A Step-by-step Guideline for Creating A Testbed for Flash Memory Research via LPC-H3131 and OpenNFM

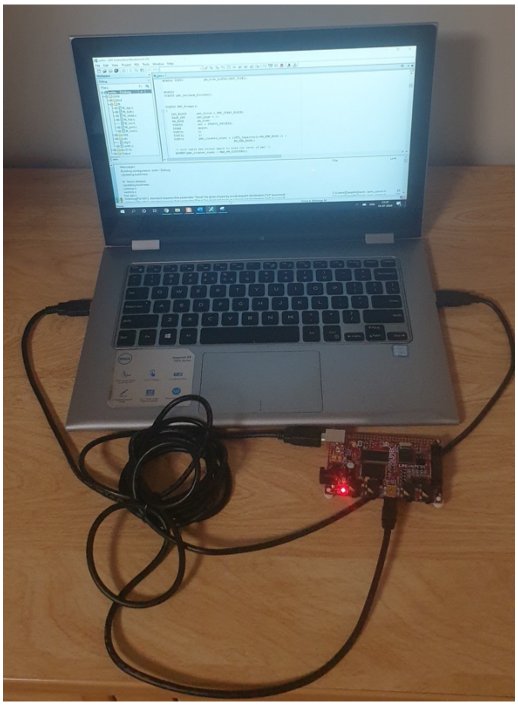
Deepthi Tankasala, Niusen Chen, Bo Chen
Department of Computer Science, Michigan Technological University
{dtankasa, niusenc, bchen}@mtu.edu

July 2020

Abstract. The Security and Privacy (SnP) Lab at Michigan Technological University has conducted extensive research on flash memory security, and most of our research relied on a flash memory testbed, built using a cheap electronic development and open-sourced flash controller. In this technical report, we provide a step-by-step guideline on how to set up this testbed. This guideline would be beneficial to any students and researchers who want to set up a cheap testbed for flash memory research.

I. Introduction

A testbed with open-source flash controller is usually expensive and unaffordable by college students as well as researchers who are interested in flash memory research. The SnP lab [1] at Michigan Technological University has been conducting flash memory research extensively recently [2-9]. Our research on flash memory security has been supported by national science foundation [10,11]. To benefit students as well as researchers in this area, we have prepared a step-by-step guideline which can help easily set up our testbed using some cheap electronic develop boards and open-sourced flash manager (or flash translation layer). Our purpose is to facilitate research in this growing area and to motivate more students and researchers in the world to contribute to this area. An overview of our testbed is shown in the below figure, which contains an electronic development board LPC-H3131, the open-source firmware OpenNFM which can be cross-compiled in IAR work bench in a Window machine and flashed to the board via Tera Term, and two USB cables which can easily be purchased online.



This report outlines a list of required hardware and software in Sec. II, the necessary system configuration and software installation in Sec III, and the detailed steps on how to cross-compile the open-source flash controller and flash it to the electronic development board in Sec. IV.

II. The Required Hardware and Software

a. Hardware:

1. LPC-H3131 USB HEADER DEVELOPMENT PROTOTYPE BOARD [link]
2. USB A to B cable
3. USB A to Mini Cable
4. A host computer equipped with USB ports

b. Software:

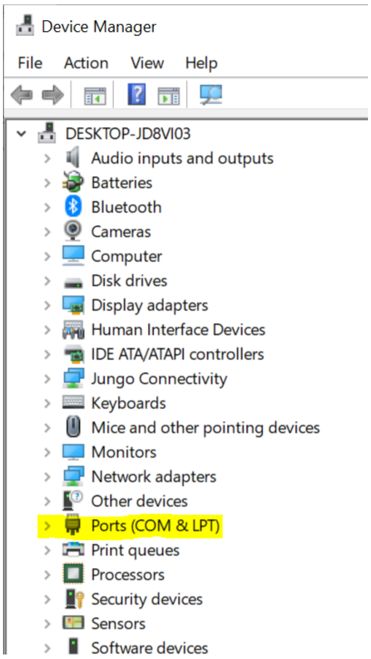
1. OpenNFM [link]: an open sourced flash memory controller
2. Tera Term [link]: a tool used to flash the flash memory controller (after compilation) to LPC-H3131. We used V4.105 in this technical report.
3. IAR Embedded workbench [link]: a cross compiler for OpenNFM. We used version 7.40.5.9739 in this technical report.
4. Microsoft Windows 7 operating system (for the host computer)

