NVMe-IP for PLDA PCIe Demo Instruction

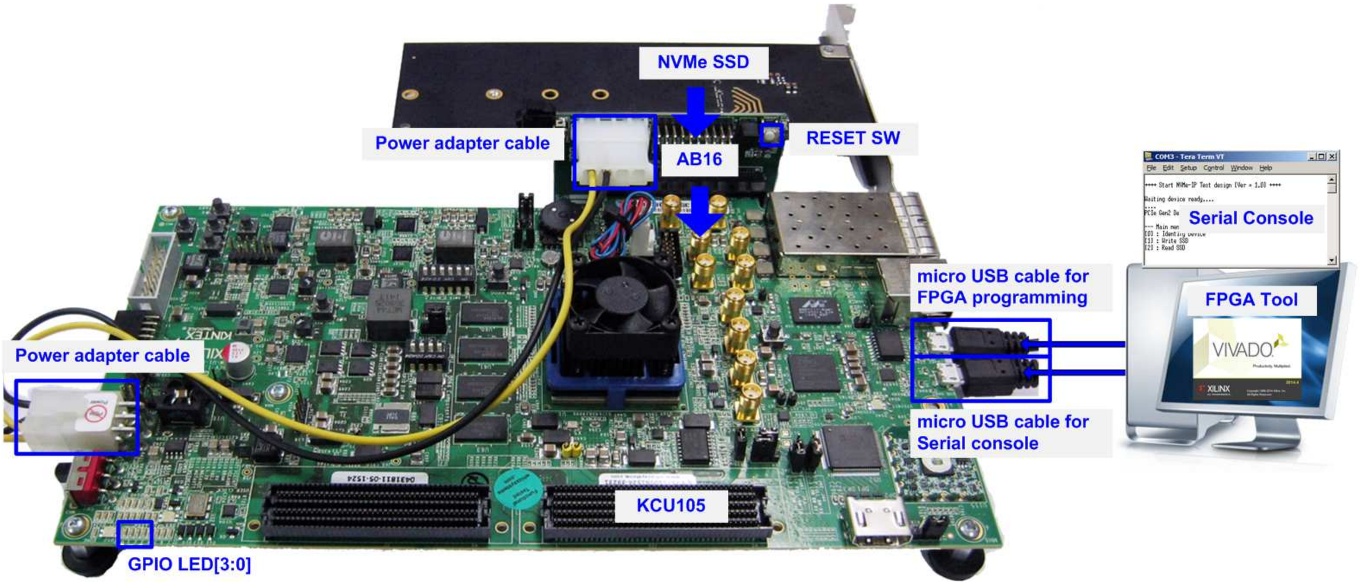
Rev1.0 6-Feb-18

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

1 Environment Requirement

To demo NVMe-IP for PLDA PCIe on FPGA development board, please prepare following hardware/ software.

1) Supported FPGA Development board: KCU105
2) PC with Xilinx programmer software (Vivado) and Serial console software
3) AB16-PCIeXOVR board + power adapter cable from AB16 delivery set
4) NVMe SSD connecting to PCIe Female connector on AB16
5) micro USB cable for programming FPGA, connecting between FPGA board and PC
6) 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 De

