

**ME 104**  
**Sensors and Actuators**  
**Fall 2003**

**Laboratory 2**  
**Hall-Effect Sensors**

Department of Mechanical and Environmental Engineering  
University of California, Santa Barbara

Fall 2003 Revision

## Introduction

In this laboratory, you will learn how to use a Hall-effect sensor to detect the presence of a magnetic field due to a permanent magnet. You will then write a *LabVIEW* program to count the number of times that a magnet is moved past the Hall sensor, in a simulation of an attendance counter at a turnstile. Finally, you will write a program to compute and display the total (cumulative) duration of low pulses. As an extra credit exercise, you will write a program to count and display the number of long low pulses, that is, low pulses that are at least 200 ms in duration.

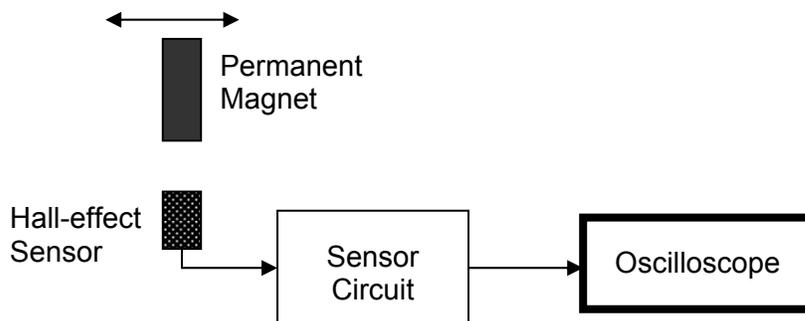
## Background Reading

Please read the following material prior to this lab:

1. Hall IC's Data Sheet, *DN6848/SE/S*, pages 1-4, Panasonic. Available online at <http://rocky.digikey.com/WebLib/Panasonic/Web%20data/DN6848,SE,S.pdf>.
2. *LabVIEW User Manual*, July 1998 Edition, *Shift Registers*, pages 3-13 and 3-14. Available online at [www.ni.com/pdf/manuals/320999b.pdf](http://www.ni.com/pdf/manuals/320999b.pdf).

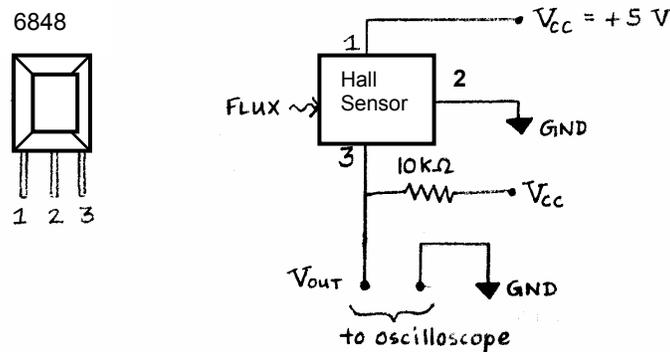
## Experiment #1: Build an Analog Circuit for a Hall-Effect Sensor

In this experiment, you will build a simple analog circuit for using the Panasonic DN6848 Hall-effect sensor as a magnetic field detector. You will then test the performance of your sensor circuit using an oscilloscope.



