

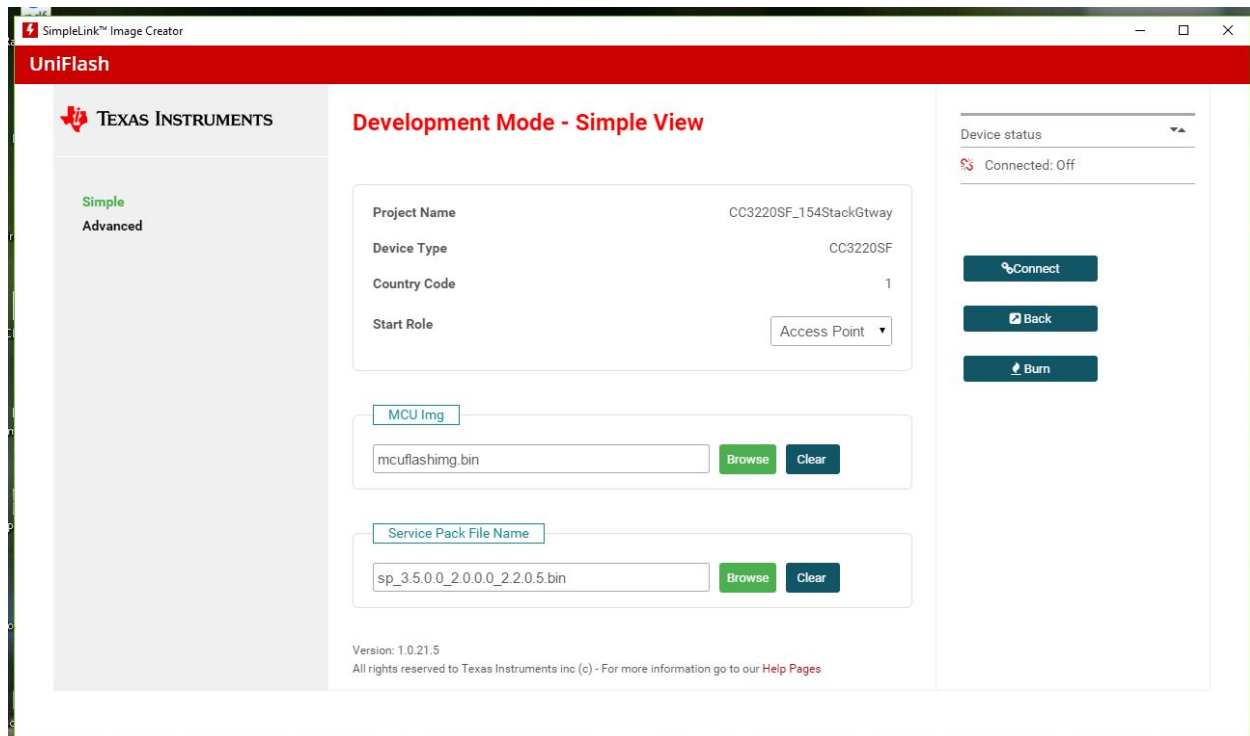
## Hardware Setup



**Figure 1. Uniflash software development**

Our Setup started with Uniflash 4.6.0 version or above. We needed to program the necessary function for our devices.

