EE360 – Project 6 (20 points) (Demo by April 14, upload code before you demo the project)

1. Using C, design a digital stop watch (on a Dragon-12 board) that will

* initially count from 00.00 to 59.99 seconds, where the first two digit represent the second count and the second 2 digits represent represents 1/10th and 1/100th of a second, respectively.
* When the one minute has passed, the stop watch with change to 1:00.0 where the first digit will represent the minute count and second 2 digits will show the seconds and the last digit will show the 1/10th second value
* after 10 minutes, the watch will show 10:00, where the first two digits will show the minute count and the last two digits will show the second count.

The counts will be displayed on the 7-segment displays. The stop watch should a single key (say, SW5 – connected to Port H pin 0) for control. The key can be used at any time to initialize start and stop the watch. Since the buttons are generally prone to bounce, you must implement a debouncing scheme that would guarantee that a button press is recognized only after 200 ms after the last button bounce. However, the seven segment output must not freeze during this time.

The key works in the following manner using Port H interrupts. If it is pressed first time it will initialize the stop watch to 00.00. The next time if it is pressed again, it will start the watch. The third press will stop the watch at the current time that would be displayed on the 7-segment display and at the same time play a 3-tone siren using Output Capture function OC5 function (pin 5 of Port T) with each tone lasting for 1 second. The frequencies of these 3-tone are 500Hz, 1000 Hz and 1500 Hz. After playing the 3 tones, the siren should stop. The next button press will initialize the stop watch.

In this problem, you must not use any delay function provided by the author of the textbook nor can you use the RTI interrupts. However, you can use another Timer using, say, OC6 function with interrupts for all other delays in the project. All delays and time intervals should use HCS12 Timer system discussed in Chapter 8. The operation cycle should repeat forever.

After you have successfully implemented the project, measure the frequencies and estimate the errors.

|  |  |  |
| --- | --- | --- |
| Desired Frequency (Hz) | Project 6 Frequency (Hz) | Project 6 percent error |
| 500 |  |  |
| 1000 |  |  |
| 1500 |  |  |

Report the byte count of your project.