EE 308 – Homework 7

Due Mar. 16, 2009

For all problems below assume your are using a MCS12DP256 chip with a 24 MHz bus clock and a 8 MHz oscillator clock.

1. Write a program to do the following: Set up bits 0, 1, 2 and 3 of Port T to capture the time of a rising edge. Use interrupts to do this. In the main program loop, wait until all four inputs have captured an edge, then print the time of edges using the DBug-12 `printf()` routine.

2. An engineer is using the HC12 to determine the speed of a motor in RPM. A pulse is generated on Bit 1 of `PORTT` 16 times every revolution of the motor. Bit 1 of `PORTT` is set up for input capture mode, and captures the time of the rising edge. The prescaler bits PR2:0 are set to 010. It is known that the time between pulses is less than the timer overflow time. When the first edge is captured, the TC1 register has a value of 0xC5A6. When the second rising edge is captured, the TC1 register has a value of 0x2794.

   (a) What it the length of time between the two rising edges?

   (b) How long does it take the motor to make one revolution?

   (c) What is the motor speed in RPM?

3. What setup do you need to do to have the HC12 set bit 3 of `PORTT` to a high on a successful output compare? Write some C code to do this.

4. You want to generate a 250 Hz square wave using Output Compare 3. What value should you add to TC3 in your `toc3_isr()`? (Assume that the prescaler bits PR2:0 = 101.) What else should you do in your `toc3_isr()`? Write the interrupt service routine `toc3_isr()` to do this.

5. Write a C program that generates a 40 ms delay after an external event (rising edge) happens. Set up Bit 4 of `PORTT` for input capture of a rising edge, and Bit 5 of `PORTT` for Output Compare. The HC12 waits for a rising edge on Bit 4 of `PORTT`. It then generates a rising edge on Bit 5 of `PORTT` exactly 40 ms after the rising edge is detected on Bit 4. Bit 5 is left high for exactly 40 ms, then is brought low.

6. Write some C code to set up the MC9S12 to capture the time of a falling edge on Bit 2 of Port T, and to generate an interrupt when the capture occurs. Set the interrupt vector to call the function `tic2_isr`. Set the timer overflow interrupt rate to about 175 ms.

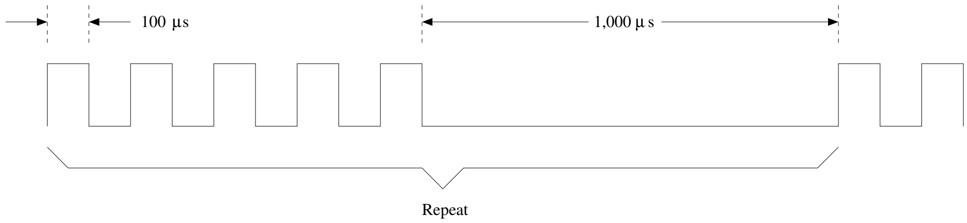
7. Write an interrupt service routine called `tic2_isr` which capture the times of two consecutive falling edges on Bit 2 of Port T, and set a global variable to the time between the two falling edges.

   - Use a `static` flag in the ISR to indicate whether you should capture the first or second edge.
   - If the flag indicates that you should capture the time of the first edge, save the time in a `static` variable (e.g., `time_1`). Set the flag to indicate that the time of the first edge has been captured, and that the next time the ISR is entered, it should capture the time of the second edge.
   - If the flag indicates that you should capture the time of the second edge, save the time difference between this time and the time of the first edge in a global variable (e.g., `time_diff`). Set the flag to indicate that the time of the second edge has been captured, and that the next time the ISR is entered, it should capture the time of the first edge.

8. Write some C code to set up the MC9S12 to set up Bits 3 and 4 of Port T as Output Compare, and to generate interrupts when the compares occur. Set the interrupt vectors to call the function `toc3_isr` and `toc4_isr`. Set the timer overflow interrupt rate to about 175 ms.

9. Write an interrupt service routine called `toc3_isr` which generates a 250 Hz square on on Bit 3 of Port T.

10. Write an interrupt service routine called `toc4_isr` which generates a signal which looks like this:


100 µs
1,000 µ s
Repeat