--- Main men
[0] : Identify device
[1] : Write SSD
[2] : Read SSD
```

Figure 1-1 NVMe-IP Demo Environment Setup on KCU105

2 Demo setup

1) Power off system.
2) Connect power adapter cable from AB16-PCIeXOVR delivery set to power connector on FPGA board, on AB16-PCIeXOVR board, and on Xilinx power adapter as shown in Figure 2-1.

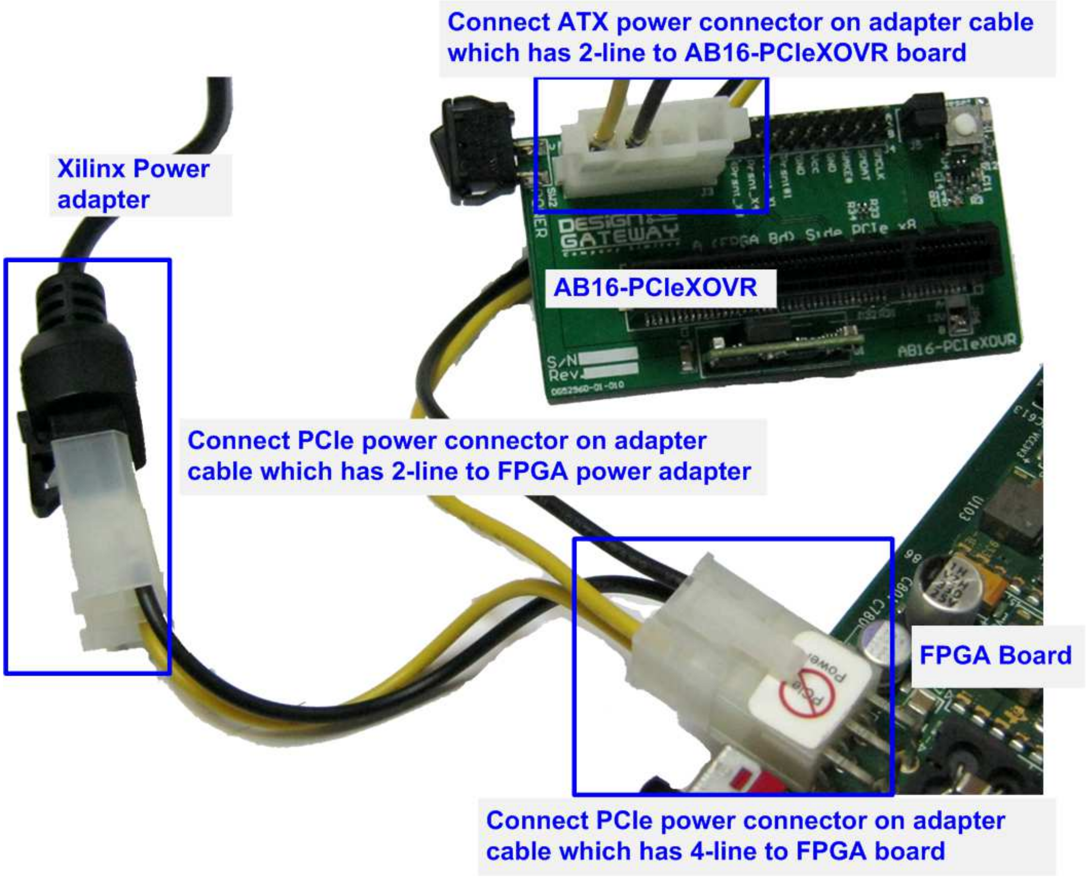

Connect ATX power connector on adapter cable which has 2-line to AB16-PCIeXOVR board
Xilinx Power adapter
AB16-PCIeXOVR
Connect PCIe power connector on adapter cable which has 2-line to FPGA power adapter
FPGA Board
Connect PCIe power connector on adapter cable which has 4-line to FPGA board

Figure 2-1 Connect power adapter cable to FPGA board, AB16, and Xilinx adapter

3) Connect PCIe connector on A Side of AB16-PCIeXOVR board to PCIe connector on Xilinx development board, as shown in Figure 2-2. Also, check that two mini jumpers are inserted at J5 connector on AB16.

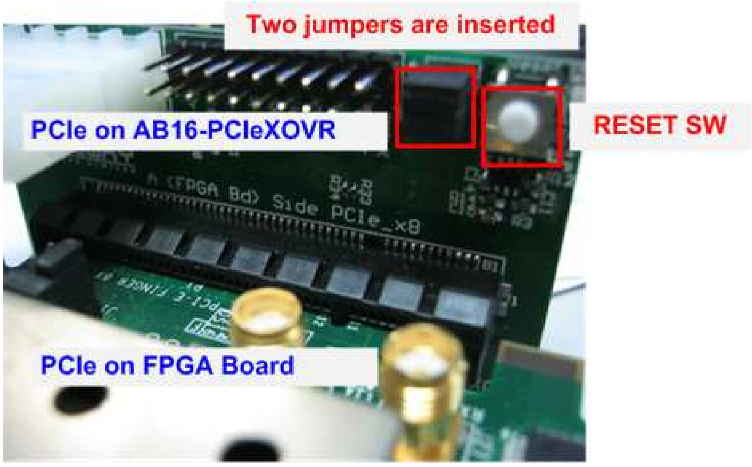


Figure 2-2 Connect PCIe connector between AB16 and FPGA board

4) Connect NVMe SSD to PCIe connector on B Side of AB16-PCIeXOVR board, as shown in Figure 2-3.

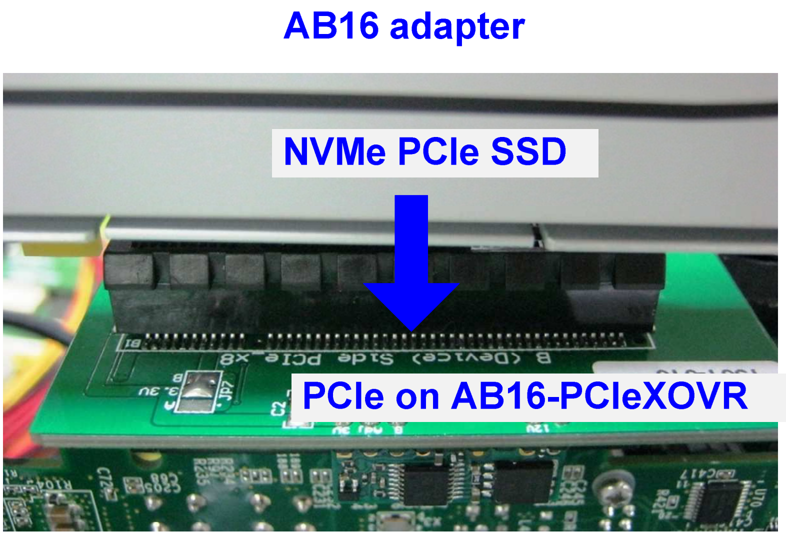