**Figure 1.** Hall-effect sensor system

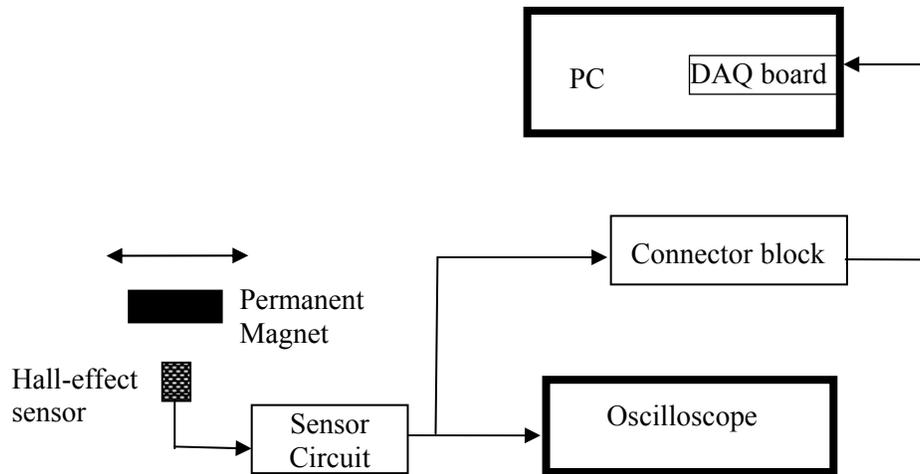
1. Build the analog circuit shown in Figure 2 on an electronic breadboard. Remember to use red wire for positive power connections and black wire for ground connections. Provide power ( $V_{CC} = 5$  volts) and ground (GND) to your circuit board using the “5 V FIXED 3 A” output from your Tektronix PS280 DC Power Supply. (Use the provided banana connectors.)
2. Turn on your oscilloscope and set its vertical scale to 5 volts/division and its horizontal scale to 1 second/division. Connect the voltage output (Hall sensor pin 3) and ground (GND) from your circuit to the oscilloscope. (Use wires with stripped ends).
3. Verify that, in the absence of a magnetic field, the voltage output from the Hall sensor circuit is of the same magnitude as the supply voltage ( $V_{CC}$ ) to your circuit. Move a magnet near the Hall sensor and observe the resulting low-voltage pulses on your oscilloscope. As indicated on page 3 of the Hall sensor data sheet, the magnetic flux should be applied from the unmarked side of the Hall sensor.



**Figure 2.** Hall-effect sensor circuit diagram.

## Experiment #2: View Hall-Effect Sensor Output on Computer Screen

In this experiment, you will build a *LabVIEW* VI to sample and display the output from the Hall sensor circuit you built in Experiment #1.



**Figure 3.** Hall-effect sensor computer interface

1. Open `yourname_lab1_ex2b.vi`.
2. Click on **Windows>>Show Diagram** to display your block diagram.
3. Using the *Labeling Tool*, change the chart label “Random Plot” to “Analog Input Chart.”
4. Using the *Labeling tool*, change the millisecond multiple control from 250 to 100. This will speed-up your display rate from 4 Hz to 10 Hz.
5. To view the analog voltage signal from the Hall sensor, connect the voltage output (Hall sensor pin 3) from your sensor circuit to the CB-68LP connector block as shown in Table 1. (Use wires with stripped ends.) Do not remove the connections to the oscilloscope.

**Table 1.** CB-68LP connector block pin assignment for measuring a voltage signal using Analog Input Channel 0.

External Signal	Connect to:
Hall sensor voltage (pin 3)	Pin 68 (ACH0)
Analog ground	Pin 67 (AIGND)

6. Click the **Run** button to see the output from the Hall sensor circuit. Verify that the low voltage pulses viewed on your computer screen are similar to the low voltage pulses (simultaneously) viewed on your oscilloscope.

## Experiment #3: Count Low Pulses

In this experiment, you will write a program to count and display the number of low pulses that have been produced by your Hall 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.\* 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.

1. Use the same VI and setup described in **steps 1 to 4** from Experiment #2 (above).
2. Using the *Positioning tool*, enlarge the area enclosed by your While Loop so that is roughly double what it originally was.
3. Using the *Positioning tool*, move the millisecond multiple control, the Wait Until Next ms Multiple function, the Stop button terminal, and the conditional terminal on the While loop towards the bottom of the While loop. Select several objects at once by drawing a box around the objects. To draw the box, click on the top-left corner of where you want your box to begin and then hold down the button while you drag to where you want your box to end. Release the button and all objects enclosed by your box will be highlighted.
4. Click down and hold the mouse button on one of the highlighted objects and drag down until all four highlighted objects have been moved near the bottom boundary. Release the button. Your block diagram should be similar to the one in Figure 4.

---

\* 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.

