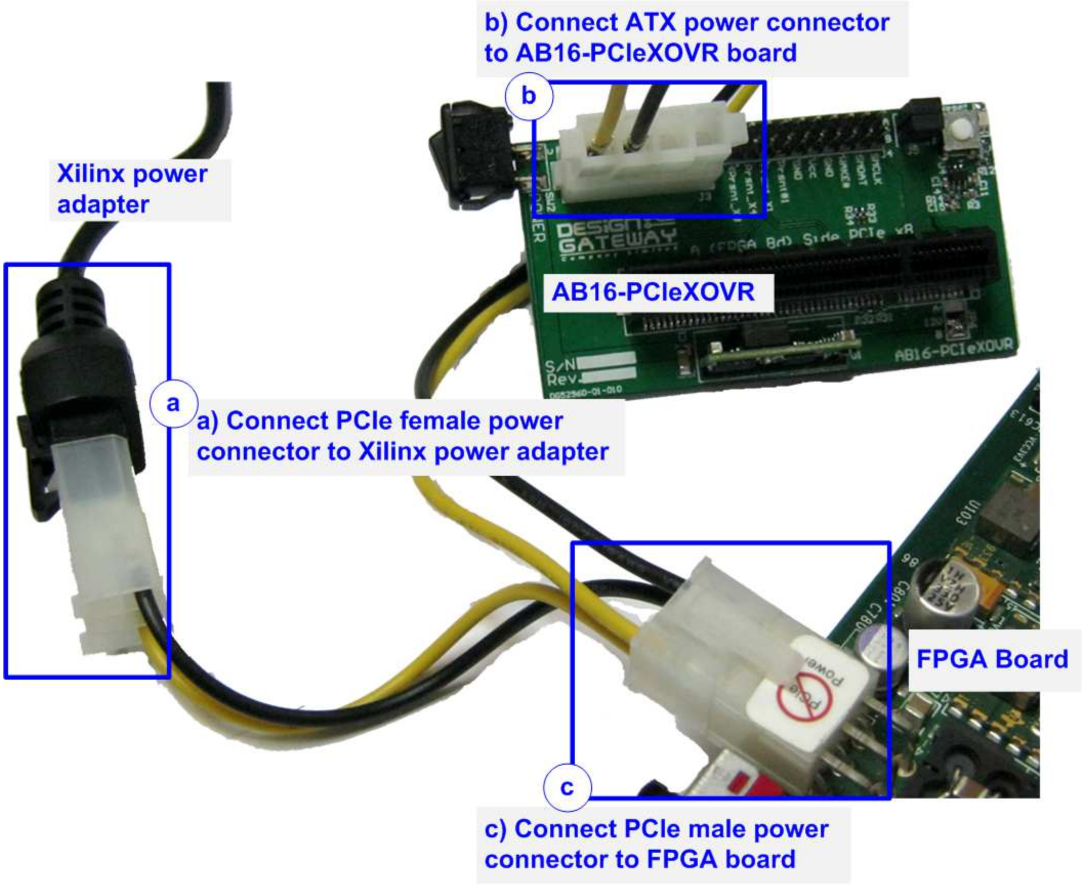

b) Connect ATX power connector to AB16-PCIeXOVR board
b
Xilinx power adapter
AB16-PCIeXOVR
a
a) Connect PCIe female power connector to Xilinx power adapter
FPGA Board
c
c) Connect PCIe male power connector to FPGA board

Figure 2-3 Connect power adapter cable to FPGA board, AB16, and Xilinx adapter

4) Connect A Side of PCIe connector on AB16-PCIeXOVR board to PCIe connector on FPGA board, as shown in Figure 2-4. Also, please confirm that two mini jumpers are inserted at J5 connector on AB16.

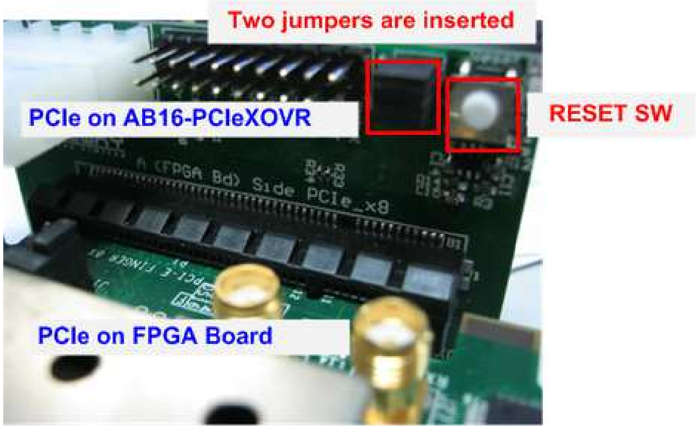


Figure 2-4 Connect PCIe connector between AB16 and FPGA board

5) Connect NVMe SSD to B Side of PCIe connector on AB16-PCIeXOVR board, as shown in Figure 2-5.

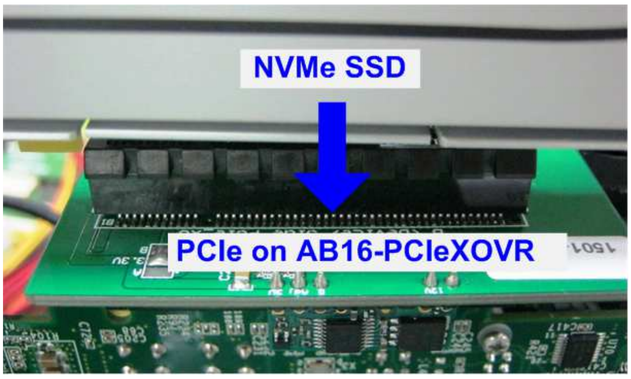


Figure 2-5 Connect NVMe SSD to AB16

6) a) For ZC706 board, connect micro USB cable between FPGA board and PC for JTAG programming, and connect mini USB cable between FPGA board and PC for Serial console.
   b) For KCU105 board, connect two micro USB cables between FPGA board and PC for JTAG programming and Serial console.

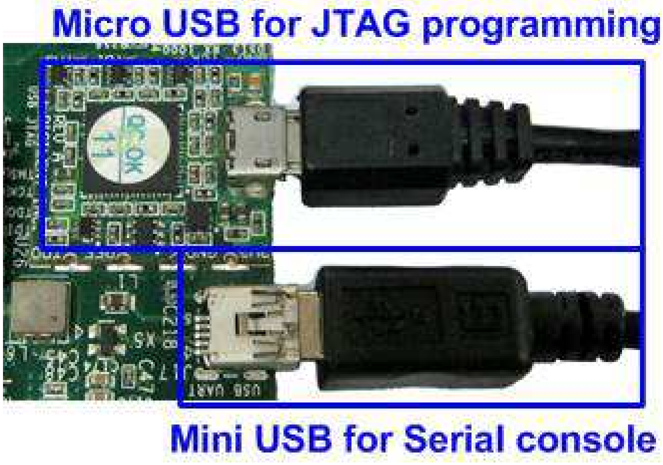


Figure 2-6 USB cable connection

7) Power on FPGA board and AB16-PCIeXOVR board.

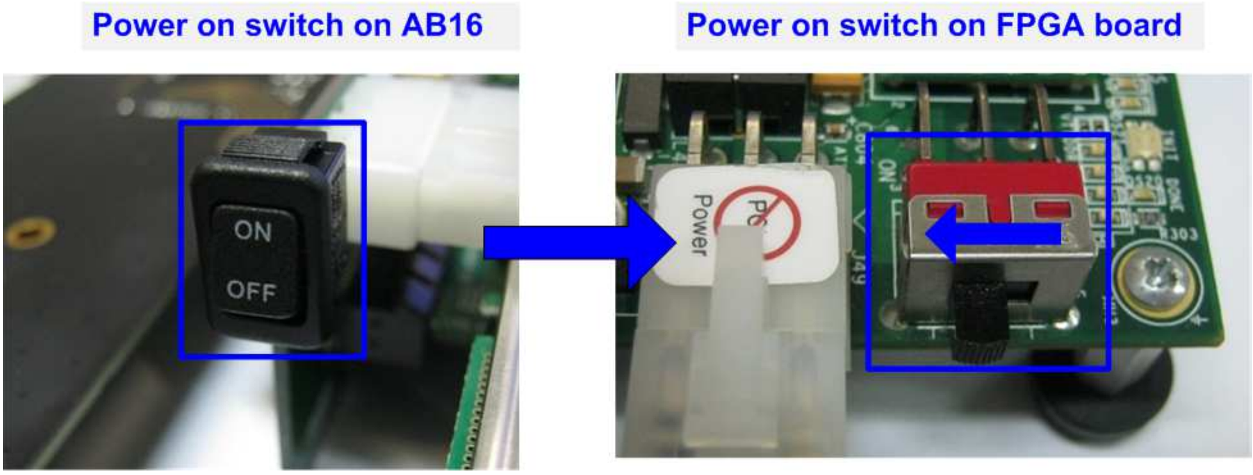
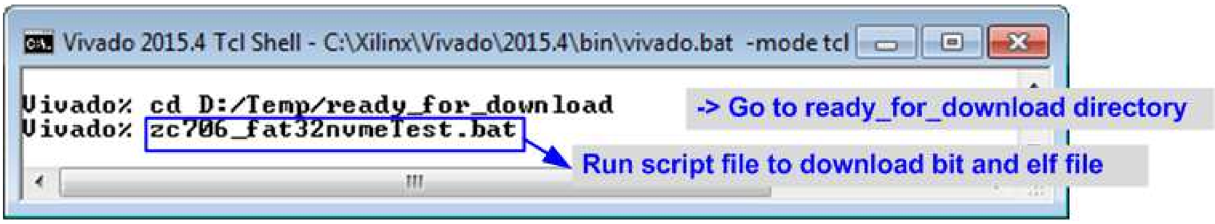


Figure 2-7 Power on AB16 and FPGA board

8) Open Serial console such as TeraTerm, HyperTerminal. Set Buad rate=115,200 Data=8 bit Non-Parity Stop=1.
9) Download bit file or bat file to configure FPGA and firmware.
   a) For ZC706 board, open Vivado TCL shell, change current directory to ready_for_download, and run zc706_fat32nvmeTest.bat, as shown in Figure 2-8


Vivado 2015.4 Tcl Shell - C:\Xilinx\Vivado\2015.4\bin\vivado.bat -mode tcl

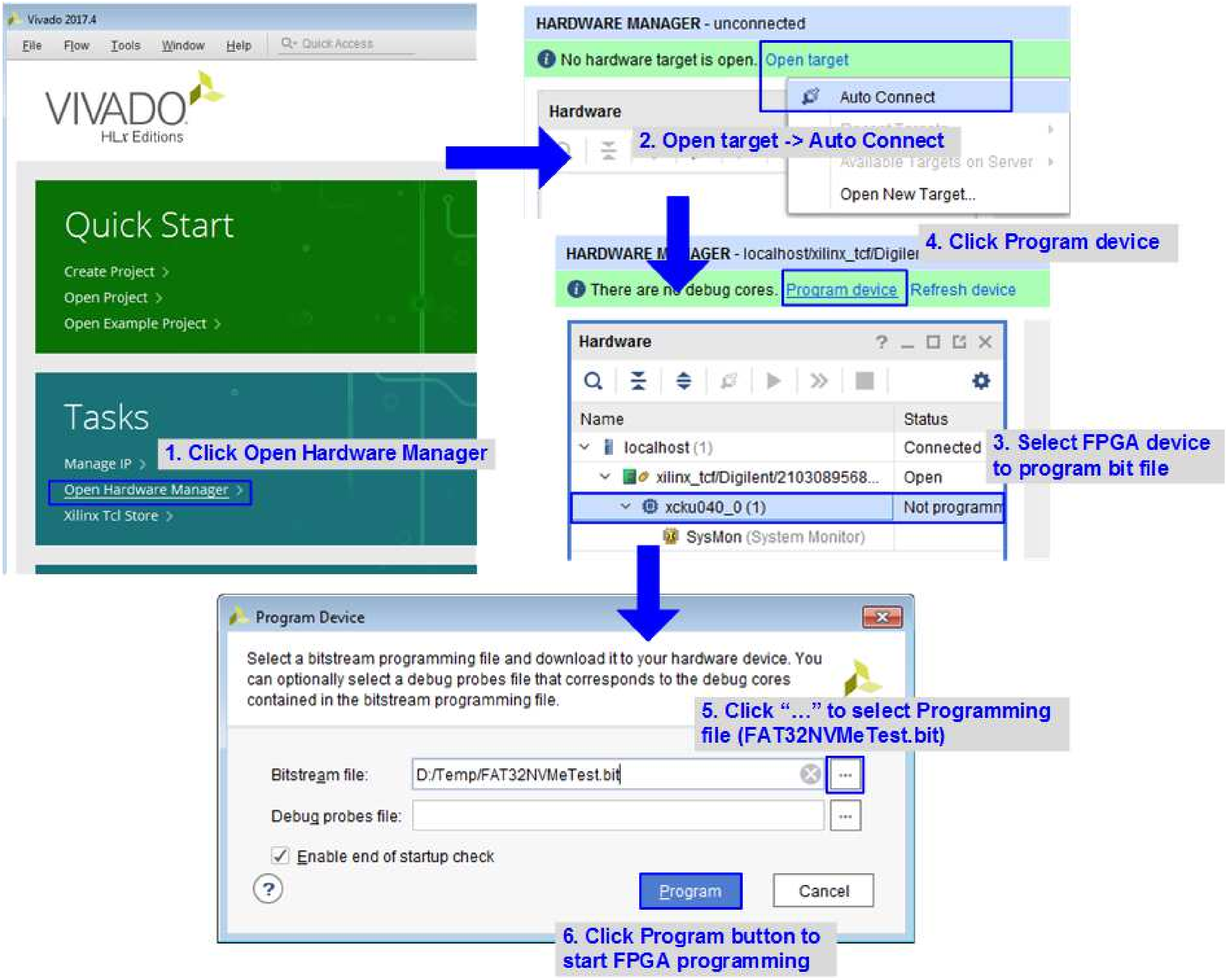
```
Vivado% cd D:/Temp/ready_for_download
Vivado% zc706_fat32nvmeTest.bat
```

-> Go to ready_for_download directory

Run script file to download bit and elf file

Figure 2-8 Command script for download demo file to ZC706 by Vivado TCL Shell

b) For KCU105 board, use Vivado tool to download configuration file, as shown in Figure 2-9



1. Click Open Hardware Manager

2. Open target -> Auto Connect

3. Select FPGA device to program bit file

4. Click Program device

5. Click "..." to select Programming file (FAT32NVMeTest.bit)

6. Click Program button to start FPGA programming

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: D:/Temp/FAT32NVMeTest.bit

Debug probes file:

☑ Enable end of startup check

Program Cancel

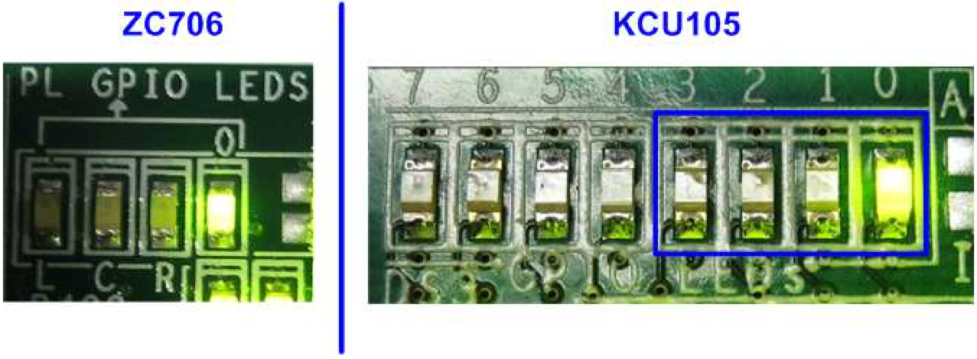
Figure 2-9 Download configuration file by Vivado tool for KCU105

10) Check LED status on FPGA board. The description of LED is as follows.

GPIO LED	ON	OFF
0	Normal operation	AB16 or SSD is not good status
1/R	System is busy	Idle status
2/C	IP Error detect	Normal operation
3/L	Data verification fail	Normal operation

Table 2-1 LED Definition

11) After programming completely, LED[0] and LED[1] are ON to show that FAT32-IP is in initialization process. LED[1] changes to OFF after FAT32-IP completes initialization process and system is ready to receive command from user. Finally, file size setting menu is displayed.


ZC706

KCU105

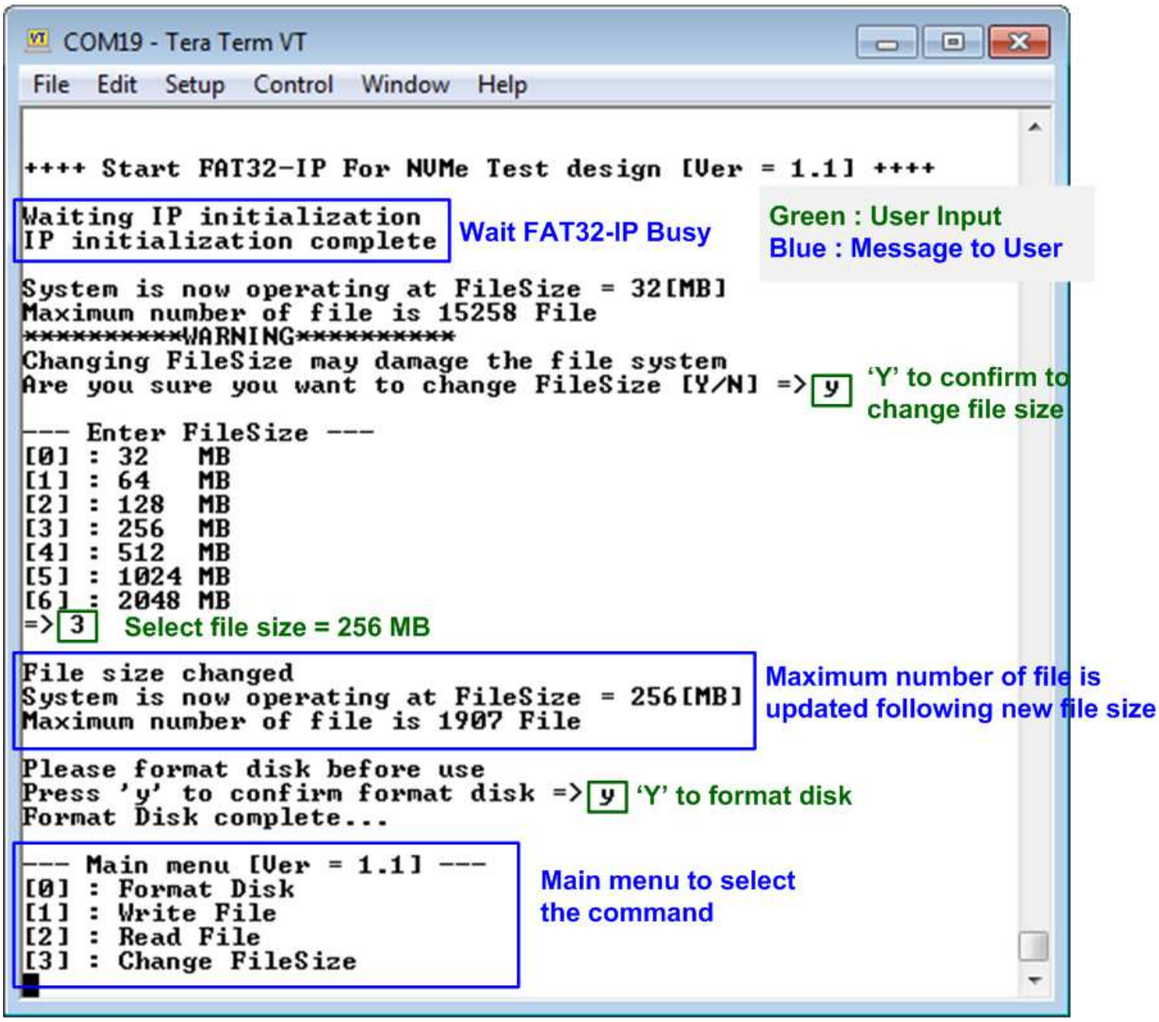
Figure 2-10 LED status after program configuration file and FAT32-IP initialization complete

12) In file size setting menu

a) Input ‘y’ or ‘Y’ to set file size. Six file sizes could be set in the submenu.

b) If user input other values, default value will be used (32 MB).