For example,

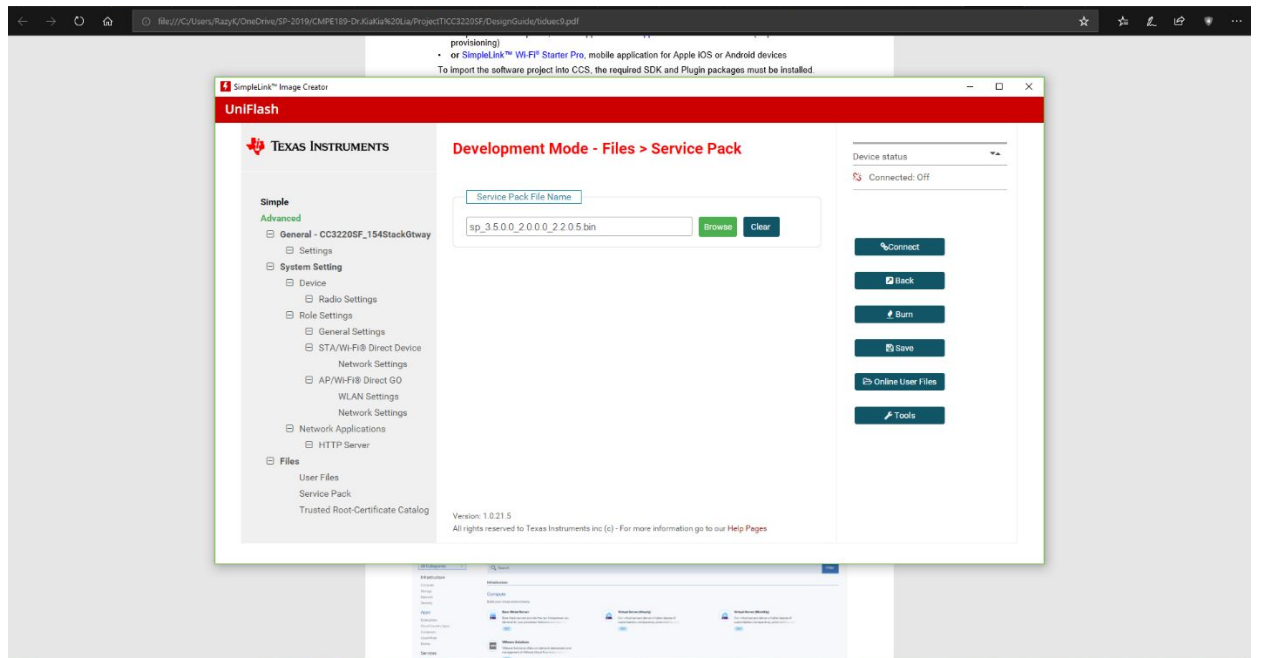
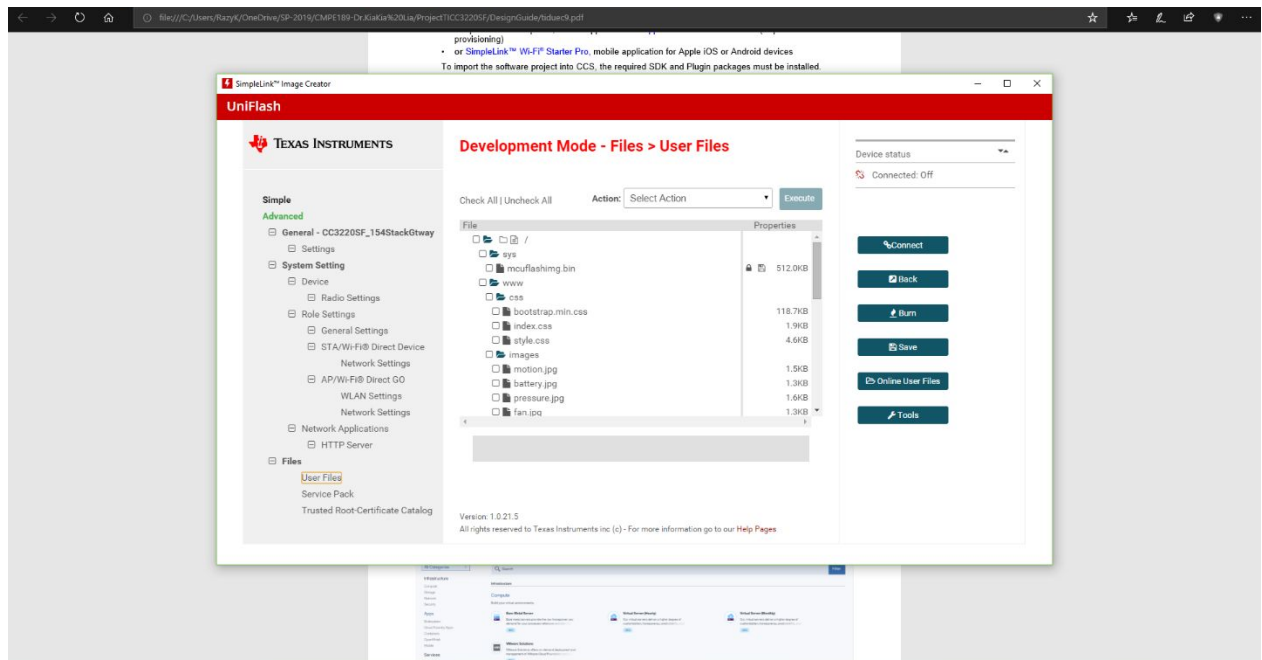