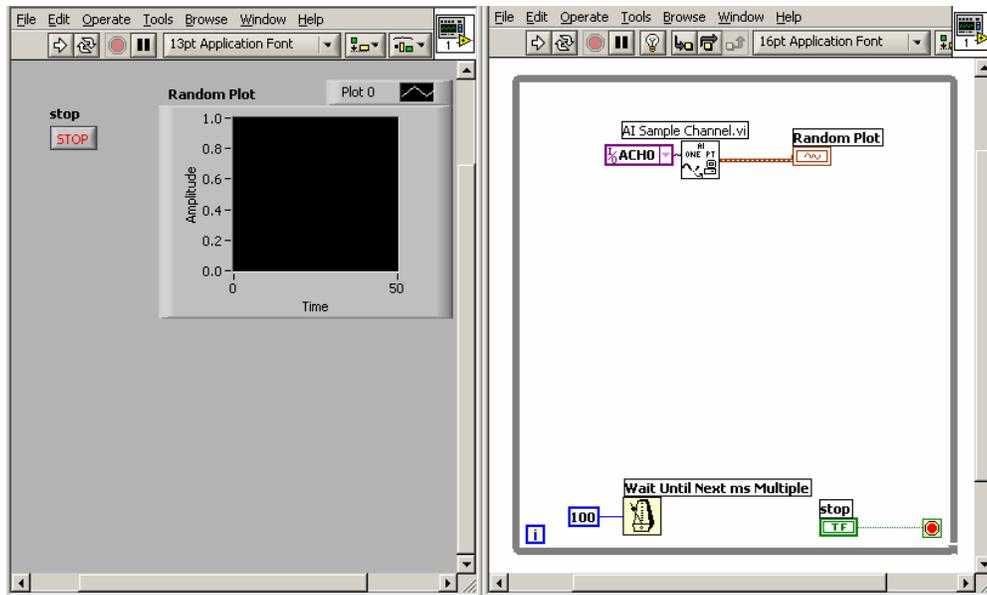


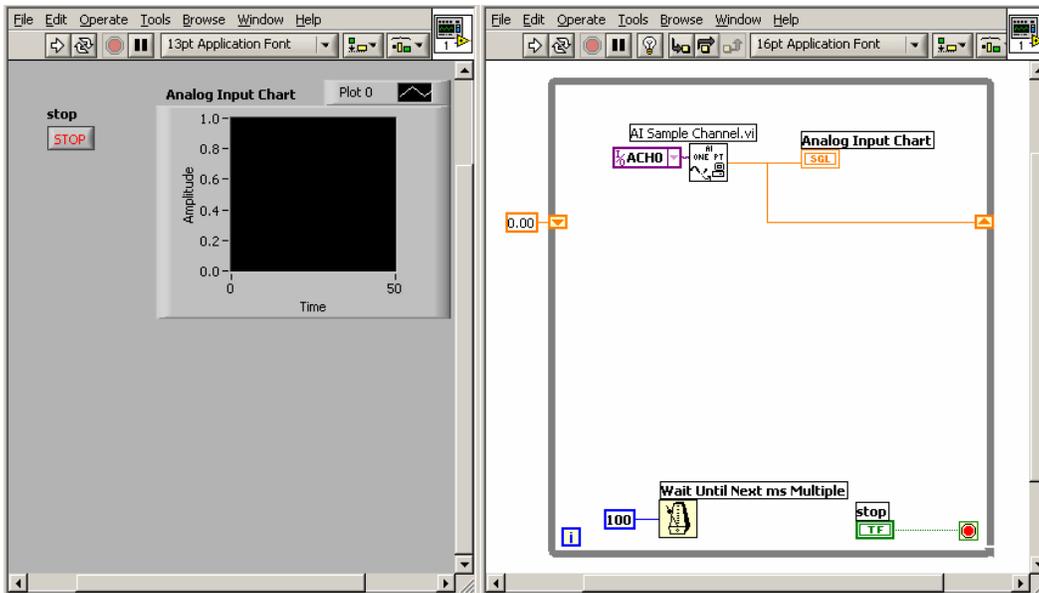
Figure 4. VI after completing steps 1- 4 of Experiment 3.

