

Cade Echevary

Dr. Muthukumar

CPE403

December 11, 2020

CPE403 – ADVANCED EMBEDDED SYSTEMS

Design Assignment #4

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

Name: Cade Echevary

Email: echevary@unlv.nevada.edu

Github Repository link (root): https://github.com/echevary/MicroController_proj

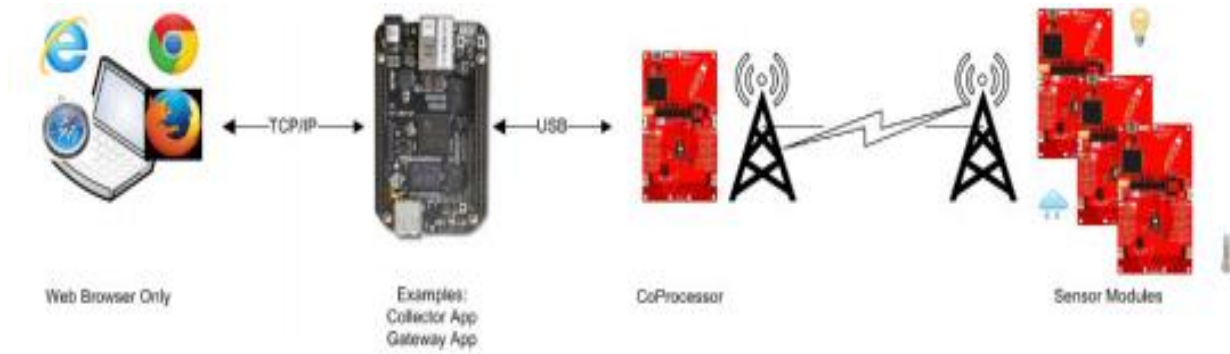
Youtube Playlist link (root):

<https://www.youtube.com/playlist?list=PLx8r8972rBxGc7qQazP1uhUSINSNA5rNC>

Youtube Link:

<https://www.youtube.com/watch?v=xDgviZoGh4I&list=PLx8r8972rBxGc7qQazP1uhUSINSNA5rNC&index=6>

TI 15.4-STACK LINUX SDK WITH CC1352 AND BBB



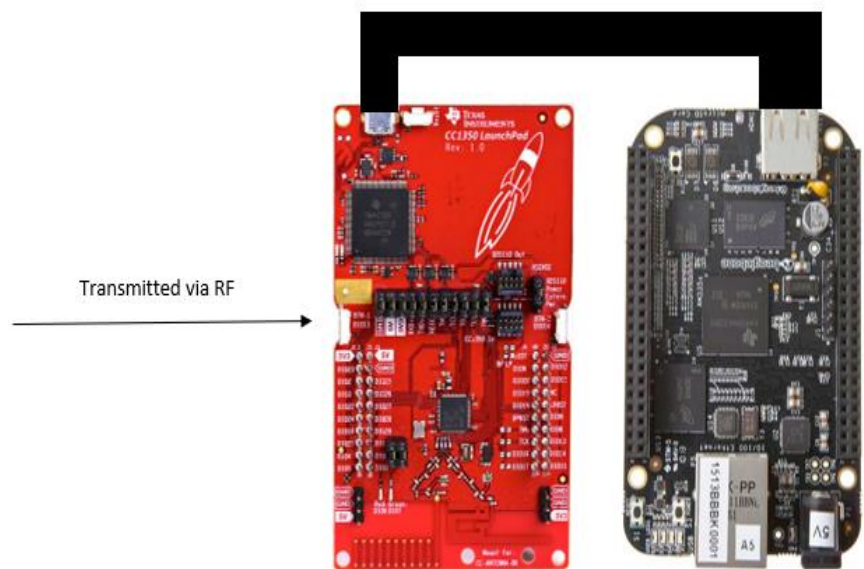
PROBLEM STATEMENT:

Our goal was to use the TI 15.4-Stack Linux SDK to create a star topology network with the BBB and two CC135x launchpads. We needed to use the lpstk to have one act as a sensor launchpad and the cc1352 as a co-processor for the BBB. The BBB would act as the embedded host.

The sensor used was the Si7021 temperature and humidity sensor.

There wasn't much code that needed to be edited. And if it was edited it was things that the project directions told you to do so ill just submit the files.

