Lab 5 Prelab

**Name**: Chimzim Ogbondah

**Lab Partner Name (if you worked together and are submitting the same document or mostly the same answers):**

**Lab Section**: 1

**Submit your prelab document as a PDF file in Canvas under the corresponding prelab assignment. Every student submits their own prelab. Lab partners are allowed to work on the prelab together and submit the same document (if there is actual collaboration on the document). For full credit, the prelab must be submitted prior to the start of lab. Text responses should be typed or printed neatly.**

1. System information

Refer to Video 12.2a (UART Background and Launchpad Support) in Chapter 11 of Valvano and Yerraballi ES online book, <http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C11_SerialInterface.htm> (section 11.2 UART).

Video 12.2a: <https://www.youtube.com/watch?v=wdOqQCNWC3c&feature=youtu.be>

The authors sketch three scenarios for using a UART on the TM4C123 LaunchPad board. Note that we have a version of the third scenario that uses a MAX3232 driver chip (you can see this chip in the Cybot baseboard schematic, photo, or actual board near the DB9 serial socket).

Sketch your own version of the third scenario, similar to that shown in the video. Your sketch should specify UART1 and the actual GPIO port pins used to communicate with the PC in the lab – this is the same UART as in Lab 4.

A close up of text on a white background

Description automatically generated

1. For a baud rate of 9600, what values should be loaded into the IBRD and FBRD registers?

UART Integer Baud Rate Divisor (UARTIBRD), e.g., UART1\_IBRD\_R == 16,000,000/16/9600 = 104.16 = **105**  
UART Fractional Baud Rate Divisor (UARTFBRD), e.g., UART1\_FBRD\_R .166667\*64 = 10.67 = **11**

Note: A 16 MHz system clock for the TM4C123 microcontroller is used in lab. Also ClkDiv = 16.

Refer to any of the following resources about setting the baud rate for a UART on the TM4C123:

* In the Bai book, see sections 8.5.3.7 and 8.5.5.2 (also in mtg08-slides-UART.pdf, page 29).
* In VY ES book, see section 11.2.2 and information about IBRD and FBRD (remember, we are using the 16 MHz system clock).
* In the Tiva datasheet, see section 14.3.2.

1. Consider the UART Line Control register (UARTLCRH).
2. Write a line of code to initialize the UART Line Control register for the following serial communication parameters.

8 data bits, 1 stop bit, no parity, FIFOs disabled, no interrupts

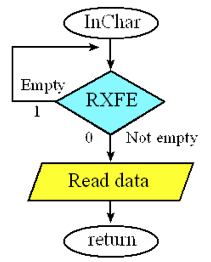
UART1\_LCRH\_R = 0b0000000001;

0x01000000;

1. What is the 32-bit memory address of UART1 LCRH register (in hex)?

O 0x4000\_D02C

1. Consider the flowchart from Figure 11.8 in section 11.2.3 in Chapter 11 of Valvano and Yerraballi ES online book, <http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C11_SerialInterface.htm> . This shows a busy-wait or polling approach to receiving data by a UART.



* 1. The RXFE flag is a bit in one of the UART registers. What is the name of the register? What is the bit number for the RXFE flag?
     1. UART1\_DR\_R , 5
  2. Instead of polling (i.e., waiting in a loop) to receive data, an interrupt approach can be used. This lets the main program do other tasks. A separate interrupt handler function is executed whenever a character is received by a UART. The main program doesn’t have to busy-wait and instead just gets data when it needs it. This is shown in Figure 12.2 from the VYES book. For example, think about an analogy of a classroom: suppose the instructor is like a processor executing a main program of teaching, including receiving student questions. Briefly explain the difference between polling and interrupts with this or another analogy. For example, how the instructor receives questions in relation to doing other tasks.
     1. Polling would be the teacher going over a topic and then waiting to see if there are any questions before moving on to the next topic, While interrupt would be understanding that the topic has been met and then moving on if there are no questions instead of being stuck in the loop.