How do you compute the difference between the  $i^{th}$  value of a variable in the While loop and the corresponding  $i-1^{th}$  value? The answer is to use a *Shift Register*.\*

5. Pop up (right click) on the right border of the While loop and select **Add Shift Register**. Two terminals should appear.
6. Using the *Wiring tool* from the Tools palette, wire the sample terminal (output) from the AI Sample Channel.vi to the right terminal of the shift-register. Right click on the sample terminal and, from the pop-up menu, choose **Select Type >> Scaled Value**. Once wiring is complete, the shift registers should turn orange to indicate that they carry floating-point data.
7. Pop up on the left terminal of the shift register and select **Create Constant**. Leave the default value 0.00 in the Create Constant control. This value is specified to provide an initial value for the voltage measurement. Your block diagram should look similar to the one in Figure 5.

---

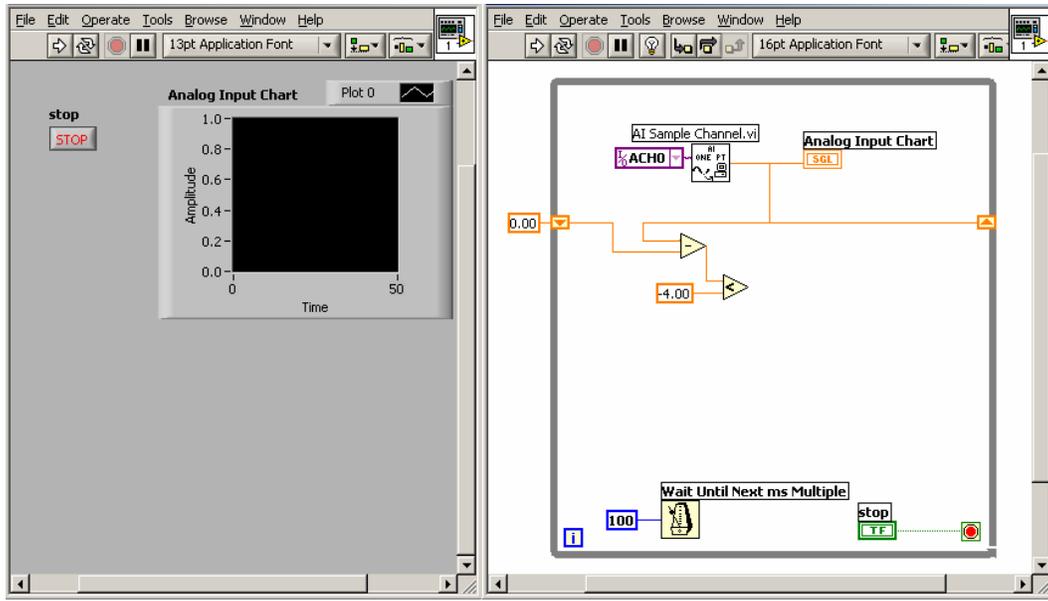
\* See *LabVIEW User Manual*, pages 3-13 and 3-14.



**Figure 5.** Your VI after completing steps 5-7 of Experiment 3.

At the end of the  $i^{\text{th}}$  iteration of the While loop, the right shift register will carry the  $i^{\text{th}}$  voltage measurement  $V_i$ , while the left shift register will carry the  $i-1^{\text{th}}$  voltage measurement  $V_{i-1}$ . To compute their difference, you can use a Subtract function.

8. Select **Subtract** from the **Functions>>Numeric** subpalette.
9. Place the Subtract Function within the While loop (near the middle).
10. Using the *Wiring tool* from the Tools palette, wire the right terminal of the shift register to the x (upper left input) terminal of the Subtract Function. To create a corner in your wire, release your mouse button and then press it down again to continue drawing the wire.
11. Wire the left terminal of the shift register to the y (lower left input) terminal of the Subtract Function.
12. Select **Less?** from the **Functions>>Comparison** subpalette.
13. Place the Less? function within the While loop below the Subtract Function.
14. Using the *Wiring tool* from the Tools palette, wire the x-y (output) terminal of the Subtract Function to the x (upper left input) terminal of the Less? Function.
15. Pop up on the y (lower left input) terminal of the Less? Function and select **Create Constant**.
16. Type the value  $-4.00$  in the Create Constant control. Your block diagram should look like the one in Figure 6.



**Figure 6.** Your VI after completing steps 8 – 16 of Experiment 3.

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.

17. Select **Boolean To (0,1)** from the **Functions>>Boolean** subpalette.
18. Place the Boolean To (0,1) function within the While loop below the Less? Function.
19. Using the *Wiring tool* from the Tools palette, wire the  $x < y?$  (output) terminal of the Less? Function to the Boolean (input) terminal of the Boolean To (0,1) Function. Your new wire should appear dotted green to indicate Boolean data. Your VI should now look like Figure 7.

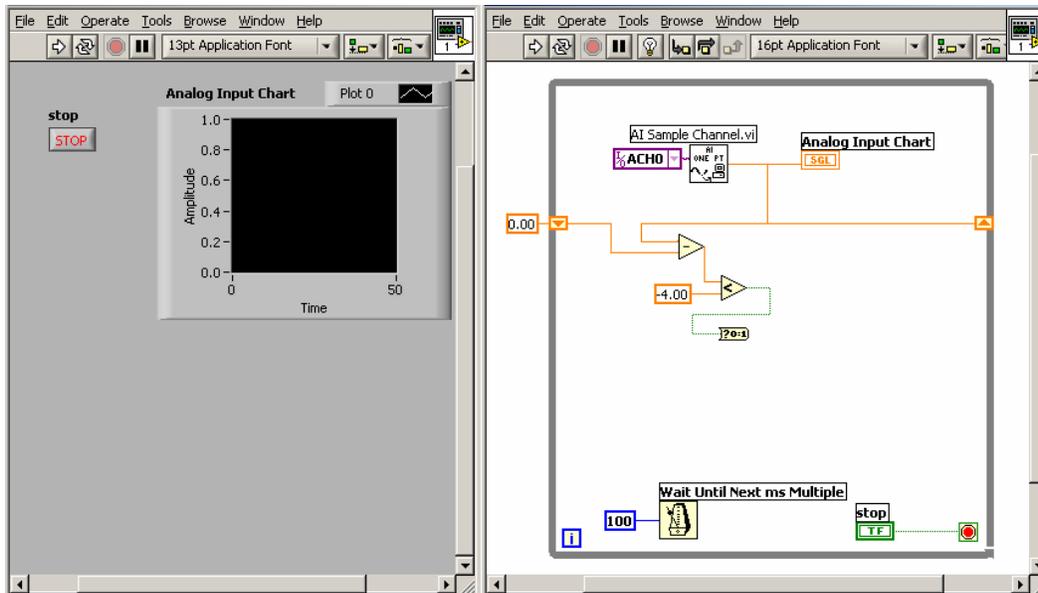
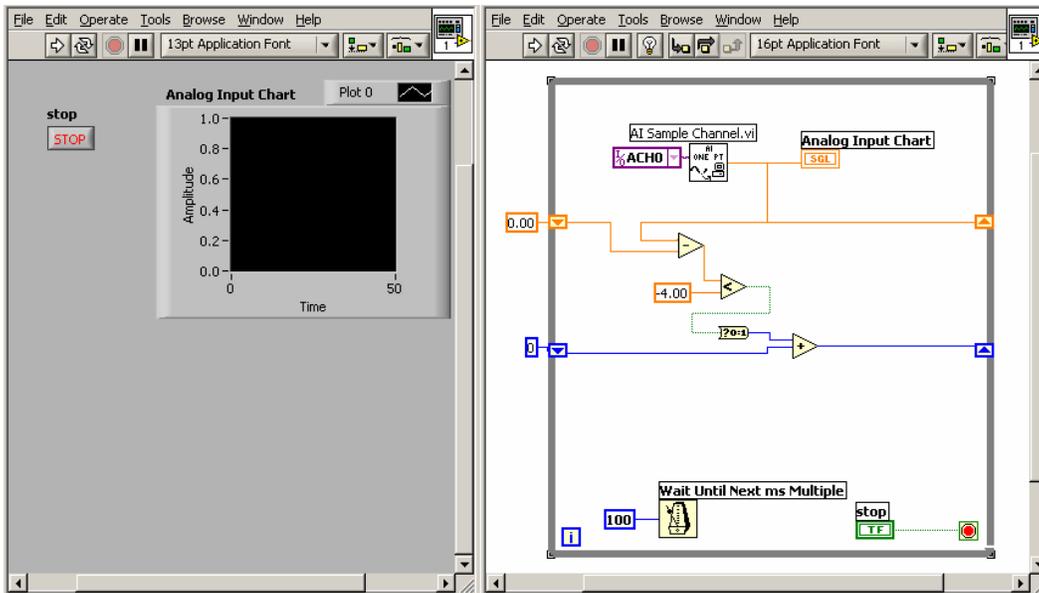