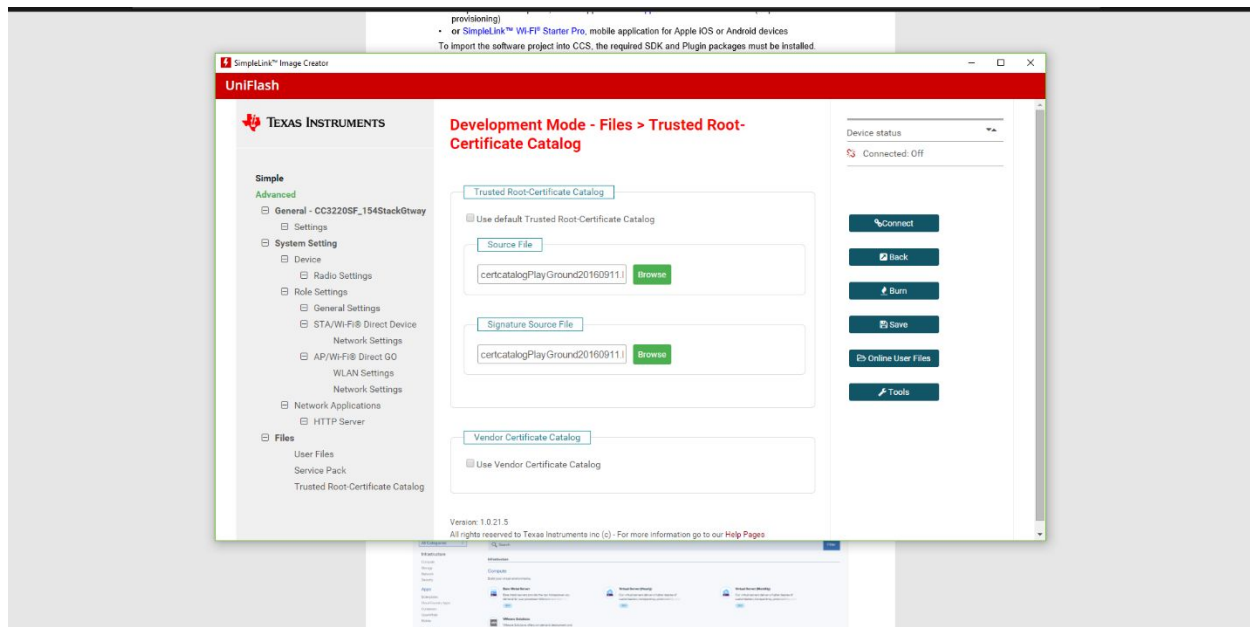
**Figure2. CC3220 board setup including materials needed for stacking and connecting to IBM Gateway**

CC3220SF needed a function called “TI 15.4 IBMStackGateway” which is provided in our project .JSON file.

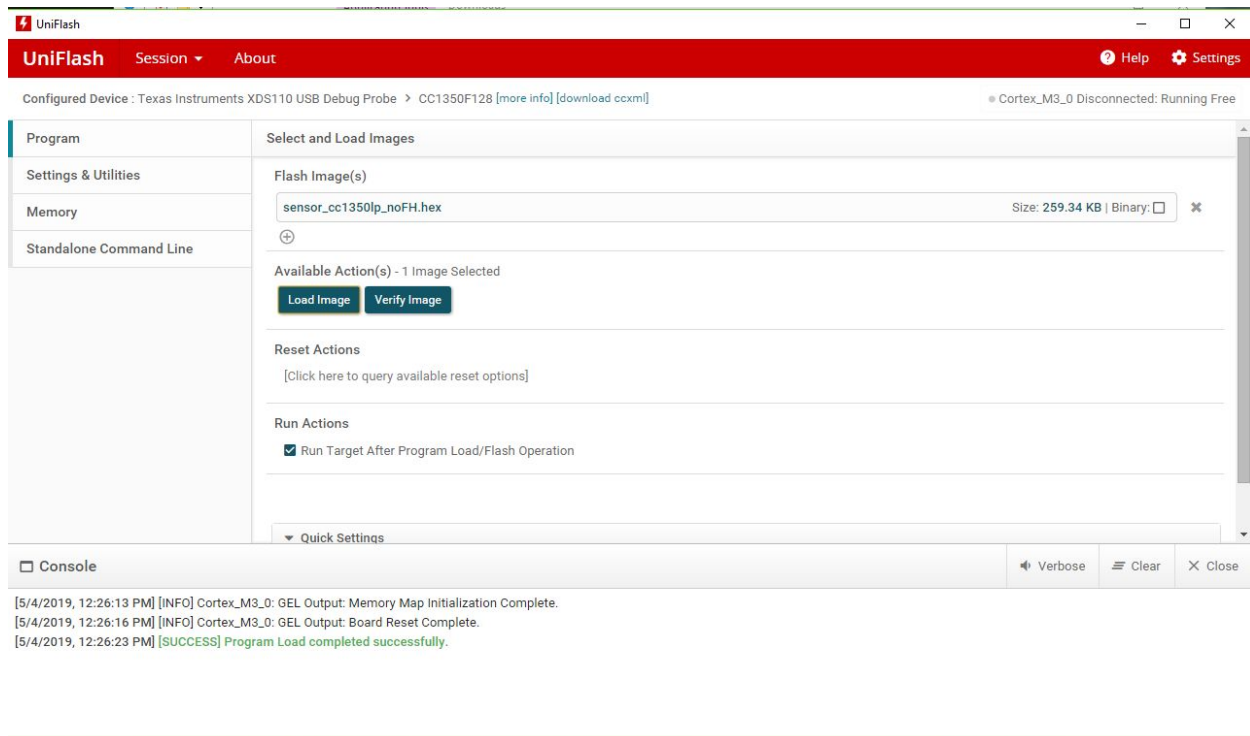
This is a very important setup step because the user needs to be careful and pay attention to the small details within this step.

Starting with the MCU img and Service Pack box .bin files from Simplelink CC32xx launchPad.





IEEE 802.15.4e/g by Using TI 15.4-Stack from the facilities designer's system

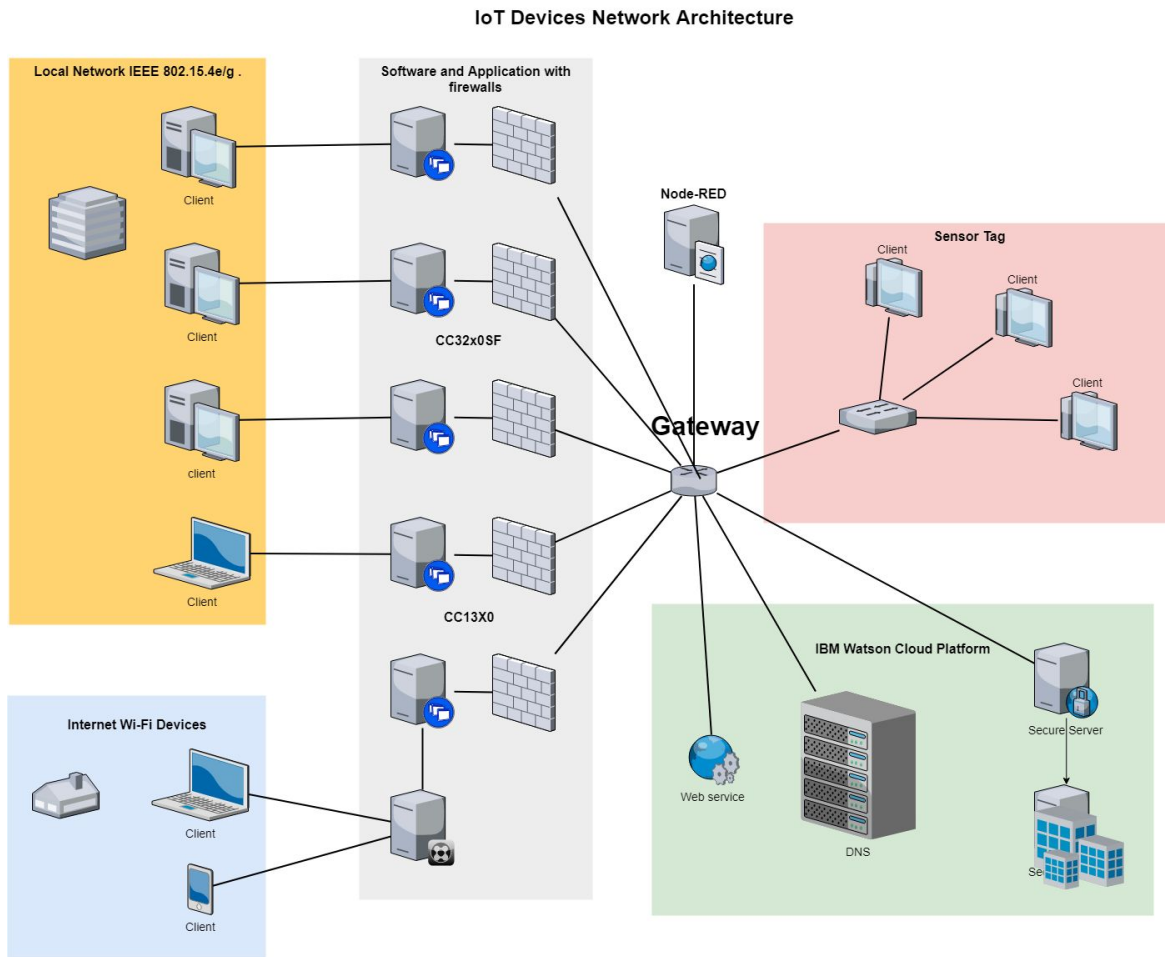


**Figure3. an example of program installation for TI CC1350 board before stacking**

CC1350 needed a function called “coprocessor\_cc13xxlp.hex” from the CC13x0\_LaunchPad.

Those two files can be found in the TI website by searching the name of the software kit. Otherwise, most of them required the users to use TI username and password to login for a download permission.

In the diagram below (**Figure4**) is the IBM StackGateway platform for all firmware development. The software and hardware components, along with the development design makes it more convenient for a sub gigahertz.



**Figure4. IoT Devices Diagram**

Also, it is easier for IoT devices to communicate with the nodes through the Gateway which handles the network for various special tasks. At the same time, it is much less resource consuming for a device to do edge computing compared to if it were to send the data to the cloud to do the computing.

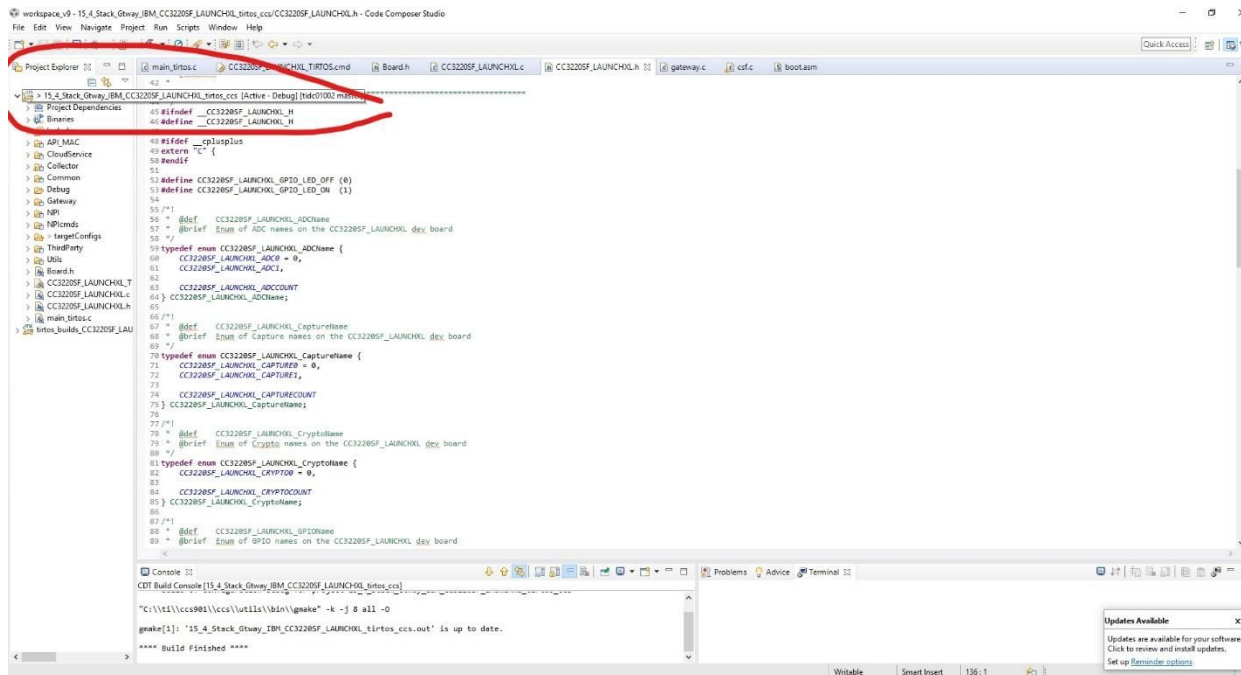
Regarding the sensor data transferring, the user can benefit from receiving the data on the local gateway instead of visiting the cloud. It is much more efficient for the user to process the data from the gateway and then go to the cloud. This is why we used Ti15.4 StackGateway for our networking as a main

function. In addition, it is able to host MCUs to communicate with CC1350 which stacks on top of the CC3220SF. This allows us the capability of running the WiFi connection.

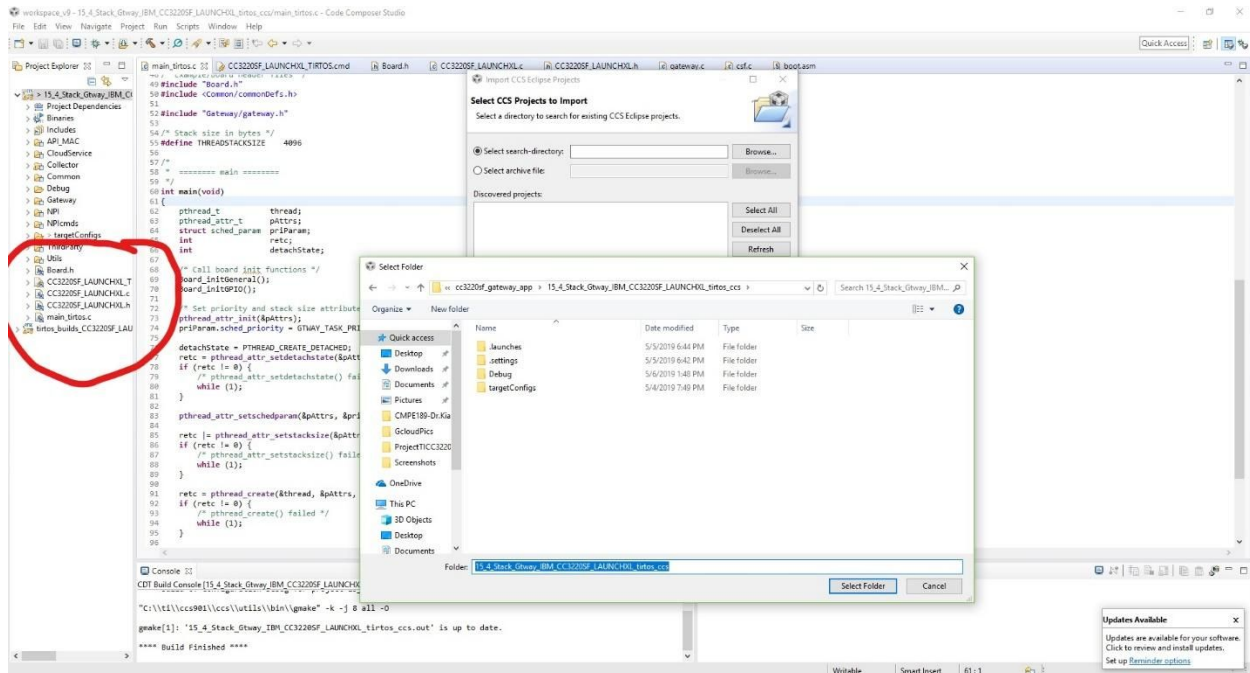
Here is the demo from Code Composer Studio 9.0.1 version (CCS).



