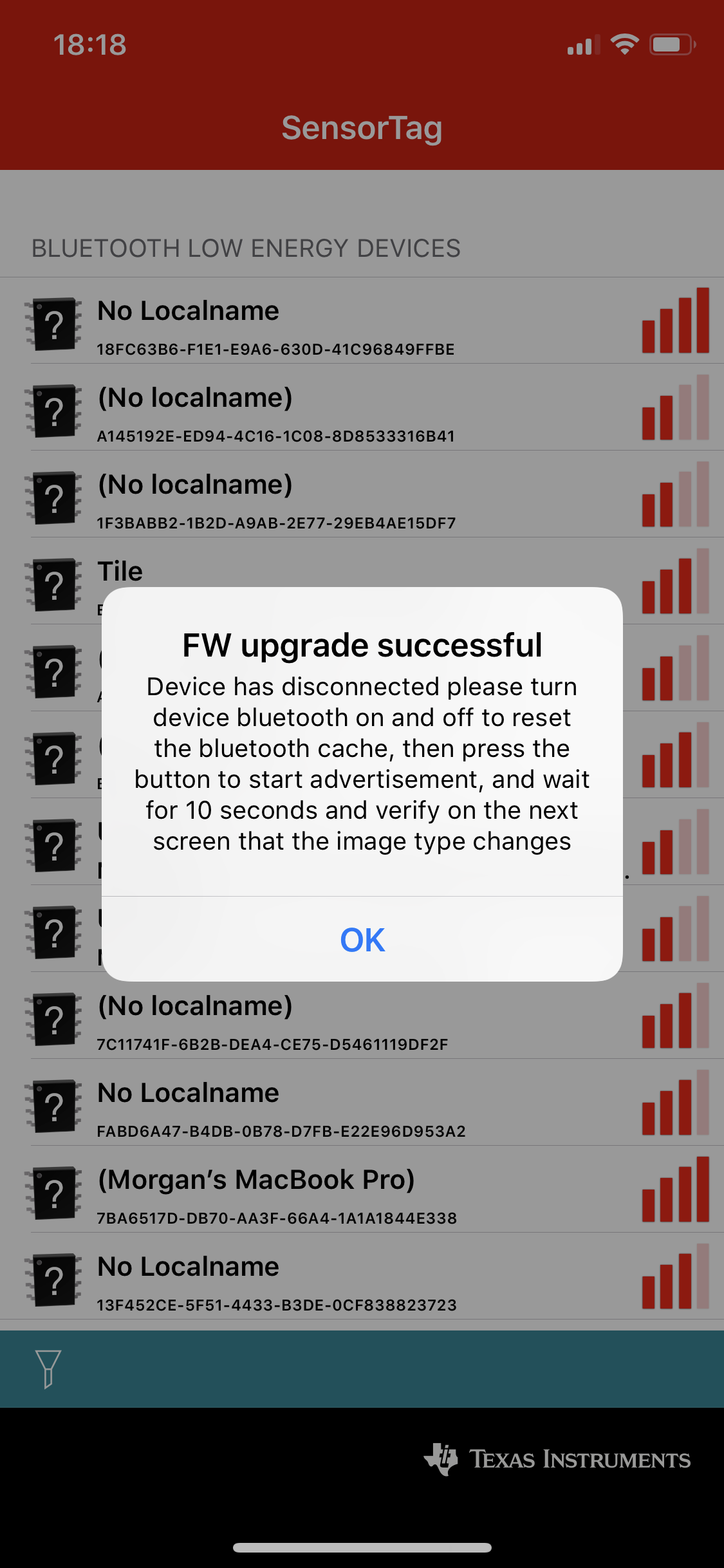
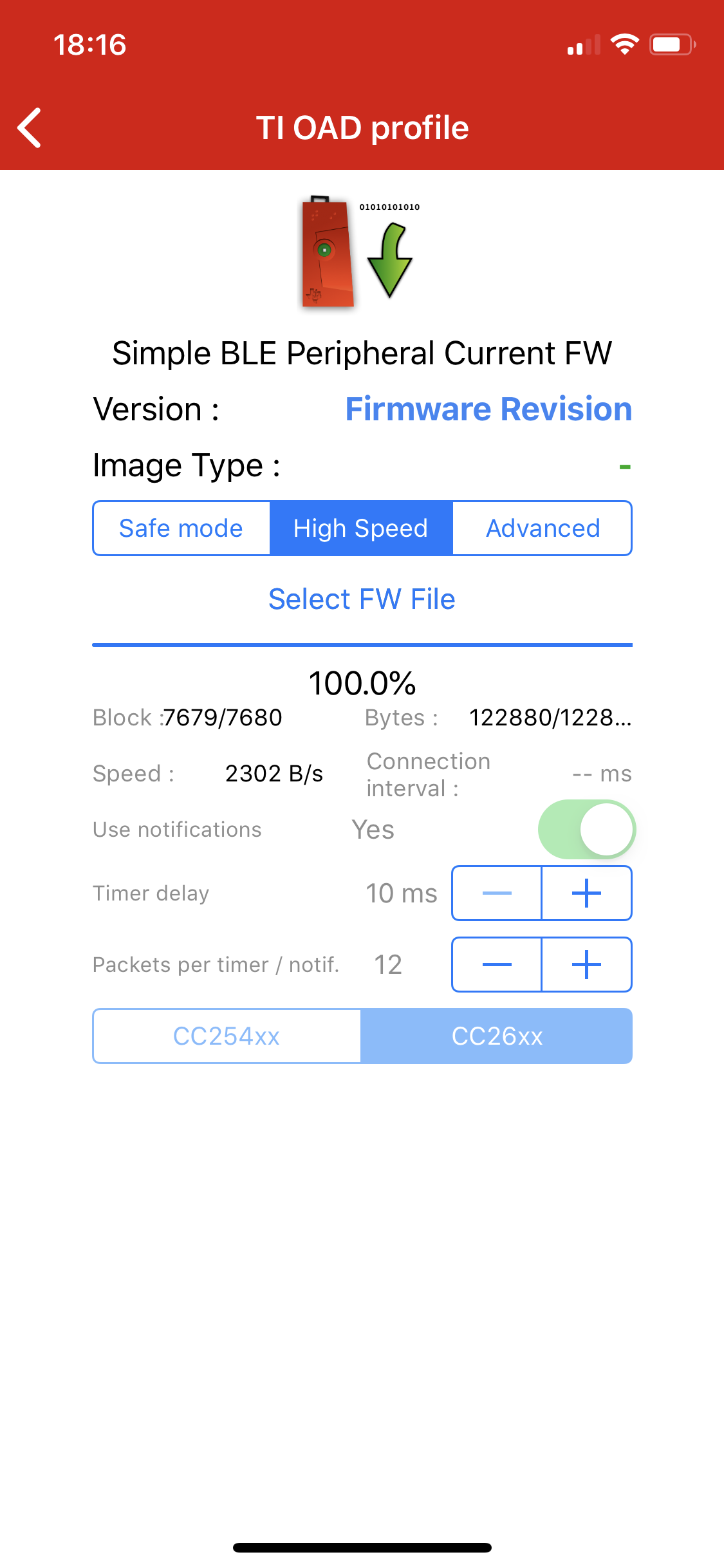
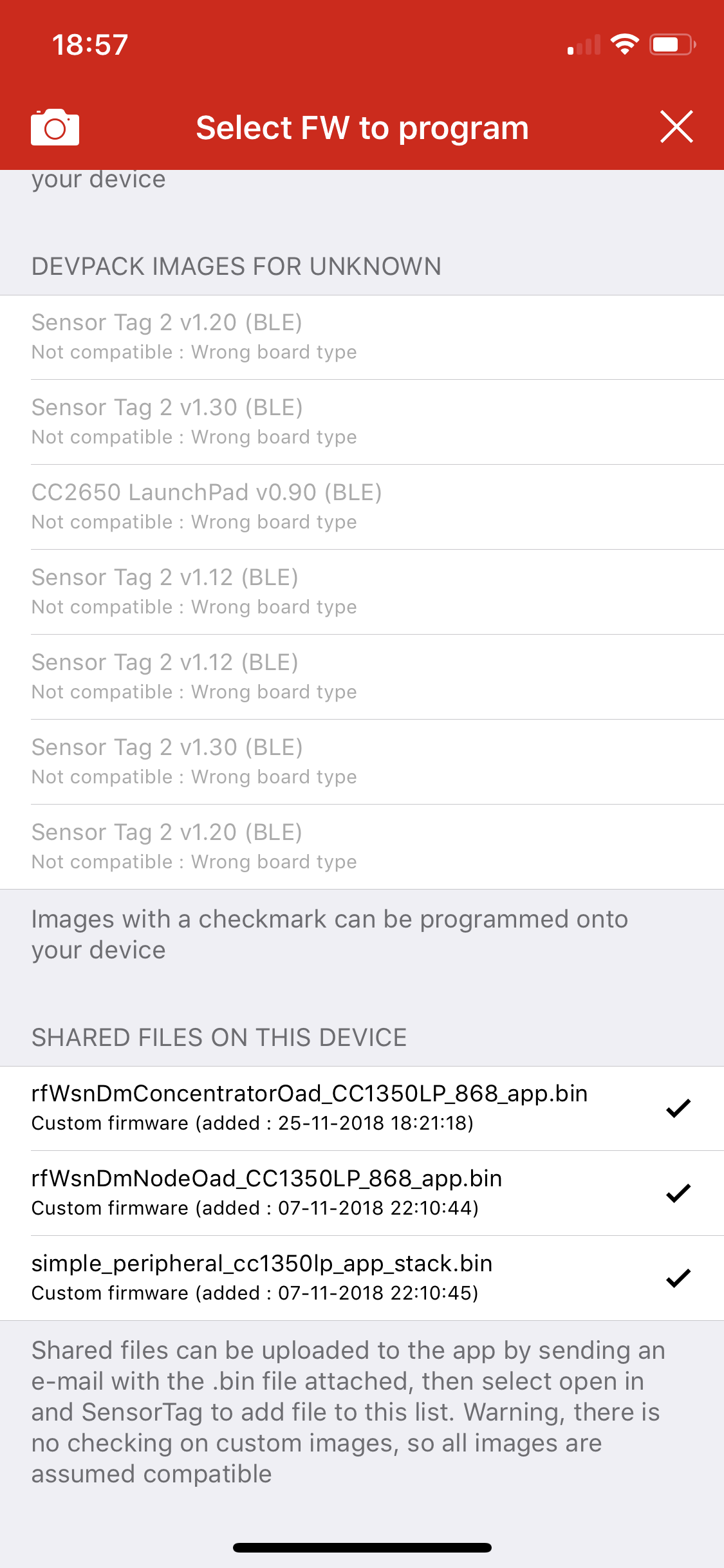
**Date Submitted: 12/4/2018**

