EE247 Programming for Embedded Systems Project #1 Spring 2016

Overlapped "Blinking Light" Program

The goal of this project is to program the HCS08SH8 microcontroller to blink two LED's (those on the demo board) in an "overlapped" pattern, similar to what is needed to drive a stepper motor. This project is essentially the same as EGR222 Lab 7, except that a report is required. When the pushbuttons are used on the demo board, the speed should change and/or the direction should reverse. The speed of blink needs to be slow enough so that the phenomenon can be observed.

See the EGR222 Lab7 write-up for directions and help information. What we are doing will differ in that we are using the HCS08SH8 instead of the HCSQG8. Our processor works on 5V, rather than 3.3V, and has a different clock speed. The connector pinouts also differ.

There are some things to do that go beyond what the EGR222 exercise does. First, you should leave the "watchdog timer" operating. So in your "busy wait" loops to give the timing delay for each step, reset (feed) the watchdog. You are also encouraged to connect a stepper motor to the demo board to see it run rather than just look at blinking LED's. You can use a 74LS244 (in pairs of buffers) to drive the motor windings instead of discrete transistors to reduce the complexity, allowing you to do it with 2 IC's: 74LS04 inverter and 74LS244 buffer.

Include a timing analysis for your delay loops. You should look at the disassembled C code (available in the debugger) and see what instructions are used to execute your delay loop. There are several ways to do the delay, but the simplest is:

```
for(i=init;i>0;i--){__RESET_WATCHDOG;}  /* loop for time determined by init */
```

Here the variable "init" is a value derived from the setting of the pushbutton(s). So the question is, how long does it take to do each full loop? Once you know the instructions that are executed to do the loop, you can look them up in the list of instructions ands see how many clocks each one takes. Knowing that, you can find the number of clocks per loop. Looking in the startup code, you can find what the clock frequency is (different for SH8 and CG8 microcontrollers; the CG8 is faster). Finally, knowing the initial value you put in from the pushbuttons, you can calculate the speed you should see for different settings of the pushbuttons. Compare with how fast you observe the LED's to blink. Is it consistent?

Your report is informal. It is to consist of an abstract (a summary paragraph which summarizes the entire report), a copy of your C program (include appropriate comments), the assembly language for one of the timing loops, your calculations to show the loop timing, and your observations of blinking light speed for various pushbutton combinations. Include schematics showing any external circuitry if you drive a stepper or use other external circuitry. The circuit can be hand drawn. Finally, include a conclusions paragraph which should include a statement saying whether it worked or not, and if not, why.