Figure 2-3 Connect NVMe SSD to AB16 board

5) Connect two micro USB cables for JTAG programming and Serial console.

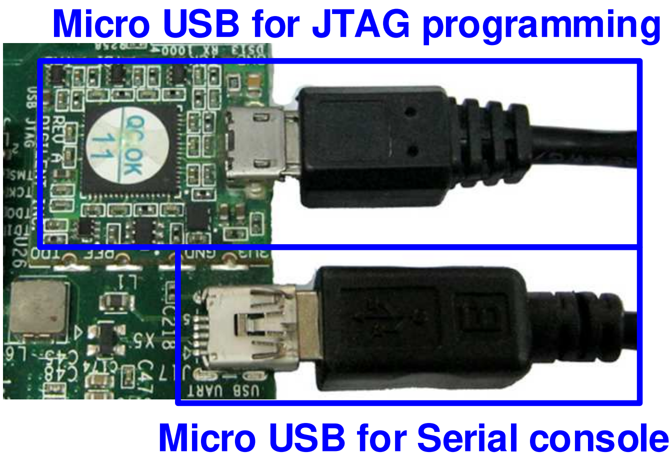


Figure 2-4 USB cable connection

6) Power on AB16-PCIeXOVR board and then power on FPGA development board, as shown in Figure 2-5.

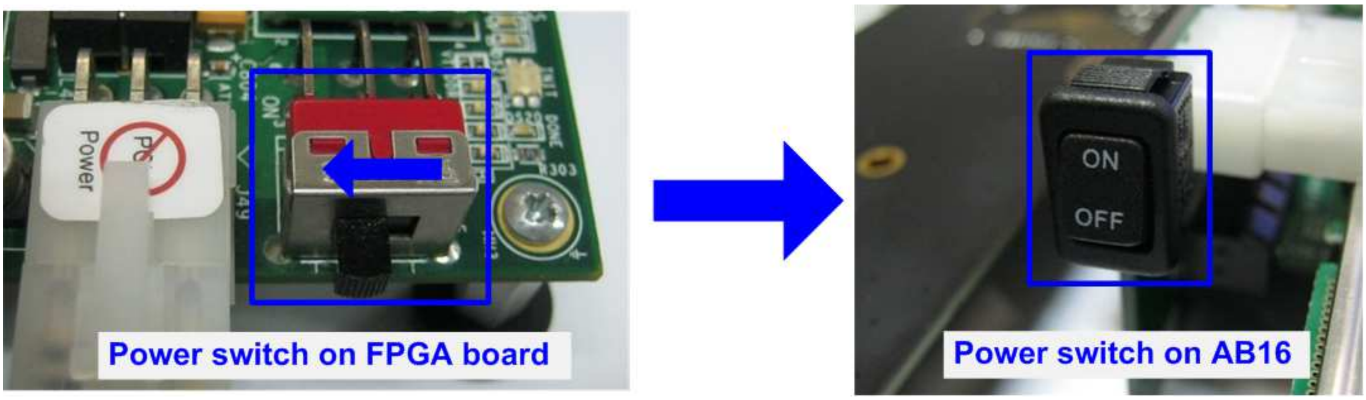
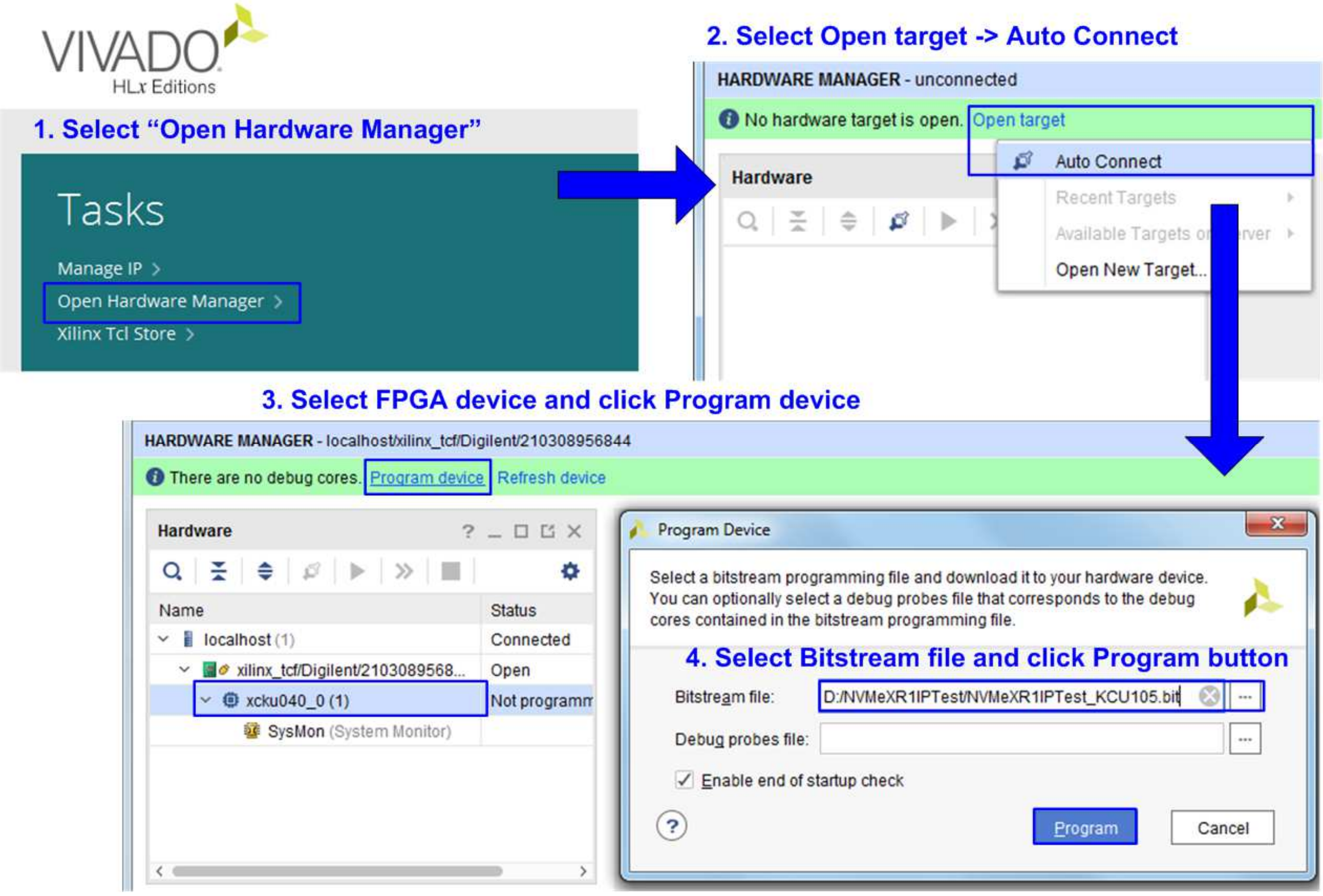


Figure 2-5 Power switch on AB16 board

7) Open Serial console such as TeraTerm, HyperTerminal. Set Buad rate=115,200 Data=8 bit Non-Parity Stop=1.
8) Download bit file to configure FPGA and firmware. Use Vivado tool to program bit file, as shown in Figure 2-6.



1. Select "Open Hardware Manager"

2. Select Open target -> Auto Connect

HARDWARE MANAGER - unconnected

No hardware target is open. Open target

3. Select FPGA device and click Program device

HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210308956844

There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/2103089568...	Open
xcku040_0 (1)	Not programm
SysMon (System Monitor)	

Program Device

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.

4. Select Bitstream file and click Program button

Bitstream file: D:/NVMeXR1IPTest/NVMeXR1IPTest_KCU105.bit