Figure 7. Your VI after completing steps 17- 19.

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.

20. Select **Add** from the **Functions>>Numeric** subpalette.
21. Place the Add function within the While loop to the right of the Boolean To (0,1) Function
22. Using the *Wiring tool* from the Tools palette, wire the 0,1 (right) output terminal of the Boolean To (0,1) Function to the x (upper left input) terminal of the Add function. Your new wire should appear blue to indicate integer data.
23. Pop up on the right border of the While loop (just to the right of the Add Function) and select **Add Shift Register**. Two new terminals should appear.
24. Wire the right terminal of the second shift register to the x+y (right) output terminal of the Add Function. The new shift registers should turn blue to indicate integer data.
25. Wire the left terminal of the blue shift register to the y (lower left input) terminal of the Add Function.
26. Pop up on the left terminal of the blue shift register and select **Create Constant**. Type the value 0 in the blue Create Constant control. This is the initial value of your low pulse counter. Your VI should now look like Figure 8



**Figure 8.** Your VI after completing steps 20 - 26.

The blue shift register's right terminal carries the number of low pulses at the end of each iteration of the While loop. To display the number of low pulses on your front panel, you should add a digital indicator.

27. Go to the front panel. Select **Digital Indicator** from the **Controls>>Numeric** subpalette.
28. Place the Digital Indicator to the right of your Analog Input Chart.
29. Using the *Labeling tool* from the Tools palette, highlight the Digital Indicator label and change the label to Low Pulse Counter.
30. Pop up (right click) on the Digital Indicator and choose **Format and Precision**. Enter 0 for the Digits of Precision. The indicator should now show zero decimals places. This is appropriate because the counter will use only integers.
31. Go to the block diagram. Using the *Positioning tool* from the Tools palette, select the Low Pulse Counter indicator and drag it into the While loop.
32. Using the *Wiring tool* from the Tools palette, wire the Low Pulse Counter indicator to the x+y (output) terminal of the Add function. Your VI should look like Figure 9.

