ECGR 4101/5101, Fall 2018: Lab 7
Internet-of-Things (IoT) and WiFi Version 1.0

Learning Objectives:

This lab will introduce the technology at the heart of many human-machine interaction tools such as VR headset. For this Lab, you are going to configure CC3220SF as an HTTP server and read integrated accelerator data wirelessly. A python code file is provided, and you should modify it in order to move a circle in a plot by tilting the device. The accelerometer data should be requested from the board through HTTP GET method.

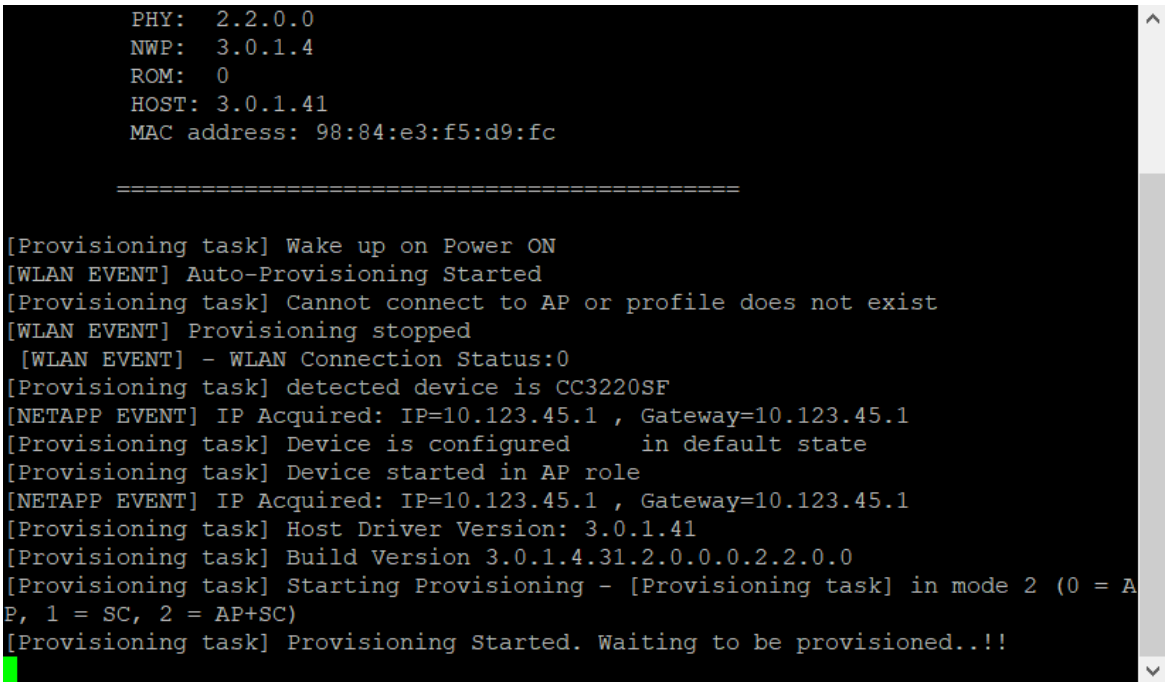
Supply List:

- TI CC3220SF (**You do not need the TIVA C board!** CC3220SF already has ARM M Series embedded processor).
- Download Python 3.6 to your PC so you can run the provided Python code file

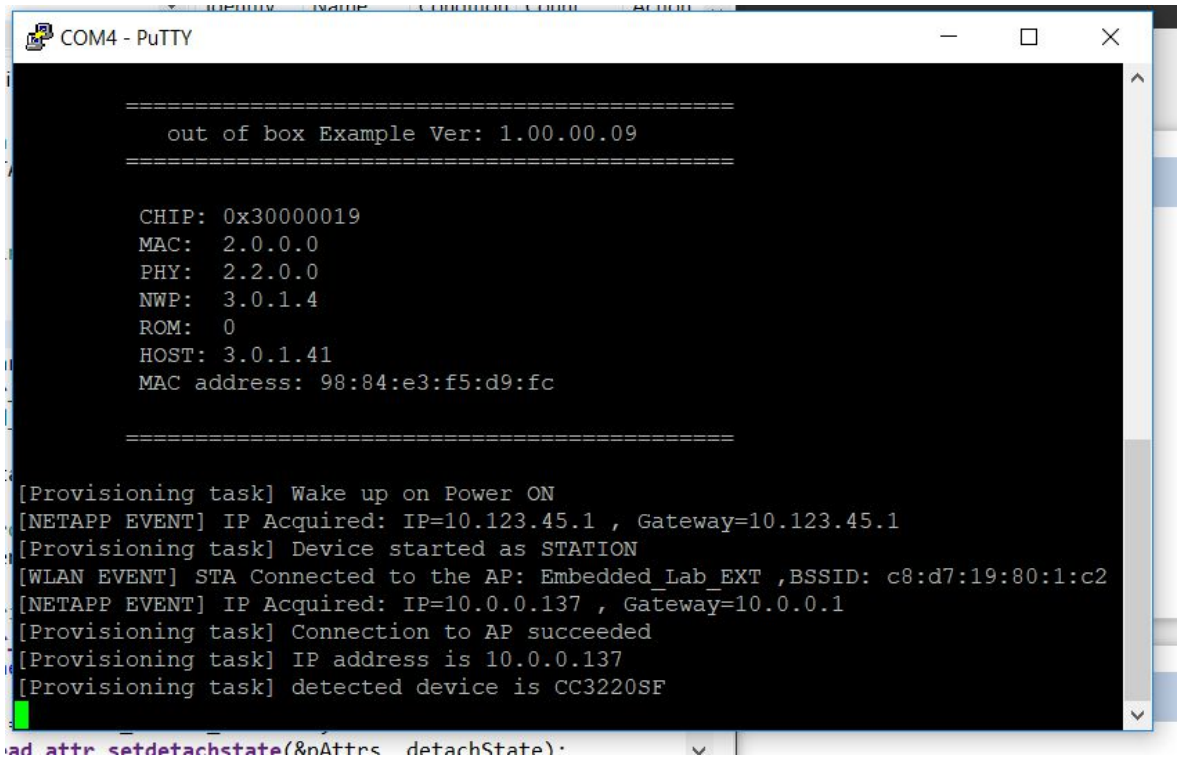
Requirements:

Specifically, students will need to translate the following requirements into specifications and then create a design to meet these requirements.

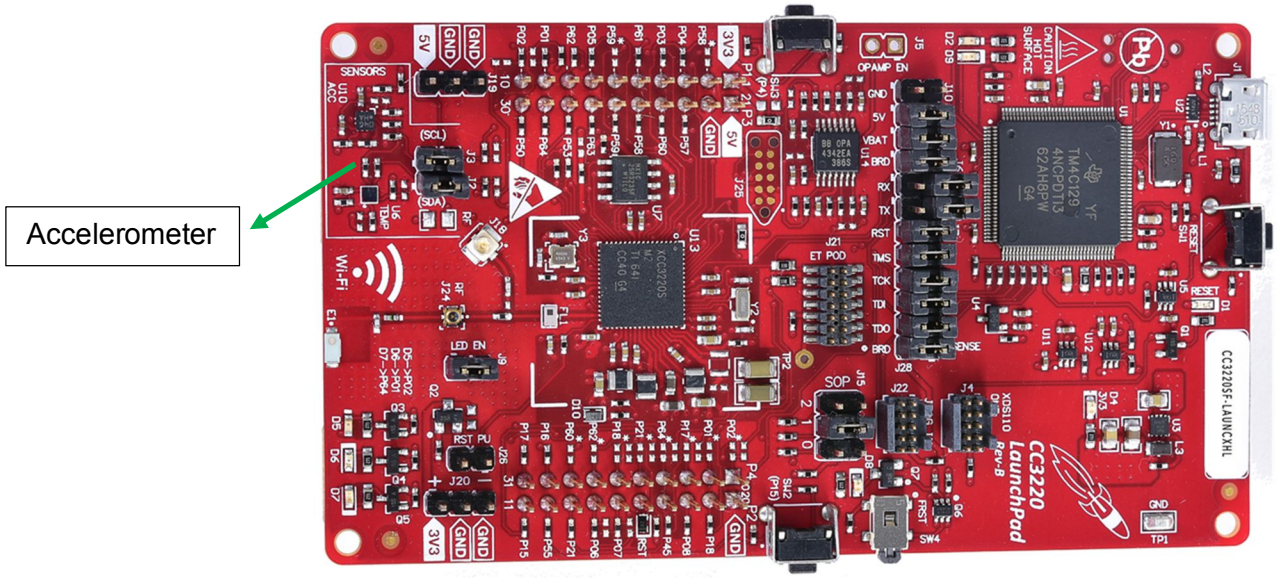
1. Import "out_of_box_CC3220SF_LAUNCHXL_tirtos_ccs" project into your CSS program. You can download it from Resource Explorer. This project will put the board in HTTP Server mode.
2. Compile the program and upload it to the board.
3. Connect to the board through your preferred terminal program and look for the acquire IP address for the first time (see following image).



