TI-15.4 Stack Linux Example Application – Quick Start Guide

1. Introduction

This Quick start guide provides instructions to quickly start using the out of box TI-15.4 Stack Linux SDK Example Application.

2. Background

The TI-15.4 Stack SDK Example Applications allows developers to create ultra-low power, very long range, star topology network solutions. The TI-15.4 Stack Linux SDK comes with the **Collector and gateway** example applications (in addition to others please see the developers Guide for more details). The Linux Collector example application interfaces with the CC13xx running the MAC CoProcessor via a UART. Collector example application builds a full-function device that performs the functions of a network Coordinator (starting a network and permitting devices to join that network) and also provides an application to monitor and collect sensor data from one or more Sensor devices. In addition it provides a socket server interface to the Linux Gateway application.

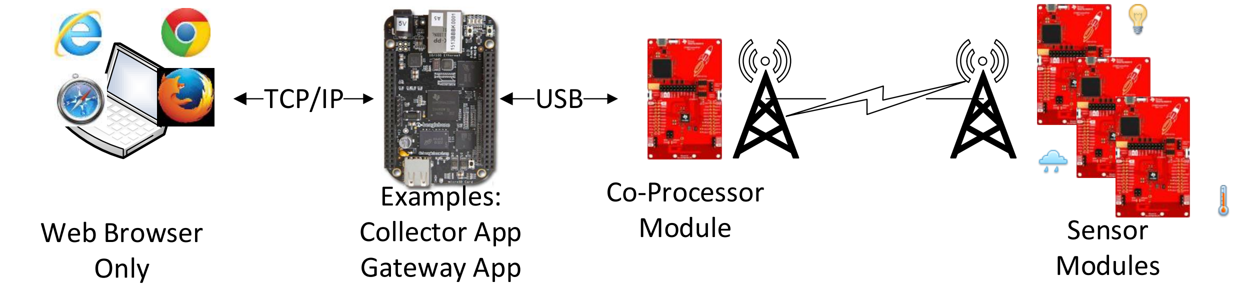
The Linux Gateway application, implemented within the NodeJs framework, connects as a client to the socket server created by the Linux Collector. In addition, it establishes a local web-server to which the user can connect via a web-browser to monitor and control the network devices.

The collector and gateway example application which provides IEEE 802.15.4 to IP Bridge is a great starting point for creating IOT applications with TI-15.4 Stack SDK.

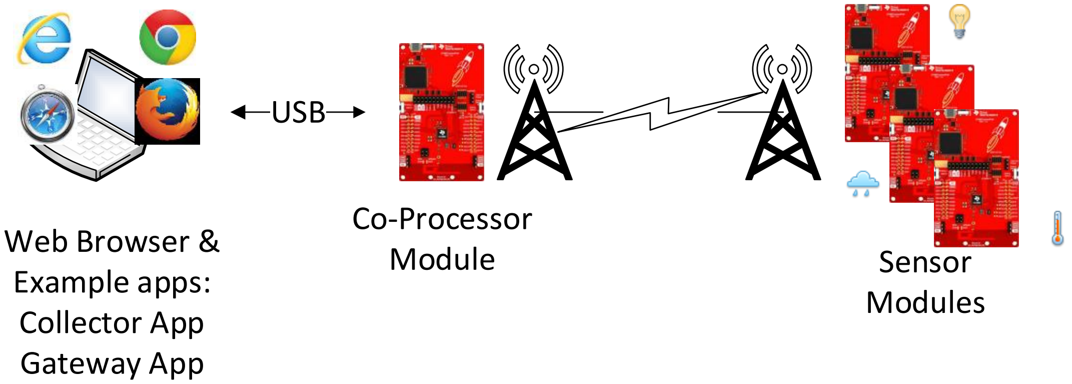
3. Supported Hardware Combinations

The out of box Linux collector and gateway example application can be run on two host hardware combinations:

HW Combination 1: Application runs on Beaglebone black. Uses a Beagle Bone Black (BBB) + CC1310LP running the MAC CoP. BBB runs the collector and gateway application while the CC1310 runs the MAC CoP application.
This combination works with any host machine. **Pro**: Once setup, only a Web Browser is required. **Pro**: No Linux machine is required right away, **Con**: a few more steps are required to setup the BBB.