III. System Configuration and Software Installation

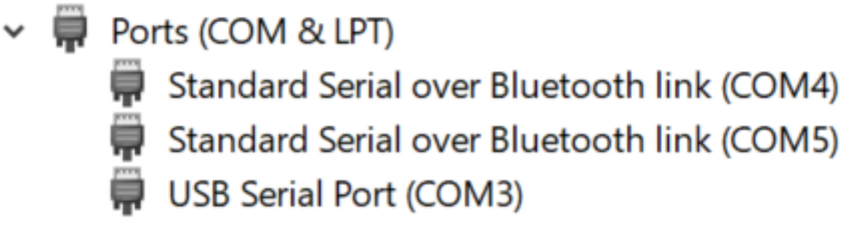
a. Tera Term

To install the Tera Term, we need to make few changes to the system settings in windows. Open Device Manager and follow the below steps:

1. Open Device Manager
2. Scroll down to find Port (COM&LPT) settings and click to expand the settings.



3. List of supported com ports are displayed



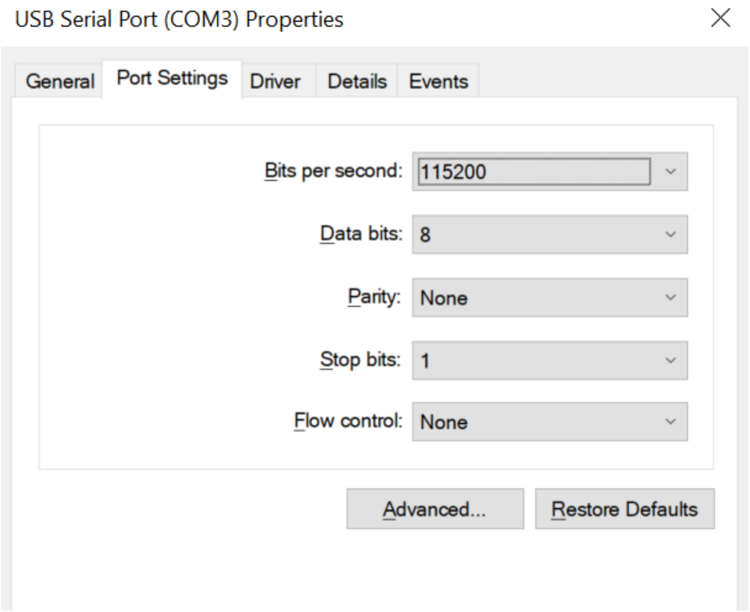
Ports (COM & LPT)

Standard Serial over Bluetooth link (COM4)

Standard Serial over Bluetooth link (COM5)

USB Serial Port (COM3)

4. Select USB Serial Port (COM3/COM1 depending on the OS version and individual system) and right click and select properties. Properties window is now open and select "Port Settings" and change "Bits per second" to 115200 and click ok.



USB Serial Port (COM3) Properties

Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

To install the Tera Term, follow the below instructions:

1. teraterm-4.105.exe file is installed by selecting "Run as Administrator"
2. Normal installation process is followed and no extra options need to be selected. Once the installation is completed, we need to setup the tera term to support the LPC board (Sec. IV-b).

b. IAR Workbench

IAR workbench is used for executing the OpenNFM Code and generating the Binary file required for getting the LPC board started.

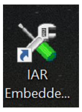
1. After having downloaded IAR workbench, follow the normal installation steps as directed by the installation. Note that free version of IAR workbench can only allow to compile a code size of 32K.
2. OpenNFM source code is added to the IAR Workbench and executed to generate "Binary File" that is used for setting up the test bed.

IV. Compiling OpenNFM and Flashing The Binary to LPC-H3131

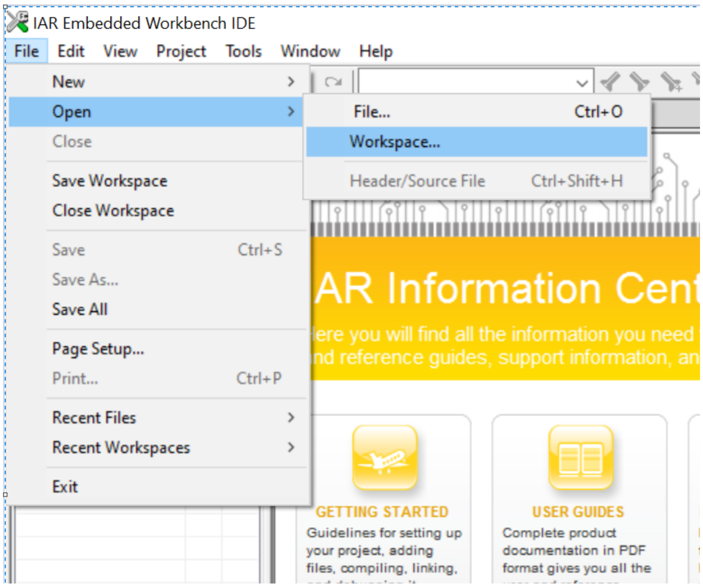
a. Cross-compiling OpenNFM

OpenNFM code base contains all the code required for creating our test-bed for NAND flash research. To compile OpenNFM, the project must be opened in IAR Workbench. Since we already have installed the IAR workbench, below are the steps for compiling and generating binary file required for our test bed.