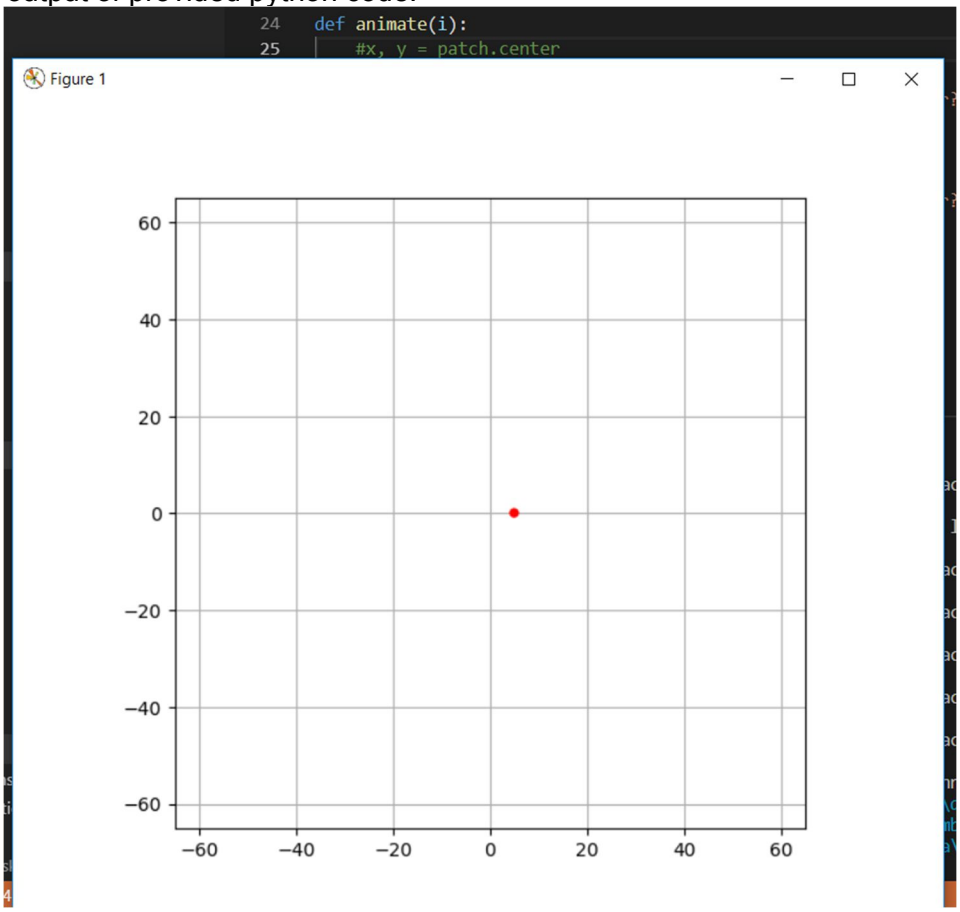
4. Look for “mysimplelink-{ID NUM}” in your wireless list. Connect to it and browse the acquired IP in your browser. From the ‘Profile’ tab, select ‘Embedded_Lab_EXT’ for SSID and configure the rest of the setting based on data provided in Lab6.
5. After adding the new profile, restart the program in the CSS and watch its output in the terminal. The board now is in station mode and acquired a new IP from ‘Embedded_Lab_EXT.’ You need this IP for provided python code.



6. Open file named “link_local_task.c” and understand how GET methods are defined to read temperature and accelerometer data. Have a look at lines 494-514 and 2070-2073. In these lines “sensor” GET method and “axisx” to “temp” variable for this GET method are defined.
7. Now modify the python code to read “axisx” and “axisy” by calling the provided GET APIs.



Here is the output of provided python code:



Specifications:

Students must come up with a set of specifications. Here are a couple of samples that will provide a guide. (Start with this list and expand!)

1. Start with how GET/POST works in the HTTP protocol and specifically how to pass a parameter with a GET.
2. After connecting to the embedded lab WIFI, it is time to learn about the I2C protocol. First, understand how its addressing works and tries to read the integrated temperature sensor.

Demo and Submit:

1. Document the specifications that were derived from the requirements. This should be in Header comments at the beginning of the source code.
2. Upload the steps that you took to make CC3220SF as HTTP server.
3. Be prepared to compile/flash your device on demand for the TA when you demonstrate the application.
4. Upload the main code file you created (with the comments, above).
5. Be sure to include all group members' names, IDs, and email addresses on the checkoff sheet.

Useful Links and Hints:

- Only modify the part of the code marked by "FIXME:" phrase.
- You should use "request.get" function to interact with your board. See here if you do not know how to use it: https://stackabuse.com/the-python-requests-module/
- This webpage might help you also: https://training.ti.com/simplelink-wi-fi-cc3120-and-cc3220-out-box-demo

Embedded Systems Lab Demonstration Validation Sheet

This sheet should be modified by the student to reflect the current lab assignment being demonstrated

Lab Number:	Lab 7 – HTTP Server
Team Members	Team Member 1: Student ID: 80 EMAIL ID: Team Member 2: Student ID: 80 EMAIL ID:
Date:	

Lab Requirements

REQ Number	Objective	Self-Review	TA Review
1	CC3220 can connect to 'Embedded_Lab_EXT' Wi-Fi network.		
2	CC3220 can make an I2C transaction and read the value from accelerometer sensor.		
3	Python code can interact with board through GET methods.		
4	The data received from server is displayed on terminal. The output of x-y axis should be displayed in the terminal.		
5	The circle on the plot moves smoothly.		