Debug probes file:

☑ Enable end of startup check

Figure 2-6 Programmed by Vivado

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

GPIO LED	ON	OFF
0	Normal operation	Clock is not locked or reset button is pressed
1	System is busy	Idle status
2	IP Error detect	Normal operation
3	Data verification fail	Normal operation

Table 2-1 LED Definition

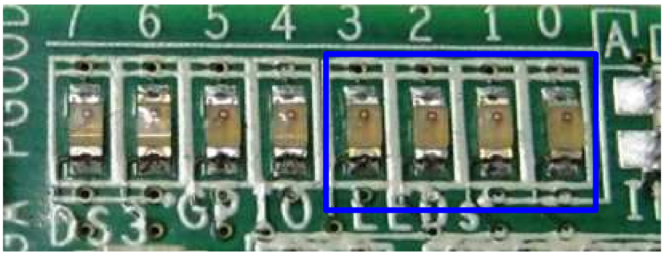


Figure 2-7 4-bit LED Status for user output

10) After programming completely, LED[0] and LED[1] are ON during PCIe initialization process. Then, LED[1] is OFF to show that PCIe completes initialization process and test system is ready to receive command from user. Main menu and PCIe speed are displayed on the console, as shown in Figure 2-8.

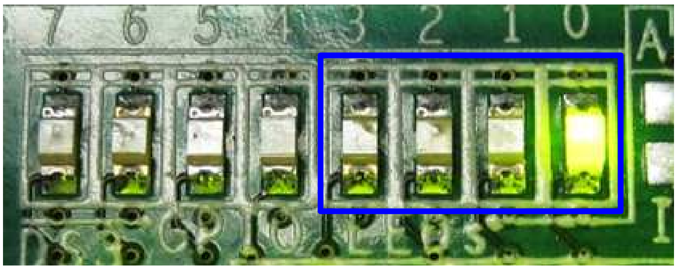
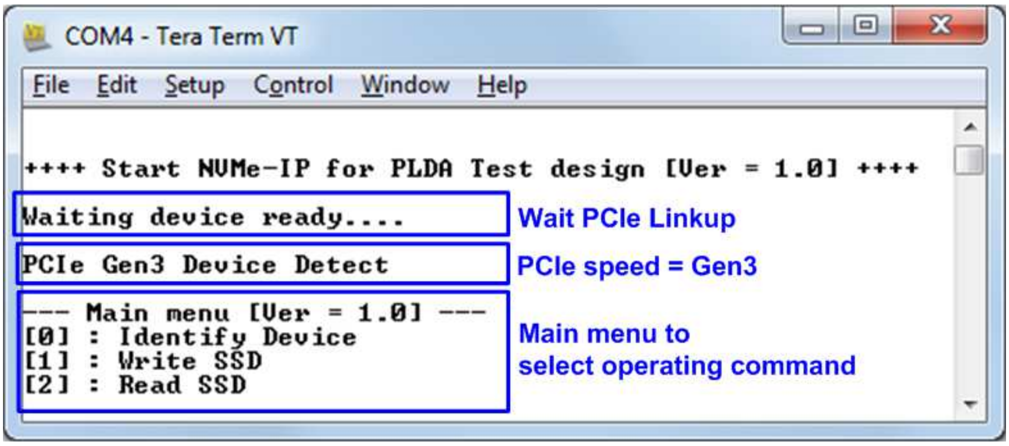


Figure 2-8 LED status after program configuration file and PCIe initialization complete



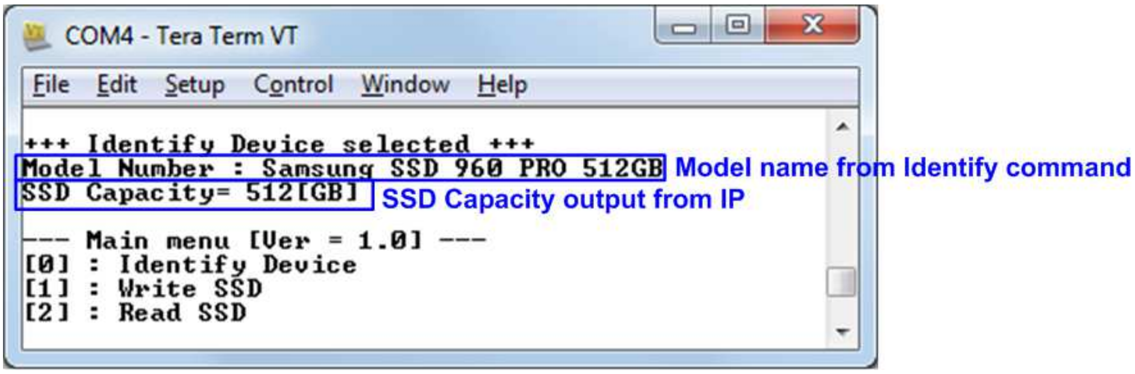
```
++++ Start NVMe-IP for PLDA Test design [Ver = 1.0] ++++
Waiting device ready....
PCIe Gen3 Device Detect
--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Figure 2-9 Main menu after program configuration file and PCIe initialization complete

