Lab 9e Capacitance Meter

This laboratory assignment accompanies the book, Embedded Microcomputer Systems: Real Time Interfacing, Second edition, by Jonathan W. Valvano, published by Thomson, copyright © 2006.

Goals
- Mixing digital and analog signals,
- Hysteresis,
- Period, pulse width, and phase measurements.

Review
- Valvano Section 6.1 on input capture interrupts,
- Valvano Section 6.7 on pulse width modulation,
- Valvano Section 11.1.1 on resistors,
- Valvano Section 11.1.2 on capacitors,
- Valvano Section 11.2.10 on voltage comparators,
- The chapter on input capture and output compare in the 9S12DP512 data sheet,
- The chapter on pulse width modulation in the 9S12DP512 data sheet

Starter files •**`OC3`** and **`IC`** projects

Background

There are three fundamental translations studied in this lab. The first translation is the conversion from capacitance to time. The typical approach is to use a resistor to create a period, pulse width or phase delay that depends on the capacitance you are trying to measure. The time constant of a simple RC circuit is $\tau = R*C$. You can look at integrated devices like the LM555 and 74LS123 to get a sense of how this translation operates. However, in this lab you will design the interface electronics with discrete resistors, capacitors and rail-to-rail op amps.

The second translation will be from analog to digital. Analog signals in an embedded system can exist at any voltage level within the power supply voltages. In our system, this will be 0 to +5V. Digital signals are characterized as high or low. If a digital input voltage is below V_{IL}, then the signal is considered low. Similarly, if a digital input voltage is above V_{IH}, then the signal is considered high. An important factor with interfacing signals to a digital input port of the microcontroller is to limit the time the signal exists between V_{IL} and V_{IH}. For example, the SPI specification states the time for rise and fall transitions for digital inputs should be less than 8 ns. A signal with a rise time on the order of msec can not be connected directly to Port T input capture.

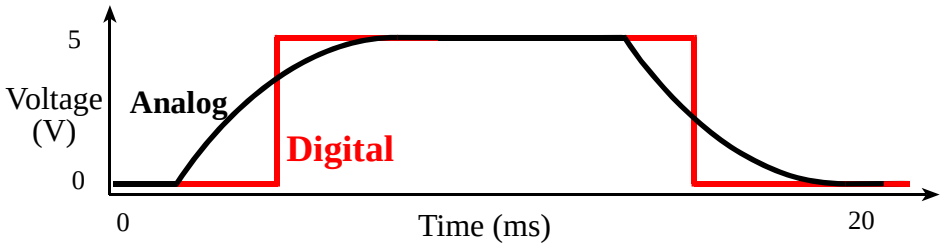

Figure 9.1. A one-bit analog to digital convertor or threshold detector.

The third translation will be from digital to analog. You can not place a large capacitive load on a CMOS digital output, because it will create output signals similar to the analog trace in Figure 9.1. Signals like the analog curve will cause excess currents to flow within the input pathways, even when the pin is defined as an output. When transmitting digital signals over long distances, we normally use line drivers or buffers to isolate the capacitive load in the cable from both the digital output on the transmitter and the digital input on the receiver, as shown in Figure 9.2. For more information about transmission lines look up RS232 and CAN in the book.

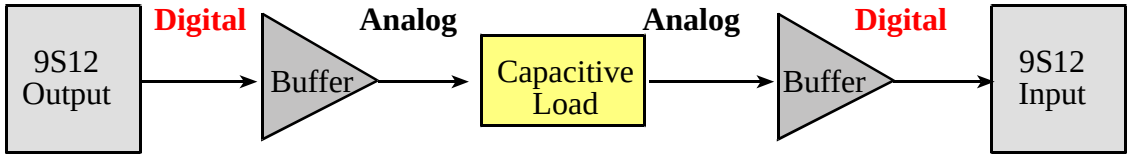

Figure 9.2. Capacitive loads need to be buffered from CMOS digital logic.