HW Combination 2: Pro: Easy to setup. **Con:** Requires a Linux machine. **Pro:** No Beagle Bone Black is required; instead all example applications run on the Linux development host machine. **Con:** Does not demonstrate the scalability of the example application. **Important Linux Host Requirements:** Prebuilt Linux binaries assume: x86 Machine running Ubuntu 14.04 64 bit LTS



Common to HW combination 1 and 2: the Network Devices: At least two CC1310 Launch Pads. Launch Pad #1 – runs the prebuilt MAC-CoProcessor application, Launch Pad #2 (to #N) act as Sensor nodes running the prebuilt Sensor application.

4. Required Hardware

Embedded Devices (common)

- 2 CC1310 Launchpad http://www.ti.com/tool/launchxl-cc1310)
- Optional – One LCD boosterpack (http://www.ti.com/tool/430boost-sharp96)

HW Combination 1: Host Machine BBB - (running the prebuilt applications directly on the Beagle Bone Black)

- Beagle Bone Black (https://beagleboard.org/black)
- 8GB SD Card (the TI Processor SDK image requires at least 8GB of space)
- A means to configure & setup the Beagle Bone Black SD Card (Windows or Linux Machine)
- A computer to host/run the Web Browser to view the example application
- In some configurations, a standard “wifi-router” might be required [see below for details]

Hardware combination 2: Host Machine Linux – (running the prebuilt applications on a Linux x86 Development host)

- Ubuntu 14.04 LTS – 64bit Linux Machine

5. Program the CC1310 Launchpads

To run the example application we need to first program one CC1310 Launchpad with MAC CoProcessor Hex file and other(s) with the sensor example application hex file. There are two ways to program the CC1310 with the desired hex files. In this quick start guide we use the Flash Programmer 2 tool running on windows machine. Developers can also use the serial flash programmer tool as described in the TI-15.4 Stack Developers Guide at /doc under the TI-15.4 Stack Linux installation directory to program the desired hex image on to CC1310LP.

Trouble shooting note: It is easy to get the Sensor & CoProcessor devices mixed up – be sure to label the devices as they are programed.

- Download and install the TIMAC SDK from - http://www.ti.com/tool/TI-15.4-STACK
- Download and install TI SmartRF Flash Programmer 2 http://www.ti.com/tool/flash-programmer
- **Program the CC1310LP #1**
  This will run the "CoProcessor" example application, label this device as "Collector"
  The LCD Booster Pack is not supported in the CoProcessor application.
  From Windows PC use SmartRF Flash Programmer2 to program a CC1310 LP MAC coP with "coprocessor_cc1310_lp.hex" file located here:
  ```
  C:\ti\simplelink\ti-15.4-stack-sdk_2_00_00_xx\examples\hexfiles
  ```

- **Program the CC1310LP #2**
  This will run the example "Sensor Application", label this device as "Sensor"
  Optional: connect the LCD Booster pack to this Launchpad.
  From Windows PC use SmartRF Flash Programmer2 to program a CC1310 LP MAC coP with sensor_default.hex file located here:
  ```
  C:\ti\simplelink\ti-15.4-stack-sdk_2_00_00_xx\examples\hexfiles\default
  ```

Important: The default hex files are built for the 915MHz band operation, to rebuild the hex files for other bands (ie: 868MHz ETSI band) please refer to:

* The Embedded Developers Guide or Quick Start Guide installed with the Windows TI-15.4 Stack installer at the location C:\ti\simplelink\ti-15.4-stack-sdk_2_00_00_xx\docs

* And the Linux SDK Developer's Guide, specifically the Example Collector Application configuration section to change the Linux example application.

Trouble shooting: If the devices (sensor/co-processor) are mixed up use the Flash Programer2 to verify the flash content: Un-check ERASE, un-check PROGRAM, and **ONLY** enable the VERIFY option along with the "read-back" feature to double-check or double-verify the flash operation.

6. Software Setup

Depending on the hardware setup combination being used: hardware combination 1 or hardware combination 2 please follow the steps in section 6.1 or 6.2 respectively to get the desired software setup to run the applications as described in section 7.

6.1 Hardware Combination 1

This section describes the required steps if running the application using the hardware combination 1.

6.1.1 BeagleBone Black SD Card Image

Program the SD card with the Processor SDK image.

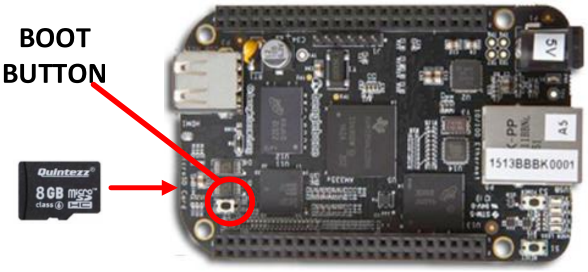
- Download the prebuild TI Processor SDK SD card image "am335x-evm-linux-02.00.02.11.img.zip" from http://software-dl.ti.com/processor-sdk-linux/esd/AM335X/latest/index_FDS.html
- Follow the instructions on the wiki-page to program the microSD memory card:
  - Using Windows Machine:
    http://processors.wiki.ti.com/index.php/Processor_SDK_Linux_Creating_a_SD_Card_with_Windows
  - Using Linux Machine:
    http://processors.wiki.ti.com/index.php/Processor_SDK_Linux_create_SD_card_script

6.1.2 Boot the Beagle Bone Black

Boot the BBB from the SD card:

- Boot from the SD Card, do the following:

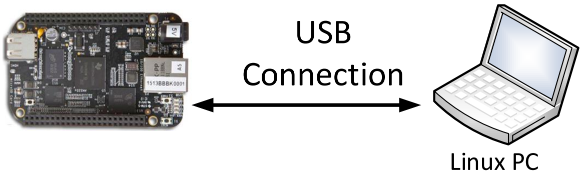
Step 1: Disconnect power and unplug USB cable from the Beagle Board
Step 2: Insert the SD Card into the BBB
Step 3: Press (and hold) the "Boot Switch"
Important: The boot switch is only detected at the initial power on.
Step 4: Provide power to the BBB – (1.5amp, 5V)
Step 5: Wait a few seconds – then Release the Boot Switch
Step 6: In about 5 to 15 seconds the LEDs will start blinking
Note: The first boot from a freshly formated SD card takes about 1 to 2 minutes longer to boot, during this extended time the BBB Linux distribution is performing some one-time-only steps.



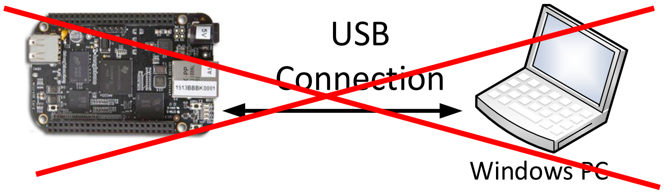
6.1.3 Determine Beagle Bone Black Network Address

There are 4 possible network permutations: (USB or NETWORK cable) times (WINDOWS or LINUX) = 4, please choose the combination that applies to your environment. All four combinations are shown below:

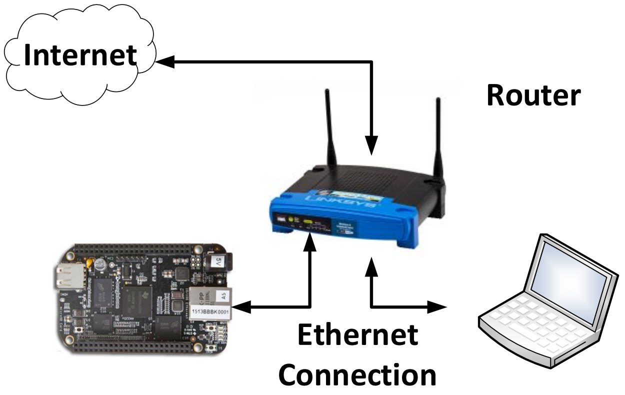
- **Connecting BBB via USB to Linux Host:** Using the TCP/IP over the USB Cable (aka: Linux USB Gadget Interface) - the USB BBB configuration always uses IP address 192.168.7.2 for the BBB (this address is hard coded in the TI Processor SDK sd-card image)



- **Connecting BBB via USB to Windows Host:** This method is not currently supported (due to windows drivers)



- **Connecting BBB via Network Router + Using a Linux or Windows Host:** This method is supported



In this configuration, IP address of the BBB can be determined in two ways:

- **Method 1**: Use the FTDI cable to connect via the serial header on the BBB and use ifconfig to determine the IP address allocated to the BBB.

