Special Topics in Computer Engineering/Software Engineering
0306-672-01, 0306-772-01, 4010-590-01 Real-Time and Embedded Systems Topics
Fall Quarter 2003 (2003-1) Drs. Roy Czernikowski and James Vallino, course instructors

Textbook: Real-Time Systems and Software by Alan C. Shaw, John Wiley & Sons, 2001 ISBN 0-471-35490-2
Reading Assignment: Textbook pages 1 – 19 inclusive – to be completed by 2 pm Wednesday, September 10, 2003
Video Assignment: View the roughly 90 minute video on the Motorola 68HC12 architecture before 1:00 pm Wednesday September 10, 2003. The M68HC12 video is available on-line. From a windows machine on campus, you can select Start -> Run and then enter "//nibbler.ce.rit.edu/pub"
You can then play the divx file directly from the server. The drivers are located there too under "DivX50Bundle". The resolution of the middle icon HC12 with the number "1" appearing at its center, is better than that viewed from the third icon at that site.

First Programming Project (Version 1.1): Write an embeddable, stand-alone 68HC12 assembly language program that will form a histogram of pulse interarrival times of one-thousand-one pulse samples. (One needs 1001 pulses to measure 1000 interarrival times.) The interarrival time between pulses is expected to average around 1.0 milliseconds, but your software should be able to count the number of interarrival times that are shorter than 980 microseconds, the number of interarrival times that are between 980 and 981 microseconds, the number between 981 and 982 microseconds, etc. in one microsecond intervals ending in the total number of interarrival times longer than 1020 microseconds. (Thus there are 42 "bins" in which your program should tabulate interrarrival times; the first bin contains the count of interrarrival times less than 980 microseconds, the second bin counts the interrarrival times between 980 and 981 microseconds, the 41st bin counts the number of interrarrival times between 119 and 120 microseconds, and the 42nd bin counts the number of interarrival times that were greater than 120 microseconds.) Your program is to display one line of text indicating the interarrival time (beginning with the shortest interarrival time ***observed*** in the range mentioned above) followed by the number of such samples. Your output is to pause until a carriage return input is accepted from the keyboard before displaying the number of interarrival times in the next interval. The reason for pausing between output categories is to allow the user to examine what's happening without possibly 42 lines of text scrolling off the screen. **Note that the graduate students in this course are to display counts in each interval as decimal integers; the undergraduates may display the counts as three hexadecimal digits.**
The first executable instruction is to begin at location $D00 in EEPROM. The stack pointer should be initialized to $A00 and descend down through roughly $9E8. You will have RAM locations $800 to whatever the maximum stack depth to use. It is important that you carefully calculate the maximum stack depth and include a memory utilization map in your report.
Note that a little later in this course you will be using this program to measure the software jitter of a commercial real-time operating system's attempt to generate pulses at exactly 1.0 milliseconds (= 1000 microseconds) between rising edges of the pulse.

This project is due to be completed and demonstrated to a course instructor no later than class time, Wednesday, September 24, 2003. Note that no demo will be conducted without a complete, hardcopy write-up given to the course instructor before officially observing the demo. Other notes: The program is to begin execution by sending a message to the screen instructing the user to enter a carriage return <CR> in order to start the data collection process. The program should echo the <CR> ($0D) followed by a line-feed character <LF> ($0A) and THEN start the data collection.