ECEN 5823
Managing Energy Mode Assignment
Fall 2018

Objective: Become familiar with the Silicon Labs' Simplicity development system as well as learn the different Blue Gecko energy modes and how to manage these energy modes.

Note: This assignment will begin with the completely Simplicity Studio Exercise Assignment project.

Due: Saturday, September 15th, at 11:59pm

Instructions:

1. Add, cut and paste, all emlib .h and .c source files into the appropriate file in platform/emlib folder of your project. The emlib .h and .c source files can be found in the following directory:
    a. For Windows: C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.3\platform

2. Update any .h files in main.c that are included in your src folder to "src/gpio.h"

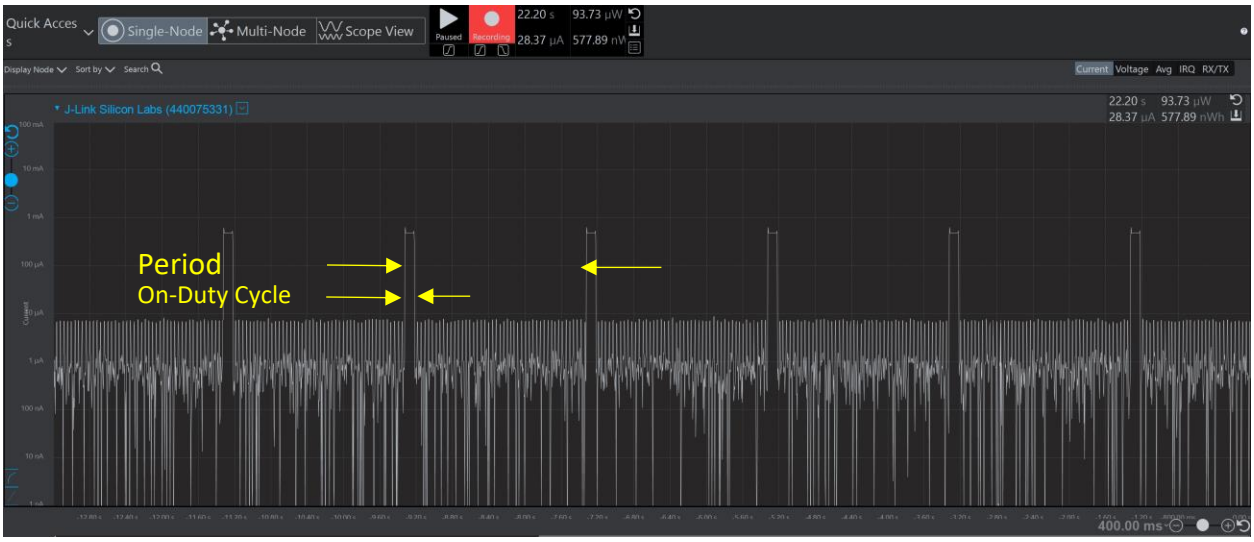
3. Develop a sleep() routine and the associated global variables and routines to manage which energy mode the Blue Gecko can enter based on the Blue Gecko peripherals that are being used?

    a. Note: For EM3, the LETIMER0 will need to be associated with ULFRCO instead of LFXO
    b. For EM0-EM2, associate the LETIMER0 with the LFXO oscillator

4. Develop the LETIMER0 interrupt handler routine to accomplish the next step

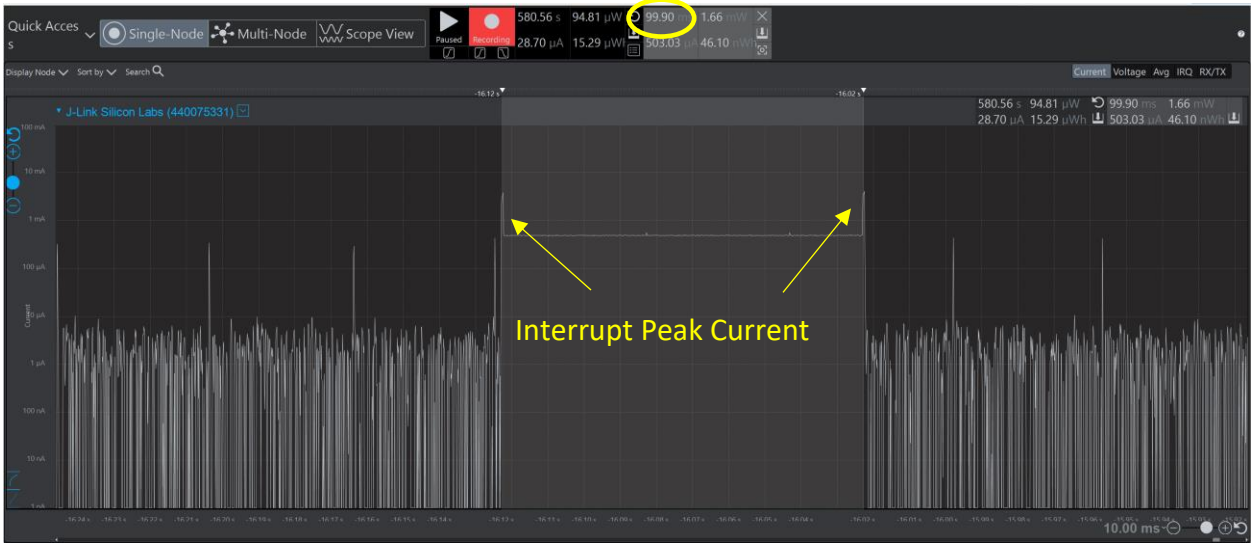
5. Program LETIMER0 peripheral to blink LED0 for 175mS every 2.25 seconds using the LETIMER0, LETIMER0 interrupt handler, and the sleep() routine.

    a. Use this routine to obtain the Average Current if LETIMER0 limits the MCU to enter the following modes while in low energy mode: EM0, EM1, EM2, and EM3
    b. A #define statement will be used to set the sleep mode limit
    c. Due to the short duration of the LED0 being on, it will appear as a quick flash



NOTE: TO HAVE SOME STANDARD TO DETERMINE WHERE TO PUT THE TIME CURSOR TO MEASURE TIME, YOU WILL BE MEASURING FROM INTERRUPT PEAK CURRENT TO INTERRUPT PEAK CURRENT. YOU WILL NEED TO ZOOM IN BOTH THE Y OR CURRENT AXIS AS WELL AS THE X OR TIME AXIS.

The below screen shot will show you where the measured time between the cursors is located, highlighted by the yellow circle, and where to fine the peak interrupt currents. The peaks are the result of the interrupt, and then the CPU going back to sleep but the current does not go all the way down due to the current draw of the LED.



Questions:

In a separate document to be placed in the drop box with the program code, please answer the following questions:

NOTE: All average currents should be taken at a time scale of 200mS/div.

1. What is the Average Current when the Blue Gecko can only go down to EM0 for a complete period and what is the average current when the LED is off? What is the average current when the LED is on?
    a. Wait 60 seconds after the Average Current is RESET to read the Average Current

2. What is the Average Current when the Blue Gecko can only go down to EM1 for a complete period and what is the average current when the LED is off? What is the average current when the LED is on?
    a. Wait 60 seconds after the Average Current is RESET to read the Average Current

3. What is the Average Current when the Blue Gecko can only go down to EM2 for a complete period and what is the average current when the LED is off? What is the average current when the LED is on?
    a. Wait 60 seconds after the Average Current is RESET to read the Average Current

4. What is the period in milliseconds of the LED blinking using EM2 using the Average Current "selected range" markers? What is the On-Duty Cycle in milliseconds of the LED using the Average Current while limited to EM2?

5. What is the Average Current when the Blue Gecko can go down to EM3 for a complete period and what is the average current when the LED is off? What is the average current when the LED is on?
    a. Wait 60 seconds after the Average Current is RESET to read the Average Current

6. What is the period in milliseconds of the LED blinking using EM3 using the Average Current "selected range" markers? What is the On-Duty Cycle in milliseconds of the LED using the Average Current while limited to EM3?

Note: The project will be tested by setting different period and on-duty cycle times for grading purposes.

Deliverables:

1. One document that provides the answers to Energy Mode Assignment.

2. Another document that contains your .c project source and header files with LETIMER0 set to EM3 mode if using the MAC or Linux version of Simplicity. If you are using the Windows version, please export the file. The main program file should be main.c.

Please note:

1. There should be a #defined statement that can be used to determine the lowest energy mode available to the LETIMER0

2. In the #defined statements, the period between LED0 being turned on should be defined in seconds. By giving a new definition of time in seconds to this #define statement, the period should reflect the new period the next time the program is compiled.

3. In the #defined statements, the On-Duty Cycle between LED0 being turned on and off should be defined in seconds. By giving a new definition of time in seconds to this #define statement, the On-Duty Cycle should reflect the new period the next time the program is ran.

4. The program should contain as a minimum, the following routines:
    a. Sleep mode routines
        i. Going to sleep
        ii. Selecting which sleep mode the system is limited to while the peripheral is required
        iii. Removing the restriction of the block sleep mode when the peripheral is no longer required
    b. Setup routines
        i. CMU
        ii. GPIO
        iii. LETIMER0
    c. An interrupt handler for LETIMER0