Test Harness Software

Augmented Installation and Quick Start Guide
# Table of Contents

1. INTRODUCTION............................................................................................................. 4
2. PREREQUISITES FOR USING THE SOFTWARE ............................................................ 4
3. REFERENCES .................................................................................................................. 6
4. OVERVIEW OF THREAD TEST HARNESS SOFTWARE ............................................. 6
5. DOWNLOAD AND INSTALLATION PROCESS OF THREAD TEST HARNESS SOFTWARE ...... 8
6. INSTALLING NXP (FREESCALE) SNIFFER DEVICE DRIVER ........................................ 15
7. HARDWARE PROGRAMMING SILABS ZM3588S-USB(-LR) STICKS ................................ 19
   7.1 INSTALLING DEVICE ZM3588S-USB(-LR) STICKS DRIVER ................................... 20
   7.2 PROGRAMMING THE ZM3588S-USB(-LR) STICKS ............................................. 21
      7.2.1 Programming Using Simplicity Studio: ......................................................... 21
      7.2.2 Programming using ISA3 Utilities ............................................................... 23
8. LOADING NXP (FREESCALE) REFERENCE DEVICE FIRMWARE .............................. 26
9. LOADING ARM REFERENCE DEVICE FIRMWARE ....................................................... 29
   9.1 INSTALLING DRIVER ............................................................................................. 29
   9.2 BOOTLOADER SETUP: ......................................................................................... 29
   9.3 UPDATE USING WINDOWS OPERATING SYSTEM .............................................. 30
   9.4 POWER DOWN, POWER UP ................................................................................. 30
10. INSTALLING OPENTHREAD FIREFLY DEVICES ......................................................... 32
    10.1 DRIVER INSTALLATION ...................................................................................... 32
    10.2 INSTALLING FIRMWARE .................................................................................... 32
       10.2.1 GIT for Windows ......................................................................................... 32
       10.2.2 Installing a Separate Copy of Python 2.7 .................................................... 33
       10.2.3 Installing firmware to Firefly devices .......................................................... 33
11. CONFIGURING WIRESHARK ....................................................................................... 35
12. TEST BED SETUP ....................................................................................................... 40
    12.1 PHYSICAL LAYOUT ............................................................................................ 40
13. DUT CONFIGURATION ............................................................................................... 40
14. USING HARNESS ...................................................................................................... 41
    14.1 SELECTING TEST BED DEVICES FOR SPECIFIC TESTS ................................. 41
    14.2 HARNESS WORK AROUNDS ............................................................................. 42
    14.3 SETTING UP ....................................................................................................... 42
14.4   RUNNING HARNESS ........................................................................................................43

14.4.1   Steps for running the software: .............................................................................44

14.5   RUNNING TESTS ............................................................................................................52

15   RESULTS AND ANALYSIS .............................................................................................54

16   USING TEST BED DEVICES AS DUTS ............................................................57

16.1   MANUAL TESTING OF TEST BED DEVICES AS DUTS ........................................57

16.2   AUTOMATION TESTING OF TEST BED DEVICES AS DUTs (AUTO DUT) ............57

17   ACCESSING SOURCE TREE BITBUCKET .................................................................58

18.   FEEDBACK AND SUGGESTIONS .................................................................................65

19   APPENDIX A: PYTHON DUT TEST SCRIPT ASSIGNMENTS .................................66

20   THE FOLLOWING ARE GENERAL NOTES BASED ON FAQ’S ................................66
1 Introduction

The Installation and Quick Start Guide provides procedures for installing, configuring, and testing using Thread Test Harness software. It also familiarizes the user with the basic operation of the Test Harness Software.

2 Prerequisites for using the software

- PC or Laptop with Windows 7 or above (32 or 64 Bit Operating system)
- NXP (FREESCALE) Sniffer USB-KW24D512
- Mozilla Firefox or Google Chrome (as default browser)
- Monitor with at least 1280 x 720 resolution is recommended

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel i5 or above</td>
</tr>
<tr>
<td>Storage (HD free space)</td>
<td>100 MB</td>
</tr>
<tr>
<td>RAM</td>
<td>4 GB or more</td>
</tr>
<tr>
<td>USB Port</td>
<td>3</td>
</tr>
<tr>
<td>OS</td>
<td>Windows 7 / 8 / 10 (32 bit or 64 bit)</td>
</tr>
<tr>
<td>USB Cable</td>
<td>See the test bed catalog. For ARM devices, in addition to the devices called out in the catalog, USB cables with the miniature connector are needed. For the small test bed version, 6 cables are needed. For the large test bed version, 32 USB cables with the miniature connector are needed.</td>
</tr>
<tr>
<td>NXP (FREESCALE) Sniffer</td>
<td>USB-KW24D512 (See Thread Test Bed Catalog for details)</td>
</tr>
<tr>
<td>Mouse (optional for laptop)</td>
<td>1</td>
</tr>
<tr>
<td>Keyboard (optional for laptop)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Test bed requirements
**Known issues**

- Wireshark must be configured manually (No automation support to change the Wireshark configuration).

- If Wireshark has not captured the trace properly then change the 'Wireshark installation path' to 'c:/programfiles/wireshark' in 'configuration.ini' file located at C:\GRL\GRL-Thread\Config.

- Supports only NXP (FREESCALE) **USB-KW24D512** as sniffer.

- For test cases, which requires 10 or more devices it is recommended to connect the reference devices to the host PC/laptop using USB 2.0 ports and hubs.
3 References

- Thread Technical Overview
- Thread Stack Fundamentals
- Thread Whitepapers
- Thread Specification Version 1.0
- Thread Test Specification Version 0.9.2.8
- UG110: EM35x Development Kit User Guide

4 Overview of Thread Test Harness Software

Thread Test Harness Software is developed using GRL Automation framework. GRL automated framework provides GUI to automate the test cases provided in the Thread Test Specification. High-level architecture of the Test Harness software is shown below:

![Figure 2 - GRL Framework License Agreement](image)

Thread Test Harness Software semi-automates the test cases defined in the Thread Test Plan. The Harness sets up the required topology using the test bed (i.e., reference) devices, and runs the tests steps within Test Plan. One or two wireless sniffers are used for capturing over the
air traffic, and the captured packets are decoded using Wireshark. The functionality of the Device Under Test (DUT) is validated by verifying the packets meet the pass criteria called out in the test plan. The typical Test Bed consists of a host computer (PC or laptop) installed with GRL Thread Test Harness Software, NXP (FREESCALE) Sniffers, and test bed devices.

All Reference Devices are connected to the Test Harness system with USB hubs. A typical test setup is shown below.

Figure 3 – Typical Test Setup
5 Download and Installation process of Thread Test Harness Software

The Harness SW package contains the Harness program, all firmware for the test bed devices, and this Quick Start Guide.

1. Download the latest software from www.graniteriverlabs.com/thread. Fill out the form using your company’s email address. This is critical for receiving your activation key for the installer. If you do not receive your key within 1 business day, contact tom@threadgroup.org.

2. Before installing any newer version of the Harness, always uninstall any previous versions. Note that uninstalling the Harness SW also deletes all sniffer files, all log files, and all Python files. If desired, back up these files before uninstalling. Be sure the \GRL folder has been deleted. If not, delete it to make ready for a clean install.

3. Locate the installer file at Thread-Test-Harness-Beta-v-X.X and double click the installer.

4. Click Next on Welcome screen.

![Harness Welcome Screen](image)

Figure 4: Harness Welcome Screen
5. Enter Harness serial number to activate the software.

![Figure 5: enter Harness serial number screen.](image)

6. Then read and agree “GRL Application Framework License Agreement”.

![Figure 6: GRL Framework License Agreement](image)

7. Read and agree to the **Python Re-Distribution License Agreement** and click Next.
The software will install in [Root Drive]\GRL\Thread folder. Click Next to proceed.

8. Click **Install** to install “Thread Test Harness Software”.

Figure 7: Python License Agreement

Figure 8 - Installation folder
9. If Microsoft Visual C++ 2012 runtime is not available in the system, installer will prompt to install it.

10. If Microsoft Visual C++ 2013 runtime is not available in the system, installer will prompt to install it.
11. Follow the wizard to install Wireshark.

12. Keep the default selection as shown and click Next.
13. Keep the default selections in Wireshark Additional Tasks.

14. Use default Destination Folder and click **Next**.
15. Install WinPcap, select the check box and click Next.

When installing for the first time or when PCAP is reinstalled, a popup window will come up asking to reboot the PC just after PCAP is installed; do not reboot at this
time. If rebooted, the harness installation stops and one must start over with the installation.

16. The GRL “Test Thread Harness Software” is successfully installed on your system.

![Installation Complete](image)

Figure 17 - Installation Complete

6 Installing NXP (FREESCALE) Sniffer Device Driver

Test harness software requires two NXP (FREESCALE) sniffers. Here are the instructions to install the drivers for these devices.

- NXP (FREESCALE) USB-KW24D512 is preloaded with sniffer firmware; however, this firmware needs to be replaced with the firmware that comes with Harness. The sniffer firmware is located in [Root Drive]\GRL\Thread\Firmwares\NXP (FREESCALE)\Sniffer for re-flashing. Using the current firmware, load in the drivers as follows below.

- Once the driver is installed, go to section 8 for instructions on how to update the FREESCALE devices’ firmware. The same procedure is used to update the sniffer firmware and the test bed device firmware.

**Steps to Install NXP (FREESCALE) Sniffer Device:**

1. Connect the NXP (FREESCALE) Device ([USB-KW24D512](#)) to one of the USB ports on your computer / laptop. (Open device manager, look for the device-connected port on your computer / laptop)
2. Double click on the unrecognized device. Following device properties dialog box appears. Click **Update Driver**.
3. In Update driver dialog box, click Browse for driver software option.

![Figure 20: Update Driver Software](image)

4. Select NXP (FREESCALE) device driver location [Root Drive]\GRL\Thread\Drivers\FSL Virtual COM Port and click Next.

![Figure 21 – Select the driver path](image)
5. Device is now ready to use.

Figure 22 - Device Manager (Virtual Com Port)

If needed, reflash the sniffer NXP device now.
7 Hardware Programming SiLabs ZM3588S-USB(-LR) Sticks

**USB Stick Hardware Overview:**

![USB Stick Hardware](image)

Figure 23 – Profile of ZM3588S-USB(-LR) showing push button switch and red and green LED

<table>
<thead>
<tr>
<th>Hardware Interface</th>
<th>EM35x I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED 0 (RED)</td>
<td>PA6</td>
</tr>
<tr>
<td>LED1 (GREEN)</td>
<td>PA7</td>
</tr>
<tr>
<td>Switch S1</td>
<td>PB6</td>
</tr>
</tbody>
</table>

Table 3: Hardware mapping to physical I/O on ZM3588S-USB(-LR)

![Programmer Port](image)

Figure 24 - Profile of ZM3588S-USB (-LR) showing programming port adapter and Pin 1

![J1 Connector](image)
Figure 25: pin out diagram for m 10 pin connector.

7.1 Installing Device ZM3588S-USB(-LR) Sticks Driver

1. Find the driver package in \[Root Drive\]\GRL\Thread\Drivers\SiLabs CEL 3588\.
2. Extract the files in the driver installation package to a folder on your local hard drive. Start the driver installation by double-clicking the appropriate driver installation executable
   - **Windows 32-bit**
     CelEM35xUsbInstaller_x86.exe
   - **Windows 64-bit**
     CelEM35xUsbInstaller_x64.exe
   **Note:** Make sure that no CEL USB Sticks are attached to the system before proceeding as this may interfere with proper UART enumeration/operation

3. Click Next >.

Figure 26: Driver installation start screen.
4. Click \textbf{Finish}.

The driver is now installed.

\section*{7.2 Programming The ZM3588S-USB(-LR) Sticks}

Application files are downloaded using a Debug Adapter (ISA3) connected via Ethernet and either the Simplicity Studio IDE or the ISA3 command line utilities. Section 7.2.1 assumes you have obtained Simplicity Studio. This is not purchased through the Thread Test Bed catalog. If you do not have Simplicity Studio, skip to section 7.2.2. If you need help connecting to the Debug Adapter (ISA3) please refer to UG110: EM35x Development Kit User Guide that is available on the Silabs website.

\subsection*{7.2.1 Programming Using Simplicity Studio:}

The following steps presume you have Simplicity Studio installed and are able to see the Debug Adapter (ISA3) in your Devices view. The ISA3 InSight Port is connected to the programming port on the ZM3588S - USB(-LR).
1. Right-click the adapter from the Devices view and select **Connect** from the pop-up menu.

![Figure 28 - Connect Selection with Ember Desktop](image)

2. Right-click the connected adapter from the Devices view and select **Upload Application** from the pop-up menu.

![Figure 29 - Upload Application Selection](image)
3. Browse to select an application image from the dialog's list, as shown in Figure 30. Note that no bootloader image is required.

![Figure 30 - Application Image Upload](image)

4. Select whether you would like the Target to Reset or Run after loading.
5. Click OK.

### 7.2.2 Programming using ISA3 Utilities

The Thread test bed catalog indicates to purchase the Debug Adapter (ISA3) without the entire EM3xx development kit. As such, section 7.2.1 of the Harness Quick Start Guide is only useful if the development kit is purchased. This section describes how to use ISA3 command line utilities to configure the Debug Adapter (ISA3) with a static IP address and to load device firmware.

Initially, the Debug Adapter (ISA3) is set up to get an IP address from a DHCP server. You can use a DHCP server. However, if the development kit is not purchased and you are not using a DHCP server, a static IP address can be configured on the Debug Adapter (ISA3) as follows:
1. Install the drivers into the PC per section 7.1.
2. Install the ISA3 utilities.
3. Connect the Debug Adapter (ISA3) with a USB cable to the PC.
4. Open a CLI window, and navigate to \Program Files\Silicon Labs\ISA3 Utilities\bin for a 32-bit OS or to \Program Files (x86)\Silicon Labs\ISA3 Utilities\bin for a 64-bit OS.
5. Run em3xx_isa.exe as follows, replacing IPADDR, NETMASK, and GATEWAY with your desired settings:

   Em3xx_isa.exe --admin “ip static IPADDR NETMASK GATEWAY”

6. Next run em3xx_isa.exe again to turn off DHCP:

   Em3xx_isa.exe --admin “ip dhcp off”

7. If the Debug Adapter (ISA3) was programmed successfully, you will receive this message, as shown in the following figure.

   Result
   Success: ip:(after reboot) DHCP off

   ![Figure 31a - Successful Debug Adapter (ISA3) Programming](image)

Programming the static IP address is complete. The device will retain the static IP address for future use; setting it up again is not required.

8. Unplug the USB cable.

   Now that the IP address is set, the firmware can be loaded into the device.
1. Connect an Ethernet cable from your PC to the Debug Adapter (ISA3). You can power the Debug Adapter (ISA3) using either POE or the USB power supply supplied with the Debug Adapter (ISA3).

2. Navigate to the ISA3 utilities folder again.

3. Execute application `em3xx_load.exe` as follows, where the IP address and firmware filename are required:

   ```
   Em3xx_load.exe --ip [IP address of the device] [Silabs Firmware binary file]
   ```

   Example: `Em3xx_load.exe --ip 10.4.176.25 c:\GRL\Thread1.1\firmwareSilabs\thread-test-app.s37`

   where the device’s IP address is 10.4.176.25 and the firmware is `thread-test-app.s37`, as shown in the following figure.

   ![Figure 31b - Successful Application Upload](image)

   Once the process is complete, you’ll see “DONE” on the CLI screen and firmware update is complete.
8 Loading NXP (FREESCALE) Reference Device Firmware


![First web page for software.](image)

Figure 32: First web page for software.

5. Click the Download button for the Test Tool for Connectivity Products entry:

![Second web page for software.](image)

Figure 33: Second web page for software.

6. Register or login to review the Software License Agreement and click I Accept if agreeing with the license terms.

7. The TestTool_Setup.exe will begin downloading after accepting the agreement.

8. Launch TestTool_Setup.exe to begin installation. Window OS Administrator privileges are required. Complete the installer steps.