- **Method 2**: To determine the IP address of the Beagle Bone Black:
  Step 1: Most routers include a built in web server to configure the device.
  Step 2: Connect the Beagle Bone Black to the router
  Step 3: Boot the Beagle Bone Black
  Step 3: Find the "DHCP Client" page, to determine the IP address of the Beagle Bone Black, below are some examples, the generic name for this feature is the: "DHCP Client Table"

Trouble Shooting Tip: The DHCP IP address is often determined by the order in which the devices boot. If your Laptop booted first, it might receive address: xx.xx.xx.100. The BBB boots second, it receives: xx.xx.xx.101 – however tomorrow when you return to the office, or if you attach another device (ie: your cellphone or tablet?) the resulting boot order might change, and thus the IP address might change.

Brand	Example Link
LinkSys	http://www.linksys.com/us/support-article?articleNum=139502

NetGear	http://documentation.netgear.com/fvs336g/enu/202-10257-01/FVS336G_RM-11-07.html
Belkin	http://www.belkin.com/pyramid/AdvancedInfo/F5D8235-4/Advance/reserveIP.htm

6.1.4 Copy the Prebuilt files to the BBB via SCP

Copy the “bbb_prebuilt.tar.gz” file to the BBB via SCP (please change the ‘xx’ below as needed)

Linux: The Prebuilt files can be found here

```
    ${HOME}/ti/simplelink/ti15.4stack_linux_64_02_00_00_xx/prebuilt
```

On Windows:

```
    C:\ti\simplelink\ti-15.4-stack-sdk_2_00_00_xx \examples\
     linux\ti15.4stack_linux_x64_2_00_00_xx-bbb_prebuilt.tar.gz
```

On Linux, the command is (recall that: ${SDK_ROOT} is the Linux SDK installation directory, substitute the appropriate address for: ${BBB_IP_ADDRESS}

```
        bash$ cd ${SDK_ROOT}/prebuilt
        bash$ scp bbb_prebuilt.tar.gz root@${BBB_IP_ADDRESS}:~/.
```

In the above command, the final: “~/.” – is a short hand notation for the “home” directory of the user, in this specific case the directory will be: /home/root

On Windows, the tool: WinSCP, Teraterm, or FileZilla can be used to transfer the file

6.1.5 Log into the BBB via SSH (get a shell prompt)

Step 1: Login to the BBB via SSH

Windows:

Use Putty, or Teraterm to make an SSH connection to the Beagle Bone Black

Linux:

```
    bash$ ssh root@${BBB_IP_ADDRESS}
```

This connects to the Beaglebone and provides the root (#) prompt, specifically: `root@am335x-em#`

Step 2: Unpack the “TAR” file

```
    root@am335x-evm# cd ${HOME}      (where the tar file is located)
    root@am335x-evm# tar xf bbb_prebuilt.tar.gz
```

Step 3: The Prebuilt binaries will be found in the “prebuilt” directory

Specifically: /home/root/prebuilt

6.2 Hardware Combination 2

This section describes the required steps if running the application using the hardware combination 2.

6.2.1 Linux Host Software Setup

- From the windows machine copy the TI-15.4 Stack Linux SDK Installer located at "*C:\ti\simplelink\ti-15.4-stack-sdk_2_00_00_xx\examples\linux*" to the x86 machine running Ubuntu OS (64-bit, Version 14.04 LTS). Then install the TI-15.4 Stack Linux SDK by going to the directory where the file was copied, and executable the following commands as a "normal user" (do not perform these steps as the ROOT user).

  ```
  bash$ cd ${where_the_run_file_is_located}
  bash$ chmod +x ti15.4stack_linux_x64_02_00_00_xx.run
  bash$ ./ti15.4stack_linux_x64_02_00_00_xx.run
  ```

  Note: "xx" is the build sequence number and can be ignored.