After that, boot menu is displayed on the console, as shown in Figure 2-11 and Figure 2-12.



```
++++ Start FAT32-IP For NVMe Test design [Ver = 1.1] ++++
Waiting IP initialization
IP initialization complete
System is now operating at FileSize = 32[MB]
Maximum number of file is 15258 File
**********WARNING**********
Changing FileSize may damage the file system
Are you sure you want to change FileSize [Y/N] =>y
--- Enter FileSize ---
[0] : 32   MB
[1] : 64   MB
[2] : 128  MB
[3] : 256  MB
[4] : 512  MB
[5] : 1024 MB
[6] : 2048 MB
=>3
File size changed
System is now operating at FileSize = 256[MB]
Maximum number of file is 1907 File
Please format disk before use
Press 'y' to confirm format disk =>y
Format Disk complete...
--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Wait FAT32-IP Busy

Green : User Input
Blue : Message to User

‘Y’ to confirm to change file size

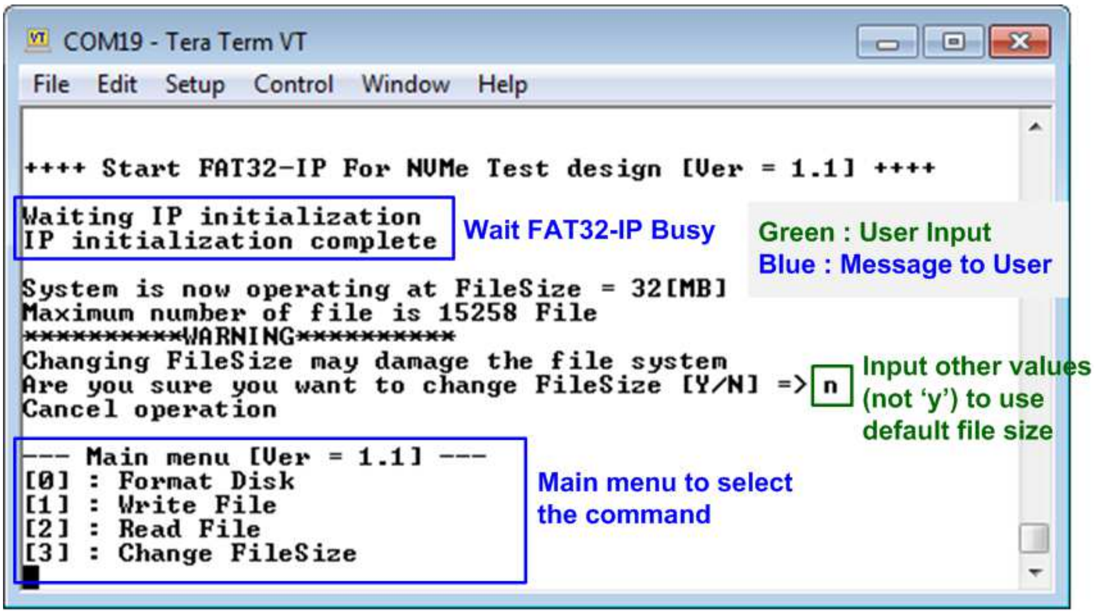
Select file size = 256 MB

Maximum number of file is updated following new file size

‘Y’ to format disk

Main menu to select the command

Figure 2-11 Main menu when confirm to set file size



```
++++ Start FAT32-IP For NVMe Test design [Ver = 1.1] ++++
Waiting IP initialization
IP initialization complete
System is now operating at FileSize = 32[MB]
Maximum number of file is 15258 File
**********WARNING**********
Changing FileSize may damage the file system
Are you sure you want to change FileSize [Y/N] => n
Cancel operation
--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Wait FAT32-IP Busy

Green : User Input
Blue : Message to User

Input other values (not 'y') to use default file size

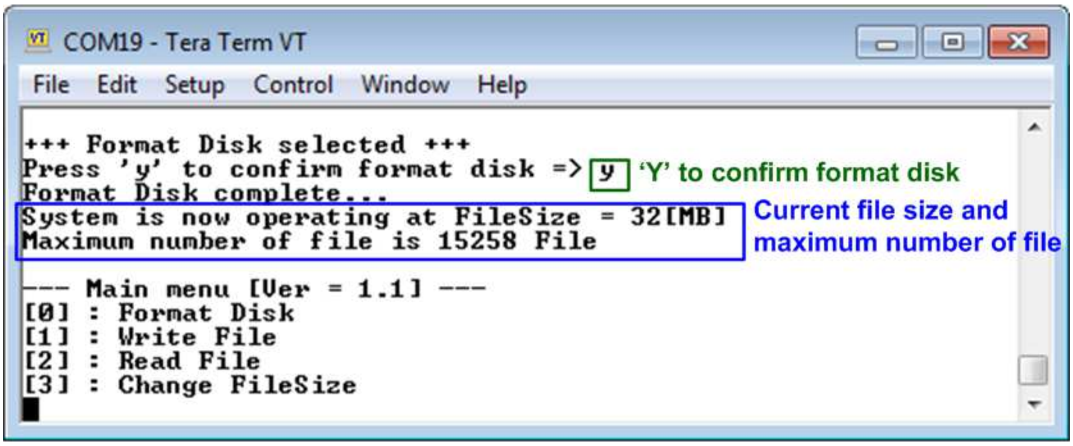
Main menu to select the command

Figure 2-12 Main menu when skip to set file size

3 Test Menu

3.1 Format Disk

Select ‘0’ to send Format disk command to FAT32-IP. When operation is completed, the system shows current operation file size and maximum number of file, as shown in Figure 3-1.



COM19 - Tera Term VT
File Edit Setup Control Window Help

```
+++ Format Disk selected +++
Press 'y' to confirm format disk => y
Format Disk complete...
System is now operating at FileSize = 32[MB]
Maximum number of file is 15258 File

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

‘Y’ to confirm format disk
Current file size and maximum number of file

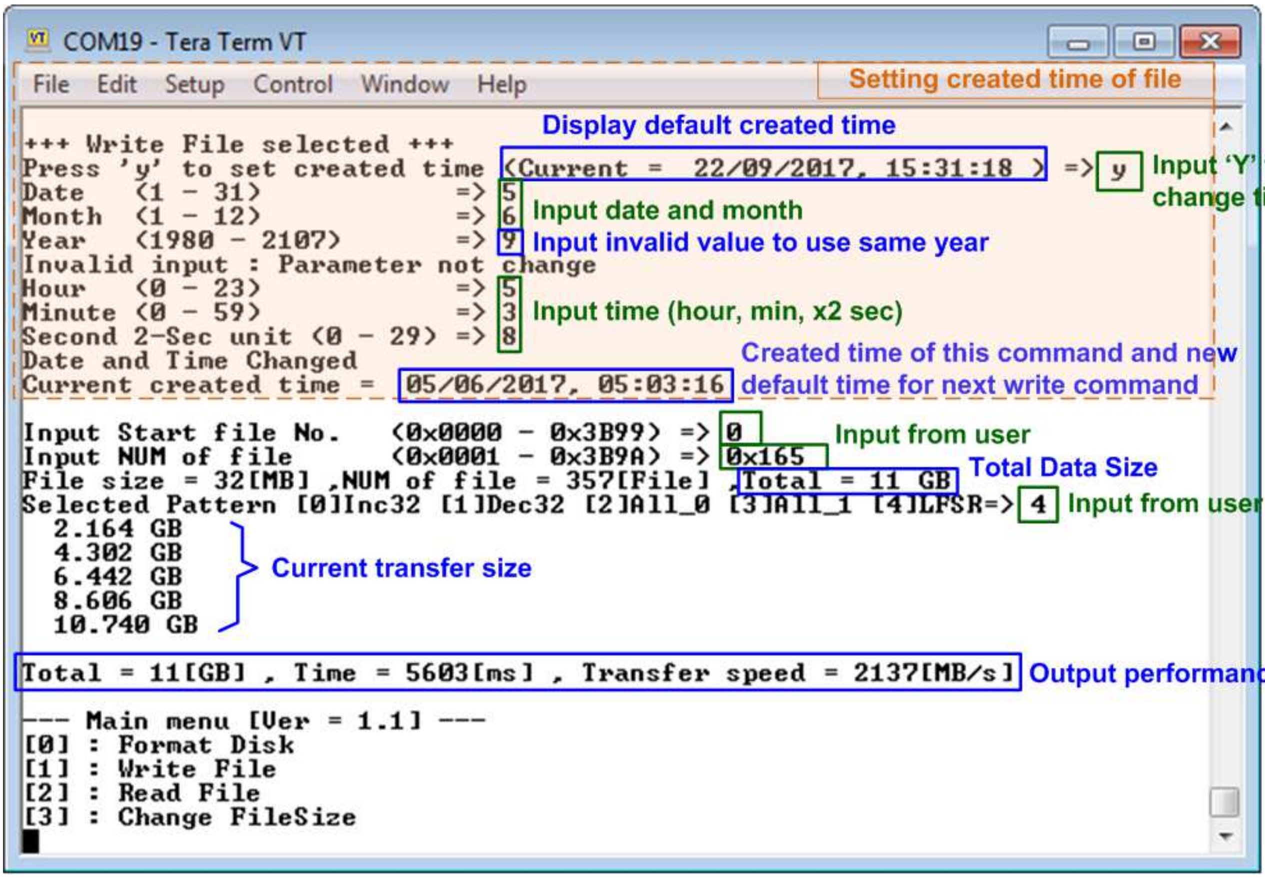
Figure 3-1 Result from Format Disk menu

It is recommended to run this menu when one of following conditions is found.
1) The disk is formatted by other system which is not FAT32-IP.
2) The disk is not formatted by FAT32-IP.
3) User needs to change file size value in the test.
4) User needs to delete file in the disk.

3.2 Write File

Select ‘1’ to send Write file command to FAT32-IP. After that, users are asked to set created time of file. Users can input ‘y’ or ‘Y’ to change created time or input others to use default created time.

<u>Note</u>: Default created time after system boot up is 22 Sep 2017, 15:31:18. When user sets new created time in Write file menu, the value is used to be created time of current Write file operation and used to update the default created time. So, default created time in the next write file operation is changed.