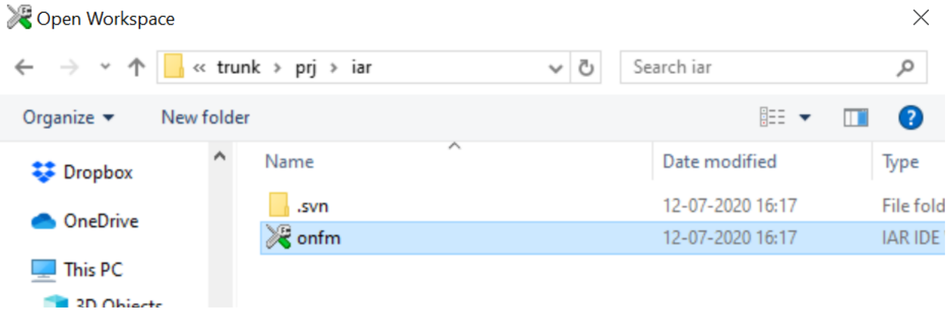
1. Open IAR Workbench by clicking on the application icon which looks like this.



2. Once the application is open, go to menu bar and click the file option to find "open" and select "Workspace" option.

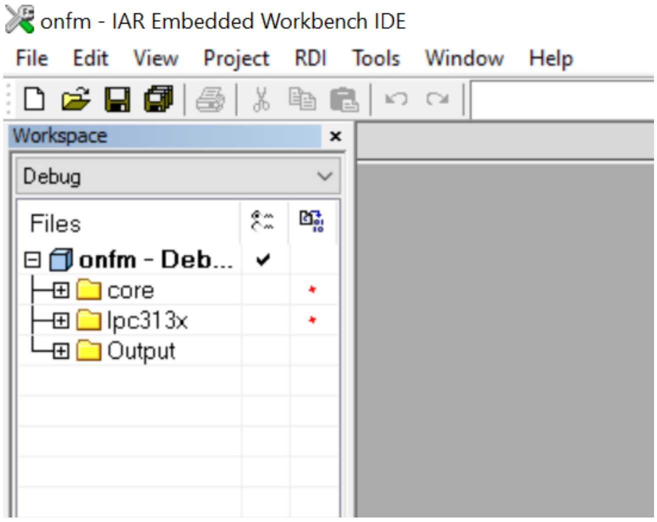

IAR Embedded Workbench IDE
File
Edit
View
Project
Tools
Window
Help
New
Open
Close
Save Workspace
Close Workspace
Save Ctrl+S
Save As...
Save All
Page Setup...
Print... Ctrl+P
Recent Files
Recent Workspaces
Exit
File... Ctrl+O
Workspace...
Header/Source File Ctrl+Shift+H
AR Information Cent
ere you will find all the information you need
nd reference guides, support information, an
GETTING STARTED
Guidelines for setting up your project, adding files, compiling, linking,
USER GUIDES
Complete product documentation in PDF format gives you all the

3. A window opens and we need to open IAR IDE Workspace type file from OpenNFM code. From the window, navigate to the OpenNFM code location and go to opennfm>>trunk>>prj>>iar and double click on: "onfm" an IAR IDE workbench file.