Diagram



PRE-REQUISITES:

Components used:

- CC1352 – co-processor
- LPSTk used as sensor
- BeagleBone Black – Used as embedded host to run web application to display sensor data

Software used:

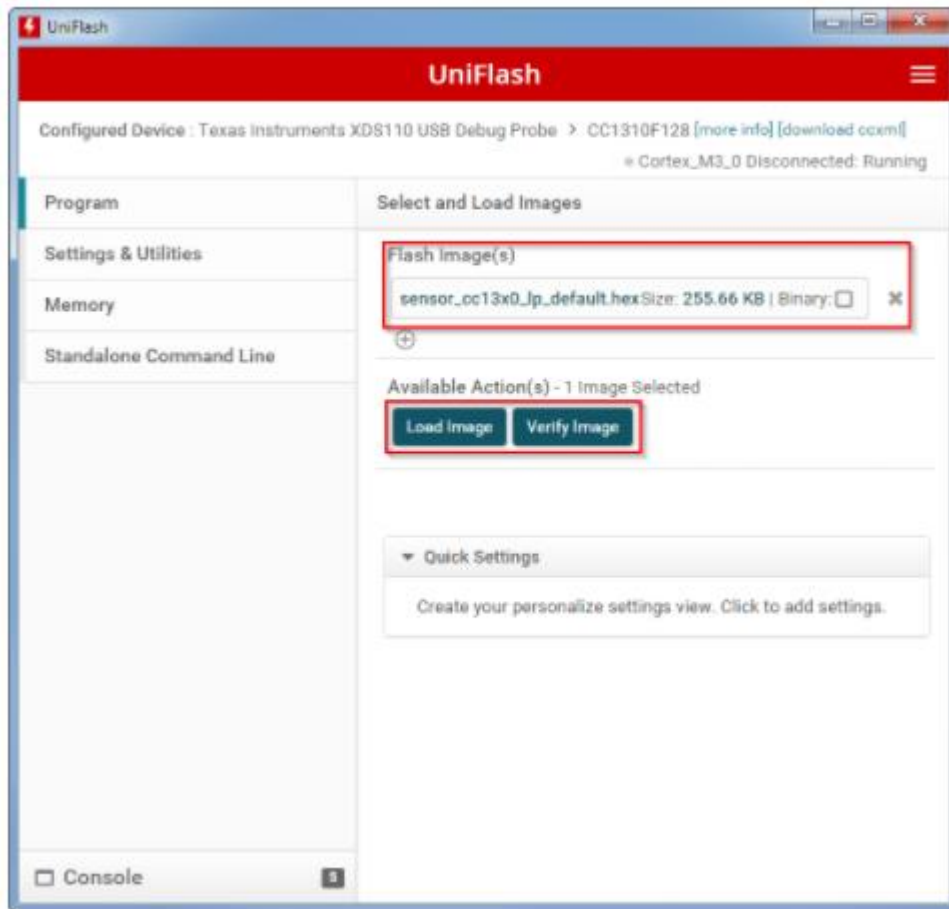
- UniFlash- used to flash the CC1350s to the correct configurations
- Code Composer Studio – The Sensor launchpad was programed in CCS
- Putty – Used to verify BBB was booted properly and to check value being sent
- Ubuntu VM – Used to set up the BBB

IMPLEMENTATION DETAILS:

Implementing Si7021 to sensor node with I2C:

1. The first step was to flash the CC1352 Board with the hex file

- MAKE SURE TO FIRST **Load image** and then afterwards **verify image**



2. The next step was to copy our bbb prebuilt tar.gz files into the beaglebone

Select text

```
cd ~/ti/simplelink/ti-15.4stack-x.xx.xx.xx/prebuilt  
scp bbb_prebuilt.tar.gz root@<bbb-ip-address>:/home/root/
```

Copy the pre-built files onto the BBB

11. On the BBB, extract the **bbb_prebuilt.tar.gz** file by executing the following commands

Select text

```
cd ~/  
tar -xvf bbb_prebuilt.tar.gz
```

Extract the pre-built content

3. After that I needed to run the gateway

version number:

Select text

```
cd ~/prebuilt
```

If using BBB host

Select text

```
cd ~/ti/simplelink/ti-15.4stack-x.xx.xx.xx/prebuilt
```

If using PC host

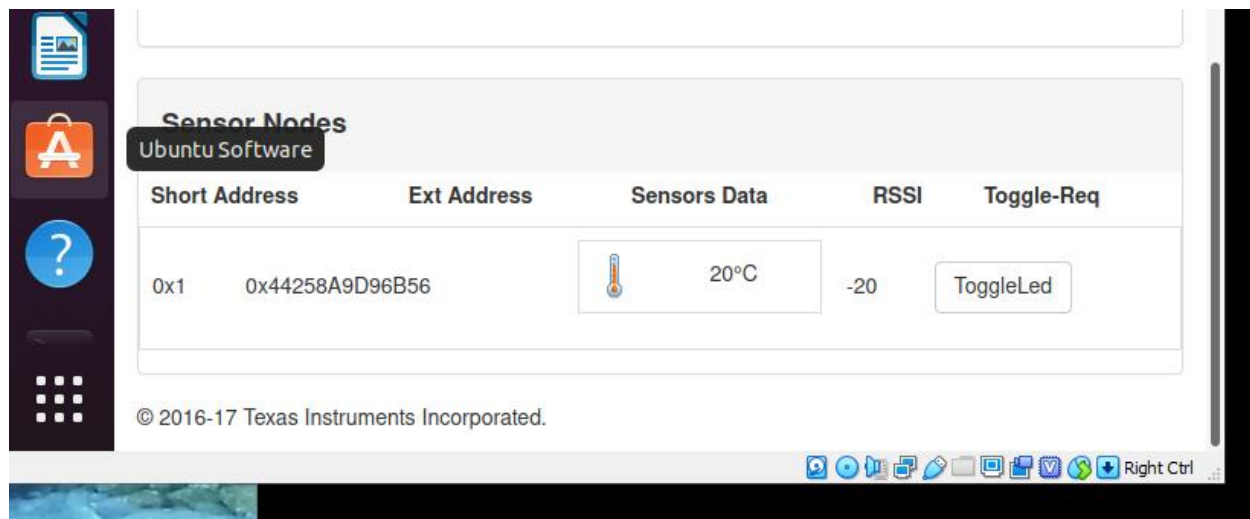
2. Execute the following commands on the host. The rm command is for getting rid of any pre-existing network configuration, and is only needed if you have already ran the gateway on this BBB before. The sh script starts the TI 15.4-Stack Gateway and Collector Application.

Select text

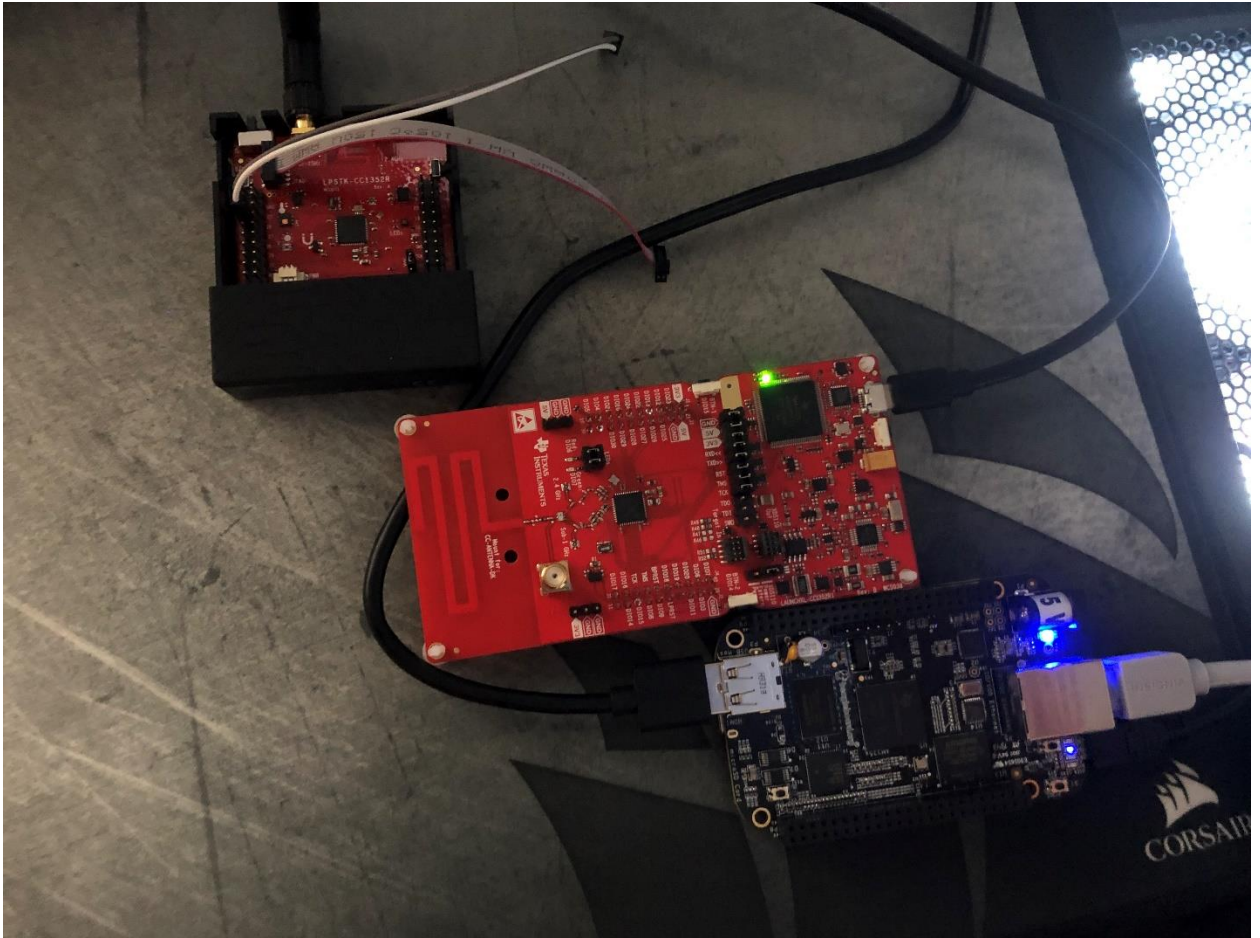
```
rm bin/nv-simulation.bin  
./run_demo.sh
```