```
+++ Write File selected +++
Press 'y' to set created time (Current =  22/09/2017, 15:31:18 ) => y
Date    (1 - 31)               => 5
Month   (1 - 12)               => 6
Year    (1980 - 2107)          => 9
Invalid input : Parameter not change
Hour    (0 - 23)               => 5
Minute  (0 - 59)               => 3
Second 2-Sec unit (0 - 29) => 8
Date and Time Changed
Current created time =  05/06/2017, 05:03:16

Input Start file No.    (0x0000 - 0x3B99) => 0
Input NUM of file       (0x0001 - 0x3B9A) => 0x165
File size = 32[MB] ,NUM of file = 357[File] ,Total = 11 GB
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
  2.164 GB
  4.302 GB
  6.442 GB
  8.606 GB
  10.740 GB

Total = 11[GB] , Time = 5603[ms] , Transfer speed = 2137[MB/s]

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Setting created time of file

Display default created time

Input ‘Y’ to change time

Input date and month

Input invalid value to use same year

Input time (hour, min, x2 sec)

Created time of this command and new default time for next write command

Input from user

Total Data Size

Input from user

Current transfer size

Output performance

<u>Figure 3-2 Result from Write File menu with changing created time</u>

Figure 3-2 shows the example when running Write File menu with setting created time of file. There are six values to set, i.e.

a) Date – Input created date of file as decimal unit. Valid range is 1 – 31.
b) Month – Input created month of file as decimal unit. Valid range is 1 – 12.
c) Year – Input created year of file as decimal unit. Valid range is 1980 – 2107.
d) Hour – Input created hour of file as decimal unit. Valid range is 0 – 23.
e) Minute – Input created minute of file as decimal unit. Valid range is 0 – 59.
f) Second – Input created x2 second of file as decimal unit. Valid range is 0 – 29. The created second is equal to input value x 2.

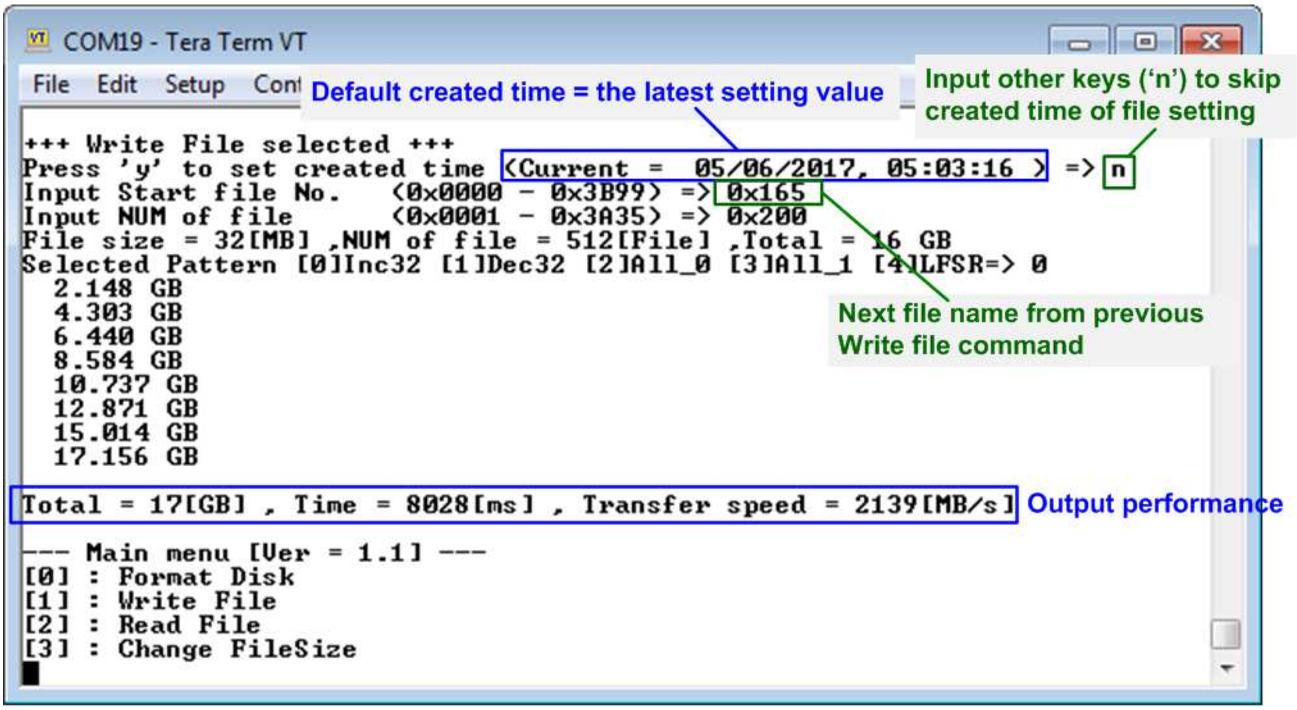
If the input is invalid, the parameter will not change and default value will be used. Only parameter that valid is changed. Finally the new created time is displayed on the console.

After setting created time process is done, three parameters are required to write file, i.e.

1) Start file No – Input file name of the first file as a number. The input is decimal unit when input only digit number. User can add “0x” to be prefix when input is hexadecimal unit.
2) NUM of file – Input total number of file to transfer. The first file name is FILE<Start file No>.BIN and the last file name is FILE<Start file No + NUM of file.BIN. The input is decimal unit when input only digit number. User can add “0x” to be prefix when input is hexadecimal unit.
   If “Start file No” and “NUM of file” are valid, total data size (calculated from File size x NUM of file) will be displayed on the console.
3) Test pattern – Select pattern of test data in written file. Five patterns can be set, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR counter.

If all inputs are valid, the operation will be started. During writing file, current transfer size is displayed on the console to show that system still be alive every second. Finally, test performance, total size, and total time usage are displayed on the console as test result.

Figure 3-3 shows the example to Write File without setting created date. In this example, default created time is updated by the value in previous Write File test. It is recommended to input “Start file No” value by the next value which continues from previous test. Previous test in Figure 3-2 writes FILE0000.BIN – FILE0164.BIN, so “Start file No” should be 0x165 (0x164 + 1).