  The default TI-15.4 Stack install directory is:

  ```
  ${HOME}/ti/simplelink/ti15.4stack_linux_64_02_00_00_xx
  ```
- The prebuilt Linux applications are in the ${SDK_ROOT}/prebuilt directory
- Install the package: build essentials

  ```
  bash$ sudo apt-get install build-essential
  ```
- Install the package: NodeJS

  ```
  bash$ sudo apt-get install nodejs
  ```
- Your user name must be a member of the group "dialout"

  ```
  bash$ sudo adduser $USER dialout
  ```
- Optional (this can be done later): To cross compile for Beaglebone black install the TI processor-SDK-Linux-am335x from http://www.ti.com/tool/PROCESSOR-SDK-AM335X . The TI AM335x Linux SDK contains all of the cross compilation tools, headers, libraries and other required files for cross compiling to the Beagle Bone Black. See the TI-15.4 Stack Linux Developer guide for more details.

6.2.2 Linux Host Prebuilt Directory

Trouble shooting note: All of these steps are performed as a normal user. Do not use the "root" user to perform these steps.

Step 1: Change to the SDK Installation Directory

```
bash$ cd ${SDK_ROOT}
```

Step 2: Change to the Linux Prebuilt Directory

```
bash$ cd prebuilt
```

7. Running the Application

7.1 Connect the CoProcessor Launch Pad

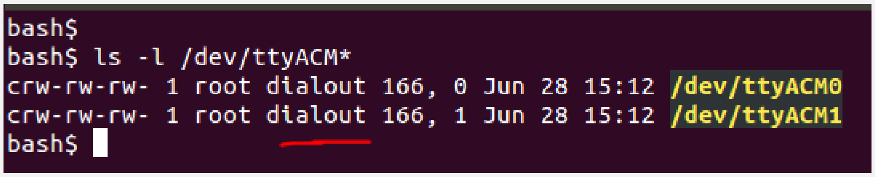
Plug the CC1310 running the CoProcessor (Coprocessor Launchpad) to the x86 machine or the BBB.

BBB Example:

Trouble Shooting Notes:

1) Is the device /dev/ttyACM0 present?

Check via this command: `bash$ ls –l /dev/ttyACM*`



```
bash$
bash$ ls -l /dev/ttyACM*
crw-rw-rw- 1 root dialout 166, 0 Jun 28 15:12 /dev/ttyACM0
crw-rw-rw- 1 root dialout 166, 1 Jun 28 15:12 /dev/ttyACM1
bash$ 
```

The Launchpad presents as two USB serial ports named: `/dev/ttyACM<somenumber>` typically these are “ACM0” and “ACM1”, and are a member of the group: “dialout”

The example Collector application uses a configuration file located here:

```
${PREBUILT}/bin/collector.cfg
```

The default configuration file uses assumes the Launch Pad is: `/dev/ttyACM0`

Sometimes the simple solution is to edit/change the ${PREBUILT}/bin/collector.cfg file

2) Multiple /dev/ttyACM devices? (Reason #1)

In some cases, an application may have “crashed” or the Launch Pad USB cable was unplugged and plugged back in. This can cause the application to hold on to a reference to an existing /dev/ttyACMx device name, when the USB device was removed and re-inserted – that old name was still in use. Linux will then use the “next available number” – ie: /dev/ttyACM2 or /dev/ttyACM3 … etc.

Another example is: “Closing your laptop lid” or “suspending your Virtual Machine” and later resuming

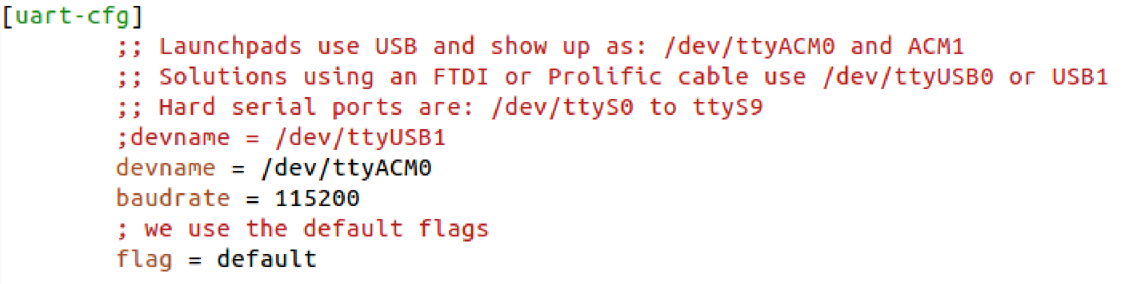
The easiest solution is to reboot the machine {the other method is to kill various process via “kill”}