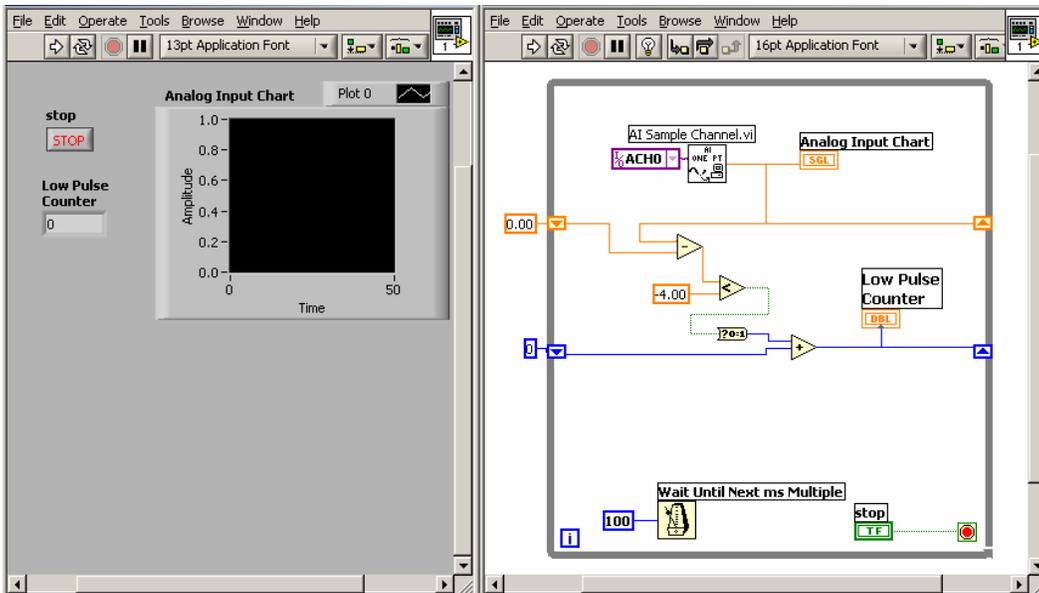


Figure 9. Your VI after completing step 27 - 32.

You are now ready to run your VI.

33. Click the **Run** button and verify that the Low Pulse Counter works. Your counter should initialize to 0 and then increment by 1 every time a new low pulse occurs.
34. Save this VI as `yourname_lab2_ex3.vi`.

## Exercise #4: Time the Total Duration of Low Pulses

In this exercise, your task is to not only count and display the number of low pulses, but also compute and display the total time during which the output from the Hall 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  $V_i$  of the top shift register is less than 1V. Every time this is true, a “low pulse timer”<sup>\*</sup> should be incremented by 100 milliseconds.

1. Open `yourname_lab2_ex3.vi`.

---

<sup>\*</sup> This name was chosen for descriptive purposes.

2. Modify (add to) this VI so that you can not only count and display the number of low pulses, but also compute and display the total time during which the output from the Hall sensor circuit was low (less than 1V).
3. Test the performance of your VI.
4. Save your new VI as `yourname_lab2_ex4.vi`.

## Saving Files

Before you leave, remember to save all of your files to **your ECI account, a floppy disk, or email it to yourself** (for later use and backup purposes). For this laboratory, you should save the following files from the Desktop:

```
yourname_lab2_ex3.vi  
yourname_lab2_ex4.vi.
```

## Laboratory Report

1. For the VI's you wrote in this laboratory (listed in the preceding section), provide a printout that shows the front panel and block diagram, similar to Figure 4 above. Hints for obtaining this printout are provided in the **Laboratory #1 Supplemental Handout**.
2. Explain how you implemented your working timer in Experiment #4.
3. Imagine you are given a Hall-effect sensor that outputs 0V in the absence of a (sufficiently strong) magnetic field and 5V in the presence of a magnetic field. In such a scenario, how would you modify your VI from Experiment #3 to work as a "high" pulse counter?

Your Lab Report should clearly state your name, Lab Report number, Lab date, and your laboratory partner's name (if any). Your lab report should be thorough, but concise. You will be graded on quality, not quantity. **Lab Report #2 is due at the beginning of Laboratory #3.**

## Additional Reading and Practice

1. Feel free to experiment with *LabVIEW* programming in the CADLab. Doing so is an excellent way to reinforce and expand on the material you learned in today's laboratory.

## Extra Credit Exercise: Count Long Low Pulses

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_ex4.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** subpalette may also be useful.)
3. Test the performance of your VI.
4. Save your new VI as `yourname_lab2_extra.vi`. (Save to a **floppy disk** before you leave).

### Lab Report

5. For your new VI, provide a printout that shows the front panel and block diagram, similar to Figure 4 above. In your own words, explain how you implemented your long low pulse counter.