Quiz 4 Study Guide

1. **Given the configuration code of a timer, know how to calculate its interval. (Know which macro refers to which clock source or counting mode).**

|  |
| --- |
|  |

1. **What are some of the properties of timers? What can they be used for? How is their power consumption? How are they affected by low power modes? How independent they are from the CPU?**

|  |  |
| --- | --- |
| * Timers are simple digital counters that in active mode increment or decrement their value at a specified clock frequency. * Initialization – Specify timer’s operation mode, clock frequency, and whether it will raise an interrupt once the counter reaches a value. * Power consumption – Using timers allows you to use power on the timer instead of counting through the CPU and leaving it on. Counting on the CPU is power hungry and cumbersome, use timers instead. * Low power mode – \_BIS\_SR(LPM0\_bits + GIE); // Enter Low Power Mode 0 * \_BIS\_SR(LPM3\_bits + GIE); // Enter Low Power Mode 3 * CPU Independence – a different clock signal than the processor, it is possible either to turn off the processor or to work on other computations while the timer is counting.  |  | | --- | |  | |

1. **Given a code for two timer channels, know how to calculate the duty cycle (The diagram representing different output modes will be provided during the quiz)**

|  |
| --- |
| **https://www.ti.com/lit/an/slaa513a/slaa513a.pdf?ts=1603998875530&ref\_url=https%253A%252F%252Fwww.google.com%252F** |

1. **Given the block diagram and the configuration code, be able to determine the source of a timer.**

|  |
| --- |
|  |

1. **What are the functions of the Watchdog timer? What is its primary function, and what can it be configured to do?**

|  |
| --- |
| - The MSP430 Watchdog Timer’s primary function is to perform a controlled-system restart after a software problem occurs.  - WDTCNT is a 32-bit up counter not accessible by software.  https://www.ti.com/lit/ug/slau399f/slau399f.pdf?ts=1603941410146&ref\_url=https%253A%252F%252Fwww.google.com%252F |

1. **What is the purpose of different registers and values like UCA0RXIFG, UCA0TXIFG, UCA0TXBUF, UCA0RXBUF, UCA0TXIE, UCA0RXIE in the code? What does it mean when they are set?**

|  |
| --- |
| * UCA0RXIFG – When a character is received in the UCA0RXBUF register, the UCA0RXIFG bit is set. * UCA0TXIFG – interrupt flag is set by the transmitter when the UCA0TXBUF is   ready to accept a new character.   * UCA0TXBUF – Register to transmit characters out to the command line. * UCA0RXBUF – Register to hold received character * UCA0TXIE – Transmit buffer interrupt enable bit * UCA0RXIE – Receive buffer interrupt enable bit |

1. **What is the size of a string or buffer created by sprintf? (Hint: Do not forget about the null character '\0')**

|  |
| --- |
| The size of the buffer should be large enough to contain the entire resulting string (see [snprintf](http://cplusplus.com/snprintf) for a safer version).  Note the function’s return type is int - it returns the length of the converted string.  [**http://cplusplus.com/reference/cstdio/sprintf/**](http://cplusplus.com/reference/cstdio/sprintf/)  **https://www.educative.io/edpresso/how-to-use-the-sprintf-method-in-c** |

1. **Know the difference between Putty/MobaXterm and UAH Serial App.**

|  |
| --- |
| * Putty – Can only display ASCII characters and can only receive characters at 8-bit size. * UAH Serial App – Allows us to view the different data types (int, float, etc.). This app translates serial packets that are sent to it, and it can graphically represent the data versus time. The UAH Serial Application expects a packet that has a 1-byte header followed by the data followed by an optional checksum. * Moba X Term - remote computing. |

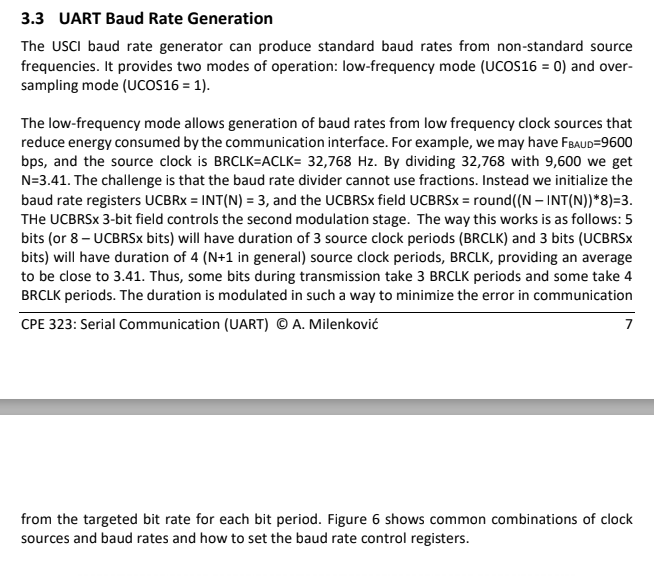
1. **Be able to describe the functionality of a given serial communication code snippet.**

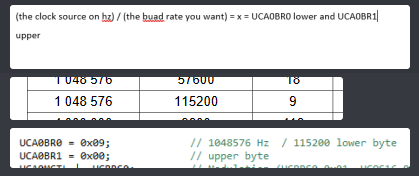
|  |
| --- |
| Pull up the Demo Code. |

1. **What is the difference between Synchronous and Asynchronous communication? What configuration needs to be shared between the nodes for asynchronous communication? How do we do that in MSP430 (Hint: divider and modulation)? What happens if the modulation value is wrong?**

|  |  |
| --- | --- |
| * Asynchronous – exchange of data between two or more parties without the requirement for all the recipients to respond immediately * Since the two devices do not share a clock signal, there should be an agreement between the devices on the speed of the communication before the actual interface starts. * Synchronous – two or more people exchange information in real-time and must share a common clock source. * Modulation value wrong – the fraction part of the division of clock frequency by the baud rate will accumulate and eventually make the two devices unable to communicate.  |  | | --- | |  | |

**11.)**

****

****