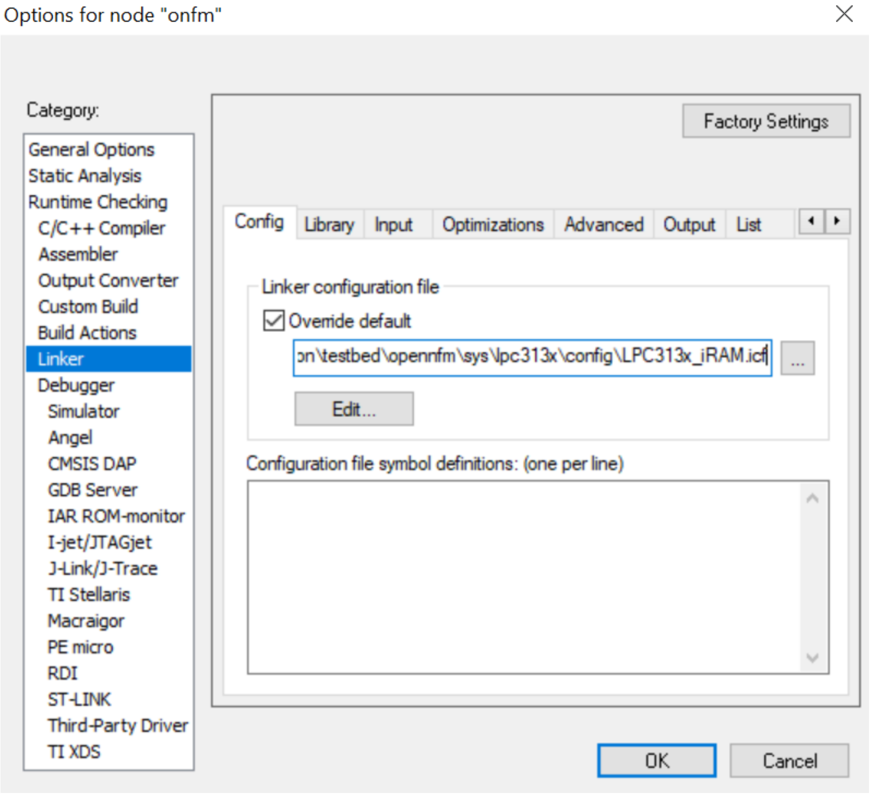


Name	Date modified	Type
.svn	12-07-2020 16:17	File fold
onfm	12-07-2020 16:17	IAR IDE

4. Now the OpenNFM is imported into IAR Workbench. And the all the workbench structure is displayed under Debug panel of IAR workbench as below.

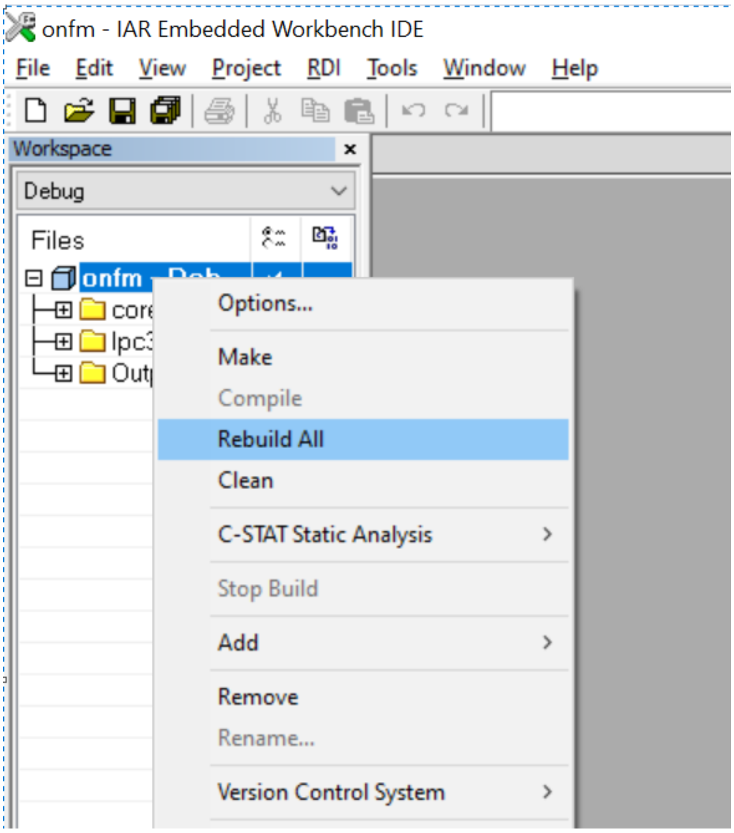

onfm - IAR Embedded Workbench IDE
File Edit View Project RDI Tools Window Help
Workspace
Debug
Files
onfm - Deb...
core
lpc313x
Output

5. We need to set the linker settings for this project. Go to Project option on the menu bar and click on linker option and under override default option, add the address location of “LPC313x_iRAM.icf” file present in the OpenNFM code folder **Opennfm>> sys>>lpc313x>>config>> LPC313x_iRAM.icf** and click ok.



Linker configuration file
☑ Override default
on\testbed\opennfm\sys\lpc313x\config\LPC313x_iRAM.icf
Configuration file symbol definitions: (one per line)

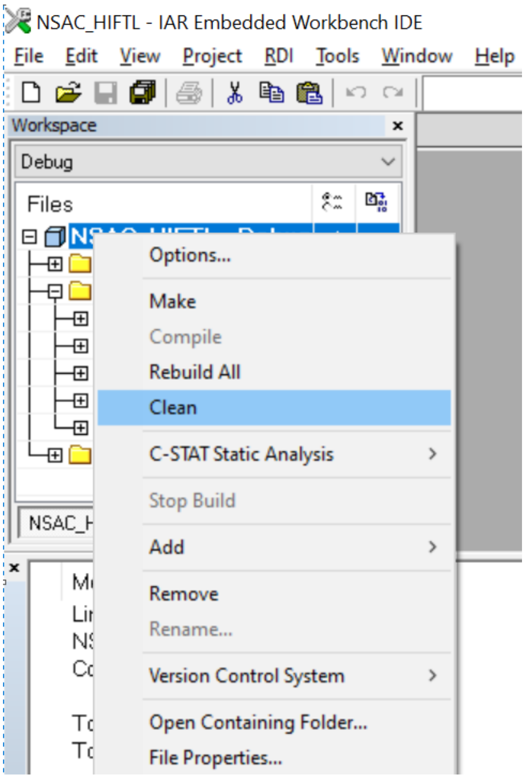
6. To make sure, all the files are present, you need to check if the code compiles by right clicking on the project present under the “Debug” panel and select “Rebuild All” option.



