

Stryder Loveday | Morgan Kiger

Dr. Venkatesan Muthukumar

CPE403/ECG603 - 84189-1001

December 12th, 2018

TI - 15.4 STACK COLLECTOR APP

PROBLEM STATEMENT:

In this project we used a BeagleBoneBlack and the CC1350 Launchpad to implement a collector app using the star-topology wireless sensor network for remote data monitoring. Our goal was to use three sensor nodes that were setup in a greenhouse to transmit data to a co-processor node and visualize the data using the BeagleBoneBlack via a local web page. The sensors are setup to read the data via I2C.

Goal Outline:

- Setup BeagleBoneBlack to run as the host
 - Increase partition size of host
 - Flash with the sdk prebuilt apps for the TI 15.4-Stack Collector App
 - Visualize the data via a local web page
- Flash an CC1350 to run as the co-processor
 - Connect to the BBB and make sure device is readable by BBB
 - Collect data from the sensor node(s)
- Flash other CC1350 to run as sensor
 - Setup multiple sensor nodes to read data via I2C
 - used sensor TSL2591 and BME280

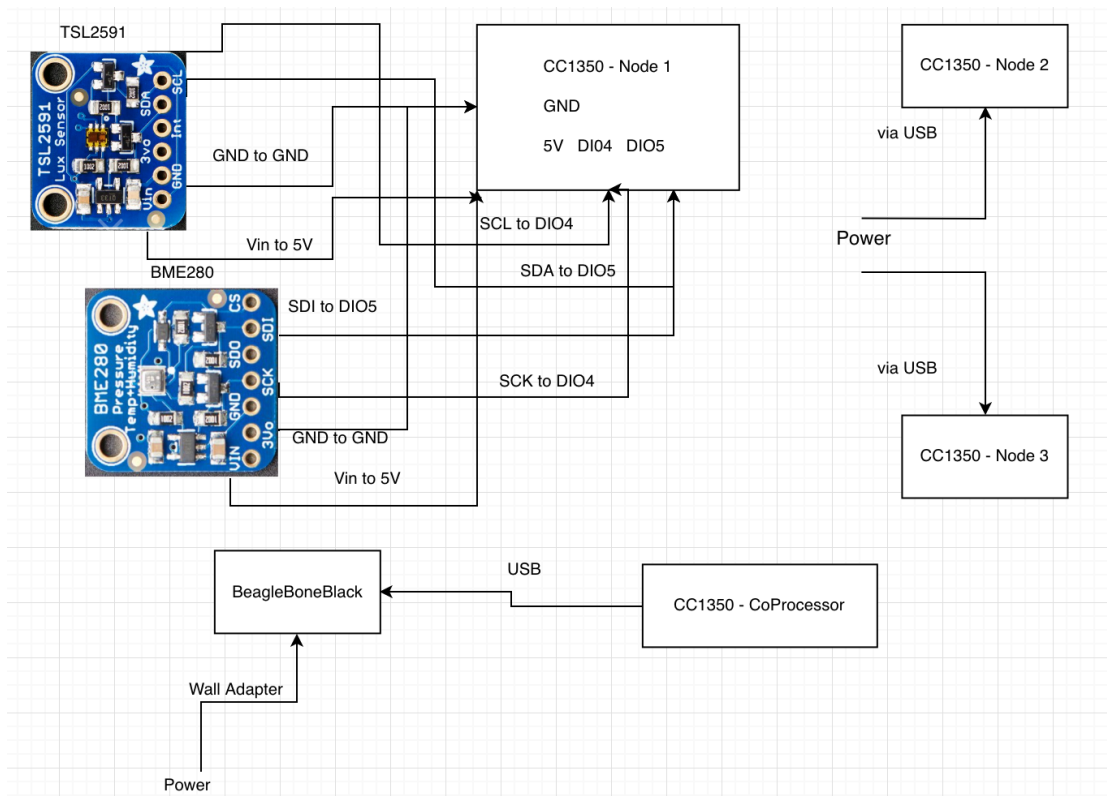
PRE-REQUISITES:

The main components involved in this project are the BeagleBone Black and the CC1350 LaunchPad. The BeagleBone Black is used as the linux host and we use the ssh protocol to connect to the BeagleBone Black which only gives you a terminal screen, no graphical user interface. We initialized the environment by installing all the packages we would need and the prebuilt apps provided via the ti 15.4 stack linux sdk. One of the CC1350 LaunchPads was flashed to run as the co-processor for the sensor data and was connected to the BeagleBone Black. The other CC1350s were flashed to run the sensor modules and send the data via I2C. One of was setup for to read data from the TSL2591 sensor, one was setup to read the data form the BME280 sensor and the last one was setup to read the data from the internal temperature sensor.

