ECEN 5823 Spring 2024
Assignment 6 - LCD Integration and Client Command Table for A7

Objective: Integrate LCD display support with your existing assignment 5. Begin learning professional design skills by building a command table. Design your Assignment 7 Client by building a command table spreadsheet (design first). Later in Assignment 7 you will then develop (code second) a BLE Client that will communicate with your BLE Server Assignment 6.

Note: This assignment will start with the completed code from Assignment 5.

References required:

Silicon Labs Bluetooth Software API Reference Manual: see https://docs.silabs.com/bluetooth/latest/

Instructions:

1. Start by creating your assignment repository using the assignment link at https://classroom.github.com/a/ShlKlfBV . You will not need to clone this from the GitHub repo you just created into a new local directory since you will be starting with your code from the previous assignment. Instead, follow these instructions and run these commands with git bash within your assignment directory. Ensure you are setup to use SSH keys before proceeding, see the instructions from github at https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/ and be sure to use Git Bash For Windows or Cygwin if using a Windows environment. (Cygwin .ssh directory will be different than Git Bash.)

    a. On your local PC, duplicate the folder created in Assignment #5 to a new folder, name it something like ecen5823-assignment6-<username> where <username> is your GitHub username.
    b. Change your current directory to this new folder ecen5823-assignment6-<username> and execute the following git commands.
    c. git remote remove origin
    d. git remote add origin <url>
        i. Where <url> is the URL for the repository created with the link above
        ii. This adds the new submission repository created in the link above as your new origin.
    e. git push origin master
    f. Import ecen5823-assignment6-<username> into Simplicity Studio
        i. Remember to perform a **Clean…** immediately after you import the project into Simplicity Studio.

2. Follow the instructions in lcd.c to integrate the display with your project. For the 1Hz timer mentioned there, use sl_bt_system_set_lazy_soft_timer(), see file lcd.c, function

displayInit(). displayInit() should be called from your boot event handler and will generate a periodic 1Hz event for you to use. We call displayInit() from the boot event handler because we're not supposed to call any BT stack function until after the boot event.

a. Add displayPrintf() calls to set the DISPLAY_ROW_CONNECTION row with the required display states.
b. Add displayPrintf() calls to set the DISPLAY_ROW_NAME with value Server (see #define in ble_device_type.h) and DISPLAY_ROW_BTADDR with the value of the 6 bytes of the Bluetooth address obtained from the structure returned by sl_bt_system_get_identity_address().
c. Add displayPrintf() calls to set the DISPLAY_ROW_TEMPVALUE with the temperature value read from the temperature sensor only when in an open connection and indications for the HTM Thermometer have been enabled by the Client. Whenever HTM Thermometer indications have been disabled by the Client, DISPLAY_ROW_TEMPVALUE shall be cleared.
d. Add displayPrintf() to display A6 on row DISPLAY_ROW_ASSIGNMENT.

3. Test your firmware implementation as shown in the "Testing" section below to ensure your implementation is complete and functional.

4. Design your Assignment 7 Client program. Start by filling out the Assignment 6 - Client Command Table for A7.xlsx spreadsheet. Leave column LCD Display State empty for now. You'll get instructions on this column in Assignment 7. Update the command table to include the events you will need to handle, and commands you will need to add to your firmware in order to implement the Bluetooth Client portion of Assignment 7.
    a. For assignment 7, the BLE Client implementation should use a build time assigned fixed Bluetooth address corresponding to the server which should be connected to when found during discovery. See ble_device_type.h
    b. The Client firmware should support on each open event:
        i. 1-time discovery of the HTM service using its service UUID
        ii. 1-time discovery of the Temperature Measurement characteristic using the HTM thermometer characteristic UUID
        iii. 1-time enabling of the HTM temperature indications. Disabling indications for the HTM characteristic is not required.
        iv. Receiving the temperature indications from the Server and displaying it on the Client's LCD.
    c. Part of your design needs to include the development of a new state machine to sequence the GATT commands used by your Client for discovery of the HTM service and characteristic in your Server, with the state machine being driven by gatt_completed_id events. Do not build your new state machine into your ble_event_handler() function!
    d. Note that you do not need to implement the firmware for this A7 functionality. You are only performing the "design work" for assignment 7 as part of assignment 6. You will develop the firmware that you designed here when we get to assignment 7.

e. Extra credit (+10 points) is available if you include a bubble state diagram of your discovery state machine in your command table spreadsheet for A7.

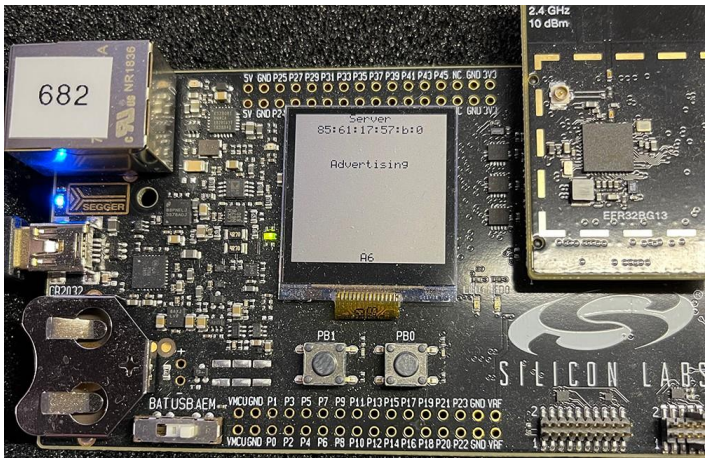
5. For the code you submit for grading, there should be no calls to __WFI() or __BKPT() or LOG_***() functions. No calls to Systick routines are allowed. However, do Log errors returned from API calls.