3) Multiple /dev/ttyACM devices – Reason #2

For the Beagle Black (Option 1) the Launch Pad is typically the **ONLY** device that is present, and thus, the launch pad will generally appear as /dev/ttyACM0 and /dev/ttyACM1

For the Linux X86 (Option 2) it is common to have many things plugged into your Linux Development host machine, these other devices might also present a /dev/ttyACM<somenumber> - for example the Beagle Bone Black when connected via USB to your Linux machine can be configured to present an /dev/ttyACM<somenumber>

If required – a simple solution is to edit the "${PREBUILT}/bin/collector.cfg" file, see below:



```
[uart-cfg]
        ;; Launchpads use USB and show up as: /dev/ttyACM0 and ACM1
        ;; Solutions using an FTDI or Prolific cable use /dev/ttyUSB0 or USB1
        ;; Hard serial ports are: /dev/ttyS0 to ttyS9
        ;devname = /dev/ttyUSB1
        devname = /dev/ttyACM0
        baudrate = 115200
        ; we use the default flags
        flag = default
```

4) **Wrong Permissions**?

For the Beagle Bone Black (Option 1) – the application generally runs as "root" – thus there is never a permission problem.

For the Linux X86 (Option 2) – the application normally runs as a "normal user" it is easy to overlook the "adduser" step.

```
bash$ sudo adduser $USER dialout
```

Remember that after adding the DIALOUT group, the change does not get propagated to other existing shell windows automatically. The easy solution is to reboot the machine and log in again.

7.2 Starting the Application – Start the network

In the PREBUILT directory is a simple shell script called: "run_demo.sh"

```
cd ${PREBUILT}          (change to the prebuilt directory)
bash ./run_demo.sh      (run the demo)
```

Beagle Bone Black (Option 1): The shell script will print the URL for your browser, select the appropriate URL and cut/paste the URL into your browser (see trouble shooting below for more details)

The Linux X86 (Option 2): The script will launch your browser automatically

Trouble shooting:

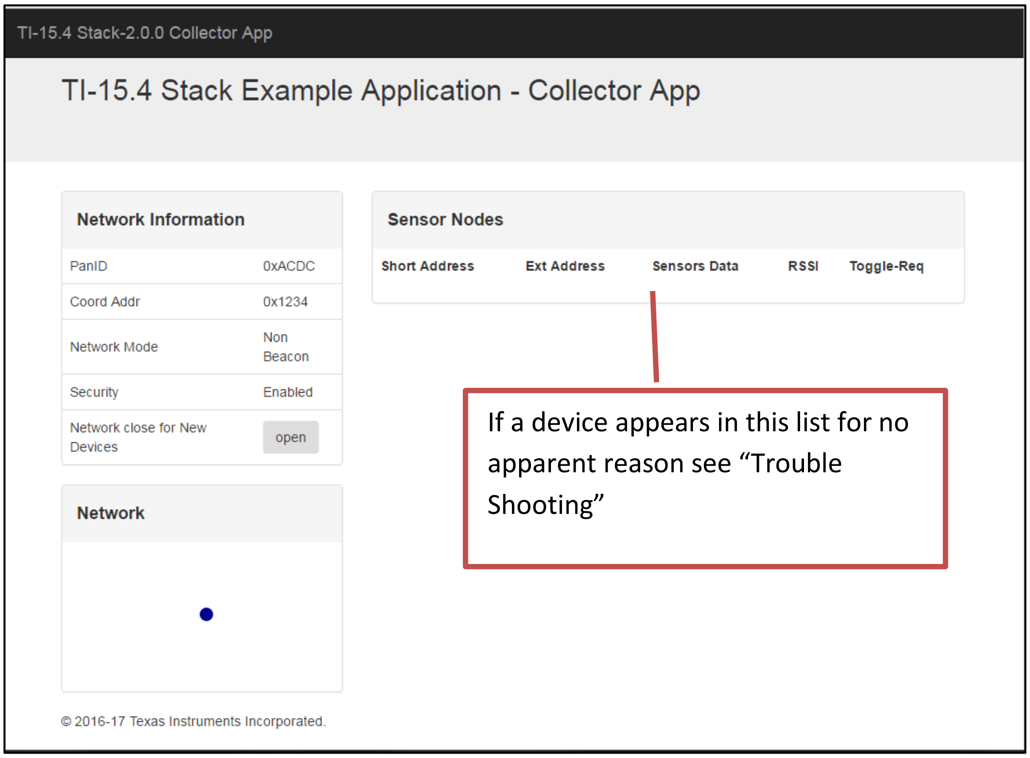
Error Messages? The application should start the applications in the background and eventually return to the bash prompt. During this process the software should print the SW Version of the CoProcessor application. If this does not occur – or if there are other error messages printed please review the “Trouble Shooting” steps earlier in this document.

Which URL to use? (Beagle Bone Black, Option 1) The Beagle Bone will print two different URL addresses. The “192.168.7.2” URL – is specific to the Linux only USB interface. The other URL is for the wired interface, and should be the ${BBB_IP_ADDRESS} used earlier.

7.3 The Gateway Application Web Page – Open Network for new device joins

The example web page is shown below:

Initially, the example application starts with (a) no devices present, and (b) the network is closed and will not accept new devices.



TI-15.4 Stack-2.0.0 Collector App

TI-15.4 Stack Example Application - Collector App

Network Information

PanID	0xACDC
Coord Addr	0x1234
Network Mode	Non Beacon
Security	Enabled
Network close for New Devices	open

Sensor Nodes

Short Address	Ext Address	Sensors Data	RSSI	Toggle-Req

Network

© 2016-17 Texas Instruments Incorporated.

If a device appears in this list for no apparent reason see “Trouble Shooting”

Figure 1: TI-15.4MAC gateway application web-application served by the local web-server after network startup

Trouble Shooting:

About the Port Number

Beagle Bone Black (option 1 only): The default "web server" service appears on Port 80, the Beagle Bone Black is running two demos: Demo #1 – is on the default service port (port 80) – this is a demo for the Processor SDK package.

Both Option 1 & Option 2: The TI 15.4 Stack SDK example gateway application uses port 1310 – thus be sure to use the ":1310" in the URL you use to connect to the gateway, for example the URLs would be:

http://192.168.7.2:1310	{Option 1 – Beagle Bone via USB interface}
http://${BBB_IP_ADDRESS}:1310	{Option 1 – Beagle Bone via Wired Interface}
http://localhost:1310	{Option 2 – Linux Only}
http://${LINUX_IP_ADDRESS}:1310	{Option 2 – Linux Only}

Device List & Restoring Previous State

The first time the application runs there are no devices in the Device List.

If the Linux example application is run a second time, the collector application will restore to the previous state by reading the file: ${PREBUILT}/bin/nv-simulation.bin. Thus, sensor devices may initially appear in the list of devices even though it is not actually present or powered off.

To Reset the Devices (Linux Collector and Sensor Application)

To reset the "Linux Collector Application" – simply remove the nv-simulation.bin file and restart the demo application.

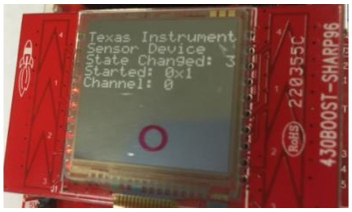
The same "restore" operation applies to the example embedded sensor applications, if they are reset or power cycled they will "resume" to their previous state. To reset the embedded sensor application on the sensor Launch Pad Step 1: press and hold BTN2, Step 2: press and release the RESET button, then Step 3: release BTN 2

7.4 Joining the devices to the network

At startup, the collector application initially has the network 'closed' thus sensor devices cannot join.

To open the network click the "Open" Button the on the web-browser. Within a few seconds (the time is depending upon the polling interval and other configuration settings) the sensor will join the network.

Once the device joins the network the RED led turns on. If the sensor launchpad has an LCD module, it will indicate the current state on the LCD: State 1 = Not joined, State 3 = Joined, State 4 = Restored, State 5 = Orphan condition. More details can be found in the embedded documentation.