Default created time = the latest setting value

Input other keys (‘n’) to skip created time of file setting

Next file name from previous Write file command

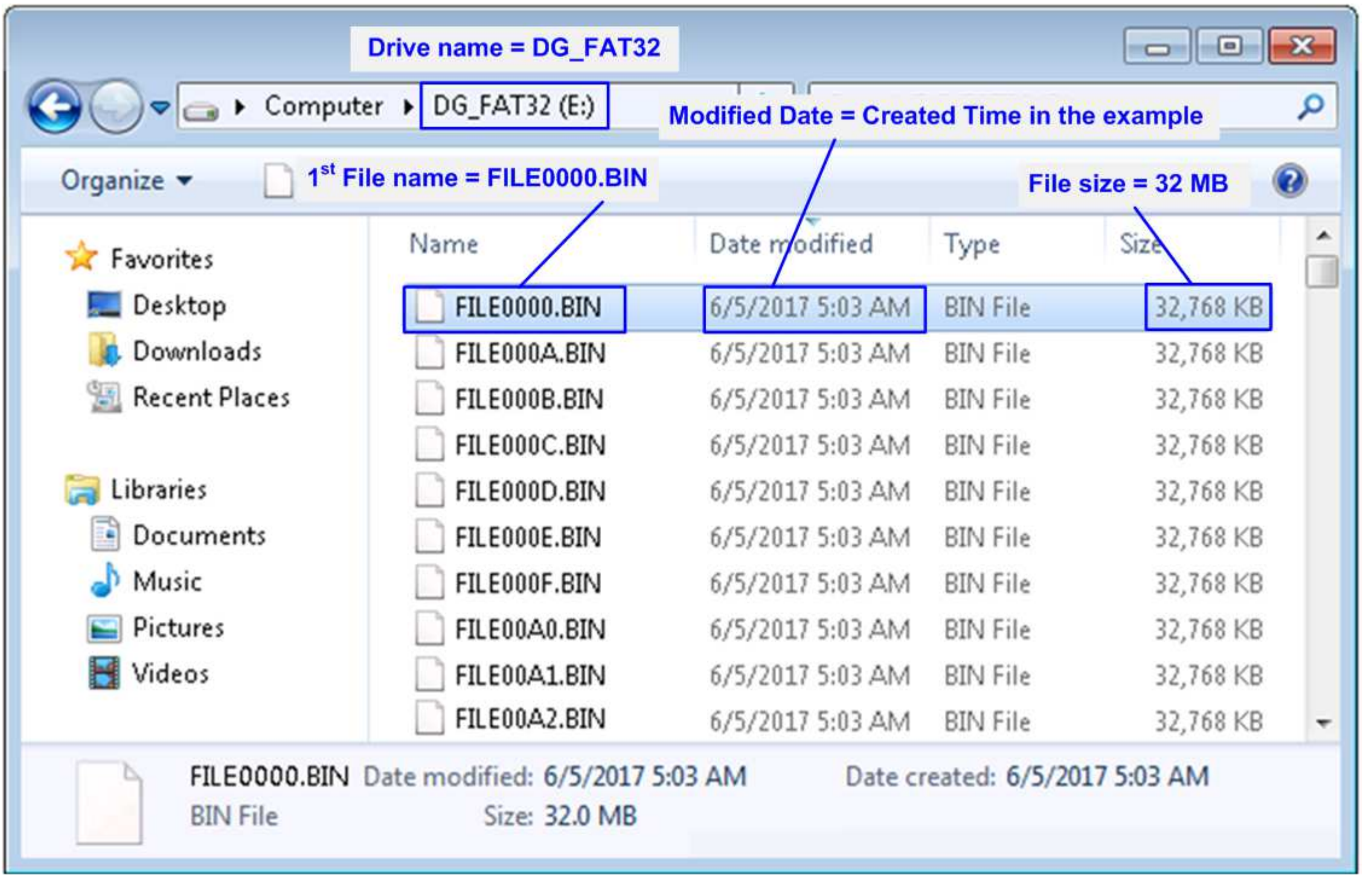
Output performance

```
+++ Write File selected +++
Press 'y' to set created time (Current =  05/06/2017, 05:03:16 ) => n
Input Start file No.      (0x0000 - 0x3B99) => 0x165
Input NUM of file         (0x0001 - 0x3A35) => 0x200
File size = 32[MB] ,NUM of file = 512[File] ,Total = 16 GB
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 0
  2.148 GB
  4.303 GB
  6.440 GB
  8.584 GB
  10.737 GB
  12.871 GB
  15.014 GB
  17.156 GB

Total = 17[GB] , Time = 8028[ms] , Transfer speed = 2139[MB/s]

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Figure 3-3 Result from Write File menu without changing created time

After completing Write File command, user can plug SSD to PC which supports FAT32 File format. On PC, the new disk “DG_FAT32” is detected. Inside the disk, FILE0000.BIN – FILE<Last file No>.BIN are stored. Also, File size and modified date must be matched to the value set in Write File test. Four numbers in File name is file number in hexadecimal unit.

Note: When connecting SSD to PC, please do not write or modify a file in the disk. If file in SSD has some modification, SSD must be formatted by FAT32-IP.


Drive name = DG_FAT32

Modified Date = Created Time in the example

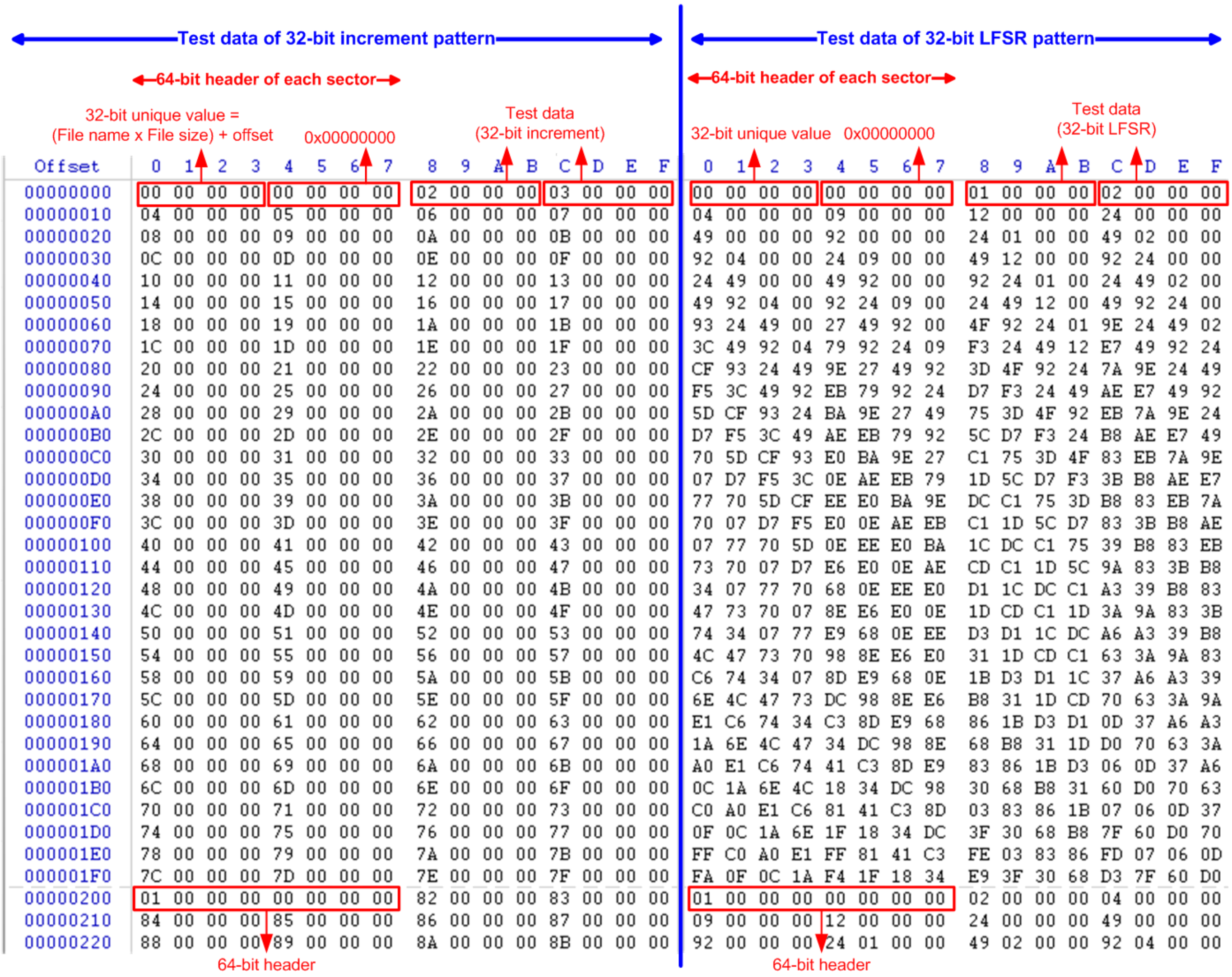
1st File name = FILE0000.BIN

File size = 32 MB

Name	Date modified	Type	Size
FILE0000.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE000A.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE000B.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE000C.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE000D.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE000E.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE000F.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE00A0.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE00A1.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB
FILE00A2.BIN	6/5/2017 5:03 AM	BIN File	32,768 KB

FILE0000.BIN Date modified: 6/5/2017 5:03 AM Date created: 6/5/2017 5:03 AM
BIN File Size: 32.0 MB

Figure 3-4 Example test files written by Write File command

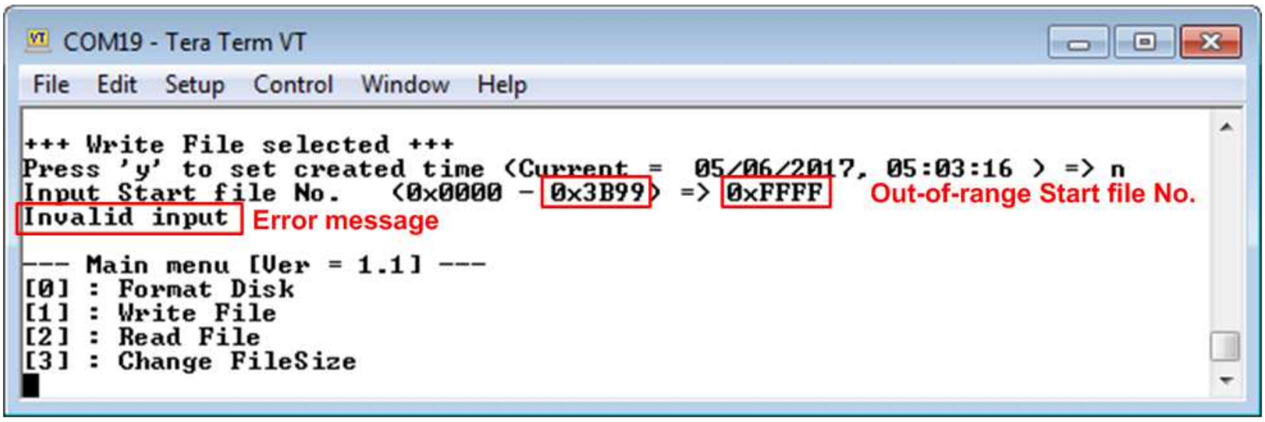

Test data of 32-bit increment pattern
Test data of 32-bit LFSR pattern
64-bit header of each sector
32-bit unique value =
(File name x File size) + offset
0x00000000
Test data
(32-bit increment)
64-bit header of each sector
32-bit unique value
0x00000000
Test data
(32-bit LFSR)
64-bit header
64-bit header

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  00 00 00 00 00 00 00 00  02 00 00 00 03 00 00 00  00 00 00 00 00 00 00 00  01 00 00 00 02 00 00 00
00000010  04 00 00 00 05 00 00 00  06 00 00 00 07 00 00 00  04 00 00 00 09 00 00 00  12 00 00 00 24 00 00 00
00000020  08 00 00 00 09 00 00 00  0A 00 00 00 0B 00 00 00  49 00 00 00 92 00 00 00  24 01 00 00 49 02 00 00
00000030  0C 00 00 00 0D 00 00 00  0E 00 00 00 0F 00 00 00  92 04 00 00 24 09 00 00  49 12 00 00 92 24 00 00
00000040  10 00 00 00 11 00 00 00  12 00 00 00 13 00 00 00  24 49 00 00 49 92 00 00  92 24 01 00 24 49 02 00
00000050  14 00 00 00 15 00 00 00  16 00 00 00 17 00 00 00  49 92 04 00 92 24 09 00  24 49 12 00 49 92 24 00
00000060  18 00 00 00 19 00 00 00  1A 00 00 00 1B 00 00 00  93 24 49 00 27 49 92 00  4F 92 24 01 9E 24 49 02
00000070  1C 00 00 00 1D 00 00 00  1E 00 00 00 1F 00 00 00  3C 49 92 04 79 92 24 09  F3 24 49 12 E7 49 92 24
00000080  20 00 00 00 21 00 00 00  22 00 00 00 23 00 00 00  CF 93 24 49 9E 27 49 92  3D 4F 92 24 7A 9E 24 49
00000090  24 00 00 00 25 00 00 00  26 00 00 00 27 00 00 00  F5 3C 49 92 EB 79 92 24  D7 F3 24 49 AE E7 49 92
000000A0  28 00 00 00 29 00 00 00  2A 00 00 00 2B 00 00 00  5D CF 93 24 BA 9E 27 49  75 3D 4F 92 EB 7A 9E 24
000000B0  2C 00 00 00 2D 00 00 00  2E 00 00 00 2F 00 00 00  D7 F5 3C 49 AE EB 79 92  5C D7 F3 24 B8 AE E7 49
000000C0  30 00 00 00 31 00 00 00  32 00 00 00 33 00 00 00  70 5D CF 93 E0 BA 9E 27  C1 75 3D 4F 83 EB 7A 9E
000000D0  34 00 00 00 35 00 00 00  36 00 00 00 37 00 00 00  07 D7 F5 3C 0E AE EB 79  1D 5C D7 F3 3B B8 AE E7
000000E0  38 00 00 00 39 00 00 00  3A 00 00 00 3B 00 00 00  77 70 5D CF EE E0 BA 9E  DC C1 75 3D B8 83 EB 7A
000000F0  3C 00 00 00 3D 00 00 00  3E 00 00 00 3F 00 00 00  70 07 D7 F5 E0 0E AE EB  C1 1D 5C D7 83 3B B8 AE
00000100  40 00 00 00 41 00 00 00  42 00 00 00 43 00 00 00  07 77 70 5D 0E EE E0 BA  1C DC C1 75 39 B8 83 EB
00000110  44 00 00 00 45 00 00 00  46 00 00 00 47 00 00 00  73 70 07 D7 E6 E0 0E AE  CD C1 1D 5C 9A 83 3B B8
00000120  48 00 00 00 49 00 00 00  4A 00 00 00 4B 00 00 00  34 07 77 70 68 0E EE E0  D1 1C DC C1 A3 39 B8 83
00000130  4C 00 00 00 4D 00 00 00  4E 00 00 00 4F 00 00 00  47 73 70 07 8E E6 E0 0E  1D CD C1 1D 3A 9A 83 3B
00000140  50 00 00 00 51 00 00 00  52 00 00 00 53 00 00 00  74 34 07 77 E9 68 0E EE  D3 D1 1C DC A6 A3 39 B8
00000150  54 00 00 00 55 00 00 00  56 00 00 00 57 00 00 00  4C 47 73 70 98 8E E6 E0  31 1D CD C1 63 3A 9A 83
00000160  58 00 00 00 59 00 00 00  5A 00 00 00 5B 00 00 00  C6 74 34 07 8D E9 68 0E  1B D3 D1 1C 37 A6 A3 39
00000170  5C 00 00 00 5D 00 00 00  5E 00 00 00 5F 00 00 00  6E 4C 47 73 DC 98 8E E6  B8 31 1D CD 70 63 3A 9A
00000180  60 00 00 00 61 00 00 00  62 00 00 00 63 00 00 00  E1 C6 74 34 C3 8D E9 68  86 1B D3 D1 0D 37 A6 A3
00000190  64 00 00 00 65 00 00 00  66 00 00 00 67 00 00 00  1A 6E 4C 47 34 DC 98 8E  68 B8 31 1D D0 70 63 3A
000001A0  68 00 00 00 69 00 00 00  6A 00 00 00 6B 00 00 00  A0 E1 C6 74 41 C3 8D E9  83 86 1B D3 06 0D 37 A6
000001B0  6C 00 00 00 6D 00 00 00  6E 00 00 00 6F 00 00 00  0C 1A 6E 4C 18 34 DC 98  30 68 B8 31 60 D0 70 63
000001C0  70 00 00 00 71 00 00 00  72 00 00 00 73 00 00 00  C0 A0 E1 C6 81 41 C3 8D  03 83 86 1B 07 06 0D 37
000001D0  74 00 00 00 75 00 00 00  76 00 00 00 77 00 00 00  0F 0C 1A 6E 1F 18 34 DC  3F 30 68 B8 7F 60 D0 70
000001E0  78 00 00 00 79 00 00 00  7A 00 00 00 7B 00 00 00  FF C0 A0 E1 FF 81 41 C3  FE 03 83 86 FD 07 06 0D
000001F0  7C 00 00 00 7D 00 00 00  7E 00 00 00 7F 00 00 00  FA 0F 0C 1A F4 1F 18 34  E9 3F 30 68 D3 7F 60 D0
00000200  01 00 00 00 00 00 00 00  82 00 00 00 83 00 00 00  01 00 00 00 00 00 00 00  02 00 00 00 04 00 00 00
00000210  84 00 00 00 85 00 00 00  86 00 00 00 87 00 00 00  09 00 00 00 12 00 00 00  24 00 00 00 49 00 00 00
00000220  88 00 00 00 89 00 00 00  8A 00 00 00 8B 00 00 00  92 00 00 00 24 01 00 00  49 02 00 00 92 04 00 00
```

Figure 3-5 Test data in sector#0 - #1 of FILE0000.BIN by increment/LFSR pattern

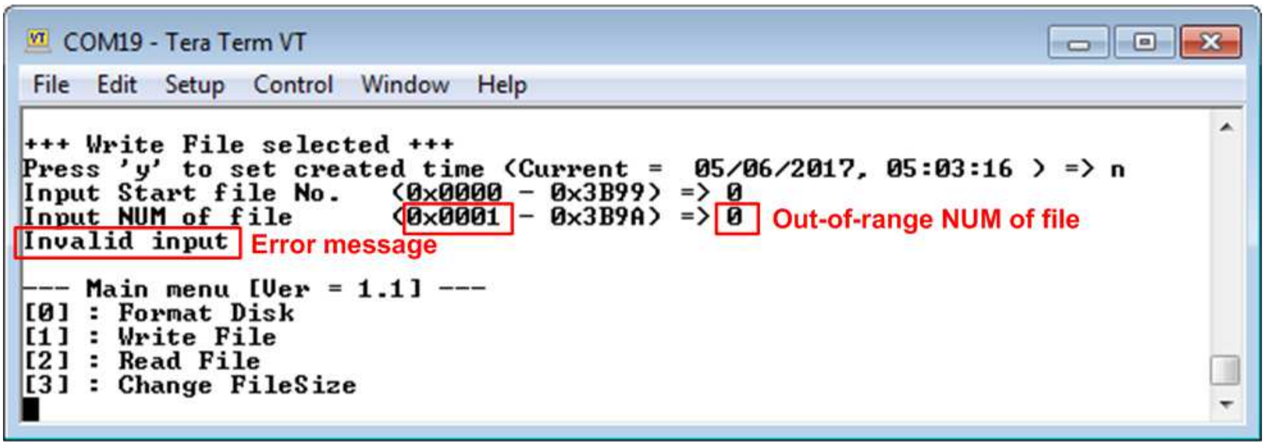
Figure 3-5 shows the example of the data in FILE0000.BIN when writing data by increment (left window) and LFSR pattern (right window). 64-bit header is inserted every 512-byte test data. The header is unique value in one SSD. The first 32-bit header is calculated by (file name x file size) + offset in the file. The remaining 32-bit data is 0. The offset value is started from 0 and increased by 1 every 512-byte data. The remaining data of one sector is test data which the pattern is defined from user.

Figure 3-6 – Figure 3-8 show error messages when user input is invalid. “Invalid input” message is displayed on the console, and then returns to main menu to receive new command.


COM19 - Tera Term VT
File Edit Setup Control Window Help
Error message
Out-of-range Start file No.

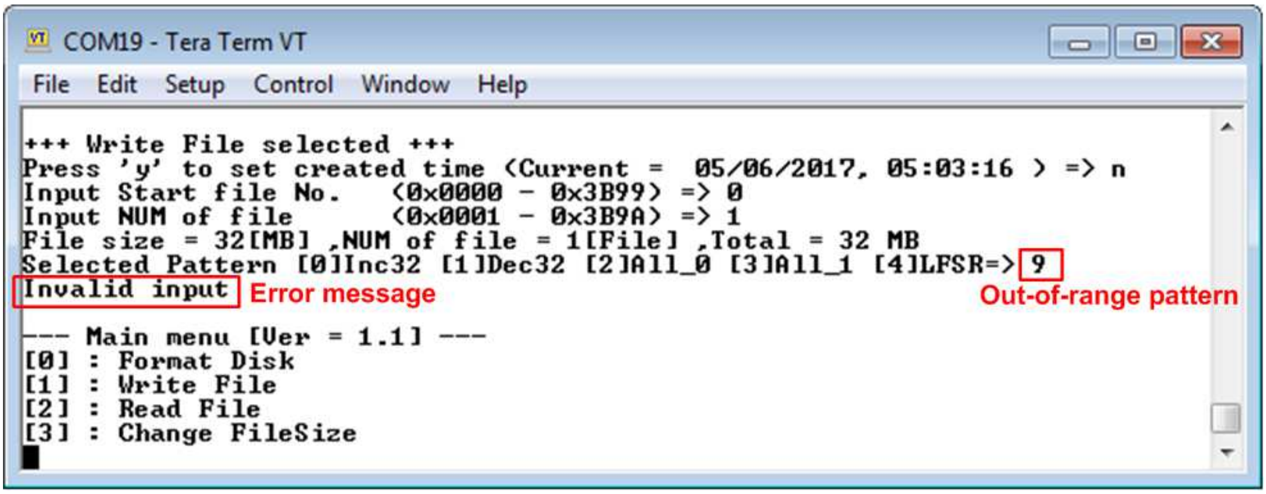
```
+++ Write File selected +++
Press 'y' to set created time (Current =  05/06/2017, 05:03:16 ) => n
Input Start file No.   (0x0000 - 0x3B99) => 0xFFFF
Invalid input

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Figure 3-6 Invalid Start file no. input



```
+++ Write File selected +++
Press 'y' to set created time (Current =  05/06/2017, 05:03:16 ) => n
Input Start file No.   (0x0000 - 0x3B99) => 0
Input NUM of file      (0x0001 - 0x3B9A) => 0
Invalid input

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Figure 3-7 Invalid Num of file input



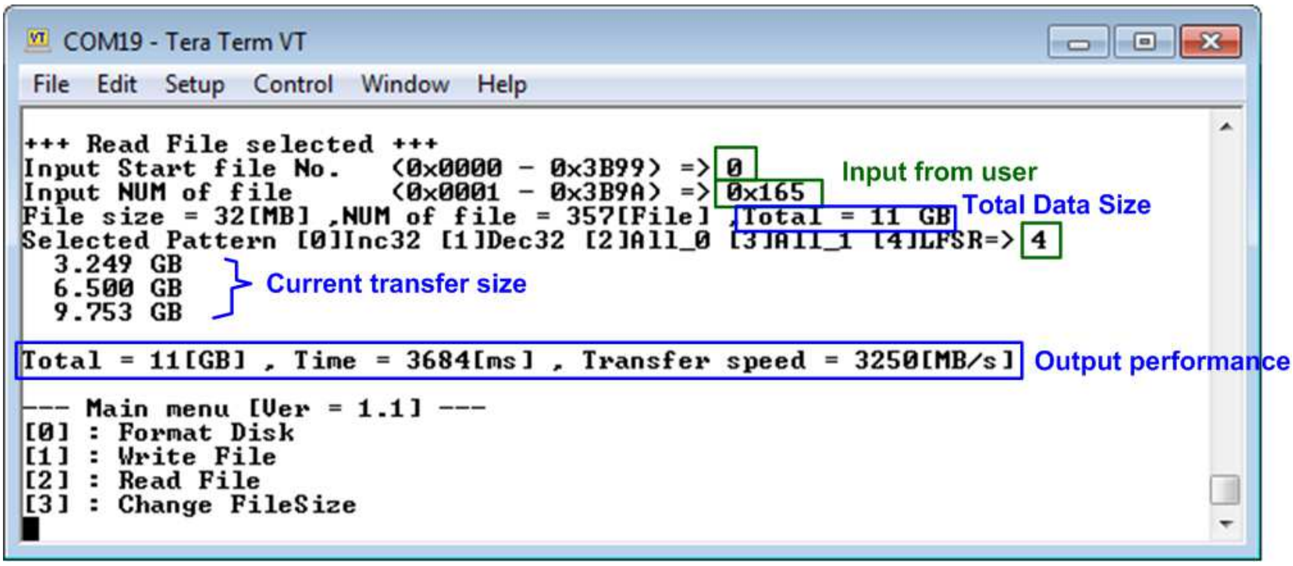
```
+++ Write File selected +++
Press 'y' to set created time (Current =  05/06/2017, 05:03:16 ) => n
Input Start file No.   (0x0000 - 0x3B99) => 0
Input NUM of file      (0x0001 - 0x3B9A) => 1
File size = 32[MB] ,NUM of file = 1[File] ,Total = 32 MB
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 9
Invalid input

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Figure 3-8 Invalid Test pattern input

3.2 Read File

Select '2' to send Read file command to FAT32-IP. Similar to Write File menu, three parameters are required to read file, i.e.

1) Start file No – Input a number of the first read file. The input is decimal unit when input only digit number. User can add "0x" to be prefix when input is hexadecimal unit.
2) NUM of file – Input total number of file to transfer. The first file name is FILE<Start file No>.BIN and the last file name is FILE<Start file No + NUM of file.BIN. The input is decimal unit when input only digit number. User can add "0x" to be prefix when input is hexadecimal unit.
   If "Start file No" and "NUM of file" are valid, total data size (calculated from File size x NUM of file) will be displayed on the console.
3) Select pattern – Select test pattern to verify data in the file. Test pattern must be matched with the test pattern using in Write File menu. Five patterns can be set, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR counter.



```
+++ Read File selected +++
Input Start file No.    (0x0000 - 0x3B99) =>0
Input NUM of file       (0x0001 - 0x3B9A) =>0x165
File size = 32[MB] ,NUM of file = 357[File] ,Total = 11 GB
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=>4
  3.249 GB
  6.500 GB
  9.753 GB

Total = 11[GB] , Time = 3684[ms] , Transfer speed = 3250[MB/s]

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Input from user

Total Data Size

Current transfer size

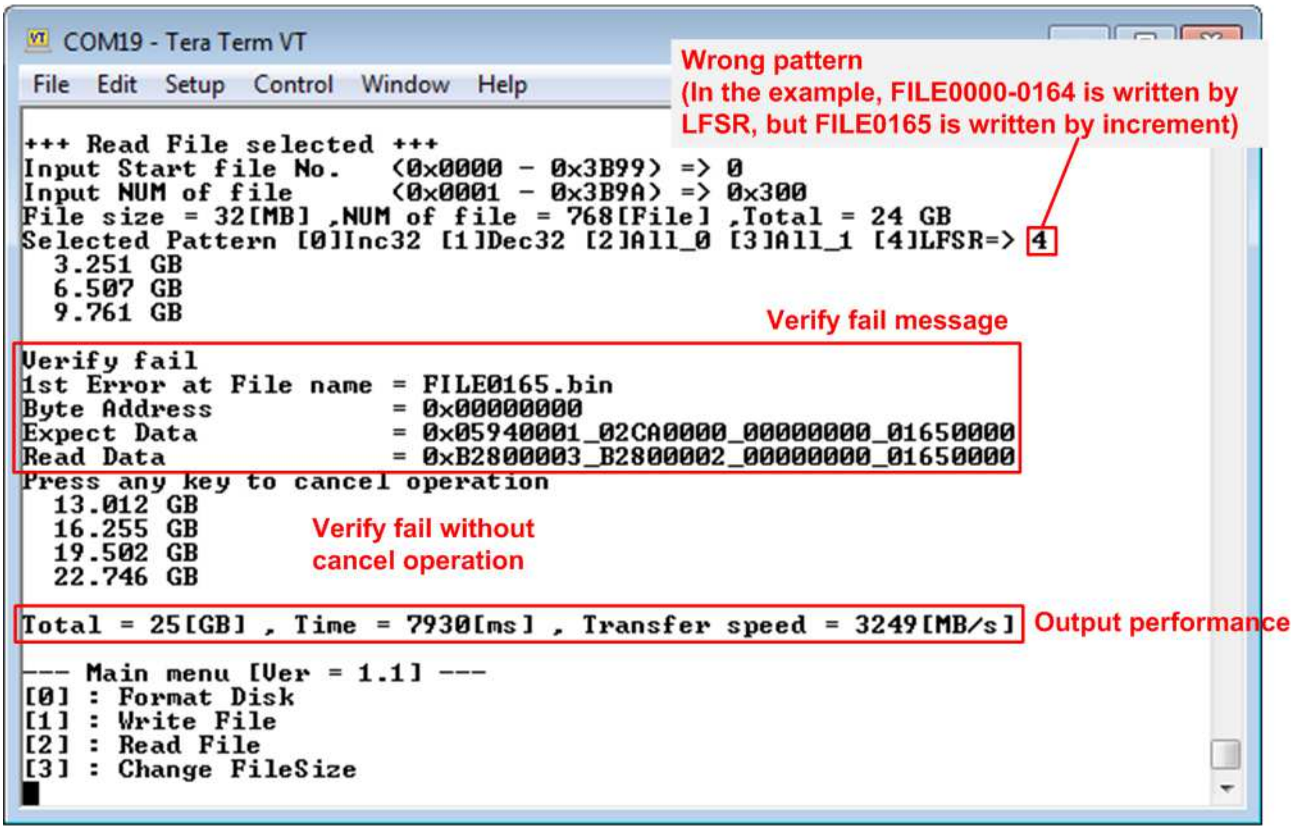
Output performance

Figure 3-9 Result and input from Read File menu with successful verification

Similar to Write File menu, if all inputs are valid, the operation will be started. During reading file, current transfer size is displayed on the console to show that system still be alive every second. Finally, test performance, total size, and total time usage are displayed on the console as test result. "Invalid input" will be displayed if some inputs are out-of-range.

Figure 3-10 and Figure 3-11 show the error message when data verification is failed. "Verify fail" message is displayed with the first file name which has error, error address of a file, expected data, and read data. User can press any keys to cancel the operation or wait until Read file complete. If read file is completed, output performance from Read file process will be displayed.

In case of cancel operation, the previous command does not complete in good sequence. It is recommended to power-off/on FPGA board and SSD, and press "RESET" button to restart system.



```
+++ Read File selected +++
Input Start file No.    (0x0000 - 0x3B99) => 0
Input NUM of file       (0x0001 - 0x3B9A) => 0x300
File size = 32[MB] ,NUM of file = 768[File] ,Total = 24 GB
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
   3.251 GB
   6.507 GB
   9.761 GB

Verify fail
1st Error at File name  = FILE0165.bin
Byte Address            = 0x00000000
Expect Data             = 0x05940001_02CA0000_00000000_01650000
Read Data               = 0xB2800003_B2800002_00000000_01650000
Press any key to cancel operation
   13.012 GB
   16.255 GB
   19.502 GB
   22.746 GB

Total = 25[GB] , Time = 7930[ms] , Transfer speed = 3249[MB/s]

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

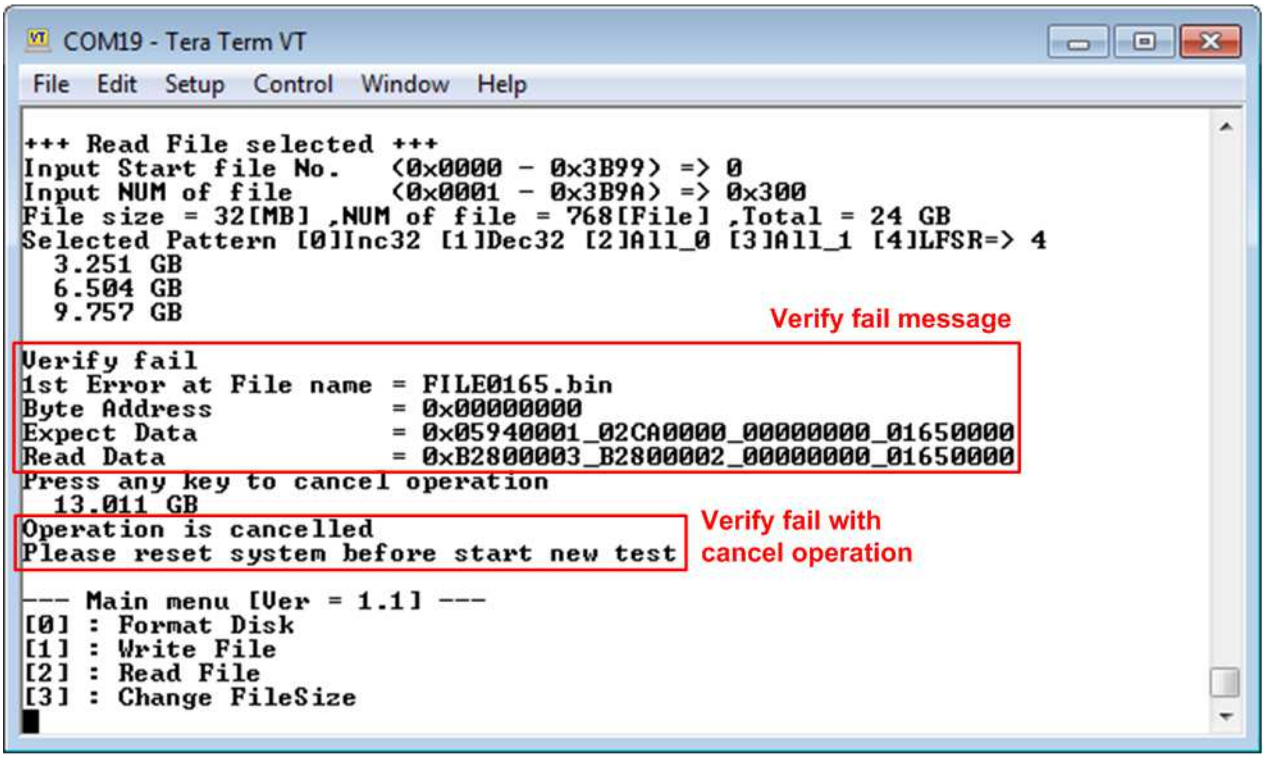
Wrong pattern
(In the example, FILE0000-0164 is written by LFSR, but FILE0165 is written by increment)

Verify fail message

Verify fail without cancel operation

Output performance

Figure 3-10 Data verification is failed but wait until read complete



