EE 525.415 Embedded Microprocessor Systems

Lab 3, quadrature encoder input

Purpose

To interface a quadrature encoder as a simulated flow rate sensor input to the cpuboard.

Background

We have been asked by a major petroleum producer to design a better flow monitor for gas pumps. Our flow rate sensor will be a small in-line turbine, connected to a quadrature encoder. Each revolution of the encoder represents 1 ounce of product.

Equipment Required

cpu board, IDE, and usb cable. Development system.
Grayhill 25LB10-Q quadrature encoder, or equivalent.
Pull up resistors – (typically 2.2 to 10k, ¼ w), 2 pieces.
No gasoline is required for this lab.

Notes

The Grayhill 25LB10-q is a quadrature encoder module.
(http://www.digikey.com/product-detail/en/25LB10-Q)

The "C" or common line needs to be grounded, and lines A and B pulled up to +3.3 volts through resistors (2.2k – 10k). The outputs A and B provide a 2-bit code.

Procedure

Interface the quadrature encoder to the cpu board. Write a program to input the states repeatedly. Send the resulting data to the pc host over the serial line. The pc should display rotation direction and a rotation rate (which implies using a timer). You spin the sensor manually. You could chuck it up in a drill..

How would you determine rotational velocity from the position data?

Convert the revolutions to gallons, accurate to 1 decimal place and display. This can be done on the host computer. Set an (arbitrary) limit to the flow rate. If the rate exceeds the limit, there might be a problem, so send a signal to an external device that can shut the pump down. Indicate this by an LED. Backflow could also be a problem, so check to see the device is not rotating in the "wrong" direction.

Here's a real-world application ripped from the world news of Fall, 2015. The "Press" says this required sophisticated software. I say it only takes a couple of lines of code. Here's the scenario, A major German auto manufacturer wants to bypass emissions testing in a clever manner. They speculate they could have two tables of engine control data, one for operation of the car, giving max performance and fuel mileage, and another to meet the emissions specs. What they need is a way to determine when an emissions test is being conducted.

A clever auto mechanic suggests that since the emissions test is done on a dynamometer, only the front (drive) wheels are turning. In normal operation, all four wheels are turning. So, using the car's quad encoders, present at each wheel, we could read 1 and 2 (front wheels) and 3 and 4 (back wheels). These are normally used for anitlock braking and traction control. But our algorithm is, when the front wheels are turning, but not the back, switch to "special" engine control table 1. Otherwise use standard engine control table 2.

Of course, this is highly illegal, and will result in massive fines, not to mention bad air...
So, how many lines of code (LOC) would it take you to do this?