This is the work in software CCS with the SimpleLink platform together with Ti-15.4 StackGateway

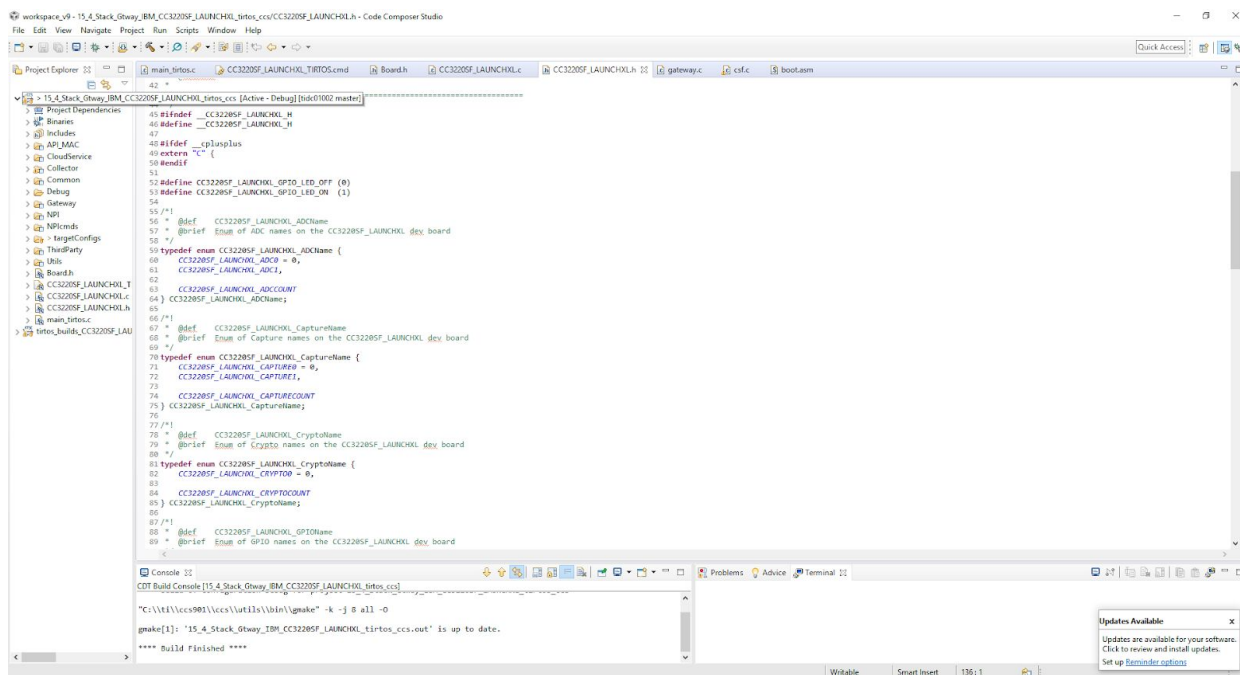


**Figure5. Import files from a local device**



**Figure6. Files component**

In the file components, there is a header file name “Board.h” which consists of a definition for the board information by default. If the program has no error on the main function, as well as in the sorted code, the user should run and debug the program as show below in the figure 7.



**Figure7. All configuration run properly**



In this section, the board should have stacked already. For more information please see the “Stack photograph” regarding plugin solution.

More importantly, the board that is on top of the CC3220 (the CC1350 board) should have all the jumpers taken out from the original position. Only the reset will remain in the same position, and we must switch the VSENSE to the blue position.