**Assignment Youtube Playlist:** **https://www.youtube.com/playlist?list=PL4oTyvRrubXdbcRALkwshuOsWGRDYeFg7**

# Task 01: Update CC1350 LaunchPad firmware

**Youtube Link:** **https://youtu.be/9t64HfOY7XM**

For Task 1 we updated our device’s firmware using UniFlash, then connected to it with my partner’s iPhone and updated the firmware with the OAD feature. This feature seemed incredibly buggy with Android devices, so all information regarding it came from my lab partner.



**-----------------------------------------------------------------------------------**

# Task 02: OAD download sub-1GHz firmware images

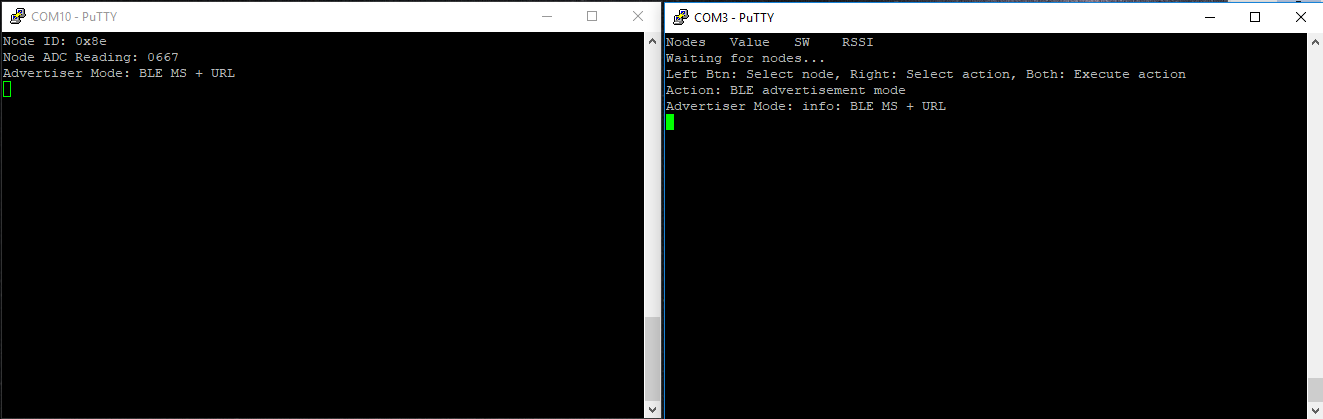
**Youtube Link:** **https://youtu.be/1TGPpE0z7XM**

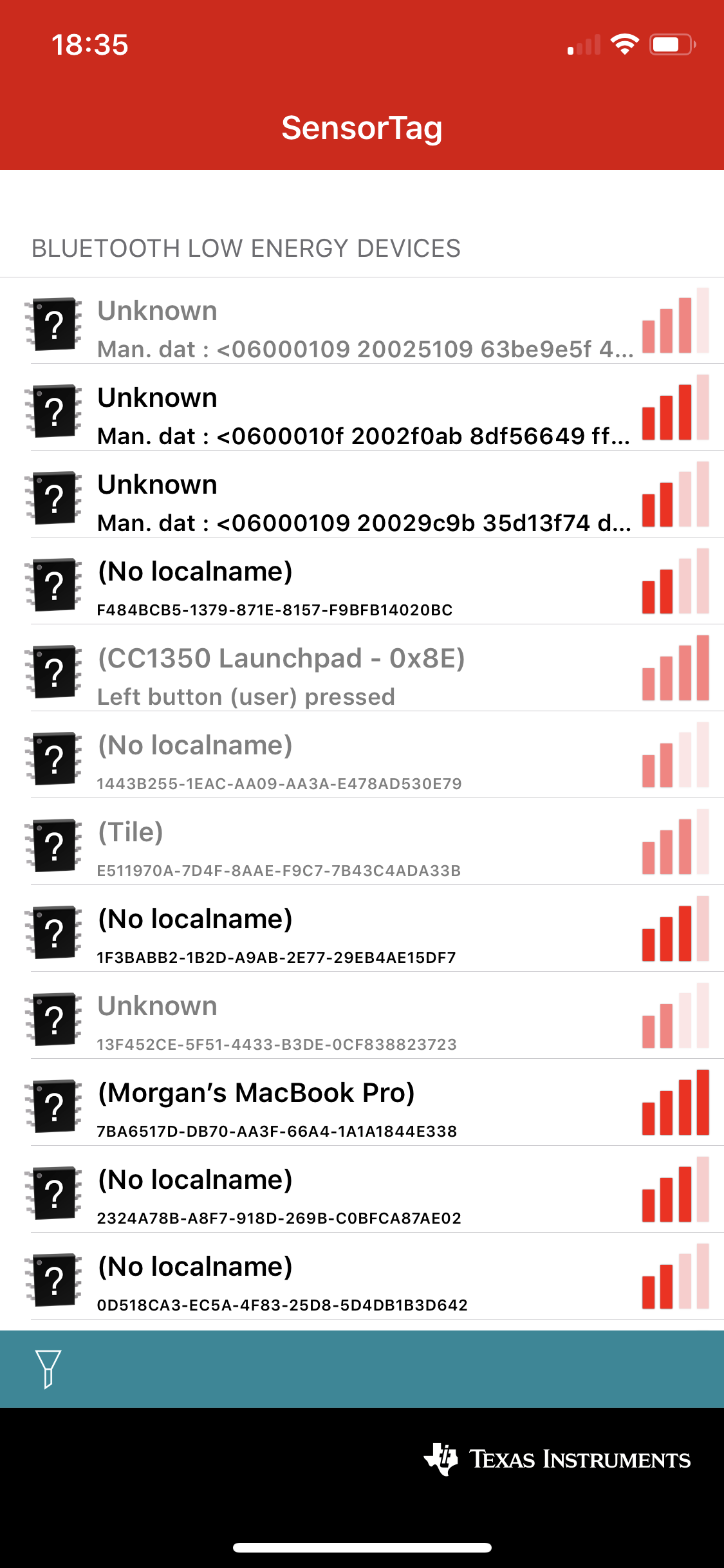
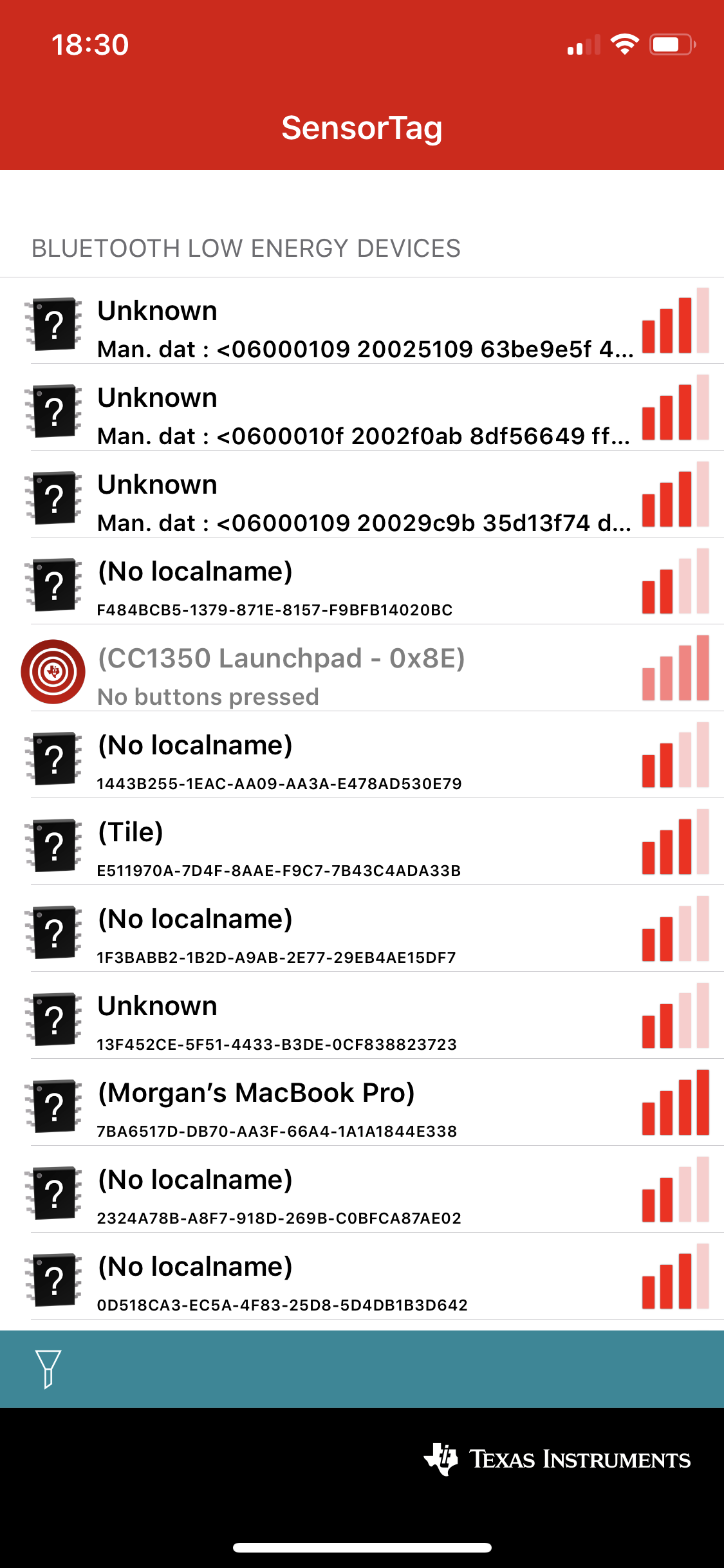
In Task 2 we load the Coordinator and Node images provided to us using the same steps as in Task 1. The images taken are identical to Task 1.