3 Test Menu

3.1 Identify Device

Select '0' to send Identify command to NVMe SSD. When operation is completed, SSD capacity and model name are displayed on the console.


COM4 - Tera Term VT
File Edit Setup Control Window Help

```
+++ Identify Device selected +++
Model Number : Samsung SSD 960 PRO 512GB
SSD Capacity= 512[GB]

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

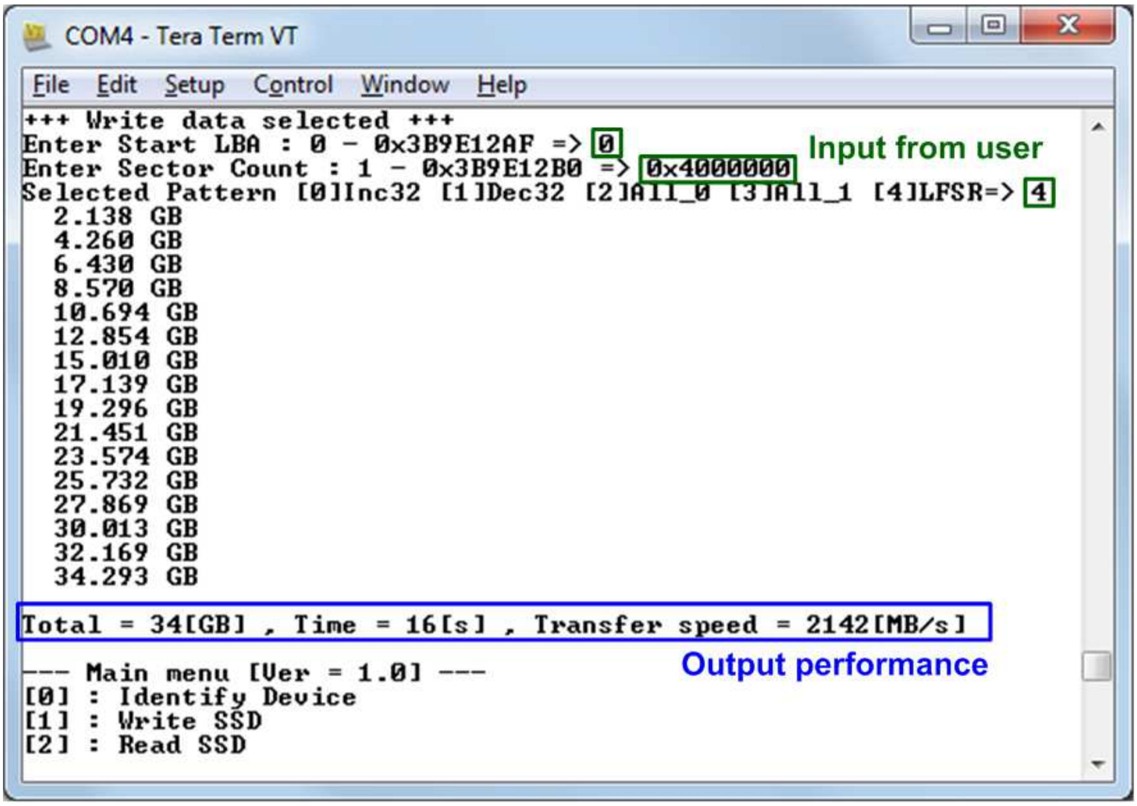
Model name from Identify command
SSD Capacity output from IP

Figure 3-1 Result from Identify Device menu

3.2 Write SSD

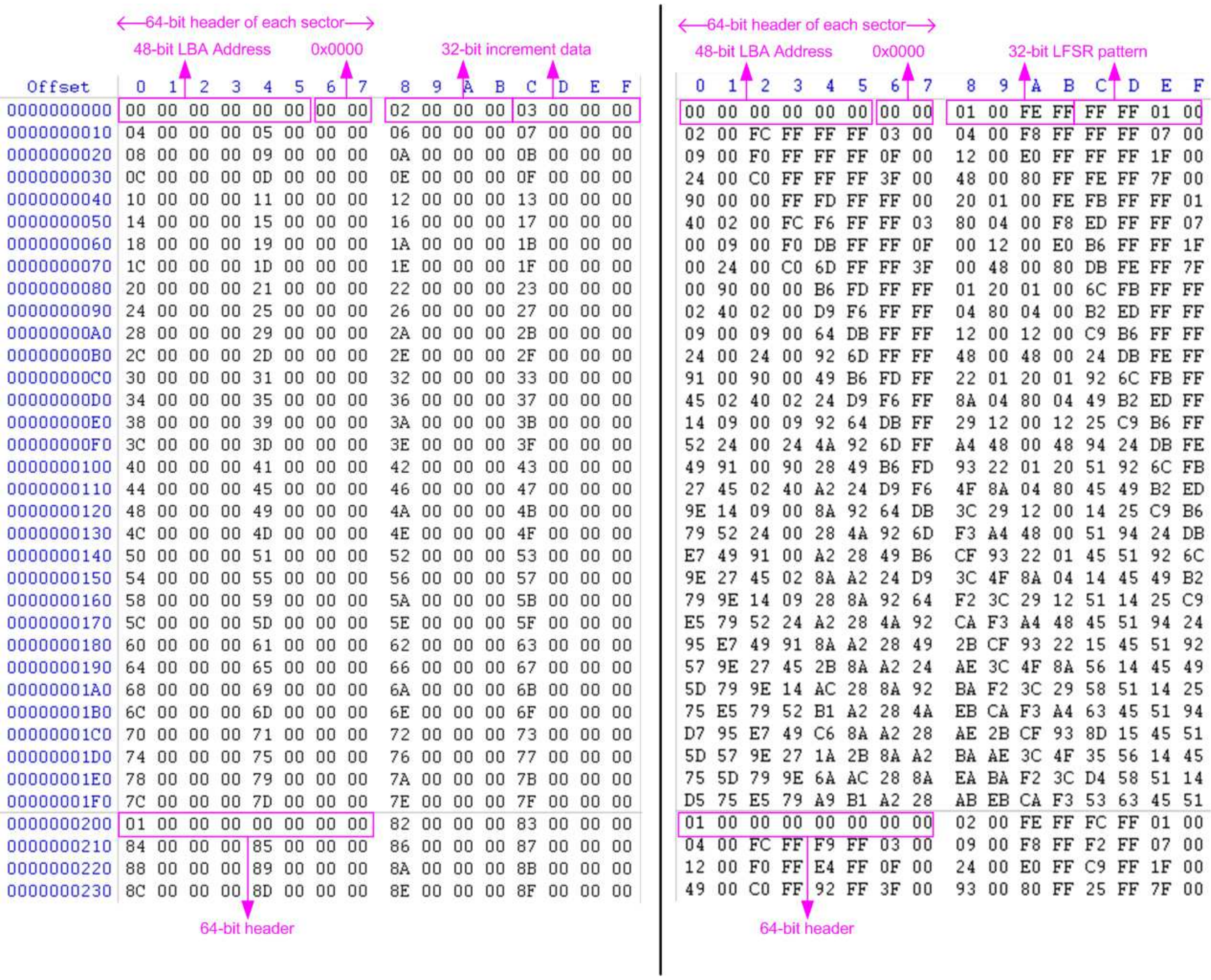
Select ‘1’ to send Write command to NVMe SSD. Three inputs are required for this menu.
1) Start LBA: Input start address of SSD in sector unit. The input can be decimal unit or add “0x” as a prefix for hexadecimal unit.
2) Sector Count: Input total transfer size in sector unit. The input can be decimal unit or add “0x” as a prefix for hexadecimal unit.
3) Test pattern: Select test pattern of test data for writing to SSD. Five patterns can be selected, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR counter.

As shown in Figure 3-2, if all inputs are valid, the operation will be started. During writing data, current transfer size is displayed on the console to show that system still be alive. Finally, test performance, total size, and total time usage are displayed on the console as test result.



```
+++ Write data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0
Enter Sector Count : 1 - 0x3B9E12B0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
   2.138 GB
   4.260 GB
   6.430 GB
   8.570 GB
   10.694 GB
   12.854 GB
   15.010 GB
   17.139 GB
   19.296 GB
   21.451 GB
   23.574 GB
   25.732 GB
   27.869 GB
   30.013 GB
   32.169 GB
   34.293 GB

Total = 34[GB] , Time = 16[s] , Transfer speed = 2142[MB/s]

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Figure 3-2 Input and result of Write SSD menu



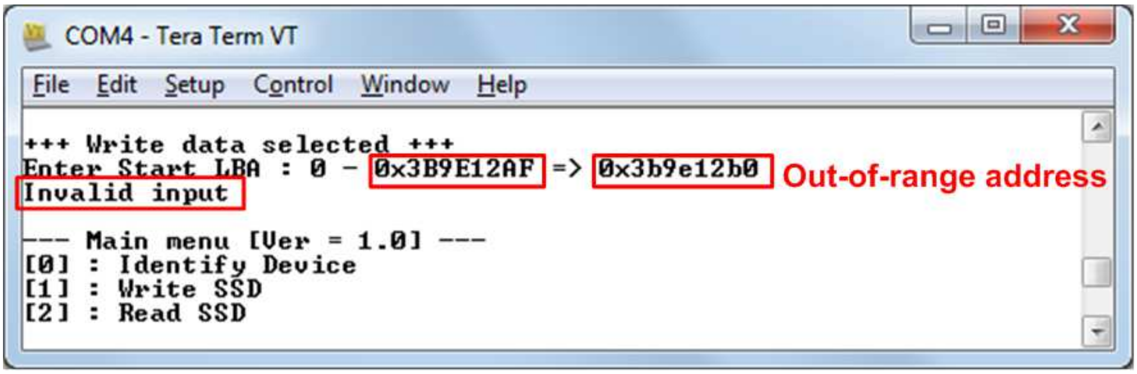
```
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0000000000  00 00 00 00 00 00 00 00  02 00 00 00 03 00 00 00
0000000010  04 00 00 00 05 00 00 00  06 00 00 00 07 00 00 00
0000000020  08 00 00 00 09 00 00 00  0A 00 00 00 0B 00 00 00
0000000030  0C 00 00 00 0D 00 00 00  0E 00 00 00 0F 00 00 00
0000000040  10 00 00 00 11 00 00 00  12 00 00 00 13 00 00 00
0000000050  14 00 00 00 15 00 00 00  16 00 00 00 17 00 00 00
0000000060  18 00 00 00 19 00 00 00  1A 00 00 00 1B 00 00 00
0000000070  1C 00 00 00 1D 00 00 00  1E 00 00 00 1F 00 00 00
0000000080  20 00 00 00 21 00 00 00  22 00 00 00 23 00 00 00
0000000090  24 00 00 00 25 00 00 00  26 00 00 00 27 00 00 00
00000000A0  28 00 00 00 29 00 00 00  2A 00 00 00 2B 00 00 00
00000000B0  2C 00 00 00 2D 00 00 00  2E 00 00 00 2F 00 00 00
00000000C0  30 00 00 00 31 00 00 00  32 00 00 00 33 00 00 00
00000000D0  34 00 00 00 35 00 00 00  36 00 00 00 37 00 00 00
00000000E0  38 00 00 00 39 00 00 00  3A 00 00 00 3B 00 00 00
00000000F0  3C 00 00 00 3D 00 00 00  3E 00 00 00 3F 00 00 00
0000000100  40 00 00 00 41 00 00 00  42 00 00 00 43 00 00 00
0000000110  44 00 00 00 45 00 00 00  46 00 00 00 47 00 00 00
0000000120  48 00 00 00 49 00 00 00  4A 00 00 00 4B 00 00 00
0000000130  4C 00 00 00 4D 00 00 00  4E 00 00 00 4F 00 00 00
0000000140  50 00 00 00 51 00 00 00  52 00 00 00 53 00 00 00
0000000150  54 00 00 00 55 00 00 00  56 00 00 00 57 00 00 00
0000000160  58 00 00 00 59 00 00 00  5A 00 00 00 5B 00 00 00
0000000170  5C 00 00 00 5D 00 00 00  5E 00 00 00 5F 00 00 00
0000000180  60 00 00 00 61 00 00 00  62 00 00 00 63 00 00 00
0000000190  64 00 00 00 65 00 00 00  66 00 00 00 67 00 00 00
00000001A0  68 00 00 00 69 00 00 00  6A 00 00 00 6B 00 00 00
00000001B0  6C 00 00 00 6D 00 00 00  6E 00 00 00 6F 00 00 00
00000001C0  70 00 00 00 71 00 00 00  72 00 00 00 73 00 00 00
00000001D0  74 00 00 00 75 00 00 00  76 00 00 00 77 00 00 00
00000001E0  78 00 00 00 79 00 00 00  7A 00 00 00 7B 00 00 00
00000001F0  7C 00 00 00 7D 00 00 00  7E 00 00 00 7F 00 00 00
0000000200  01 00 00 00 00 00 00 00  82 00 00 00 83 00 00 00
0000000210  84 00 00 00 85 00 00 00  86 00 00 00 87 00 00 00
0000000220  88 00 00 00 89 00 00 00  8A 00 00 00 8B 00 00 00
0000000230  8C 00 00 00 8D 00 00 00  8E 00 00 00 8F 00 00 00
```

```
 0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00 00 00 00 00 00 00 00  01 00 FE FF FF FF 01 00
02 00 FC FF FF FF 03 00  04 00 F8 FF FF FF 07 00
09 00 F0 FF FF FF 0F 00  12 00 E0 FF FF FF 1F 00
24 00 C0 FF FF FF 3F 00  48 00 80 FF FE FF 7F 00
90 00 00 FF FD FF FF 00  20 01 00 FE FB FF FF 01
40 02 00 FC F6 FF FF 03  80 04 00 F8 ED FF FF 07
00 09 00 F0 DB FF FF 0F  00 12 00 E0 B6 FF FF 1F
00 24 00 C0 6D FF FF 3F  00 48 00 80 DB FE FF 7F
00 90 00 00 B6 FD FF FF  01 20 01 00 6C FB FF FF
02 40 02 00 D9 F6 FF FF  04 80 04 00 B2 ED FF FF
09 00 09 00 64 DB FF FF  12 00 12 00 C9 B6 FF FF
24 00 24 00 92 6D FF FF  48 00 48 00 24 DB FE FF
91 00 90 00 49 B6 FD FF  22 01 20 01 92 6C FB FF
45 02 40 02 24 D9 F6 FF  8A 04 80 04 49 B2 ED FF
14 09 00 09 92 64 DB FF  29 12 00 12 25 C9 B6 FF
52 24 00 24 4A 92 6D FF  A4 48 00 48 94 24 DB FE
49 91 00 90 28 49 B6 FD  93 22 01 20 51 92 6C FB
27 45 02 40 A2 24 D9 F6  4F 8A 04 80 45 49 B2 ED
9E 14 09 00 8A 92 64 DB  3C 29 12 00 14 25 C9 B6
79 52 24 00 28 4A 92 6D  F3 A4 48 00 51 94 24 DB
E7 49 91 00 A2 28 49 B6  CF 93 22 01 45 51 92 6C
9E 27 45 02 8A A2 24 D9  3C 4F 8A 04 14 45 49 B2
79 9E 14 09 28 8A 92 64  F2 3C 29 12 51 14 25 C9
E5 79 52 24 A2 28 4A 92  CA F3 A4 48 45 51 94 24
95 E7 49 91 8A A2 28 49  2B CF 93 22 15 45 51 92
57 9E 27 45 2B 8A A2 24  AE 3C 4F 8A 56 14 45 49
5D 79 9E 14 AC 28 8A 92  BA F2 3C 29 58 51 14 25
75 E5 79 52 B1 A2 28 4A  EB CA F3 A4 63 45 51 94
D7 95 E7 49 C6 8A A2 28  AE 2B CF 93 8D 15 45 51
5D 57 9E 27 1A 2B 8A A2  BA AE 3C 4F 35 56 14 45
75 5D 79 9E 6A AC 28 8A  EA BA F2 3C D4 58 51 14
D5 75 E5 79 A9 B1 A2 28  AB EB CA F3 53 63 45 51
01 00 00 00 00 00 00 00  02 00 FE FF FC FF 01 00
04 00 FC FF F9 FF 03 00  09 00 F8 FF F2 FF 07 00
12 00 F0 FF E4 FF 0F 00  24 00 E0 FF C9 FF 1F 00
49 00 C0 FF 92 FF 3F 00  93 00 80 FF 25 FF 7F 00
```

Figure 3-3 Example Test data in sector#0/#1 by increment/LFSR pattern

Test data of each sector has different 64-bit header which consists of 48-bit LBA address and 16-bit all 0 value. 48-bit LBA address is unique value for each sector. After that, the test pattern is filled following user selection such as 32-bit increment pattern (left window of Figure 3-3), 32-bit LFSR pattern (right window of Figure 3-3).

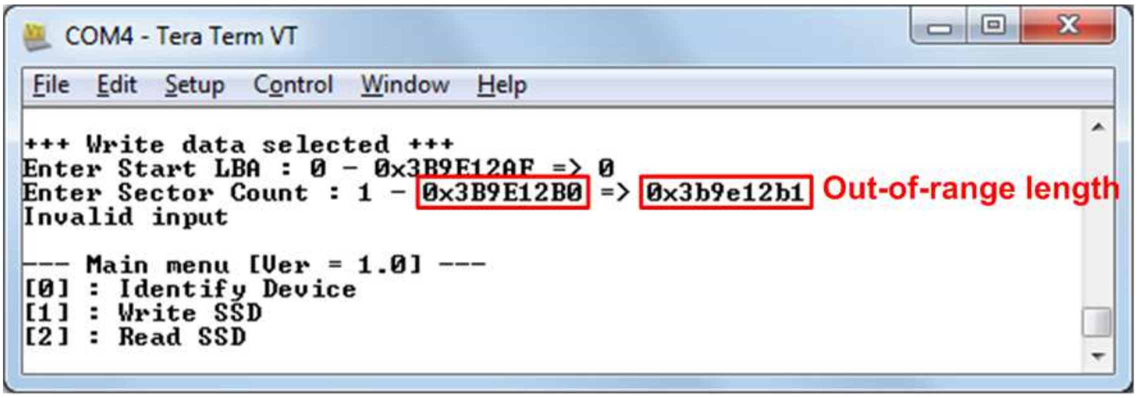
Figure 3-4 – Figure 3-6 show the error message when user input is invalid. “Invalid input” message are displayed on the console. Then, it returns to main menu to receive new command.



```
+++ Write data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0x3b9e12b0
Invalid input

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Out-of-range address

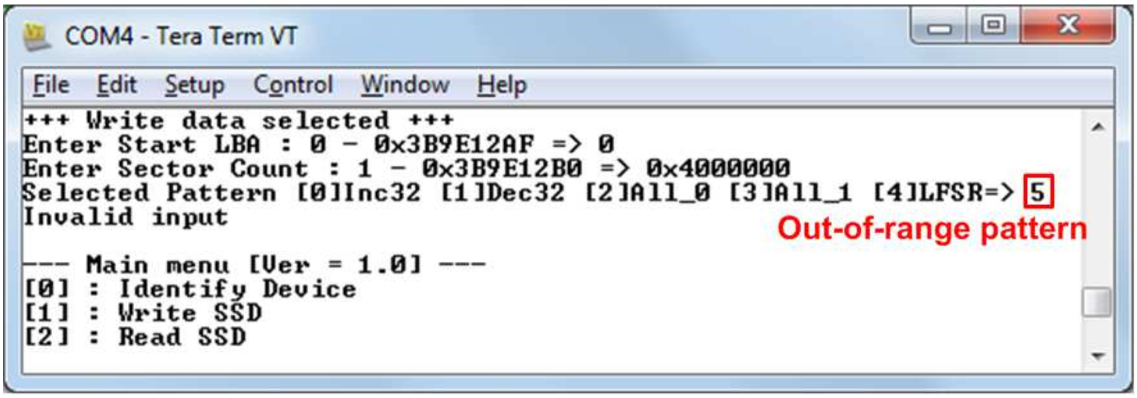
Figure 3-4 Invalid Start LBA input



```
+++ Write data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0
Enter Sector Count : 1 - 0x3B9E12B0 => 0x3b9e12b1
Invalid input

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Out-of-range length

Figure 3-5 Invalid Sector count input



```
+++ Write data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0
Enter Sector Count : 1 - 0x3B9E12B0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 5
Invalid input

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

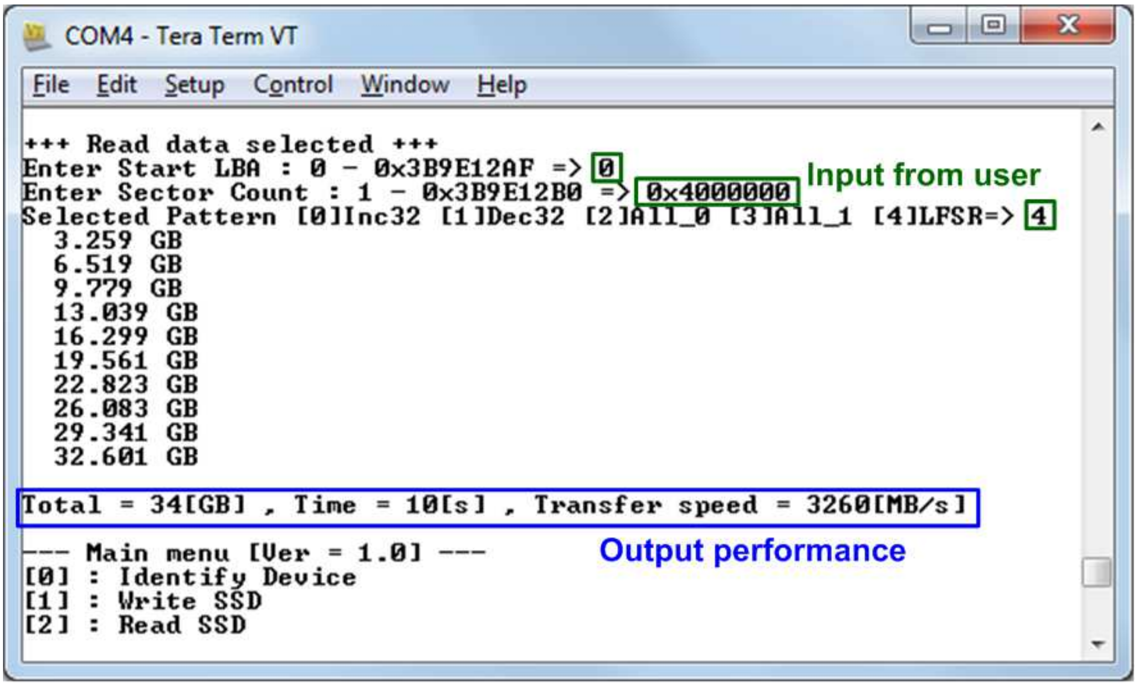
Out-of-range pattern

Figure 3-6 Invalid Test pattern input

3.3 Read SSD

Select ‘2’ to send Read command to NVMe SSD. Three inputs are required for this menu.
1) Start LBA: Input start address of SSD in sector unit. The input can be decimal unit or add “0x” as a prefix for hexadecimal unit.
2) Sector Count: Input total transfer size in sector unit. The input can be decimal unit or add “0x” as a prefix for hexadecimal unit.
3) Test pattern: Select test pattern to verify data from SSD. Test pattern must be matched with the test pattern which is used during write test. Five types can be selected, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR counter.

Similar to write test, if all inputs are valid, test system will read data from SSD. Test performance, total size, and total time usage are displayed after end of transfer. “Invalid input” will be displayed if some inputs are out-of-range.

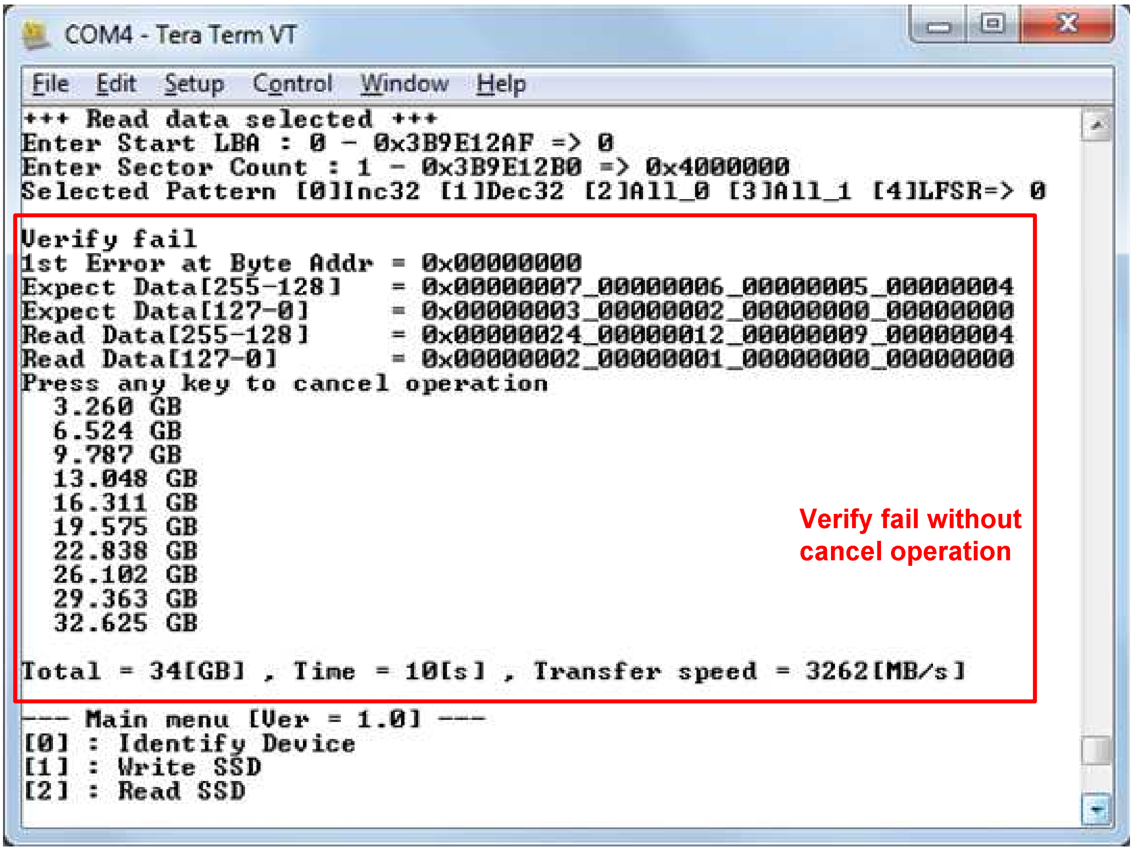


