

# ECEN 3360

## Digital Design Lab #8

### Capsense Driver to Turn-on/off Temp Readings

### Spring 2019

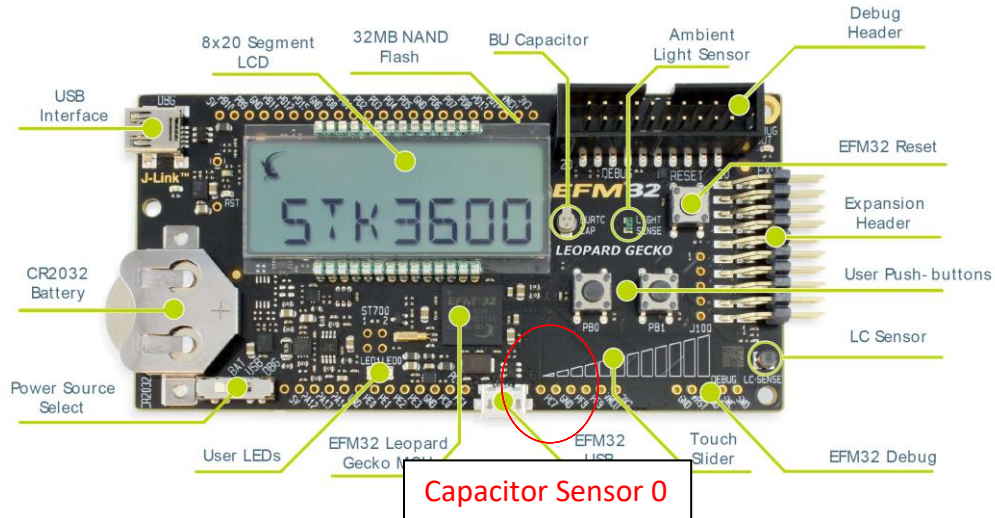
**Objective:** This assignment is to use a capacitor sensor as an on/off switch to turn-on/off the temperature readings. Using the capacitor sensor to enable the processor to remain in the lowest energy state to monitor the button using the Capsense driver. While the temperature readings are disabled, the CPU will still monitor, decode, and act on commands from the BLE module such as changing the degrees sent to Fahrenheit.

**Note:** This assignment will begin with the completed Lab 7 assignment.

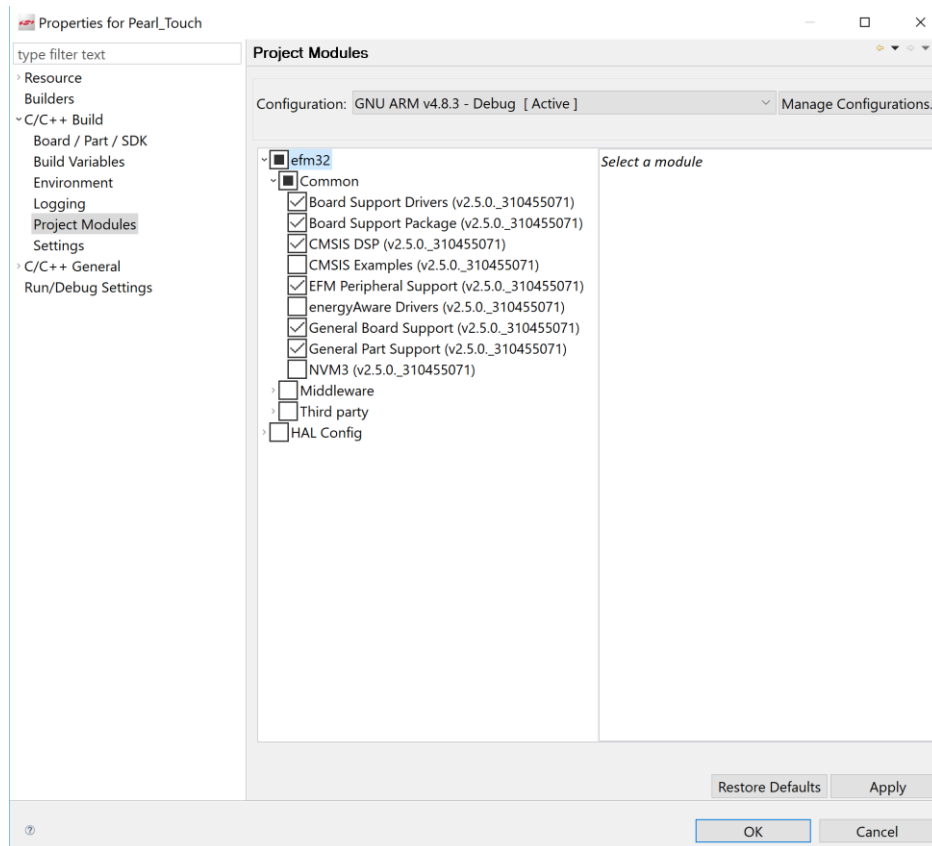
**Lab 8 Due:** Thursday May 2<sup>nd</sup>, 2019

#### Lab 8 Instructions:

1. Make any changes required to Lab 7, Using the LDMA for LEUART receive. Please work with the Instructing Team to get your code working to Lab 7.
2. To better match the energy measurements from your project to the expect results, all projects must enable HFXO to wake up and run after coming out of sleep energy modes. The following line of code should be added in your cmu.c function after all the calls to initialize the HFXO oscillator.
  - a. `CMU_HFXOAutostartEnable(true, false, false);`
3. LETIMER0 should be set to the following conditions at startup / reset.
  - a. Si7021 temperature read period = 3.0 seconds
  - b. No LED heart beat requirement
4. You will be using the Cryotimer to determine when to sense the Capacitor Sensor
  - a. Set up and initialize the Cryotimer to interrupt in a period close to 1sec
  - b. Choose the oscillator that will enable the Cryotimer to work all the way down into EM4.
    - i. What oscillator will you use?
  - c. Upon the Cryotimer Interrupt, you will call a function to read/sense whether the Capacitor Sensor at location 0 is being activated



- To fully utilize the supported Capsense driver, you will need to add the appropriate links to the driver .h and .c files to your project. To add these links, right click your project and select "Properties." Then work your way to C/C++ Build/Project Modules/efm32/Common. To add the board supported libraries, select General Board Support and then click on Apply.



6. Copy the Capsense.c driver located in the “Lab Reference Files” folder in the Course Canvas website to your project /src folder
7. You can use these two resources to learn how to use the Capsense driver.
  - a. SLSTK3402A\_touch project example found in Simplicity
    - i. You can find it by going to the Launcher view, select your dev kit, and then scroll down under Software Examples until you find the touch example
    - ii. Click on this example and it will import the project into your Simplicity IDE project space
  - b. The Pearl Gecko HAL software documentation
8. The Capacitor Sensor will be acting as a single Toggle switch per touch to turn-on or off the temperature reading
  - a. Your software must perform logic to indicate a single touch event
    - i. A single touch event is once the capacitor sensor has sensed a touch, it will not execute another touch/toggle until it senses a non-touch event
  - b. Per touch event, if temperature readings are disabled, your software will enable temperature readings
  - c. Per touch event, if temperature readings are enabled, your software will disable temperature readings
9. When temperature readings are disabled, the following should be disabled:
  - a. LETIMER0 should be disabled
  - b. SENSOR\_ENABLE pin to the SI7021 should be deasserted/disabled
  - c. No writes of temperature data should be sent to the BLE module
10. Upon enabling the temperature reading, the following should occur:
  - a. LETIMER0 should start at a known condition of **LETIMER0->CNT = 0**
  - b. Lab 7 operations should occur as normal
11. The Pearl Gecko should be able to receive and transmit data from the BLE module simultaneously
12. Both the transmit and receive functions of the LEUART0 should be using DMA instead of polling or interrupts.
13. While temperature readings are disabled, BLE commands sent to the Pearl Gecko should still be decoded and acted upon.

Deliverables:

1. Project code exported to Canvas for grading

2. Energy Profiler screen shot showing the first instance of a temperature reading after the temperature readings have been re-enabled after being disabled for 10 seconds.
3. Lab 8 worksheet to be completed in Canvas
4. Deductions:
  - a. Magic numbers - 3 pts
  - b. Not unique files per peripheral - 3 pts
  - c. No comments before each function documenting its function, input arguments, and outputs returned (LDMA, LEUART, I2C, CRYOTIMER) - 6 pts
  - d. No acknowledgement of IP - 2 pts
5. Late Submission:
  - a. Due date to Friday, May3rd , at 11:59pm - 10 pts
  - b. After May 3<sup>rd</sup>, no credit will be given
6. If code does not work, corresponding Quiz questions will be marked as 0 if answered correctly.