9. After installation, launch the Test Tool 12 application using desktop shortcut (if created) or from the start menu at NXP (FREESCALE) Test Tool ➔ Test Tool 12

![Test Tool monogram](image)

Figure 34: Test Tool monogram
10. In the Test Tool button bar select the **Firmware Loaders → Kinetis Firmware Loader** option:

![Figure 35](image)

11. Connect a J-Link pod via USB to the PC. Install drivers if required. J-Link drivers can be downloaded as part of the **J-Link Software and documentation pack** at:

   [https://www.segger.com/jlink-software.html](https://www.segger.com/jlink-software.html)

   When downloading J-Link SW from the website, choose the SW and Documentation package. This choice will give you the entire SW package and documentation.

12. If the J-Link is successfully installed, it shows up under J-Link devices in the Test Tool Kinetics Firmware Loader view:

![Figure 36](image)

13. **Connect the J-Link pod to the mini-JTAG connector on the USB-KW24D512.**

![Figure 37](image)

   Figure 37: Shows how to connect J-Link to NXP device.

14. In the Test Tool Kinetics Firmware Loader, click the **Browse...** button then navigate and select the s-record firmware file (KTSTB*.srec) located in [Root Drive\GRL\Thread1.1\Firmwares\NXP (FREESCALE)\Reference Device\](Root Drive\GRL\Thread1.1\Firmwares\NXP (FREESCALE)\Reference Device\)
15. Click the **Upload** button.
16. Select the **KW2XD512** option if prompted:

![Figure 38: Development Board Option](image)

17. Firmware programming will begin and take about 10s of seconds to complete. Note the Upload Progress bar at the bottom of the Test Tool window to indicate completion.
18. Remove the J-Link connection, then unplug and plug back the USB-KW24D512 board. Note both blue LEDs are flashing.
19. Repeat the steps for all the NXP (FREESCALE) USB-KW24D512 Reference Devices.
9  Loading ARM Reference Device Firmware

Before programming, put the daughter board on the mother board.

9.1  Installing Driver

The driver must be installed in the PC first.

1. Connect a USB cable as shown in the picture Fig 39, red arrow.
2. Plug board into PC. A message will appear that there is no driver for this device.
3. Find the device in the Device Manager and open up properties.
4. Click on Update driver.
5. Navigate to the folder containing the ARM driver

C:\GRL\Thread1.1\Drivers\ARM - FRDM K64F\mbedSerialDriver

6. Double click the exe file to install driver.

9.2  Bootloader setup:

The bootloader need not be installed with each firmware upgrade. If needed, subsequent releases of Harness will include upgraded firmware bootloaders.

Figure 39: Bootloader setup button on motherboard.

To check your firmware version:

- Open the DETAILS.TXT file if present.
- If not present, open the MBED.HTM file on your mbed microcontroller with a text editor

Get the latest bootloader firmware from
“ROOT_DRIVE:\GRL\Thread1.1\Firmwares\ARM\BootLoader”
Enter Bootloader Mode

1. Disconnect all power, external circuits and USB cables.
2. Press and hold the button marked as RESET, (Refer figure 39).
3. Connect a USB cable as shown in the picture Fig 39, red arrow.
4. The drive should mount named BOOTLOADER.

The reset button can now be released.

Figure 40

9.3 Update using Windows Operating system

- Download the interface firmware file to your hard disc.
- Drag and drop the file onto the USB drive named BOOTLOADER.

9.4 Power Down, Power Up

Disconnect the USB cable and reconnect. The drive should now be named MBED or DAPLINK.

Figure 41: popup after bootloader is installed.

1. Find the mass storage device “MBED” when you connect the device to PC.
2. Copy or drag and drop ARM Thread stack file from Firmware’s folder with extension (*<filename>*.bin) to MBED device mass storage.

3. Find ARM com ports in Device Manager. See section 11.1 on how to find the COM ports used by each test bed device plugged into the PC.

4. If more than 14 devices are to be connected to the harness bootloader is to be flashed on to the ARM devices. This bootloader flash image is included in harness package [“ROOT DRIVE:\GRL\Thread1.1\Firmwares\ARM\BootLoader”]. When flashing large number of ARM devices, the entire device set might not show up at once as windows may run out of drive letters, so devices might need to be flashed in smaller batches.
10 Installing OpenThread Firefly Devices

10.1 Driver installation

1. Plug a Firefly device into a USB port. Windows may install the driver automatically.
2. If Windows does not install the driver automatically, you will need to install the driver manually. The driver is provided with Harness at [Root Drive]\GRL\Thread\Drivers\.
3. To install manually, open Device Manager and find the Firefly device; it will have the yellow triangle with the exclamation mark in it showing the driver is not installed.
4. Right click on Firefly, and select Update Driver.
5. Navigate to the folder containing the driver.
6. Click install. Driver should install and Firefly device will now show up in the port section of Device Manager.

10.2 Installing Firmware

10.2.1 GIT for Windows

To install firmware into the Firefly devices, GIT for Windows is needed. Download from https://git-scm.com/download/win.

Please install GIT according to the installation wizard, making sure the Git Bash mode is checked as below:

Figure 42b: GIT install wizard setup requirements.
10.2.2 Installing a Separate Copy of Python 2.7

**Note** It is **RECOMMENDED** to install a separate Python2.7 to avoid possible unexpected results even though Test Harness has built-in python2.7.

Download Python from [https://www.python.org/downloads/](https://www.python.org/downloads/)

Please choose to install pip and add Python to PATH in the installation wizard.

![Python Setup for OpenThread firmware install.](image)

Launch Git Bash to fix python console issue as follows:

```bash
echo "alias python='winpty python.exe'" >> ~/.bashrc
```

Install the serial connection for Python:

```bash
pip install pyserial
```

10.2.3 Installing firmware to Firefly devices

- The firmware file has the name with the structure `ot-YYYYmddd-xxxxxxxx.bin` where `xxxxxxxx` is the git commit id of the source, and `YYYYmddd` is the building date.
- All commands below should be run in GIT Bash, and in order to smoothly finish the whole process follow the below steps under the directory where the extracted files are.
1. Launch GIT Bash
2. Download firmware install utility from GitHub
   
   ```
   git clone https://github.com/JelmerT/cc2538-bsl.git
   ```
   
3. Launch Device Manager and note which COM port the Firefly device is using.
4. Install firmware on Firefly device using the serial port serial port (eg COM4):
   
   ```
   python cc2538-bsl/cc2538-bsl.py -e -w -v -b 460800 -p COM4 ot-YYYYmmdd-xxxxxxx.bin
   ```
   
   Here’s some sample output using COM4 from GIT:

   ```
   Opening port COM4, baud 460800
   Reading data from ot-20161209-a67ada0.bin
   Cannot auto-detect firmware filetype: Assuming .bin
   Connecting to target...
   CC2538 PG2.0: 512KB Flash, 32KB SRAM, CCFG at 0x0027FFD4
   Primary IEEE Address: 00:12:4B:00:06:0D:B3:50
   Erasing 524288 bytes starting at address 0x00200000
   Erase done
   Writing 524288 bytes starting at address 0x00200000
   Write 16 bytes at 0x0027FFF0F8
   Write done
   Verifying by comparing CRC32 calculations.
   Verified (match: 0x248a89f0)
   ```

   5. Repeat step 4 for all the Firefly devices.
11 Configuring Wireshark

Steps to configure Wireshark:

1. Open Wireshark application from Program Files.
2. Click Edit and select Preferences.

![Wireshark Network Analyser](image)

Figure 43: Wireshark Network Analyser

3. Click Protocols (It expands a list of protocols).
4. Find **CoAP** Protocol and configure it as given below:
   i. CoAP port number as 5683
   ii. Additional CoAP port number (1) as 19789
   iii. Additional CoAP port number (2) as 61631
   iv. Additional CoAP port number (3) as 0

5. Select IEEE 802.15.4 and click **Enter**.
<table>
<thead>
<tr>
<th>Hardware</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.15.4 Ether type (in hex)</td>
<td>809a</td>
</tr>
<tr>
<td>TI CC24xx FCS format</td>
<td>Deselect</td>
</tr>
<tr>
<td>Dissect only good FCS</td>
<td>Select</td>
</tr>
<tr>
<td>Automatically acquire Thread sequence counter</td>
<td>Select</td>
</tr>
<tr>
<td>Use PANID as first two octets of master key</td>
<td>Deselect</td>
</tr>
<tr>
<td>Thread Sequence counter</td>
<td>000000000</td>
</tr>
<tr>
<td>Decryption key 1</td>
<td>00112233445566778899aabbccddeenf</td>
</tr>
<tr>
<td>Decryption key index 1</td>
<td>1</td>
</tr>
<tr>
<td>Security suite (802.15.4-2003)</td>
<td>AES-128 Encryption, 32 bit integrity protection</td>
</tr>
<tr>
<td>Extend authentication data (802.15.4-2003)</td>
<td>Select</td>
</tr>
<tr>
<td>Key hash</td>
<td>Thread hash used to derive key</td>
</tr>
</tbody>
</table>

Table 4: Configuration for IEEE 802.15.4 protocol
6. Click Thread CoAP and select Decode CoAP for Thread as shown below.

![Figure 47 - Thread CoAP](image)

7. Set the 6LoWPAN Context as shown below. Set context 0 value as fd00:0db8::/64.

![Figure 48: 6LoWPAN](image)
8. Disable ZigBee NWK protocol (and any other unwanted protocols):
   i. Click Analyze and then select Enabled Protocols and Uncheck the following:
      o LwMesh
      o ZigBee
      o ZigBee Green Power

![WireShark Enabled Protocols - Profile Default](image)

Figure 49 – Enabled Protocols
12 Test bed Setup

Figure 50: block diagram of the test bed hardware. The “small bed” configuration has 6 test bed devices, and the “large bed” has 32 test bed devices.

12.1 Physical Layout

There are no physical layout requirements at this time. Enough space between devices should allow for adequate ventilation as the devices do generate heat.

Note that the antennas on the devices are traces on the circuit boards. Thus, the RF radiation is highly polarized in the plane of the circuit boards. Best RF communication between devices is obtained by making all devices’ circuit boards parallel to each other.

13 DUT Configuration

The member owner shall supply all necessary instructions and equipment required to run certification testing.

The Device Under Test (DUT) shall have no physical connection to the PC that runs Harness and no connection to the test bed Golden Devices. If an external power supply (e.g., USB) or external control (e.g., USB, Ethernet, Serial Interface) is needed to power or configure the DUT, this external connection shall be to equipment (e.g., PC, laptop) that is not associated to Harness or the test bed Golden Devices.
14 Using Harness

14.1 Selecting Test Bed Devices for Specific Tests

The Harness is capable of selecting the vendor devices for specific nodes in a given test. To enable device selection, go to ‘%HOMEDRIVE%:\GRL\Thread1.1\config’ and set "EnableDeviceSelection" to True. If it is set to false, Harness will randomly select devices for the test bed device roles. If set to True, on starting Harness, Harness will scan the TopologyConfig.txt file present in folder ‘%HOMEDRIVE%:\GRL\Thread1.1\Thread_Harness\TestScripts’

This configuration is made using the file “TopologyConfig.txt”. To decide on which vendor device to select while forming the topology, each row in the file corresponds to a test case and follow the syntax mentioned below:

Testnumber-Role:devicename,role:devicename, …

For example:

5.1.2-Leader:Freescale,ED_1:ARM,SED_1:SiLabs

- **5.1.2** represents the test case id and must be followed by a “-“
- **Leader/ED_1/SED_1** represent the devices roles required by the test case which will be controlled by the harness [Naming convention is case sensitive].
- **Freescale/ARM/SiLabs** represent the required vendor device for that role [These names are case-insensitive]
- “,” [Comma] is used as a delimiter between each device role and “:” [Colon] is used for separating the device role and vendor name.