Testing:

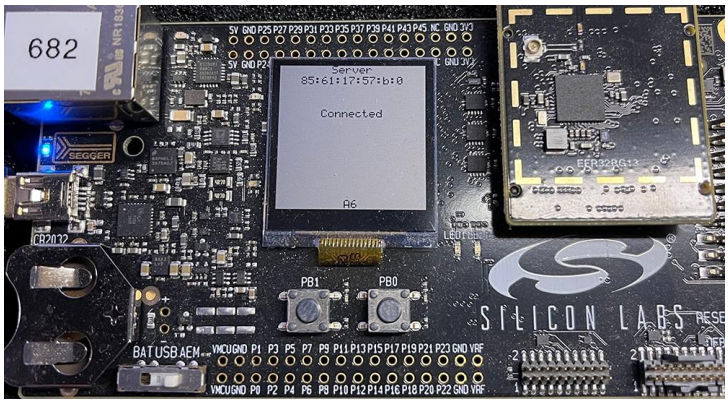
Your firmware should pass the following tests:

1. When powered on the first line should display Server. The Server's bd_addr should be shown in the second line of the display, displayed left-to-right from array index [0] to array index [5]. The last row of the display, DISPLAY_ROW_ASSIGNMENT, should show A6.
2. On initial power up the DISPLAY_ROW_CONNECTION row should show Advertising.
3. When connected to the Client, DISPLAY_ROW_CONNECTION row should show Connected.
4. When connected to the EFR Connect mobile application:
    a. The display should show updating temperatures as an integer value in celsius, on row DISPLAY_ROW_TEMPVALUE of the display from the onboard temp sensor only when HTM indications are enabled and in an open connection. When HTM indications are disabled DISPLAY_ROW_TEMPVALUE should be blank.
    b. The EFR Connect mobile application values should show the same updating temperatures when HTM indications are enabled.
    c. The display should show Connected on DISPLAY_ROW_CONNECTION row.
5. When disconnected from the EFR Connect application, the display should show Advertising once again on DISPLAY_ROW_CONNECTION and clear row DISPLAY_ROW_TEMPVALUE.
6. It should not be necessary to reset your device to reconnect with the EFR Connect application.

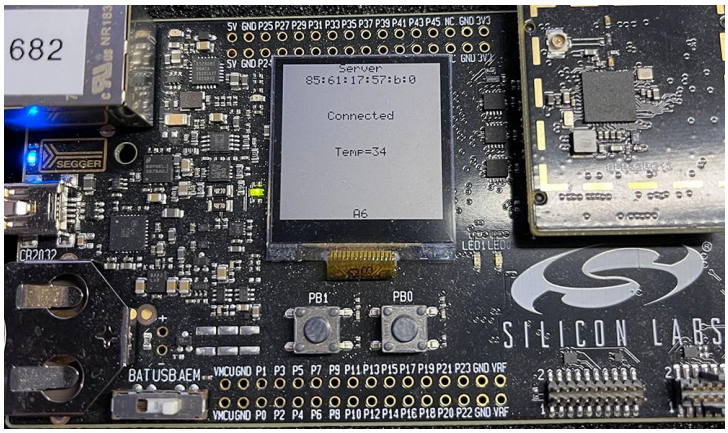
Here are pictures of assignment 6 in the advertising state when not connected:



When connected and HTM indications are disabled:



When connected and HTM indications are enabled:



Deliverables:

- An updated Assignment 6 - Client Command Table for A7.xlsx spreadsheet in the questions folder that describes your A7 design. *Although not required for the grade, a bubble diagram of your state machine may be very useful (and extra credit). You could draw your bubble diagram in PowerPoint and then paste that into your Client Command Table spreadsheet off to the side of the table for reference.*
- The completed assignment 6 program project to be run on the instructing team's computer for grading.
- Submission via **github classroom** and your repository setup via the classroom link at above. Use the following git command to submit your assignment: git push origin master
- Verify your ecen5823-assignment6-<username> repository contains your latest/ intended code and files in the questions folder. It is your responsibility to push the

correct version of your project/code. Use a web browser to verify the correct version of your project/code has been uploaded to GitHub.

- In **Canvas**, submit the URL to your github repository and in **GitHub** create a tag of your GitHub submission prior to the due date with the text of A6-final. The date and time of the tag creation is what we use to determine whether your submission was on time or late. These 2 steps will complete your assignment submission.

Guidance:

DISP_ENABLE and SENSOR_ENABLE are connected together on the PCB, so remove the code that turns off SENSOR_ENABLE for load power management and have the Si7021 sensor "ON" all the time. Otherwise your LCD display will flash on and off.

Write code to implement GPIO configuration for EXTCOMIN, port, pin and push-pull mode. Develop code for the routine gpioSetDisplayExtcomin() called from display.c. Put the code for the gpioSetDisplayExtcomin() function in your gpio.c file. **Note**: All of your GPIO configuration and manipulation functions should be in gpio.c/.h.

Files in your src/ directory should look like: