ECGR 4101/5101, Fall 2018: Lab 6
Internet-of-Things (IoT) and WiFi Version 1.0

Learning Objectives:

For many students, this lab will introduce the technology at the heart of Internet-of-Things; specifically, the integration of sensors, wireless communication, and an HTTP server. The server is configured and there is a wireless access point is available in the EPIC 2126 lab which is reachable from EPIC2130. Students will need to use a CC3220SF to communicate a temperature sensor value to the server. Any web browser will be able to check that the result is correct. They also learn how to interact with sensors which are connected to the I2C bus.

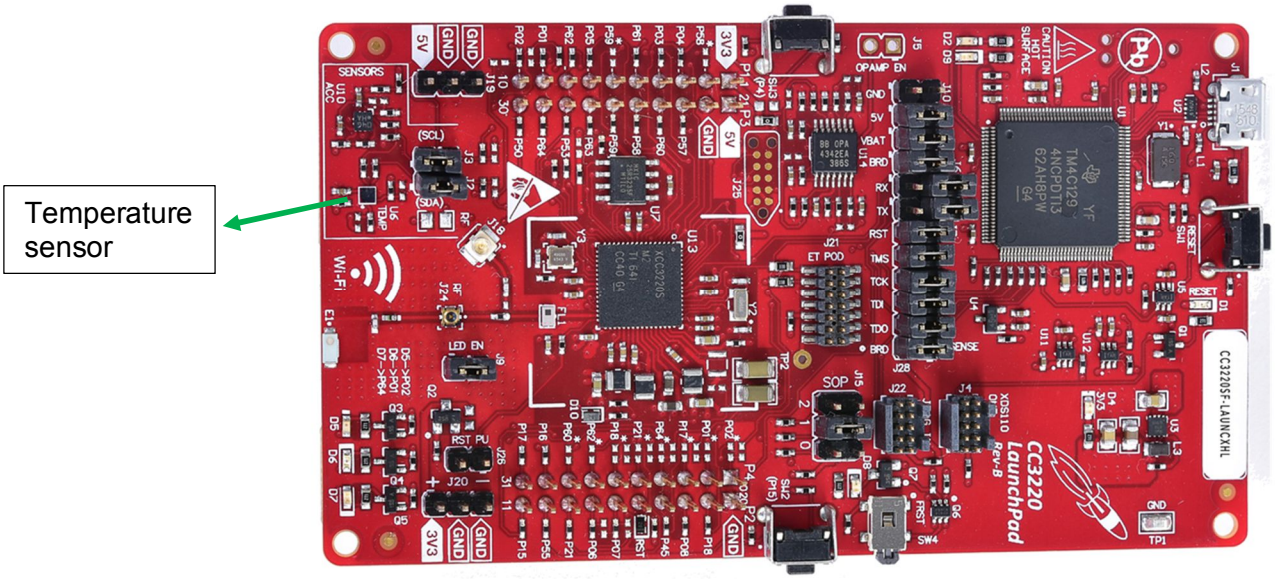
Supply List:

- TI CC3220SF (**You do not need the TIVA C board!** CC3220SF already has ARM M Series embedded processor).

Requirements:

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

1. Connect to "Embedded_Lab_EXT" wireless by using CC3220SF. Its password is 'embedded' and its encryption type is WPA2. It should be your first step. Refer to 'hint' section for more information.
2. The integrated temperature sensor is connected to the I2C bus. You should make an I2C transaction to communicate to the sensors.



3. When you successfully connected to wireless, Communicate to HTTP server. Its IP address is 10.0.0.31. Use following URLs to send and receive data from server.
    a. 10.0.0.31/lab6/?Action=Save&Student_ID=200&Tempr=24
    b. 10.0.0.31/lab6/?Action=Show&Student_ID=200

Remember the above URLs are case sensitive. ‘Action’ defines the functionality, ‘Student_ID’ is your student ID, and ‘Tempr’ is temperature. Use ‘Save’ for logging data into the server and ‘Show’ for controlling your source code correctness. Sever shows you recent temperature values that it has been saved. You can try it out in any browser.

After calling ‘lab6/?Action=Save&Student_ID=200&Tempr=24’ URL, server will send back a text regarding the request. If it returns ‘Server: New record created successfully’, it means the transaction has been completed successfully otherwise, it specifies the error you had.

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 read the temperature sensor (also in the header comments).
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:

1. If you are not able to upload your compiled code to CC3220SF and the CCS gives you an error, it means the CC3220SF is in ‘product’ mode and you can update or debug its firmware. At first, you should change its mode to ‘develop’. Refer to here to learn how to do that.
2. From CCS menu please click on “**View > Resource Explorer**”. Then, you will find a set of great documents and examples from “**Software > SimpleLink CC32XX SDK – V:2.30.00.05”** about how to work with CC3220SF. Two ‘httpget’ and ‘i2ctmp006’ example and driver will help you a lot.

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 6 – HTTP client
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 temperature sensor.		
3	CC3100 can communicate with HTTP server with aid of GET method. An appropriate GET URL is generated based on current temperature value and it is sent to the HTTP server.		
4	The data received from server is displayed on terminal.		
5	The correctness of recent saved temperature values is tested by using `Show` functionality.		