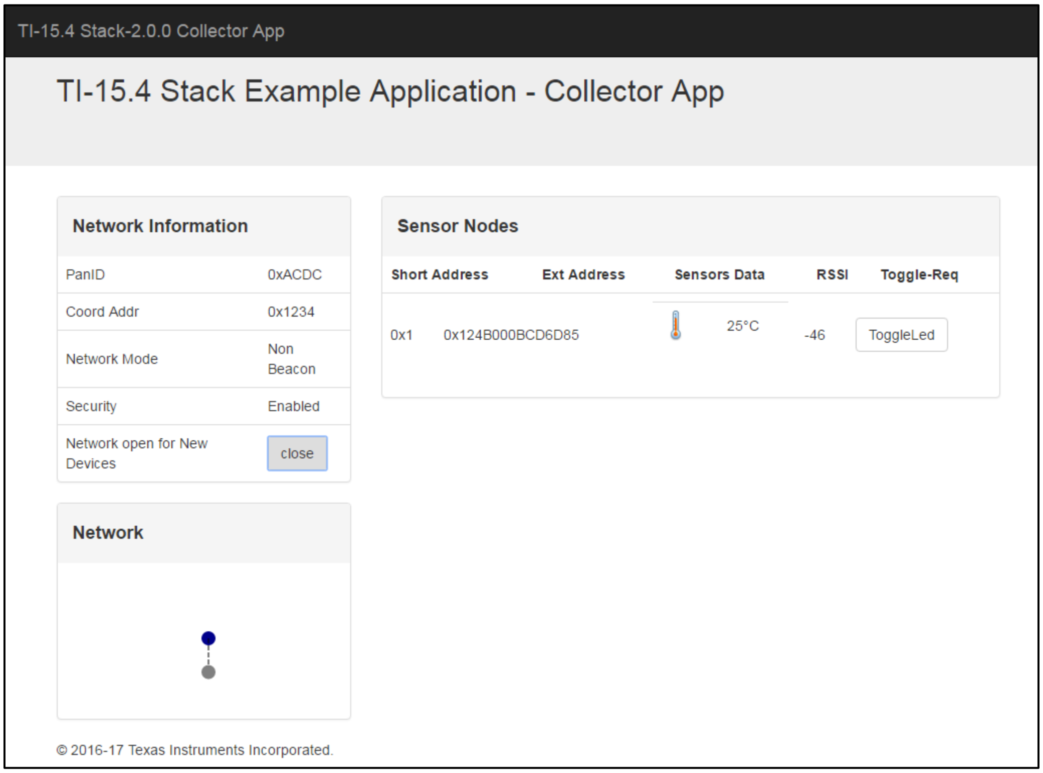


7.5 Data Communication

After the new device appears, initially only the short & extended address will appear. The data fields with be displayed as --- (hyphens) indicating no data.

Sensor Data reports: After about 1 minute data should appear on the screen (the exact interval is configured in the collector application via a #define value) see the TI-15.4 Stack Embedded Developers Guide and/or the Linux Example Collector source code for more details. The sensor nodes after this time periodically report the sensor data.

Actuation: Click on the Toggle LED button will send a message to the Sensor module to toggle the LED. There may be slight delay (few seconds) in toggle operation on the desired sensor LaunchPad. This is because the sensor nodes are "sleepy" and only wake up periodically to the poll and get the command buffered on the collector.



TI-15.4 Stack-2.0.0 Collector App

TI-15.4 Stack Example Application - Collector App

Network Information

PanID	0xACDC
Coord Addr	0x1234
Network Mode	Non Beacon
Security	Enabled
Network open for New Devices	close

Network

Sensor Nodes

Short Address	Ext Address	Sensors Data	RSSI	Toggle-Req
0x1	0x124B000BCD6D85	25°C	-46	ToggleLed

© 2016-17 Texas Instruments Incorporated.

Figure 2: TI-15.4 Stack gateway application web-application after device has joined to the network

8. Next Steps

For additional information developers should refer to the TI-15.4 Stack Linux Developers Guide installed with the TI-15.4 Stack Linux SDK and also for details on the stack operation, setting up the packet sniffer, etc please refer to TI-15.4 Stack Developers Guide installed with the TI-15.4 Stack Windows Installer. Also refer to other documents included with the TI-15.4 Stack SDK installation.

Other useful links:

- Find answers to your questions and common issues, post your questions and answer questions from other developers at the TI e2e forums: http://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/

- TI-15.4MAC Wiki (www.ti.com/ti-15.4-stack-wiki)

Thank you for being a customer of Texas Instruments and evaluating our products.