7. If all the files are intact, the project should compile successfully without errors.

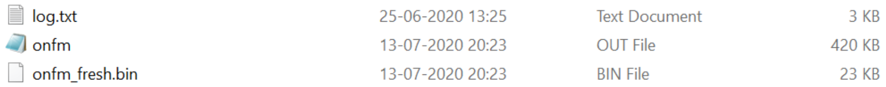
8. If any errors are present, the details of the error are displayed under the build panel of the IAR workbench.

9. To clear the error messages under the build panel and to rebuild the entire project, right click on the project and click on “clean” option



10. On successful compilation of the OpenNFM code, binary file with the name “onfm” will be generated in the Exe folder present in the below location **Opnfm>>prj>>iar>>Debug>>Exe**

11. To verify whether the binary file is latest or not, check the Date modified property of the opennfm (BIN File type). The time should match the time when OpenNFM project was rebuild from IAR Workbench.



log.txt	25-06-2020 13:25	Text Document	3 KB
onfm	13-07-2020 20:23	OUT File	420 KB
onfm_fresh.bin	13-07-2020 20:23	BIN File	23 KB

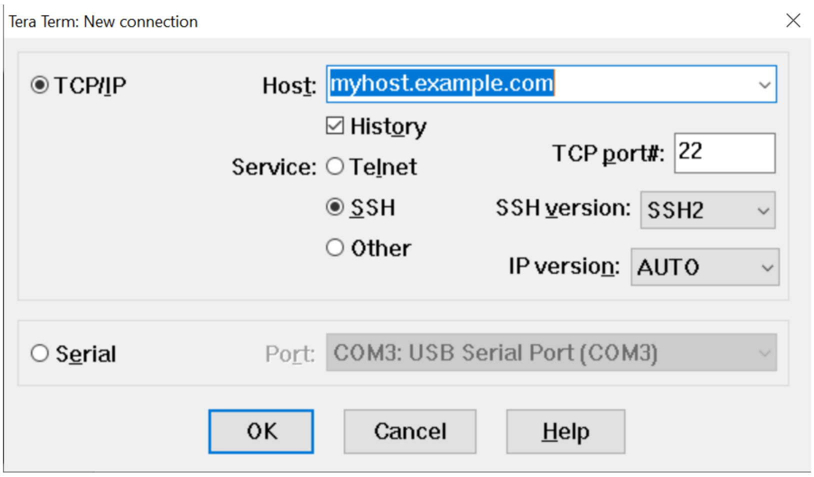
12. Whenever any changes are made to the files in the project folder, the entire code should be re-compiled and then newly generated “onfm” binary file should be uploaded in the Tera Term to see the updated changes in action on the OpenNFM test bed simulation.

b. Flashing The Binary of OpenNFM to LPC-H3131

For windows we are using Tera Term application for serial console and network file transfer. Below are the steps for installing and setting up Tera Term:

1. Open Tera Term app from the start menu of the windows 7 operating system.

2. A new connection terminal is displayed as below.



Tera Term: New connection

● TCP/IP Host: myhost.example.com

☑ History

Service: ○ Telnet ● SSH ○ Other

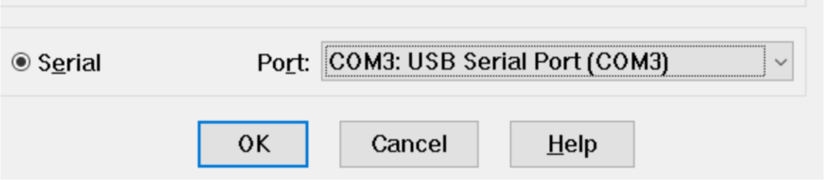
TCP port#: 22

SSH version: SSH2

IP version: AUTO

○ Serial Port: COM3: USB Serial Port (COM3)

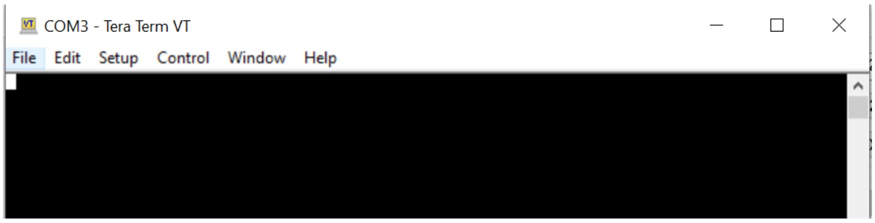
3. Select "serial" option and then choose COM3: USB Serial PORT option and click ok