Requirements

In this lab you will design, construct and test a capacitance meter. The input range is 1 to 220 nF. The capacitor type will be ceramic with leads. The capacitor under test will be plugged into a special spot on your protoboard. The measurements will be displayed on a LCD as fixed point numbers in either nF or μF. There is no particular sampling rate necessary for this machine; however, the measurements should occur repeatedly. The measurement processes will occur in the background and the LCD display processes will occur in the foreground. You must use input capture, but you are free to use period, pulse width or phase measurements. The average measurement accuracy should better than 10% of reading. Your TA will have a set of standard capacitors he/she will use to test your machine. You will have access to these standards during testing. The calibration experiments and measurement accuracy experiments must be performed on separate days.

You will need to design an analog to digital interface as illustrated in Figure 9.1. Hysteresis will be required to eliminate extra input capture edges that might occur as the slowly varying input crosses the threshold. Notice the two threshold voltages in Figure 9.1. The analog to digital buffer needed in this lab has the following transfer function:

- V_t is the threshold voltage
- Input can handle low slew rate signals
- If the digital output is low and the analog input goes above (V_t+x) then the digital output goes high
- If the digital output is high and the analog input goes below (V_t-y) then the digital output goes low
- x and y are about 0.1 to 0.2V

The digital to analog buffer needed in this lab has a seemly simple transfer function:

- Digital low converted to analog 0V
- Digital high converted to analog +5V
- Output can drive capacitive loads
- No hysteresis is desired because the time in transition is very short

Preparation (do this before your lab period)

Part a) Explain how your system will convert capacitance to time. Develop the fundamental equations your system will use to measure capacitance. In particular, the system should use input capture to measure period, pulse width or phase.

Part b) Design the hardware interface between the capacitor connection spot (two places on your protoboard) and the 9S12. Label all hardware chips, pin numbers, and resistor values. Draw the circuit using PCB Artist. Make a printout of your SCH file. We expect you to use resistors, capacitors and rail-to-rail op amps.

Part c) Write the low-level measurement driver, creating **meter.h** and **meter.c** files. Put the prototypes for public functions in the **meter.h** file. Decide whether to use a FIFO or a mailbox to link the measurements occurring in the ISR with the display process occurring in the foreground. You may not use shared global variables to pass data between modules. For example, you could define a function **`Meter_In`** (which will be called by the main program) that reads from a private FIFO or mailbox and returns a new capacitance measurement. Figure 9.3 shows a potential data flow graph of the system. There needs to be a simple and obvious way to enter calibration data into the module.

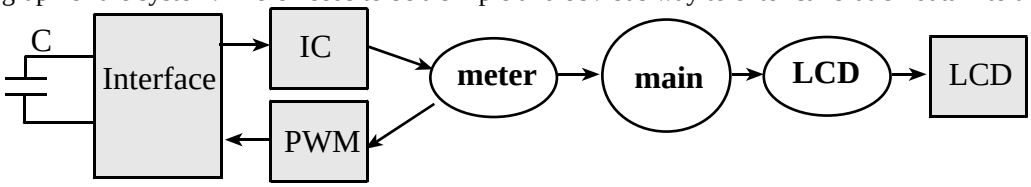

Figure 9.3. Data flows from the keyboard to the main program. The main program can use the LCD.

Part d) Write a main program that implements the capacitance meter. Figure 9.4 shows a possible call graph of the system. Dividing the system into modules allows for concurrent development and eases the reuse of code. Notice the details of input capture, port pin assignments and the FIFO queue are hidden from the main program.

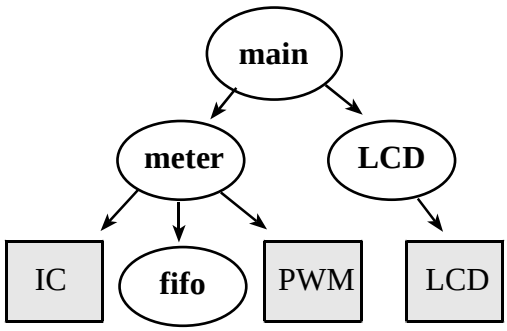

Figure 9.4. A call graph showing the capacitance meter.

Procedure

Part a) Using a dual trace oscilloscope measure the important signals in your system. Print screen shots for at least two capacitor values. Use these measurements to verify your basic approach described in preparation a)

Part b) Using about 5 capacitance standards, collect calibration data. Enter these data as constants in your system. If your system were to be mass produced, a simple yet effective calibration procedure will greatly affect the profitability of the product.