The above line in the configuration file would cause a Freescale [NXP] device to become the leader, an ARM device to be the End Device and a SiLabs device will be used for the Sleepy End Device when running test case 5.1.2. Device types, ARM, Silabs, NXP, can be selected, but specific devices within the type cannot. For example, if there are more than one ARM device in the test bed, Harness will select any one of the ARM devices for the role.

If the required number of devices [From any vendor] is not connected to Harness, Harness will not display the tests or if the “display all test” check box is checked, the tests will be grayed out indicating they cannot be selected. If the file, "TopologyConfig.txt", does not exist in the designated folder, Harness will not display any tests or all tests will be grayed out.

**WARNING:** the topo… config file is read by Harness at start up. Any changes to topo… config file only take effect when Harness is restarted.
14.2 Harness Work Arousnds

At the present time, Harness has an error that occurs when the large (32 test bed devices) test bed architecture is used. Figure 51 shows the error message:

Figure 51: error message from Harness using large test bed architecture.

The system is Windows 7, 32 bit operating system. The work around to prevent this error is as follows:

1. Open this file using Notepad: C:\GRL\Thread1.1\Config\Configuration.ini
2. Find line 16: Device History =…
3. Remove all text after the “=” sign to remove the error. Be sure to remove all the text after the “=” sign. If not, the error will continue to occur.

14.3 Setting Up
When plugged into the USB hub, each device is assigned a COM port number. Before running any test, the COM port number needs to be known for the test bed devices, the DUT and the sniffers. The COM port number is found in the Device Manager under PORTS (COM & LPT). To find device’s COM port number, plug in each device one by one to the USB hub and note the new COM port number appearing in the Device Manager.

![Device Manager](image)

Figure 52: example of COM port designation for one of the Thread test bed devices.

### 14.4 Running Harness

Harness software is compatible with Google Chrome or Mozilla Firefox; ensure you open the application in Google Chrome or Mozilla Firefox. Select **Thread Harness** from start menu and follow the given steps.

Currently, there is no automation to run tests consecutively without human intervention; see section 17 for the special case of AutoDUT using the test bed devices as a DUT. Tests can
be queued up, but a person will need to react to requests from Harness for information such as the Factory MAC address, the ML64 address, etc.

14.4.1 Steps for running the software:

1. Disconnect all Thread devices from the computer. Connect Thread Sniffer dongles to the computer.
2. Select Start Menu > All Programs > Thread Test Harness Software > Thread Harness.

![Opening Thread Test Harness Software](image)

Figure 53 – Opening Thread Test Harness Software

3. Click **Thread Harness** to open the application.

The test harness software opens and displays in your browser window. If the default browser is not Google Chrome or Mozilla Firefox, copy the link and open in any one of the supported browser.
If the test bed devices are plugged into the USB hub(s), Harness will now automatically scan for these devices. If the devices are going to be connected manually as described in step 9 below, then click on the skip button outlined in red in figure 54a. Once skipped, Harness displays the screen shown in figure 54b.

If any “Security certificate” dialog box pops-up as shown below. Click “OK” and add the self-signed certificate to harness software.
4. Setting Global Values

In the Harness web page, on the left is the button to pull up the menu for the Harness settings. In this menu, the primary and secondary channels can be set, the network key, the PANID, etc.

In the middle of the web page is the Capture Settings. Here is where the information for the sniffer can be found and modified if needed. If Harness does not detect the sniffer COM port automatically, it can be detected here manually.

On the right is the General Setup. Usually nothing needs to be changed here.

Figure 56: Thread Setup

5. In figure 57, specify the primary and secondary channels the test needs to be conducted. Do not change anything else. Click Save settings and exit.
6. Click **Capture Setup/ Sniffer Settings**.

Figure 58, if one of the sniffer USB radios is not found automatically, use the “Auto Detect” button to find its COM Port. Once found, click Verify to validate the results and SAVE. The second sniffer will be detected when the tests needing it are run. No detection is needed at startup.

If the sniffer firmware is not correct in the NXP sniffer dongle, then an error message is posted as shown in figure 58b. To correct, reflash the sniffer dongle with the firmware that comes with the latest Harness release. Both sniffer dongles must be flashed.

Figure 57: Thread Setup channel selection

Figure 58a - Capture Setup
7. Click **General Setting**.

Configure the Child Update Timeout [Keep Alive Timeout] and the SED polling rate these values are DUT Specific. These will default to values that are used for the test bed devices as the DUT.
8. Click Start. Harness will display the Test Bed configuration window.

Figure 60 – Setup Configuration

9. Connect the thread devices one by one. Identify the port number from the device manager. Drag and drop the required device from the Supported Hardware list (on your left) to the Test Bed (on your right). If auto-detect was allowed to complete, all test bed devices should be already populated in the right column.

Figure 61 - Configure Test Bed
10. Specify the port number in the **Serial Line** text box and select **Blue** box with arrows to connect.

![Figure 62 - Test Bed](image)

11. The line present to the left of the device is the **Status indicator** line. It turns green once the setup is successful.

![Figure 63 - Green Status indicator](image)
12. The Status indicator line turns Red in case of setup failure. In such cases verify the port number specified in the Serial Line text box and try again. A failure will also occur if the device’s firmware version does not match the version Harness expects. In this case, flash the device with the firmware that comes with Harness.

Figure 64- Red Status Indicator

13. Once all the status indicators turn Green, the Next button gets activated. Click Next to start the test.
Figure 65 – Navigating to Test Page

14.5 Running Tests

Steps to run the test (Figure 66):

1. Select DUT role (Leader, Router, etc.) (1).
2. Select one or more tests case from Test Selection box (2).
3. Click Green button (3) to run.

Figure 66 - Thread Compliance Test Suit

The test progress and results appear in the Test Results window.
4. While running a test if information (e.g., ML64) of the DUT is required, a window pops up requesting the user to provide the required inputs to perform the tests.

Figure 68 – DUT specific details input
15 Results and Analysis

PDF, Excel, and CSV files can be generated automatically for each test. For these results to automatically be generated, go to GRL\Threadx.x\config folder and open the configuration.ini file. At the bottom make sure AutoGeneratePDF = True. Then the pdf, Excel, and CSV will be generated in GRL\Thread1.1\Thread_Harness\temp folder.

Alternatively, Harness provides links to the logs and PDF reports in the GUI. All Wireshark sniffer captures are available in the folder Sniffer captures in .GRL\Threadx.x\Captures.

Steps to see the result:

1. To see stepwise result, click the test name in the Test Result list.

![Figure 69-Thread Compliance Test Suite (DUT Selection)](image_url)

2. To generate the PDF Report for the tests performed click PDF Icon and the provide the details prompted in a test information window. Sample information is shown in Figure and click on Generate PDF Reports.
The PDF Report contains the test information, information of the devices used by the harness, and step by step results of the test performed.

3. Click on the Excel Icon to download the results of the tests performed in Excel format, a sample excel report is shown in Figure 72.
Figure 72- Excel Report

To exit the Harness application, close the command consoles and the browser tab.
16 Using Test Bed Devices as DUTs

16.1 Manual Testing of Test Bed Devices as DUTs

Deprecated with use of AutoDUT.

16.2 Automation Testing of Test Bed Devices as DUTs (AutoDUT)

Granite River Labs has provided methods for running the test bed devices as DUTs in an automated fashion. Automation incorporates all of the DUT scripts into Harness thus allowing the test to run without human intervention to provide any of the information needed in the manual testing mode.

To enable the Auto DUT feature, select the check box on bottom right corner on the test bed selection page (see figure 6).

![Figure 74: Harness test bed device selection screen showing how to enable the AutoDUT feature.](image)

Once AutoDUT is enabled, each device will have a radio button to select it as the DUT. To configure one of the test bed device as the DUT select the “Set as DUT” radio button in the device panel of the respective device (see figure 7).
Figure 75: shows how to select the test bed device as the DUT.

Continue to the test selection page by selection the next button to continue testing as normally done in the manual testing method. Tests can be queued up in succession, and Harness will run them without any human intervention in most cases. There are 3 cases, tests 5.6.7, 9.2.9, 9.2.10, that require that devices be moved into and out of RF shield enclosures. These tests will require human intervention even in using AutoDUT.

17 Accessing Source Tree bitbucket

The latest Harness scripts and complete Harness installation can be obtained from Source Tree bitbucket. To get started with using bitbucket, you will need an Atlassian sign in. If you do not have an Atlassian sign in, you can create one when running the install SW for Source Tree.

The Source Tree application must be downloaded from Source Tree and installed on your PC. Go to https://www.sourcetreeapp.com/ to download the application. During the installation you will be asked to either signup or sign into Atlassian. Use your Atlassian credentials to login and finish the installation process, or signup to create an Atlassian account then finish the installation. If you see any checkout/clone dialog boxes, skip them for now and finish installation.

After installation when launching the Source Tree application, you might see a dialog box asking you to download and install Mercurial; Please do so.

STEP 1: Open your web browser and navigate to https://bitbucket.org/threadgroup/v1.1-test-scripts you might need to log into bitbucket using you Atlassian login. Once logged in, you will see a dashboard as seen in the below screenshot. Don’t worry if you don’t see all the repositories, the one to focus on is the “v1.1-test-scripts”.

Copyright © 2015 Granite River Labs. All rights reserved.
STEP 2: Click on the “v1.1-test-scripts” repository to open it up and copy the repositories link. You will need this link to clone the repository using Source Tree.
STEP 3: Once you have the link to the repository, Launch Source Tree and you will be shown the following configuration tab. Go ahead and click on the “Clone” button.

Figure 78
**STEP 4:** Clicking on the clone button will launch a new dialog tab.

- Enter the repository link that you copied from the bitbucket website in STEP 2 in the first test box.
- In the second test box specify the location on your local machine where you wish to store the repository files.
- In the advanced setting select the branch you wish to checkout, if you’re not sure which one to select you’re better off checking out the master branch.
- Next click the blue “Clone” button on the bottom.

![Clone dialog](Figure 79)
Once you hit the clone button, you will be taken to a new tab which displays the cloning progress.

![Cloning progress](image)

Figure 80

Hit on the “**close**” button once it finishes downloading all the files into your local repository. You should now be able to navigate to the local folder to see all the files downloaded. Ignore the `.git` files.

![Local folder with files](image)

Figure 81
STEP 5: To pull latest code from bitbucket, go back into Source Tree, it should load with “Thread_Harness” Tab [Depending on what you name the local repository] open and displays all its details. Now click the pull button that will open up a new dialog box that lets to pull updated code from the remote repository.

![Figure 82](image_url)
STEP 7: Select the branch you wish to download, the master branch will contain tested and regressed test cases whereas the dot releases [V45.1 in this case] will have test cases that have been signed off by GRL and are ready for UL regression/testing. In case the drop down menu does not list any branches, click the “Refresh” button.

Figure 83

STEP 8: One you have all the latest files you will need to copy them into “C:\GRL\Thread1.1\Thread_Harness” to start using them with installed harness.
18. Feedback and Suggestions

Thread values your feedback on our products. To help us serve you better on “Thread Test Harness Software”, please send us your valuable feedback, suggestions, ideas or comments via e-mail to jeff@threadgroup.org along with the following information:

**General Information**

1. Your name, company, mailing address, e-mail id, phone number, FAX number.
2. Please indicate if you would like to be contacted by Granite River Labs about your suggestions or comments.

**Program-Specific Information**

2. Description of the problem.
3. Device firmware version number.
4. Sniffer firmware version number.
5. Configuration used in the application.
6. If possible, send the trace file and the report files for further analysis.

When you use e-mail, please type in the subject line “Thread Test Harness Software”
19 Appendix A: Python DUT Test Script Assignments

Deprecated with use of AutoDUT.

20 The following are general notes based on FAQ’s.

**Router Attachment sequence**: parent request packet, parent response packet, child ID request packet, child ID response packet, address solicit packet, address solicit response packet.

Address solicit packet and address solicit response packet are sent when the device requests to become a router. They are CoAP messages with a CON /a/as/ and an ACK /a/as/ in the Wireshark info column.

**End Device Attachment sequence**: parent request packet, parent response packet, child ID request packet, child ID response packet. (End Devices do not send address solicit packets)

The Child ID Request packet’s mode TLV of each attaching device can be used to assess the role of the device. Routers will be FFD (fully functional device). EDs will be RFD (reduced function device) with Receive on when idle:True. SEDs will be RFD with Receive on when idle: False. SEDs also send keep-alive messages to their parent in the form of data request every few seconds.

FFDs are full functional devices – routers and leaders.
FEDs are full end devices – end devices that accept non-stable network data and respond to address queries.
MEDs [minimal end device] do not accept non-stable data and do not respond to address queries.

**Requirements for the Beacon Payload**

when “B” bit is set to 1, then the Protocol ID = 3 and Version = 2 is required and the payload shall be a Beacon payload.

When “B” bit is set to 0, Protocol ID ≠ 3 and Version ≠ 2 are required. Also, the payload shall be empty payload or have an application specific value (not a beacon payload). See section 8.4.4.2 of the thread specification.

Specification sections 5.2.2.4, 5.2.2.5, 5.3.1 and 5.3.2 explain link local addresses and mesh-local addresses. The names in the Python scripts were changed in the specification:

```
LL64 -> link-local IPv6 address (there is no acronym for this one)
```
ML64 -> ML-EID

Test 9.2.18 SED, decoding the ping packets after the master key is changed.

Settings in Wireshark to view the SED ping packets:

- A menu pops up on clicking Profile:Default text in the bottom right corner of the wireshark (shown in green box of figure 8 below).

Figure A: Wireshark screen with capture.

- From the menu select Profile: 9_2_18_MasterKey, this shall apply new master key on the packets, so only the packets with new master key are decrypted
- To view the ping reply from SED we need to add the **Static Address** in the preferences,
- In wireshark goto **Edit > Preferences > Protocols > IEEE 802.15.4**
- Click on the Edit button corresponding to Static Address
A Profile adding menu pops up, Click on the 'New' button (shown in red box in the screenshot2 attached)

A new window appears, Enter the short address, PAN ID and 64 EUI of DUT and Click Okay

In the Static address window, click on Apply and then on Ok.

After this change the packets with the new master key shows up and the Ping response from the SED will appear. This has to be done when validating manually and has to be reverted (remove the static address and change profile to default) else may cause problem to other test cases.

NXP failing to reset with new Harness or FW load.
You might get the following error message when you try to start a test:

WARNING: Device Reset Failed (Device THCI object is 'None') Device ID 0
The following could fix it. If not, a reinstall of Harness is necessary.

1. Go to Harness "C:\GRL\Thread1.1\Config" folder.
2. Find the Configuration.ini file.
3. Find the property named "DevicesHistory" and delete its entry completely; make sure there is no character after the = sign.

--------------------------------------------------------------------------------------------------------------------

Clearing COM Ports.

As devices are added to the test bed, the COM port numbers might increase and multiple COM port numbers will be generated that are not used. COM ports can be cleared in WIN7 as follows:

1. Right-click “Command Prompt” in Accessories and choose “Run as Administrator”
2. Enter “set devmgr_show_nonpresent_devices=1” – without the quotes
3. Enter “start devmgmt.msc”
4. In the box that opens, select “Show hidden devices” in the ‘view’ menu.

Now if you expand the section on COM ports, all the COM ports that have ever been created will be displayed, the non-present (i.e., those not used) ones being in grey. You can uninstall away anything that you don’t want (right click, select uninstall).

Ref: http://www.fettesps.com/how-to-remove-com-ports-which-are-in-use/

--------------------------------------------------------------------------------------------------------------------