Sonar Ranging

Objective:
Students will demonstrate ability to design, write and implement a Sonar Ranging program. They will use the microcontroller's internal timer (Enhanced Capture Timer) Input Capture feature to interface with sonar device and measure signal travel time. They will demonstrate ability to utilize data sheets to calculate timer input capture values using integer math.

Lab 10 description and requirements
Program Specification
This Program will use the Enhanced Capture Timer (ECT), but will not use the interrupts. This program will require a regular build….”not the small”

Write a C program to measure distance using a sonar device, and display it on the LCD display. Label the feet (FEET) and inches (INCHES) on the display. For example, an object 5½ feet away should be reported as *5 FEET, 6 INCHES*. The sonar device transmits ultrasound from a small speaker, and picks up echoed (reflected) sound on a microphone. The program will essentially (but not directly) multiply the time between the sound transmission and the echo reception by the speed of sound to yield the distance.

Sonar Implementation

The MC9S12 timer input capture feature will be used to measure the time for the sonar signal return trip. Port T (PTT) bit 6 is connected to the *init* input on the sonar ranging module (speaker). The sonar *echo* line is connected to Port T (PTT) bit 7 (microphone).

To determine the round trip time you need to capture the clock times for the IC6 and IC7 (init and echo) events. You can monitor the flag on the *echo* input capture signal to determine when the echo has returned. The time between *init* going high and *echo* going high represents the roundtrip time to the target.

Set the capture edge for both input captures to rising edge capture in register TCTL3 (ECT_TCTL3). Be sure to reset flags C6F and C7F in register TFLG1 (ECT_TFLG1) by writing ones to the corresponding bits before every measurement. Turn on the timer module by setting the TEN bit in register TSCR1 (ECT_TSCR1). Program the input capture designator bits in register TIOS (ECT_TIOS).

Start the sonar by setting the *init* signal (low-to-high transition). Do not reset it until you have detected that the echo has returned (C7F). The sonar roundtrip time is represented by the difference in the two input capture clock counts TC6 and TC7 (ECT_TC6 & ECT_TC7).

The clock speed for the system is 8 MHz. The speed of sound in dry air is ~331.45 meters/second, or 1116 feet/sec. Write the program so that a distance of up to 20 feet can be measured before the compare registers overflows. Do not forget that the roundtrip is twice the actual distance (hence 40 feet for the maximum roundtrip – about 36 milli sec). The sonar ranging module cannot detect objects closer than 1.33 feet (2.66 ft. roundtrip, or about 2 msec).

You may need to slow down the timer clock speed by writing to the *prescale* bits in register TSCR2 (ECT_TSCR2).

The pushbutton connected to bit 4 of PORTS (PTIS) must be used to start the measurement. Make a measurement whenever the switch goes low (pressed). Be sure to wait for the switch to go back high before starting the next measurement. Debounce the rising edge of the switch signal by delaying approximately 10 milliseconds after the switch is first detected in the high state. You can use the delay routine from the LCD display lab. There is no need to debounce the falling switch edge since the distance measurement test takes several milliseconds. Note, that running the sonar device continuously will cause power to drop below an acceptable value, so only run it once when the switch gets pressed.

IMPORTANT NOTE: The sonar module plugs into the top left side of the trainer board.

Display Implementation

To get the display data, convert the echo travel time count to inches, using a pre-determined scaling factor. Calculate this scaling factor in advance and use integer arithmetic with the scaling factor in your program. The scaling factor can be a simple division. Note: You must use integer arithmetic.

The inches to feet conversion is easy. The number of inches beyond the nearest foot can be determined by using the C language modulus operator **%** to get the remainder after a division by 12.

To convert the feet and inches (integer variables) to ASCII to display, the *sprintf(char *s, char fmt, ...)* can be used. When using *sprintf()*, You will need to add the `#include <stdio.h>` at the top of the .c file. Function s*printf()* writes a formatted string to the buffer pointed to by s, and can be used in the same way as *printf* to form a formatted ASCII string for output to the LCD.

For example:
```
char buffer[33];
sprintf(buffer, “D= %d Feet %d Inches”,  feet, inches);
```
The address of the buffer can then be sent to your LCD display code written in a previous laboratory.
(Research and make sure you understand how these two lines work).

Debugging

Put a break-point after the start-button check and verify that the code waits for the switch to go low. Check to make sure the sonar echo (input) line goes high after the init line goes high. Hardware test points are available on the board, and memory locations and ports can be examined using the debugger. Check to make sure that CF6 and CF7 (TFLG1) are low before the start of the distance procedure and are high after the echo line goes high.

Before you code!

Use the provided document, “Set up Timer for lab 10”, to outline your code. You must complete this document and understand your program flow before you begin to code your solution.

Submission: your complete zipped project and your completed outline document.