Part c) Write a main program to measure the probability density function (pdf) of your noise process. We will assume the noise added to the data is independent from sample to sample. Place a capacitor in the system. Since the input is fixed, variations in measurements arise from noise processes. First, measure the capacitance 1000 times and determine the minimum and maximum of these measurements. Define from 10 to 50 bin counters (for example, if the data ranges from 11.6 to 13.9 nF, then bins could be 11.6 to 11.7, 11.8 to 11.9, …, 13.8 to 13.9 nF). Define a global variable for each bin. Measure the capacitance 1000 times again, and increment the corresponding bin counter for each measurement. Using the counts in each bin plot a frequency distribution of your noise. This frequency distribution is an experimental estimate of the noise pdf. What kind of noise do you have? For example, white noise has a Gaussian-shaped probability density function. What noise process could have caused the noise pdf in Figure 9.5? Do not expect your data to look like Figure 9.5. If most of your data are identical, then you could perform this analysis on your raw input capture measurements.

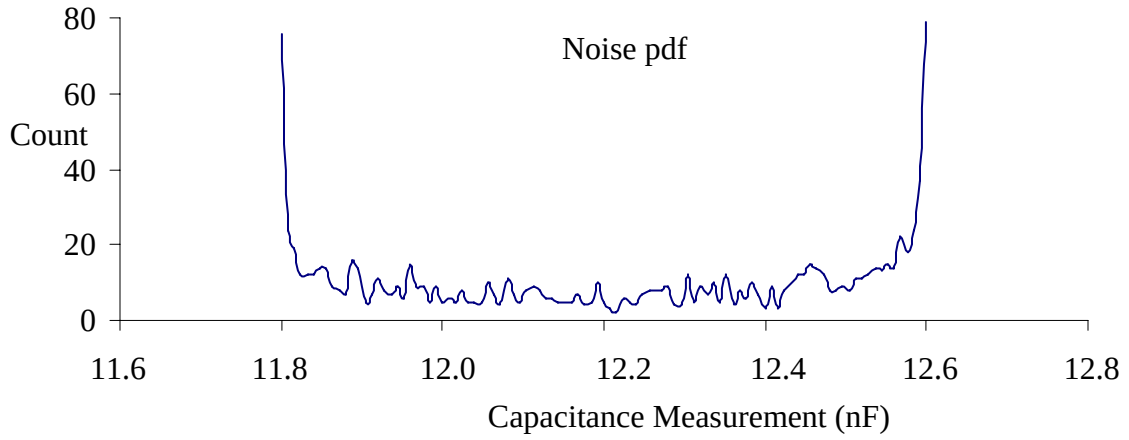

Figure 9.5. Probability density function caused by a particular type of noise.

Part d) Using at least 5 capacitance standards, collect accuracy data. Perform this experiment on a separate day from the calibration. Calculate accuracy of reading as a percentage.

$$\text{Average accuracy (with units in percent)} = \frac{100}{n}\sum_{i=1}^{n}\left|x_{ti} - x_{mi}\right| / x_{ti}$$

Deliverables (exact components of the lab report)

A) Objectives (1/2 page maximum)

B) Hardware Design

Hardware interface, showing all external components (Preparation b)

C) Software Design (a hardcopy software printout is due at the time of demonstration)

Explain how your software measures capacitance (Preparation a), including calibration

If you organized the system different than Figure 9.3 and 9.4, then draw its data flow and call graphs

D) Measurement Data

Signal measurements on scope (Procedure a)

Calibration data (Procedure b)

Measured noise pdf (Procedure c)

Accuracy data (Procedure d)

E) Analysis and Discussion (1 page maximum)

Discuss your noise process and the one in Figure 9.5

What limits accuracy?

Checkout

Demonstrate the functionality of your capacitance meter including calibration. Convince your TA that your FIFO implementation has no critical sections (could be theoretical proof or experimental observations.) You will be asked to describe the two ways the critical section could have been removed from the bad FIFO implementation.

A hardcopy printout of your software will be given to your TA, and graded for style at a later time.