Start the TI 15.4-Stack Linux Gateway

4. Then we extracted the data from the sensor and sent it to the co-processor:and this gives you your data through the gateway



OUTCOMES, RESULTS AND CONCLUSIONS:



This is an image of the board setup

```

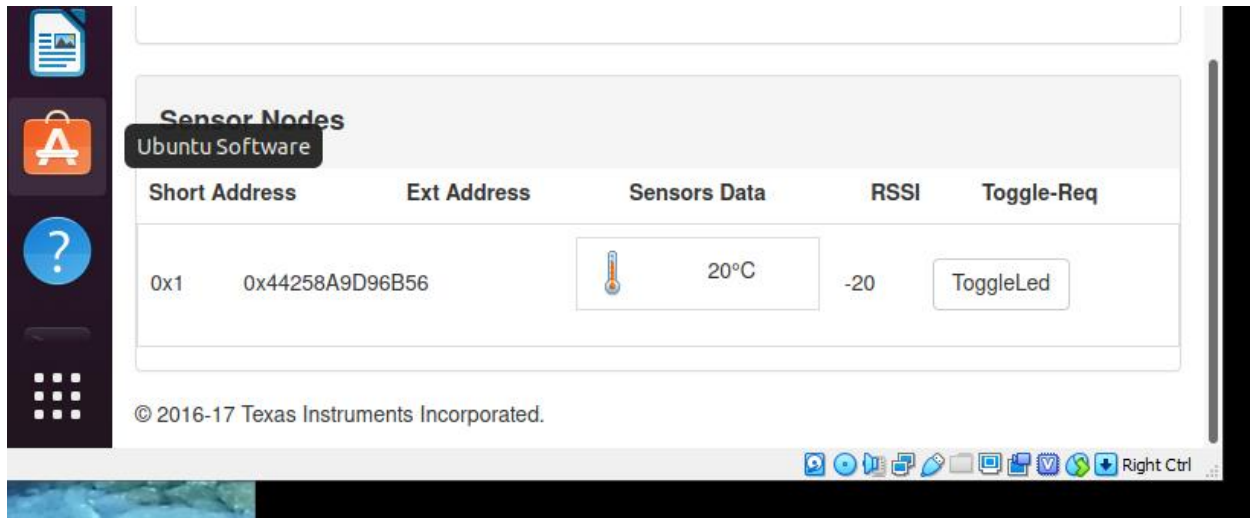
File Edit Setup Control Window Help
Login incorrect
beaglebone login: ddeebbiaann
X
Password: temppwd
nt
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
debian@beaglebone:~$
debian@beaglebone:~$ ifconfig
eth0: flags=28669<UP,BROADCAST,MULTICAST,DYNAMIC> mtu 1500
    ether 04:79:b7:f1:c5:51 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 55
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1120 bytes 77280 (75.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1120 bytes 77280 (75.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.7.2 netmask 255.255.255.252 broadcast 192.168.7.3
    inet6 fe80::679:b7ff:fef1:c553 prefixlen 64 scopeid 0x20<link>
    ether 04:79:b7:f1:c5:53 txqueuelen 1000 (Ethernet)
    RX packets 1362 bytes 105151 (102.6 KiB)
    RX errors 0 dropped 4 overruns 0 frame 0
    TX packets 51 bytes 11210 (10.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
usb1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.6.2 netmask 255.255.255.252 broadcast 192.168.6.3
    ether 04:79:b7:f1:c5:56 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
debian@beaglebone:~$

```

Login to ssh beaglebone

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permissible by applicable law.  
Last login: Sun Oct 7 20:25:11 2018  
debian@beaglebone:~$ ls -l /dev/ttyACM*  
crw-rw---- 1 root dialout 166, 0 Oct 7 20:23 /dev/ttyACM0  
crw-rw---- 1 root dialout 166, 1 Oct 7 20:23 /dev/ttyACM1  
debian@beaglebone:~$ cd /usr/src/linux
```

Checking for ttyACM 0 and 1



Succession for a reading

Video Demo:

<https://www.youtube.com/watch?v=xDgviZoGh4I&feature=youtu.be>