**-----------------------------------------------------------------------------------**

# Task 03: Running the sub-1GHz firmware images

In Task 3 we run the newly-loaded firmware images and display the results through UART. As a side note, it appears that Eddystone beacons may be discontinued. We can also see when buttons on the CC1350 are pressed on the TI SensorTag App.





**-----------------------------------------------------------------------------------**

Lab 6 Part 2

# Task 01: Importing the WSN Examples

**Youtube Link:** **https://youtu.be/yRzpP88teiU**

In Task 1 we import the Sensor Network Concentrator Example. No adjustments are made to the code, we load it as imported and see that it is waiting for node, as we do not have any nodes available yet.

**-----------------------------------------------------------------------------------**

# Task 02: Putting it all to work

For Task 2 we turn on our node device and allow it to connect to the collector. The results would be shown on our concentrator LCD, however we do not have that pack to install.

**-----------------------------------------------------------------------------------**

# Task 03: Change RF Channel

**Youtube Link:** **https://youtu.be/1g3lIVoRwe4**

In Task 3 we change our RF channel to learn how to select different frequencies to avoid interfering with other devices. Here we opted to change the frequency to 869MHz, keeping it close to the sample code. Please note that is is outside our allocated band for US-based frequencies centered at 915MHz.

EasyLink\_setFrequency(869000000);

This is adjusted for both the collector and node.

**-----------------------------------------------------------------------------------**

# Task 04: Switch from 2-GFSK 50 kbps to Long Range Mode (LRM)

**Youtube Link:** **https://youtu.be/9xSsz\_dloN8**

For Task 4 we adjust our settings to allow communication in long range mode, which allows longer range communication at the expense of a lower rate. We adjust the following code in RadioProtocol.h on both our concentrator and node:

**#define** RADIO\_EASYLINK\_MODULATION EasyLink\_Phy\_50kbps2gfsk

**-----------------------------------------------------------------------------------**

# Task 05: Measure Power consumption on the node

For Task 5 we measured power consumption on the node, which averaged approximately 5mA when the LEDs were off and 16mA when the LEDs were on. My equipment was not sensitive enough to record extremely accurate values with a quickly-changing load. My video did not save for this task, but is similar to task 6.

**-----------------------------------------------------------------------------------**

# Task 06: Modify Code Running on the Sensor Controller

**Youtube Link:** **https://youtu.be/P7ESndyoEQs**

In Task 6 we adjust our code on our sensor controller to read the ADC every two seconds rather than every second. This results in less often readings and less power consumption as indicated in our video. Although the resolution of our measurements was not very high, there is a distinct lowering in the values obtained.

