ME 104
Pre-lab 2

1. **`yourname_lab2_AcquireAI.vi`**
   Write a VI to acquire and display the voltage output from the Hall-Effects sensor circuit (use DAQ exercise from Lab 1 as a template) with the following specifications:
   a. Acquire analog voltage from Analog Input Channel 0, using Referenced Single-Ended (RSE) terminal configuration.
   b. Acquire analog voltage until stopped by the user and at a sampling rate specified by the user.
   c. Specify the input voltage range (maximum and minimum voltages) corresponding to output voltages of the Hall Effects sensor.
   d. Display the acquired voltage on a chart as well as on a digital indicator.
   e. Track errors and indicate if an error occurs.
   f. Save VI as **`yourname_lab2_AcquireAI.vi`**

Extra
   g. Allow users to choose the analog input channel.

2. **`yourname_lab2_AcquireAI_CountLP.vi`**
   Modify (add to) **`yourname_lab2_AcquireAI.vi`** to also count and display the number of low pulses that have been produced by your Hall-Effects sensor circuit.

   Since the Hall sensor outputs a voltage of (approximately) either 0 or 5 volts, every time a new low pulse occurs, the present voltage measurement V_i will be less than the immediately preceding voltage measurement V_{i-1} by approximately 5 V. Therefore, every time the voltage difference $V_i - V_{i-1}$ is less than, say, -4 V, you know that a new pulse has occurred.[1] Therefore your strategy for detecting a new pulse will be to compute the voltage difference $V_i - V_{i-1}$ at each iteration of the While loop and then check whether that difference is less than -4 V. If the difference is less than -4 V, a low pulse counter is incremented by one. Where do you get the *i-1* value? The answer is to use a *Shift Register*[2] to store the *i-1* value.

   Your low pulse counter should reset to 0 every time your restart your VI. To do so, initialize the shift register for the counter to 0.

   Hint: The **Less?** Function,  , will output a Boolean TRUE value whenever a new low pulse occurs and a Boolean FALSE value otherwise. To convert the Boolean TRUE or FALSE values to counting numbers (integers) 1 or 0, you can use the **Boolean To (0,1)** function, .

[1] The threshold voltage magnitude of 4 V was chosen for convenience. In practice, a threshold voltage magnitude of anywhere between 0.5 V to 4.5 V should work just as well.
[2] See *LabVIEW 7* User Manual April 2003 Edition, pg. 8-6 to 8-7.

The **Boolean to (0,1)** function will output a 1 whenever a new low pulse occurs and a 0 otherwise. By adding the outputs from this function, you can create a low pulse counter. The easiest way to perform this addition is to use an Add function and another shift register.

Save your VI as **`yourname_lab2_AcquireAI_CountLP.vi`**.

3. **`yourname_lab2_AcquireAI_CountLP_TimeLP.vi`**
   Modify (add to) **`yourname_lab2_AcquireAI_CountLP.vi`** to also compute and display the total time during which the output from the Hall-Effect sensor circuit is low (less than 1V).

   A straightforward way to do this is--at each iteration of the While loop--to check whether the voltage output V_i is less than 1V. Every time this is true, a "Low Pulse Timer" should be incremented. (By what value should the Low Pulse timer be incremented?)

   Save this VI as **`yourname_lab2_AcquireAI_CountLP_TimeLP.vi`**.

Extra Credit Exercise: Count Long Low Pulses (5-10 points)

In this extra credit exercise, your task is to count and display the number of "long low pulses" that are being produced by your Hall sensor circuit. For the purposes of this exercise, a "long low pulse" is defined as any low pulse that has a measured duration of 200 ms or longer. Note that, since the VI's you have been using in this lab sample at the rate of once every 100 milliseconds, the shortest measured duration of a single low pulse is 100 milliseconds.

1. Open **`Yourname_Lab2_AcquireAI_CountLP_TimeLP.vi.`**
2. Modify (add to) this VI so that you can not only count and display the number of low pulses, but also count and display the number of long low pulses. (Hint: See *Shift Registers*, page 3-14. The **And** function from the **Functions>Booleans** sub-palette may also be useful.)