ECEN 3360
Digital Design Lab #7
Using the LDMA for LEUART receive
Spring 2019

Objective: This assignment's goal is to realize energy savings by offloading the highest energy consuming peripheral, the CPU, from receiving bytes of data to the LEUART from the BLE module. While the Direct Memory Access, DMA, is occurring, the processor should be in the lowest energy sleep mode, EM2.

Note: This assignment will begin with the completed Lab 6 assignment.

Lab 7 Due: Thursday April 18th, 2019

Lab 7 Instructions:

1. Make any changes required to Lab 6, Using the LDMA for LEUART transmit. Please work with the Instructing Team to get your code working to Lab 6.

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, true, true);

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. Implement a function to set up the structures required for LDMA reception of data from the BLE module
    a. What should the source of the DMA receive be?
        i. What will be the address increment size of the source?
        ii. Please refer to the LDMA chapter in the Pearl Gecko Reference Manual
    b. What should the destination of the DMA be?
        i. What will be the address increment size of the destination?
        ii. Please refer to the LDMA chapter in the Pearl Gecko Reference Manual
    c. There are predefined structures that you can use for the different descriptor modes in the Pearl Gecko HAL documentation
        i. Which one should you use for this assignment?
            1. LDMA_DESCRIPTOR_SINGLE_M2M_BYTE
            2. LDMA_DESCRIPTOR_SINGLE_M2M_HALFWORD

3. LDMA_DESCRIPTOR_SINGLE_M2M_WORD
4. LDMA_DESCRIPTOR_SINGLE_P2M_BYTE
5. LDMA_DESCRIPTOR_SINGLE_M2P_BYTE
6. LDMA_DESCRIPTOR_SINGLE_P2P_BYTE

d. There are predefined structures for the LDMA configuration as well
   i. Which one should you use for this assignment?
      1. LDMA_TRANSFER_CFG_PERIPHERAL
      2. LDMA_TRANSFER_CFG_PERIPHERAL_LOOP
      3. LDMA_TRANSFER_CFG_MEMORY
      4. LDMA_TRANSFER_CFG_MEMORY_LOOP

5. A helpful example can be found at Silicon Labs Git Hub site:
   a. https://github.com/SiliconLabs/peripheral_examples/blob/public/usart/spi_dma_master/src/main_s1_pg1_efr.c

6. To enable DMA from the LEUART down into the EM2 mode, you will need to properly enable this function in the LEUART0 peripheral
   a. Refer to section 17.3.10 DMA Support in the Pearl Gecko reference manual
   b. You can enable this functionality by writing directly to the LEUART0 register or use the proper HAL eml routine

7. With the DMA descriptor and configuration functions created, you must assign a DMA channel # for this DMA operation
   a. No specific number is required
   b. All DMA channels need to be unique, must be different than the transmit DMA channel

8. The above sets up the basic data structures and peripheral to enable the DMA to occur. Now it is time to modify you program code.
   a. You will replace the code to receive commands from the BLE UART with a DMA transfer
   b. You will need to create an input buffer, string, to handle the DMA reception from the LEUART0
      i. Your DMA input buffer should be large enough to handle any expected input size
      ii. Assume that during grading that a size greater than 4 will be entered

9. The basic procedure to use the DMA to receive is the following:
   a. Enable the LEUART0 to support DMA in low energy modes
   b. Set up the LEUART0 to use STARTF with the RXBLOCKEN to begin receiving the command and thus initiate the DMA transfer
      i. Interrupts for RXDATAV should be disabled. Instead of the processor handling the data reads, we want the DMA to handle the transfer to the memory input buffer

c. Call the function to set the DMA descriptors and configuration for the LEUART receive channel
d. Call the function to enable the DMA, LDMA_StartTransfer, for the LEUART receive channel
    i. Pease refer to the Pearl Gecko HAL documentation for additional details
e. Upon the last byte transferred, #, the SIGF will generate an interrupt
    i. Set RXBLOCKEN
    ii. Copy data that was received in the DMA input buffer to the string that you use to decode the message
    iii. Once the DMA buffer is available for a new command
        1. Call the function to set the DMA descriptors and configuration for the LEUART receive channel
        2. Call the function to enable the DMA, LDMA_StartTranfer, for the LEUART receive channel
    iv. The DMA is now set for the next command
f. With the data in the string that you had previously used to decode the incoming message, you can now call the command decode function

10. The Pearl Gecko should be in sleep mode during the DMA transfers, reducing the energy expenditures of receiving the commands from the BLE module for the Pearl Gecko

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.

Deliverables:

1. Project code exported to Canvas for grading
2. Energy Profiler screen shot of the DMA reception of the command from the BLE module to the Pearl Gecko.
3. Lab 7 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 files) - 3 pts
    d. No acknowledgement of IP - 2 pts
5. Late Submission:
    a. Due date to Friday, April 19th, at 11:59pm - 5 pts
    b. After the 19th by before Sunday the 20th at 11:59pm - 10 pts
    c. After April 20th - 15 pts

6. If code does not work, corresponding Quiz questions will be marked as 0 if answered correctly.