Brian Kiaer

Vincent Tuason

Dr. Venkatesan Muthukumar

CPE 403 -1001

December 8, 2018

Final Project

**Problem Statement:**

The goal of this project is to create a Wireless Sensor Network that implements a star network topology using two SimpleLink CC1350 boards and one BeagleBone Black board. One CC1350 board will act as a transmitter and the other CC1350 will be connected to the BeagleBone Black and will be used as a co-processor module.

The objective of this project is to create a deeper understanding of creating real life applications that can implement the use of the CC1350 microcontrollers and the BeagleBone Black board. By creating a Wireless Sensor Network (WSN) we will able to understand the functionalities of all three boards working in unison simultaneously. Furthermore, the objective goes into understanding C based code for the microcontrollers while setting up a host system for embedded application development. Next, the BeagleBone Black will have to be set up for application development as well and applications for the BeagleBone Black will have to be developed in C/C++.

Problem, goal, objectives, and outcomes with project block diagram

**pre-requisites:**

Components:

* Two SimpleLink Dual-Band CC1350 LaunchPad Kits.
* One BeagleBone Black
* 32GB microSD
* 2 micro USB cables
* 1 mini USB cable
* TSL2561 Lux Sensor
* Ethernet Cable

Software:

* UniFlash
* Code Composer Studio
* TeraTerm/PuTTy/Terminal
* Debian/Ubuntu
* Win32 Disk Imager

Components, tools, software used in the design, install steps etc.

**implementation details:**

Step 1:

The first part in creating the Wireless Sensor Network is to program one of the CC1350 LaunchPads to act as a “Sensor”. To do this a program called UniFlash provided by Texas Instruments will be used. Prior to programming the Launchpad, we made sure that the Launchpad had been erased by going to Settings & Utilities > All Unprotected Sectors > Erase Entire Flash. After, the image that corresponds to the Software Development Kit (SDK) is loaded into UniFlash and will program the CC1350 as the Sensor Launchpad.

Step 2:

The next step is to program the other Launchpad to become the “MAC-Cop” Launchpad. This will act as the co-processor module. The same steps will be performed, similar to step 1. However, a different image file within the SDK will be used named coprocessor\_<target\_board>.hex this will also program the second Launchpad using UniFlash. These steps are imperative to creating the star topology mentioned earlier.

Step 3:

Step 3, will consist of installing the SDK for the Linux Gateway. The installer can be found on the Texas Instrument website using valid credentials. After downloading the installer, the next step is to go on the linux terminal and running the following commands:

chmod +x ti15.4stack\_linux\_x64\_x\_xx\_xx\_xx.run

./ti15.4stack\_linux\_x64\_x\_xx\_xx\_xx.run

This will run the Linux Gateway SDK installer which is imperative to bridge a connection between the boards.

Step 4:

Step 4, increased the difficulty of the project. The next step was to flash the am335x image file onto an SD card. This image will allow us to boot our BeagleBone Black to “Arago Project” seen below:



Fig. 1.

After logging in as the root user (by typing root) the next step is to find the assigned IP address that corresponds to the Ethernet port connection:

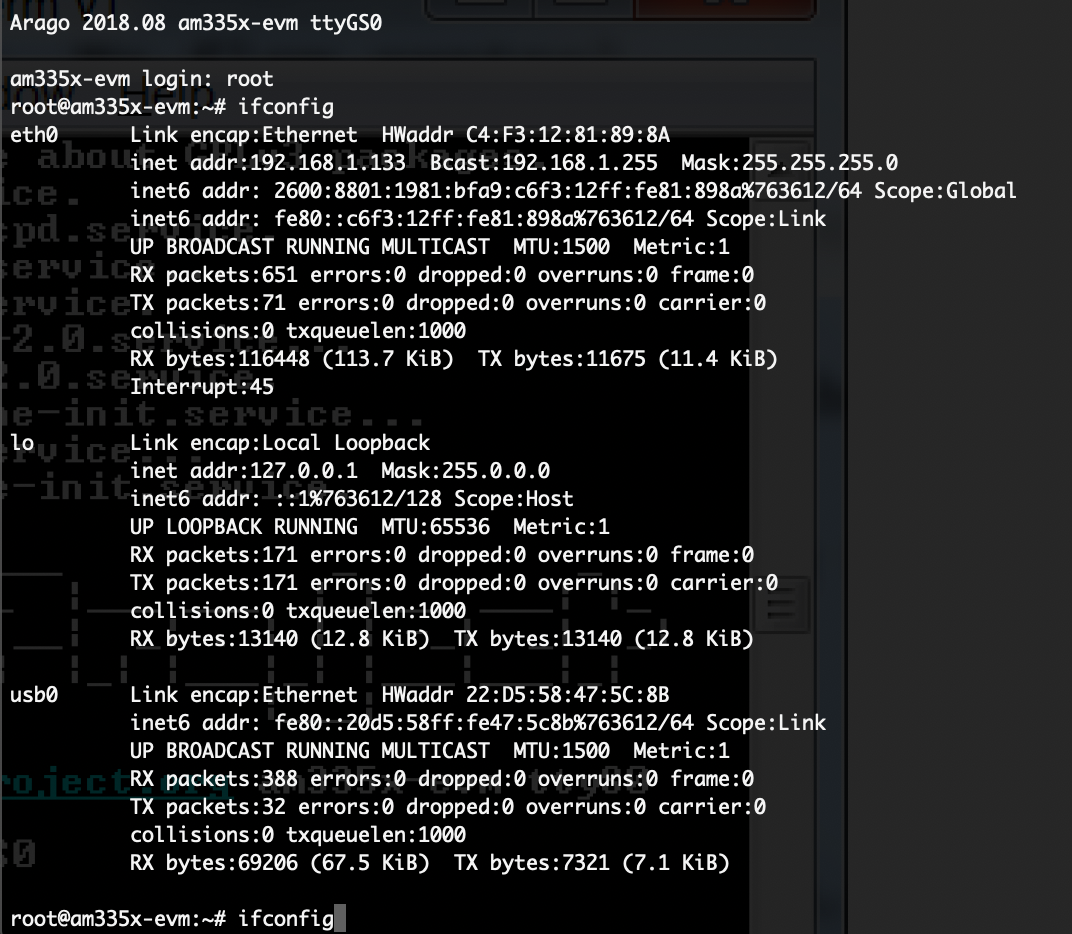


Fig. 2.

We can see that on the left hand side of the terminal, there is a category called “eth0” this will be used to find the IP address. The section called “inet addr” will tell us where the IP address is, which in this case is : 192.168.1.133. Once the IP address has been found, we next have to login using another Terminal window by using the login root@192.168.1.133 with no password.

After logging into the BeagleBone with Secure Shell, the .tar.gz file that was uploaded on our Ubuntu Virtual Machine will need to be copied onto it.

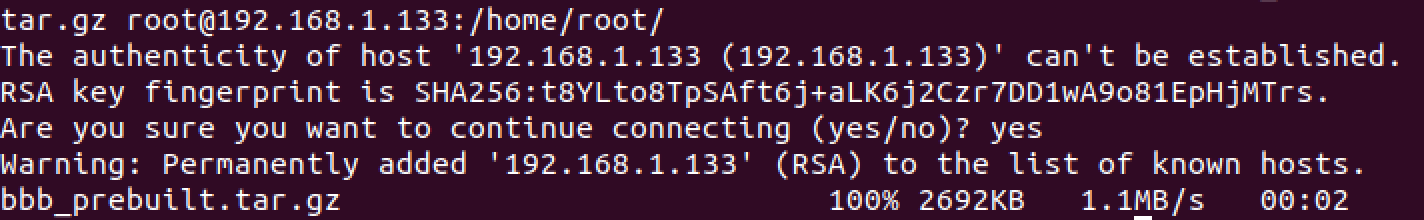


Fig. 3.

