ECEN 3360
Digital Design Lab #3b
I2C and Load Power Management
Spring 2019

Objective: Use the Si7021 to take a temperature measurement and turn on LED0 if the temperature falls below a defined set point. Milli-Amps do matter. To minimize power, the concept of Load-Power Management will be utilized to reduce the average current of the design from 4.5mA into the micro amp range.

Note: This assignment will begin with the completed Lab 3a, I2C and Load Power Management, Assignment

Due Date 3a: February 23rd, 2018 at 11:59pm

Instructions:

1. If you have not completed LAB 3a assignment, please contact the professor to help you complete LAB 3a so that you can begin LAB 3b.

2. Your project should be structured based on "Best Design Practices."
    a. Replace "magic numbers" with #define statements
    b. Create a .c and .h for each peripheral for ease of readability and code reuse, best practices. For example:
        i. cmu.h/cmu.c
        ii. gpio.h/gpio.c
        iii. letimer.h/letimer.c
        iv. i2c.h/i2c.c
        v. sleep.h/sleep.c
        vi. si7021.h/si7021.c

3. Set LETIMER0 should be set to the following conditions at startup / reset
    a. Period = 4.0 seconds
    b. No LED to be blinked based on the LETIMER0 interrupts
    c. COMP0 and COMP1 will be used for other functionality
    d. COMP0 should be set to the desired period of taking temperature measurements

e. COMP1 should be set to the minimum time required to power up the Si7021 to take a temperature measurement from power on. You can find this value in the Si7021 datasheet
    i. Assuming your code turned on the LED in the Energy Mode assignment in COMP0 and turned it off in COMP1
f. Lowest Energy mode of operation = EM2; block EM3

4. Develop software ladder flow chart for I2C driver to read the temperature from the Si7021. This flow chart should not use pseudo instructions, but the actual C-lines of code.

5. Develop a software ladder flow chart of the sequence to Load Power Management ENABLE the Si7021 using pseudo code instructions

6. Develop a software ladder flow chart of the sequence to Load Power Management DISABLE the Si7021 using pseudo code instructions

7. You can read/take a temperature measurement either through the HOLD or NO HOLD command. Your ladder chart should match which mode that you implement

Measure Temperature, Hold Master Mode	0xE3
Measure Temperature, No Hold Master Mode	0xF3

8. Si7021 configuration:
    a. Use 14-bit temperature resolution
    b. 14-bit resolution is the default reading resolution, so no requirement to write to USER Register 1

9. Pearl Gecko I2C initialization configuration changes from LAB 3a.
    a. Do not use CLTO or BITO interrupts. Comment out or remove these lines of code.
    b. With the SCL and SDA GPIO pins being controlled by the LETIMER0 COMP0 and COMP1 interrupts, you can comment out or remove these lines from your I2C initialization routine

10. The initial defined set point to compare the Si7021 temperature against will be defined as 15C.

11. Implementing Load Power Management
    a. You should initialize all of your peripherals before you enter main.c's while(1) loop
    b. The SENSOR_ENABLE pin should be set to '0' to isolate and power down the Si7021 when you initializae its GPIO configuration

c. The SCL and SDA pins should be set to DISABLE and not WiredAND while SENSOR_ENABLE is set to '0'
d. On COMP0 interrupt, you will ASSERT the SENSOR_ENABLE pin to '1' to connect the Si7021 to the Pearl Gecko and to power up the Si7021.
e. On COMP1 interrupt you will need to:
    i. Use your block_sleep_mode to the lowest energy mode for I2C master operation
    ii. Change the SCL and SDA pins from DISABLED to WiredAND
    iii. Reset the I2C state machines of both the Si7021 and the Pearl Gecko similar to what you have done is assignment 3a
    iv. Read the temperature from the Si7021
    v. Convert the value from the Si7021 to degrees C
    vi. If the temperature is below the defined temperature, turn-on LED0
    vii. If the temperature is above the defined temperature, turn-off LED0
    viii. Change the SCL and SDA from WiredAND to DISABLE
    ix. Turn-off SENSOR_ENABLE by clearing it to 0
    x. Use your unblock_sleep_mode to release the I2C energy mode hold

12. Development / Debug suggestions:
    a. DO NOT IMPLEMENT load power management yet!
    b. Set the desired LETIMER0 COMP0 and COMP1 values
    c. Focus on getting your I2C read of the Si7021 temperature first
        i. Create the Software Ladder flow chart with c-code to read the Si7021 temperature
        ii. Implement the Si7021 temperature read function from the Software Ladder flow chart
        iii. Read the Si7021 temperature during each LETIMER0 COMP1 interrupt
        iv. Take the result of the read, a 16-bit value, and convert the value to degrees C
            1. The equation to convert the 16-bit Si7021 value to degrees C can be found in the Si7021 data sheet
            2. It should be a separate function compared to the actual I2C read function
            3. Turn-on LED0 if temperature is below the defined temperature set point
            4. Turn-off LED0 if the temperature is above the defined temperature set point
        v. Use the debugger to validate the result of the temperature calculated by setting a breakpoint after the temperature has been converted to degrees C.
            1. Note: 70F is equivalent to 21C
            2. If you are getting values that appear correct, put your fingers around the device to warm it up, and use the debugger to take a

new measurement. Does the temperature read back show a temperature rise?

d. With the read functionality now working, you can now consider implementing Load Power Management
    i. Create the Software Ladder Flow chart for Load Power Management ENABLED
    ii. Make the required code changes per the Software Ladder flow chart for COMP0 interrupt
    iii. Make the required code changes per the Software Ladder flow chart to COMP1 interrupt
    iv. Note: You will need to remove code that you previously had that is being superseded by these code changes
    v. Validate that you are still taking correct temperature readings
        1. Use the debugger to validate the result of the temperature calculated by setting a breakpoint after the temperature has been converted to degrees C.
            a. If you are getting values that appear correct, put your fingers around the device to warm it up, and use the debugger to take a new measurement. Does the temperature read back show a temperature rise?
e. Now it is time to complete the Load Power Management by disabling the Si7021 to save energy when not required
    i. Create the Software Ladder Flow Chart for Load Power Management DISABLED
    ii. Implement the above Software Ladder Flow Chart in your code in the LETIMER0 COMP1 interrupt
    iii. Validate that you are still taking correct temperature readings
        1. Use the debugger to validate the result of the temperature calculated by setting a breakpoint after the temperature has been converted to degrees C.
            a. If you are getting values that appear correct, put your fingers around the device to warm it up, and use the debugger to take a new measurement. Does the temperature read back show a temperature rise?
f. Validate your design by reviewing the energy profile. Is your code at the energy / current level as expected in the different times within the LETIMER0 period?

Questions:

1. Answer questions to LAB 3B quiz

Deliverables:

1. Submit 3 Software Ladder Flow Charts
    a. I2C read of the Si7021 temperature
    b. Si7021 Load Power Management ENABLE
    c. Si7021 Load Power Management DISABLE

2. Export you project as an archive, .zip file, and upload into canvas

Rubric: (Base grade is 20 pts)

1. I2C read and write flow charts		6 pts
2. I2C software project		14 pts
	a. Total	20 pts
3. Deductions:		
a. Magic numbers		- 3 pts
b. Not unique files per peripheral		- 3 pts
c. No acknowledgement of IP		- 2 pts