```
+++ Read File selected +++
Input Start file No.    (0x0000 - 0x3B99) => 0
Input NUM of file       (0x0001 - 0x3B9A) => 0x300
File size = 32[MB] ,NUM of file = 768[File] ,Total = 24 GB
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
   3.251 GB
   6.504 GB
   9.757 GB

Verify fail
1st Error at File name  = FILE0165.bin
Byte Address            = 0x00000000
Expect Data             = 0x05940001_02CA0000_00000000_01650000
Read Data               = 0xB2800003_B2800002_00000000_01650000
Press any key to cancel operation
   13.011 GB
Operation is cancelled
Please reset system before start new test

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

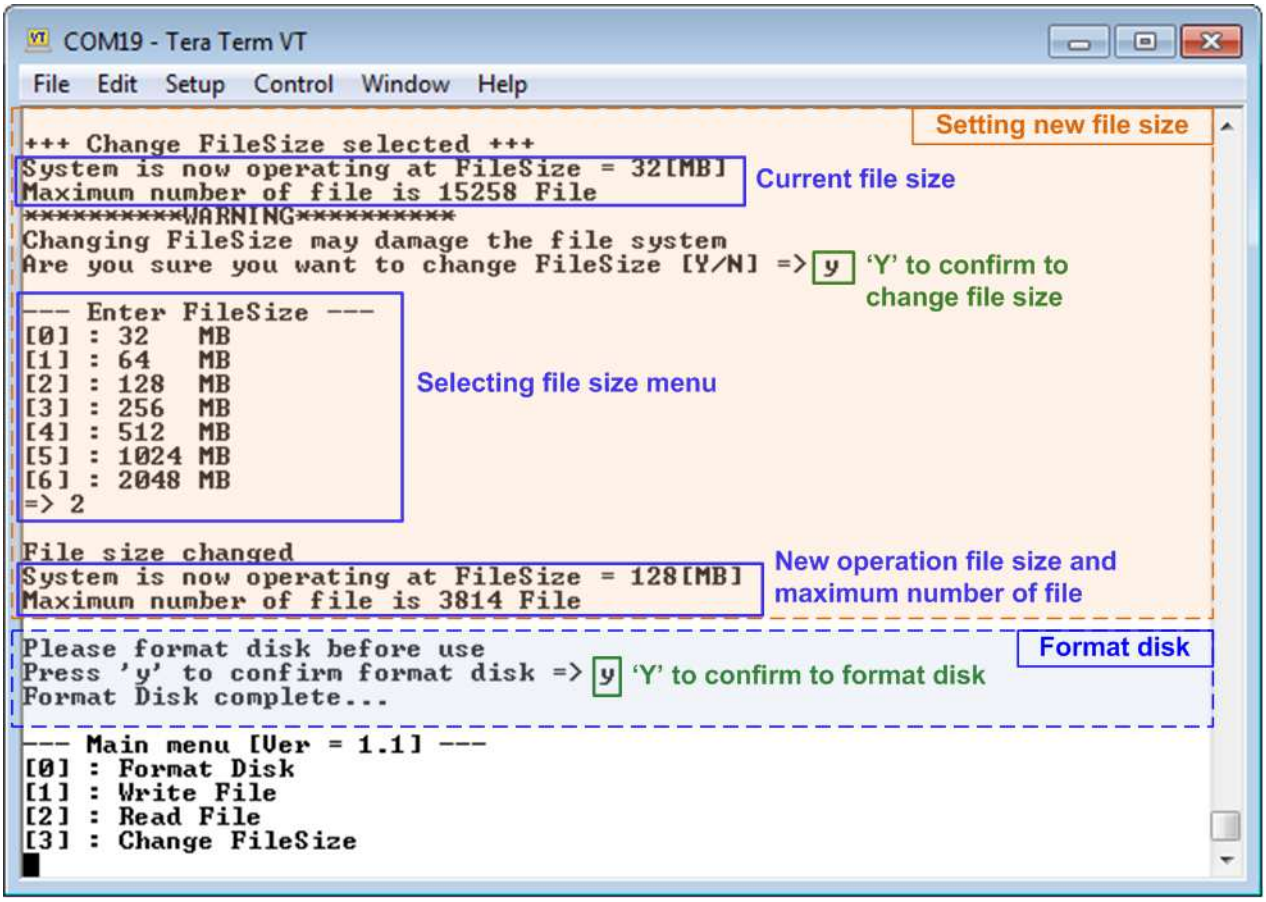
Verify fail message

Verify fail with cancel operation

Figure 3-11 Data verification is failed and press any keys to cancel operation

3.3 Change File Size

Select ‘3’ to change file size. After selecting menu, current file size and warning message are displayed on the console. Then, user input ‘y’ or ‘Y’ to confirm to change file size or input others to cancel the operation.



COM19 - Tera Term VT
File Edit Setup Control Window Help
Setting new file size
Current file size
‘Y’ to confirm to change file size
Selecting file size menu
New operation file size and maximum number of file
Format disk
‘Y’ to confirm to format disk

```
+++ Change FileSize selected +++
System is now operating at FileSize = 32[MB]
Maximum number of file is 15258 File
**********WARNING**********
Changing FileSize may damage the file system
Are you sure you want to change FileSize [Y/N] => y
--- Enter FileSize ---
[0] : 32   MB
[1] : 64   MB
[2] : 128  MB
[3] : 256  MB
[4] : 512  MB
[5] : 1024 MB
[6] : 2048 MB
=> 2
File size changed
System is now operating at FileSize = 128[MB]
Maximum number of file is 3814 File
Please format disk before use
Press 'y' to confirm format disk => y
Format Disk complete...
--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

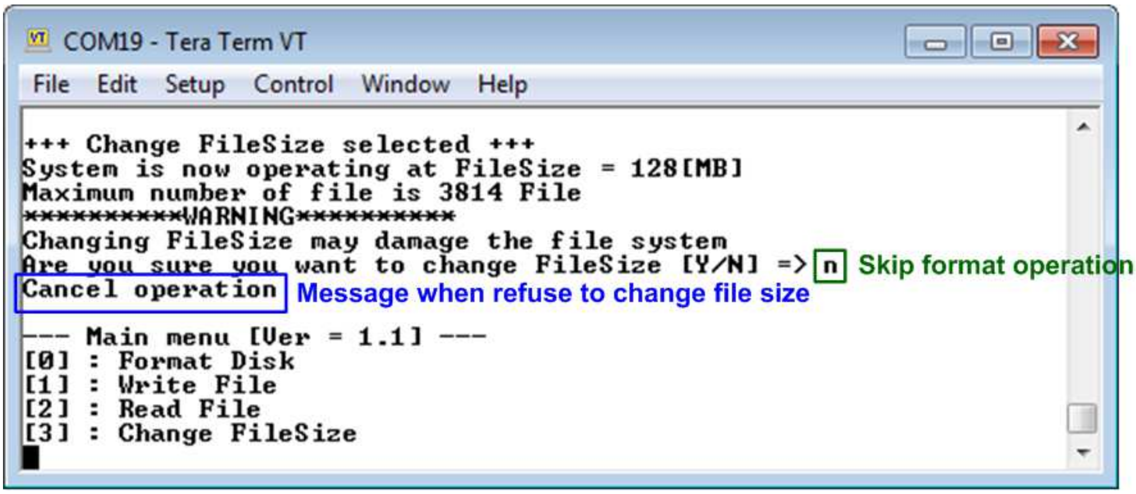
Figure 3-12 Result from Change FileSize command

After confirming to change file size, the selecting file size menu is displayed. There are six file sizes to select, i.e. 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB, and 2 GB. After setting new file size, updated file size and maximum number of file are displayed on the console.

After that, a request to format disk message is asked. User can input ‘y’ or ‘Y’ to accept format disk operation or input others to refuse.

Note: Changing file size may damage file system. Therefore, user should format disk after complete file size changing menu to delete all files in the device.

The example to refuse changing file size by input others (not ‘y’) is shown in Figure 3-13.



```
+++ Change FileSize selected +++
System is now operating at FileSize = 128[MB]
Maximum number of file is 3814 File
**********WARNING**********
Changing FileSize may damage the file system
Are you sure you want to change FileSize [Y/N] => n
Cancel operation

--- Main menu [Ver = 1.1] ---
[0] : Format Disk
[1] : Write File
[2] : Read File
[3] : Change FileSize
```

Skip format operation

Message when refuse to change file size

Figure 3-13 Cancel Change FileSize command

4 Revision History

Revision	Date	Description
1.0	9-Nov-17	Initial version release
1.1	25-May-18	Update firmware version and description