```
+++ Read data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0
Enter Sector Count : 1 - 0x3B9E12B0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
  3.259 GB
  6.519 GB
  9.779 GB
  13.039 GB
  16.299 GB
  19.561 GB
  22.823 GB
  26.083 GB
  29.341 GB
  32.601 GB

Total = 34[GB] , Time = 10[s] , Transfer speed = 3260[MB/s]

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Figure 3-7 Input and result of Read SSD menu

Figure 3-8 and Figure 3-9 show the error message when data verification is failed. “Verify fail” message is displayed with error address, expected data, and read data. User inputs any keys to cancel read operation or wait until all read process complete.

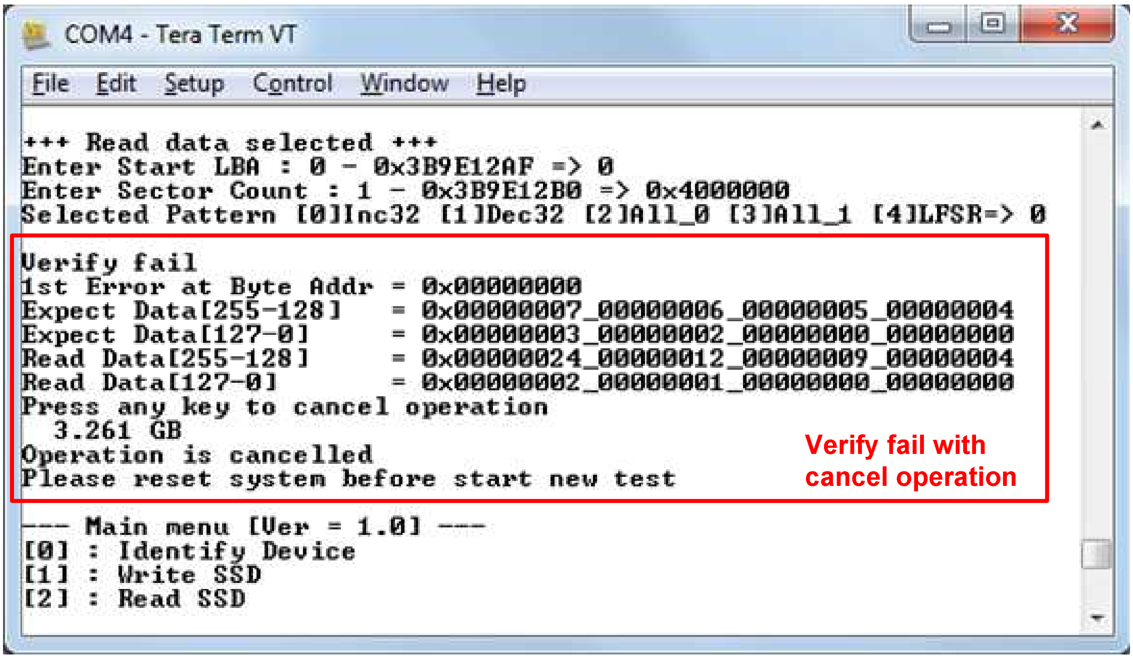
“RESET” button must be pressed to restart the system when user cancels the operation.



```
+++ Read data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0
Enter Sector Count : 1 - 0x3B9E12B0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 0

Verify fail
1st Error at Byte Addr  = 0x00000000
Expect Data[255-128]    = 0x00000007_00000006_00000005_00000004
Expect Data[127-0]      = 0x00000003_00000002_00000000_00000000
Read Data[255-128]      = 0x00000024_00000012_00000009_00000004
Read Data[127-0]        = 0x00000002_00000001_00000000_00000000
Press any key to cancel operation
  3.260 GB
  6.524 GB
  9.787 GB
  13.048 GB
  16.311 GB
  19.575 GB
  22.838 GB
  26.102 GB
  29.363 GB
  32.625 GB

Total = 34[GB] , Time = 10[s] , Transfer speed = 3262[MB/s]

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Verify fail without cancel operation

Figure 3-8 Data verification is failed, but wait until read complete



```
+++ Read data selected +++
Enter Start LBA : 0 - 0x3B9E12AF => 0
Enter Sector Count : 1 - 0x3B9E12B0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 0

Verify fail
1st Error at Byte Addr  = 0x00000000
Expect Data[255-128]    = 0x00000007_00000006_00000005_00000004
Expect Data[127-0]      = 0x00000003_00000002_00000000_00000000
Read Data[255-128]      = 0x00000024_00000012_00000009_00000004
Read Data[127-0]        = 0x00000002_00000001_00000000_00000000
Press any key to cancel operation
  3.261 GB
Operation is cancelled
Please reset system before start new test

--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
```

Verify fail with cancel operation

Figure 3-9 Data verification is failed, and input any keys to cancel operation

4 Revision History

Revision	Date	Description
1.0	6-Feb-18	Initial version release