Lab 2; you will demonstrate the code here in class. The quiz answers are to be entered in the blank test on BlackBoard.

|  |
| --- |
| Watch the video and try out the code first. |
|  |
| 1) On line 13, the variable I is defined as volatile. <https://en.wikipedia.org/wiki/Volatile_(computer_programming)> |
| The root word “vol” means to fly, flying or flies.  Here is an example:  volplane  1. To glide toward the earth in an airplane with the engine turned off.  2. To move by gliding or to travel, or to move, by gliding. |
| Which of the following is wrong?  a) the variable defined as volatile may change between different accesses  b) Prevents an optimizing compiler from optimizing read/write operations  c) Volatile values primarily arise in hardware access (memory-mapped I/O) because in memory mapped I/O reading/writing to an I/O port is no different than reading/writing to a RAM location.  d) In multi-threading, two or more threads cannot modify the same variable. |
| 2) On line 13, we see uint32\_t. How can it be broken up? Make the wrong answer  a) t stands for type and u stand for unsigned  b) int stands for integer  c) 32 stands for 32 bits  d) not of the above |
| 3) on line 15, the vertical line “|” stands for  a) AND  b) OR  c) EXOR  d) NOR |
| 4) You may open the header file ccsv6\ccs\_base\arm\include\msp432p401r.h with notepad. Make sure you do not modify it. Line 15 of the code will place which value in WDTCTL?  a) 0x805A  b) 0x5A000080  c) 0x5A80  d) 0x8000005A |
| 5) We decided to double check our answer above by opening the variables windows and tracing our code. As you can see the contents of the watchdog timer control register is not what we thought it to be.    Chapter 15 of MP432 technical reference manual (<http://www.ti.com/lit/ug/slau356a/slau356a.pdf>) explains this.  Which of the following is wrong.  a) This is how the microcontroller is designed.  b) You 69h is the password for writing; there is a mistake in our code.  c) it is ok to read 069h.  d) you will always read |
| 6) On line 10 what value is loaded into R1?  a) 0x40004C05  b) 0x40004C03  c) 0x40004C08  d) 0x40004C99 |
| 7) On line 11, we are what into R0?  a) value stored in R1  b) 0x40004C03  c) port two’s out register  d) it depends on the state of the processor. |
| 8) The EOR instruction stands for (save this locally: <http://www.ti.com/lit/ug/spmu159a/spmu159a.pdf>)  a) EXNOR b) EXOR  c) OR d) NOT |
| 9) On line 13, we are  a) storing the contents of R0 into memory location pointed to by R1  b) storing the contents of R1 into memory location pointed to by R1  c) storing the contents of R1 into memory location pointed to by R0  d) storing the contents of R0 into memory location pointed to by R0 |
| 10) The commented out line 15 can accomplishes the same as the line 16; they both go back to the main calling program. Please try it out yourself – don’t ask someone else for the anser.  a) true  b) false  c) it depends  d) it happens |