The tools used was sd card flasher, virtual machine, TI smart rf flash programmer, code composer studio, and Beagle Bone Black

IMPLEMENTATION DETAILS:

- Steps used in design:
 - In order to implement the co-processor and the BeagleBone Black
 - we had to flash the coprocessor with the out of the box co-processor hex file using SmartRF
 - install the ti 15.4 stack linux sdk to the BeagleBone Black
 - run the provided ./build_all.sh script within BeagleBone Black
 - In order to get our three sensor nodes working
 - code the BME280 sensor to transfer temperature, humidity data
 - code the CC1350 to transfer data from the internal temperature sensor
 - code the TSL2591 sensor to transfer lux data
 - used provided code from Dr. Venki as a starting point
 - In order to link the sensor nodes with the co-processor
 - we had to update the config.h file within the sensor project in CCS to the desired channel and how frequently we wanted it to transfer over data
 - for the co-processor there was a collector.cfg housed on the BBB that held the config settings for the co-processor, we had to make sure the desired channel matched the sensor config.h file
 - We ran the ./run.sh file within the ti 15.4 stack linux sdk to launch the local web page that displayed the data of the sensor nodes
 - opened the coprocessor network, connected the three sensor nodes, and watched the data flow through



OUTCOMES, RESULTS AND CONCLUSIONS:

Table 1

Example data from our three sensor nodes

	Internal Temp Sensor	TSL2591 Sensor	BME280 Sensor
Temperature	12 C and 7 C	N/A	8 C
Humidity	N/A	N/A	46% RH
Lux	N/A	956 Lux	N/A

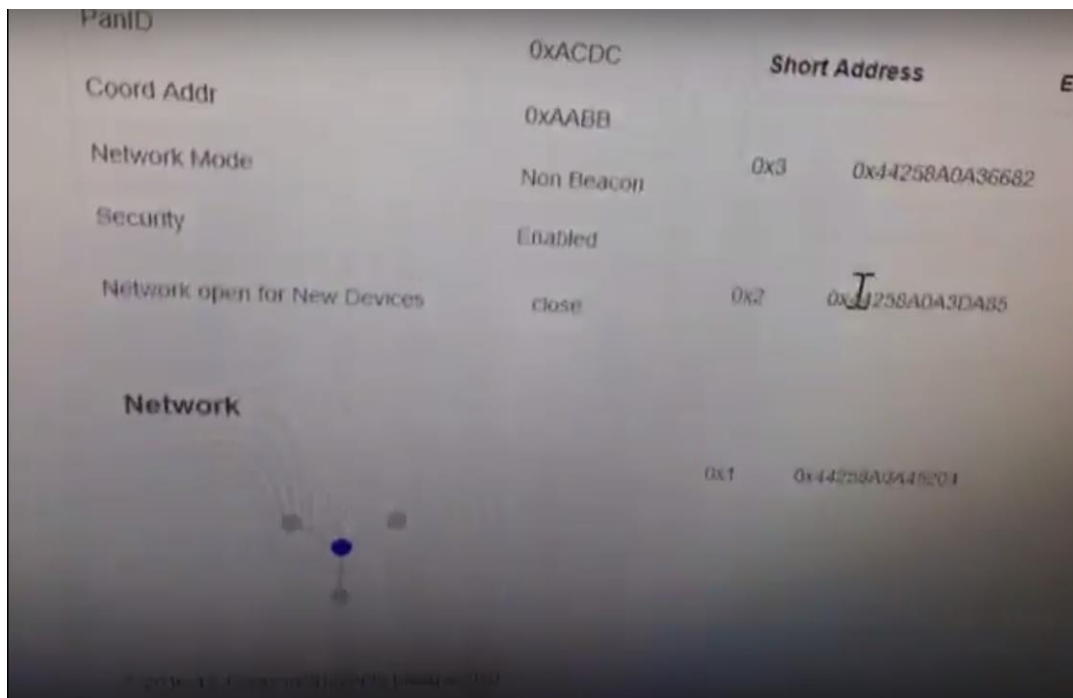


Fig 1: you can see the network connection on the bottom left and the sensor node address on the right

Address	Ext Address	Sensors Data	RSSI	Toggle-Req
0x44258A0A36682		12°C	-78	ToggleLed
0x44258A0A3DA85		7°C	-66	ToggleLed
		46% RH		
0x1	0x44258A0A45204	8°C	.70	ToggleLed
		956lux		

Fig 2: picture of the data displayed on the local web page taken from the sensor nodes

In conclusion, we implemented a star-topology wireless sensor network using three CC1350 as sensor nodes, one CC1350 as the co-processor/collector node and a BBB as the data visualizer. We were able to read and transfer temperature, humidity, and light data and display on a web page. The temperature, humidity, and light data was taken from a greenhouse, while the other two sensors used their internal temperature sensor. One was setup in a garage and the other was setup in a wellhouse. We wanted to implement more sensors like a current sensor for example, however, we did not have enough time to expand/development the framework to get different kind of sensor data. Furthermore, we brainstorm what additional we could have done or how we could have made the project better. Some of the things we came up with are further modularization of sensor code design and expanding the polling functionality to allow additional control settings

REFERENCE:

Instruments, T. (2018). *Example Applications — TI 15.4-Stack 0 documentation*. [online]

Dev.ti.com. Available at:

http://dev.ti.com/tirex/content/simplelink_cc13x0_sdk_1_30_00_06/docs/ti154stack/ti154stack-sdg/ti154stack-sdg/Example%20Applications.html#linux-collector-and-gateway-application

[Accessed 12 Dec. 2018].

abrain (2018). *Sub-1 GHz Sensor to Cloud IoT Gateway - Review*. [online] Element 14.

Available at: <https://www.element14.com/community/roadTestReviews/2590/1/sub-1-ghz-sensor-to-cloud-iot-gateway-review> [Accessed 12 Dec. 2018].