Step 5:

The next step is to add the MAC CoProcessor CC1350 Launchpad onto the BeagleBone Black via USB A to micro-USB cable. After plugging in the device, the next command to run would be ls -l /dev.ttyACM\*

This command will show if the MAC CoProcessor was successfully connected to the BeagleBone Black. Lastly the next step is to run the run\_demo.sh script file.

\*Due to changes throughout the distribution of the programs. Modifications have been created for the script file due to incorrect paths. We were able to change the gateway path to home/gateway/gateway. It is one level further now. Once, that script has been executed a window like this should appear:

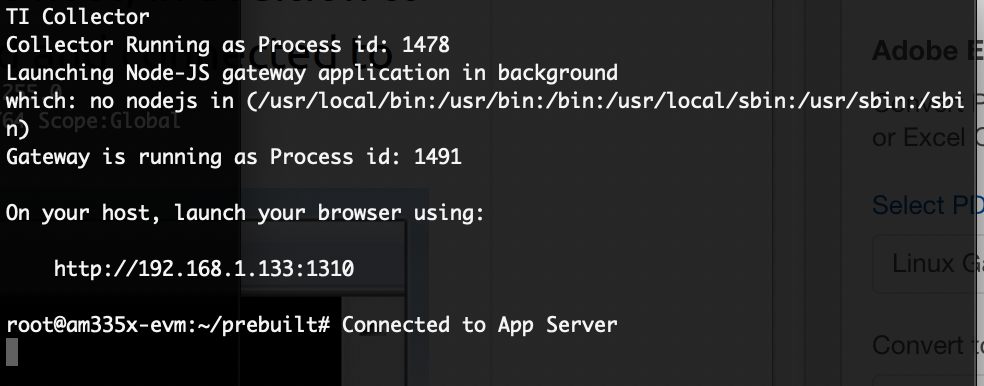


Fig. 4.

After running the script file run\_demo.sh, the unit will connect to the App Server and provide a link to the TI Collector App using an id number and the IP number. The example above is <http://192.168.1.133:1310> when visiting the link, the appropriate webpage appears:

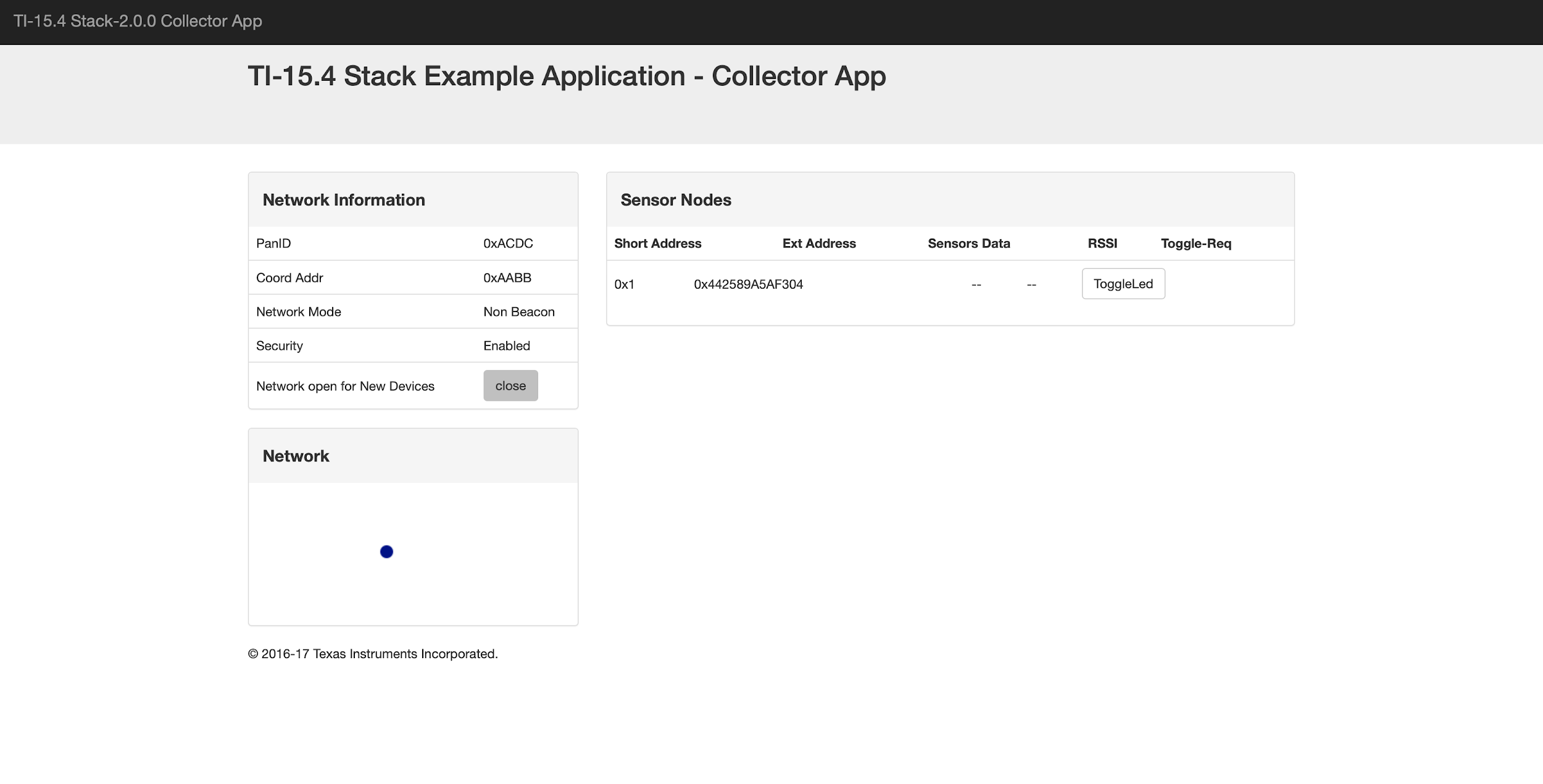


Fig. 5.

Step 6:

After this, we had to add a Sensor to the TI-15.4 Collector App. By doing this, we used the Sensor TI CC1350 Launchpad that was previously flashed using the UniFlash software. Once powered on, resetting the device is necessary for it to appear on the Collector App.

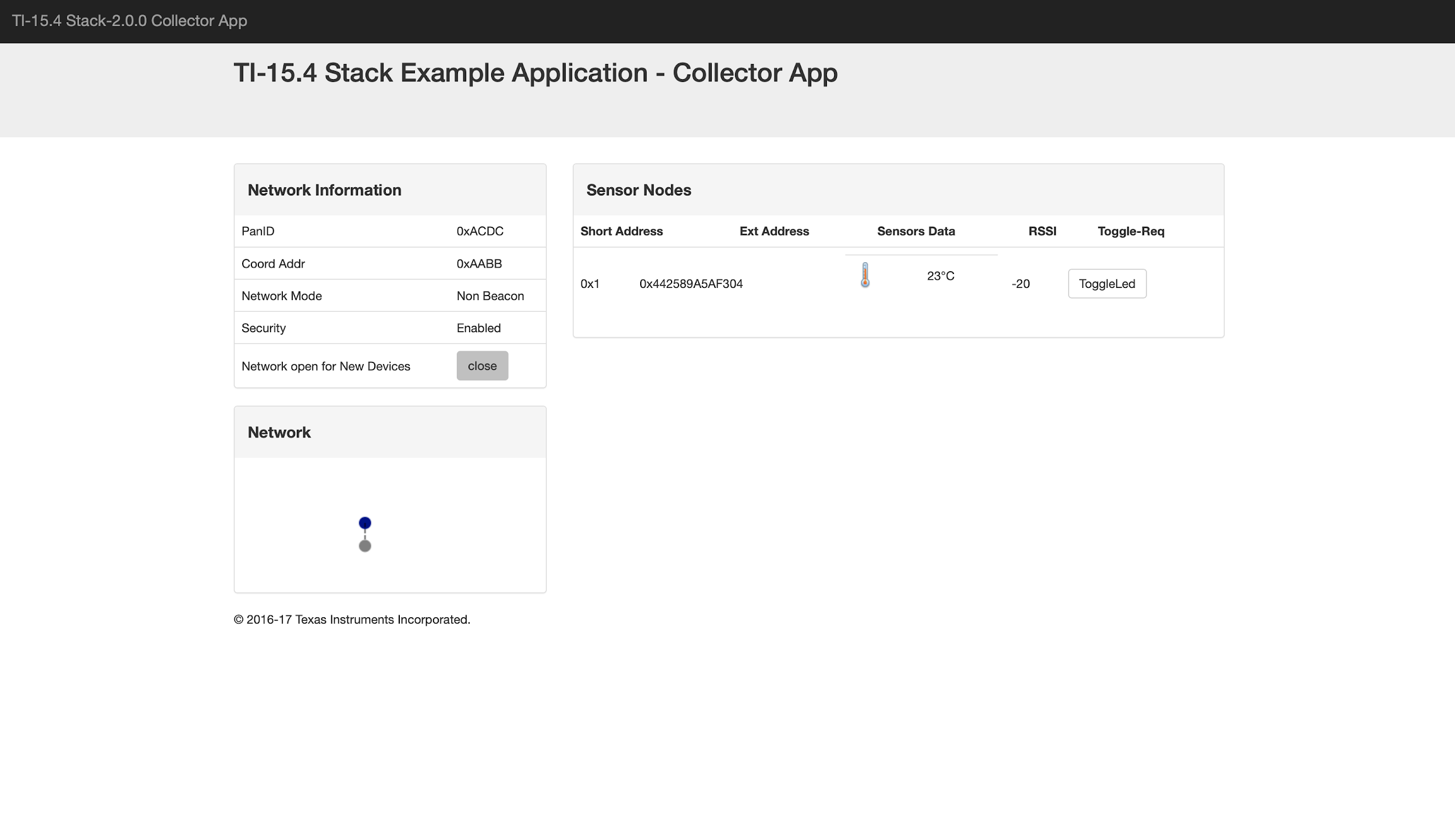


Fig. 6

This is the final step when re-creating the TI-15.4 Gateway project. This is imperative for the final project and is necessary for understanding the fundamental design of a Wireless Sensor Network.

Step 7:

The next step is to add sensor values to the existing TI 15.4 Collector Application. One of the more fundamental sensors we have been using and are able to implement would be the TSL2561 Lux Sensor. This sensor has the capabilities of recording light (lux) values using I2C interface. What we had to do was first add onto the existing project an interface that allows I2C communication to be active. This was done within the sensor.c file in the sensor\_cc1350lp project. I2C libraries and fundamental initializing functions have been incorporated. Once that is done, the next step was to create a predefined symbol in the project properties in order for the sensor.c file to recognize that the Lux Sensor icon has been activated. Below shows an example of the lux sensor turned on and functioning:

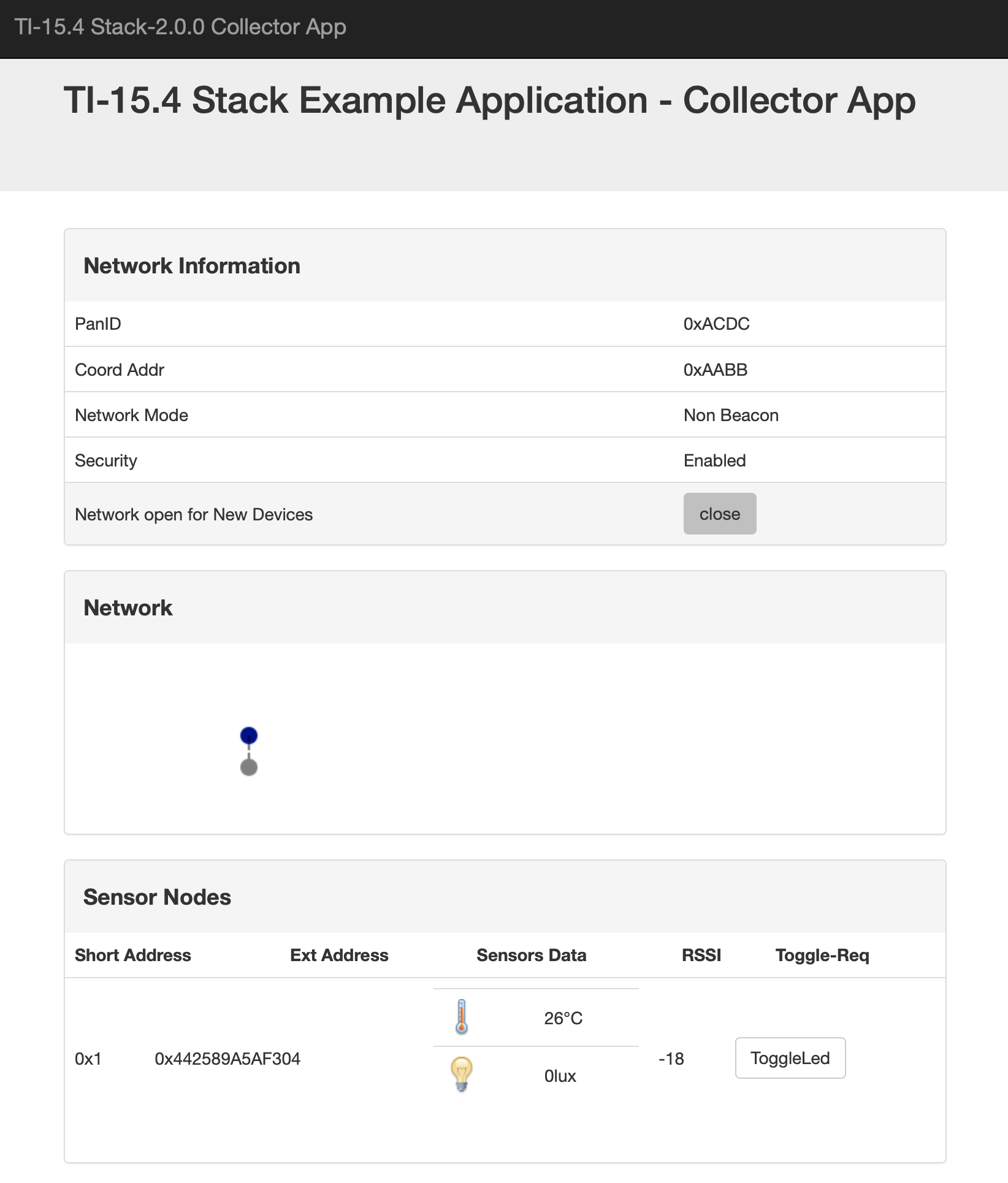


Fig. 7.

**outcomes, results and conclusions:**

Table 1

[This sample table is formatted to follow MLA guidelines. To add a new table, on the Insert tab, tap Table. When you create a new table in this document, it will automatically use MLA formatting.]

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Heading | Column Heading | Column Heading |
| Row Heading |  |  |  |
| Row Heading |  |  |  |
| Row Heading |  |  |  |

Source: [This source text uses a style named Table Source.]

1. [This note text uses a style named Table Note. Table notes use a lowercase letter instead of Arabic numerals to differentiate them from the notes to body content.]

**reference:**

1. “Adding New Sensor Support To Sub1GHz Sensor To Cloud Linux Gateway.” *MSP430 LaunchPad Tutorials - Texas Instruments Wiki*, processors.wiki.ti.com/index.php/Adding\_New\_Sensor\_Support\_To\_Sub1GHz\_Sensor\_To\_Cloud\_Linux\_Gateway.
2. 三妹 . “Basic Example to Use OPT3001 on CC2650 LaunchPad.” *Sun May Sky*, 1 Jan. 1970, sunmaysky.blogspot.com/2016/03/basic-example-to-use-opt3001-on-cc2650.html.