● Serial Port: COM3: USB Serial Port (COM3)

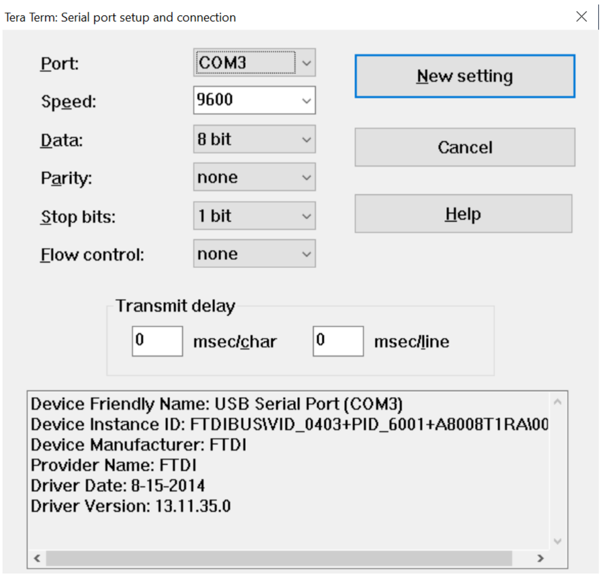
4. If the connection is successful, a blank terminal with COM3/COM1

5. title is displayed as below else an error message will be displayed "unable to connect"



COM3 - Tera Term VT

6. Click on "setup" option from the menu and select "Serial Port" option. A serial port settings and connection window is displayed.



Tera Term: Serial port setup and connection

Port: COM3

Speed: 9600

Data: 8 bit

Parity: none

Stop bits: 1 bit

Flow control: none

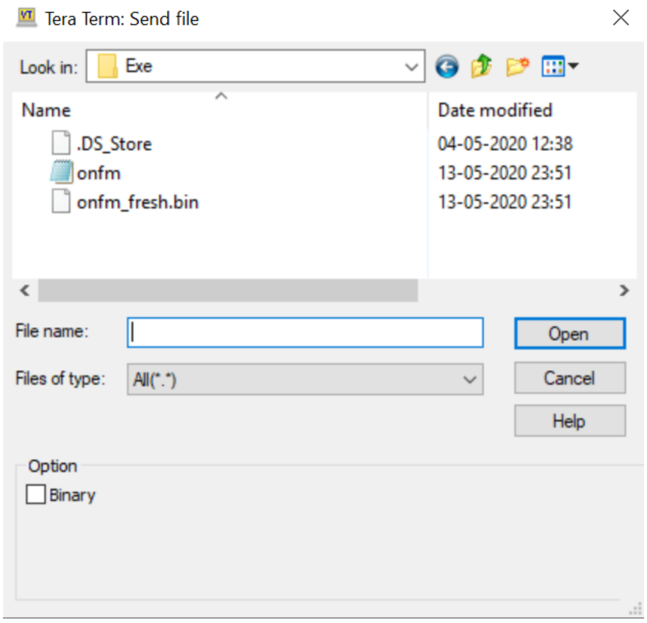
Transmit delay

0 msec/char 0 msec/line

Device Friendly Name: USB Serial Port (COM3)
Device Instance ID: FTDIBUS\VID_0403+PID_6001+A8008T1RA\00
Device Manufacturer: FTDI
Provider Name: FTDI
Driver Date: 8-15-2014
Driver Version: 13.11.35.0

7. Under speed, change 9600 to 115200 and click on “New Connection”

8. Now open the “File” option on the menu and select “Send File”. A file selection window is displayed to select the Binary file(opennfm) that needs to be transferred to the LPC board.



Tera Term: Send file

Look in: Exe

Name	Date modified
.DS_Store	04-05-2020 12:38
onfm	13-05-2020 23:51
onfm_fresh.bin	13-05-2020 23:51

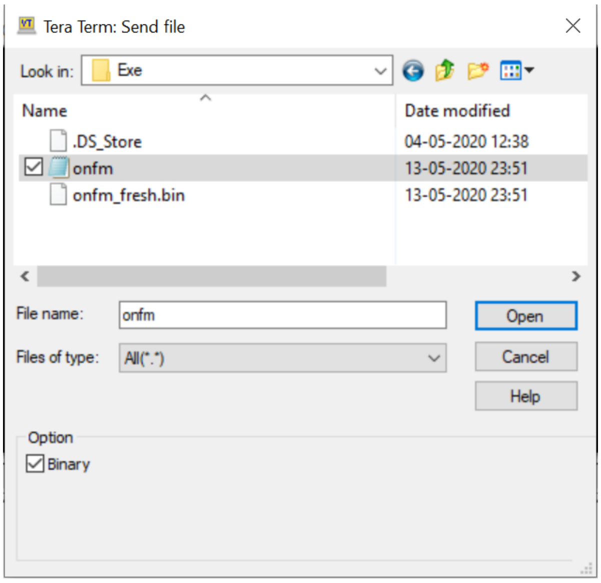
File name:

Files of type: All(*.*)

Option

□ Binary

9. Select the “Binary File” and select the Binary check box on the window and click on open.



Tera Term: Send file

Look in: Exe

Name	Date modified
.DS_Store	04-05-2020 12:38
☑ onfm	13-05-2020 23:51
onfm_fresh.bin	13-05-2020 23:51

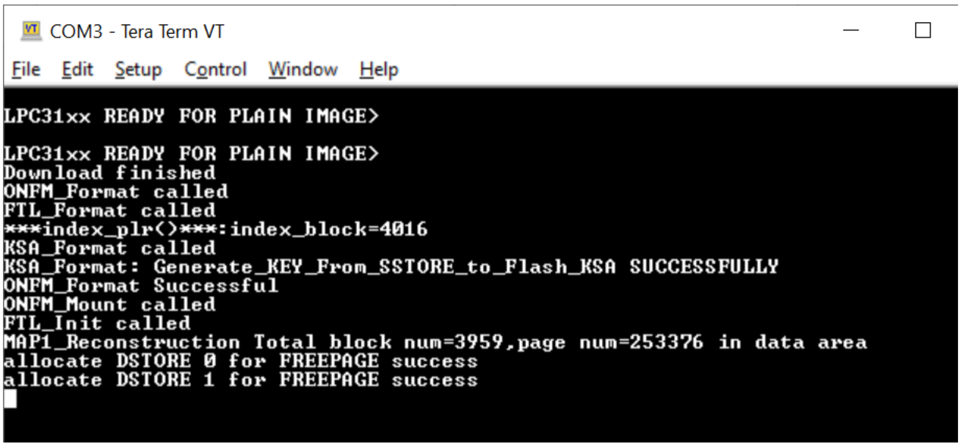
File name: onfm

Files of type: All(*.*)

Option

☑ Binary

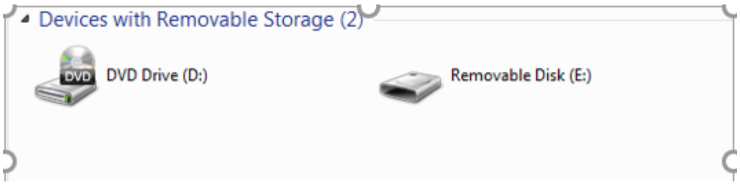
10. File is transferred to the LPC board and instructions showcasing the board being mounted is displayed on the Tera Term terminal



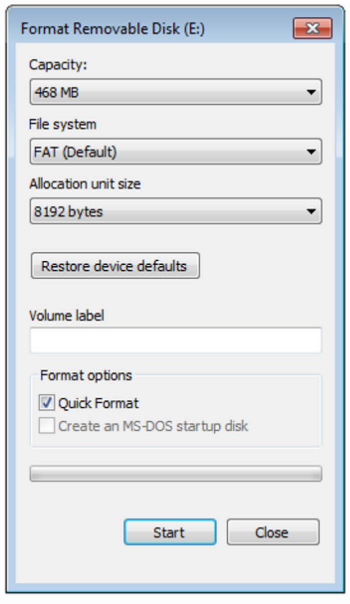
```
LPC31xx READY FOR PLAIN IMAGE>

LPC31xx READY FOR PLAIN IMAGE>
Download finished
ONFM_Format called
FTL_Format called
***index_plr()***:index_block=4016
KSA_Format called
KSA_Format: Generate_KEY_From_SSTORE_to_Flash_KSA SUCCESSFULLY
ONFM_Format Successful
ONFM_Mount called
FTL_Init called
MAP1_Reconstruction Total block num=3959,page num=253376 in data area
allocate DSTORE 0 for FREEPAGE success
allocate DSTORE 1 for FREEPAGE success
```

11. Once the device is successfully mounted, USB drive is displayed under the devices and drives section on windows PC as below.



