ECGR 4101/5101, Fall 2017: 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 TCP server. Students will need to use a Tiva-C (TM4C1234G) and the add-on CC3100 to communicate a with a python code.

Supply List:

- TI Tiva-C TM4C1234G
- TI CC3100 BOOSTER
- a potentiometer (each group has a different one)

Do not solder onto your board! *You will need to reuse these pins in the future.*

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" or any wireless which is possible for you by using CC3100. Its password is 'embedded' and its encryption type is WPA2. It should be your first step. You can start the project again by importing 'getting_started_with_wlan_station' or 'tcp_socket'.
2. Modify tcp_socket example so it shows you the assigned IP. Write the IP address in the format of 'X.Y.W.Z' on CCS terminal.
3. Modify tcp_socket example so it starts only TCP server. At this point, it will write *Starting TCP server* and it waits for incoming connection.
4. Download 'client.py' file. It is a simple python code for TCP communication. Run it on your PC as follows:

   Code 1: Command for executing python code

   ```
   $ python client.py 192.168.2.18 5001
   ```

   The first argument is the IP of CC and second one is socket port number. If it is successfully connected to the CC board, it will print 'LED1:' and it will ask you an integer value. Enter '0' if you want to turn the LED1 off, otherwise '1'. Do the same for LED2. When you enter the values, it will send the following string to CC:

   Code 2: Transmitted string value to CC board

   ```
   LED1:0&LED2:1
   ```

5. Parse the transmitted string and active or de-active the LEDs based on values. You can use your LED bar or any other LEDs.
6. Enable ADC peripheral and its ISR to read Pot voltage.
7. Python code will stop for receiving an integer value from CC board. It is blocking mechanism. At this time you should send the pot value to your computer.
8. If everything goes well, python will show you the following line and it will exit:

Code 3: CC has sent the value of 100 as POT value

```
POT:100
```

Demo and Submit:

1. Document the specifications that were derived from the requirements. This can be a simple text file (i.e. README) document that accompanies the source code.
2. Upload the steps that you took to make CC3100 pull the values from the POT.
3. Be prepared to compile/flash your device on demand for the TA when you demonstrate the application.
4. Be sure to include all group members' names, IDs, and email addresses.

Useful Links and Hints:

Following links can be used for both WLAN station and server:

- User's Guide
- TCP Socket Application
- CC3100 SDK Micro. Make sure to define the discussed micros in your CCS project.
- Acquired IP address
- Example folder of CC3100 SDK folder both for TIVA C and MSP430.
- Remember to include and link both TivaWare and SimpleLink SDKs to your CCS solution.
- If you run into some 'undefined' errors, search other SimpleLink SDKs header files to find them.
- Learn how to use sl_Recv and sl_Send functions