12. USB drive needs to be formatted to be used for writing files.



Format Removable Disk (E:)

Capacity:
468 MB

File system
FAT (Default)

Allocation unit size
8192 bytes

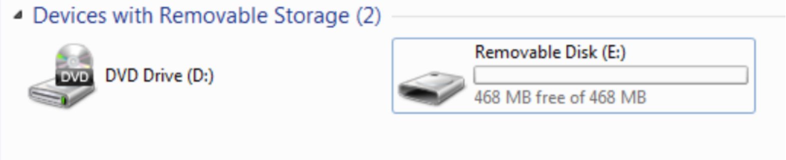
Volume label

Format options

☑ Quick Format

☐ Create an MS-DOS startup disk

13. Now the removable drive is accessible for writing files.

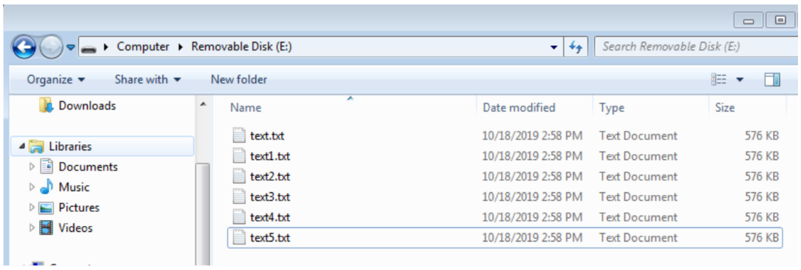


Devices with Removable Storage (2)

DVD Drive (D:)

Removable Disk (E:)
468 MB free of 468 MB

14. Files can be written to the Removable Drive as below.



Computer ▸ Removable Disk (E:)

Name	Date modified	Type	Size
text.txt	10/18/2019 2:58 PM	Text Document	576 KB
text1.txt	10/18/2019 2:58 PM	Text Document	576 KB
text2.txt	10/18/2019 2:58 PM	Text Document	576 KB
text3.txt	10/18/2019 2:58 PM	Text Document	576 KB
text4.txt	10/18/2019 2:58 PM	Text Document	576 KB
text5.txt	10/18/2019 2:58 PM	Text Document	576 KB

15. To remove the drive, click on the disconnect usb and the tera term connection is automatically closed.

Note: Depending on the version of the operating system and the system settings, COM port for Tera Term also change. To know which port works for the system one is using, connect the device to the system and open Device manager. Device manager displays the COM port for the attached NAND simulator. The other way to find the COM port is by trial and error method and try all the available ports till it gets connected. But the settings of the serial port remain same though out. The OpenNFM test bed setup is done for Win 7 operating system and the setup is estimated to remain similar for all the versions of Windows operating system.

Acknowledgment

This work was supported by National Science Foundation under grant number 1938130-CNS and 1928349-CNS. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Security and Privacy (SnP) lab at Michigan Technological University. https://snp.cs.mtu.edu.

[2] Bo Chen, and Niusen Chen. A Secure Plausibly Deniable System for Mobile Devices against Multi-snapshot Adversaries. 2020 IEEE Symposium on Security and Privacy (S&P '20), 2020.

[3] Wen Xie, Niusen Chen, and Bo Chen. Incorporating Malware Detection into The Flash Translation Layer. 2020 IEEE Symposium on Security and Privacy (S&P '20), 2020.

[4] Peiying Wang, Shijie Jia, Bo Chen, Luning Xia and Peng Liu. MimosaFTL: Adding Secure and Practical Ransomware Defense Strategy to Flash Translation Layer. The Ninth ACM Conference on Data and Application Security and Privacy (CODASPY '19), 2019.

[5] Le Guan, Shijie Jia, Bo Chen, Fengwei Zhang, Bo Luo, Jingqiang Lin, Peng Liu, Xinyu Xing, and Luning Xia. Supporting Transparent Snapshot for Bare-metal Malware Analysis on Mobile Devices. 2017 Annual Computer Security Applications Conference (ACSAC '17), 2017.

[6] Shijie Jia, Luning Xia, Bo Chen, and Peng Liu. DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer. 2017 ACM Conference on Computer and Communications Security (CCS '17), 2017.

[7] Bo Chen, Shijie Jia, Luning Xia, and Peng Liu. Sanitizing Data is Not Enough! Towards Sanitizing Structural Artifacts in Flash Media. 2016 Annual Computer Security Applications Conference (ACSAC '16), 2016.

[8] Shijie Jia, Luning Xia, Bo Chen, and Peng Liu. NFPS: Adding Undetectable Secure Deletion to Flash Translation Layer. The 11th ACM Asia Conference on Computer and Communications Security (ASIACCS '16),2016.

[9] Bo Chen, and Radu Sion. HiFlash: A history independent flash device. arXiv preprint arXiv:1511.05180, 2015.

[10] EAGER: Enabling Secure Data Recovery for Mobile Devices against Malicious Attacks. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1938130&HistoricalAwards=false.

[11] SaTC: CORE: Small: Collaborative: Hardware-assisted Plausibly Deniable System for Mobile Devices. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1928349&HistoricalAwards=false.