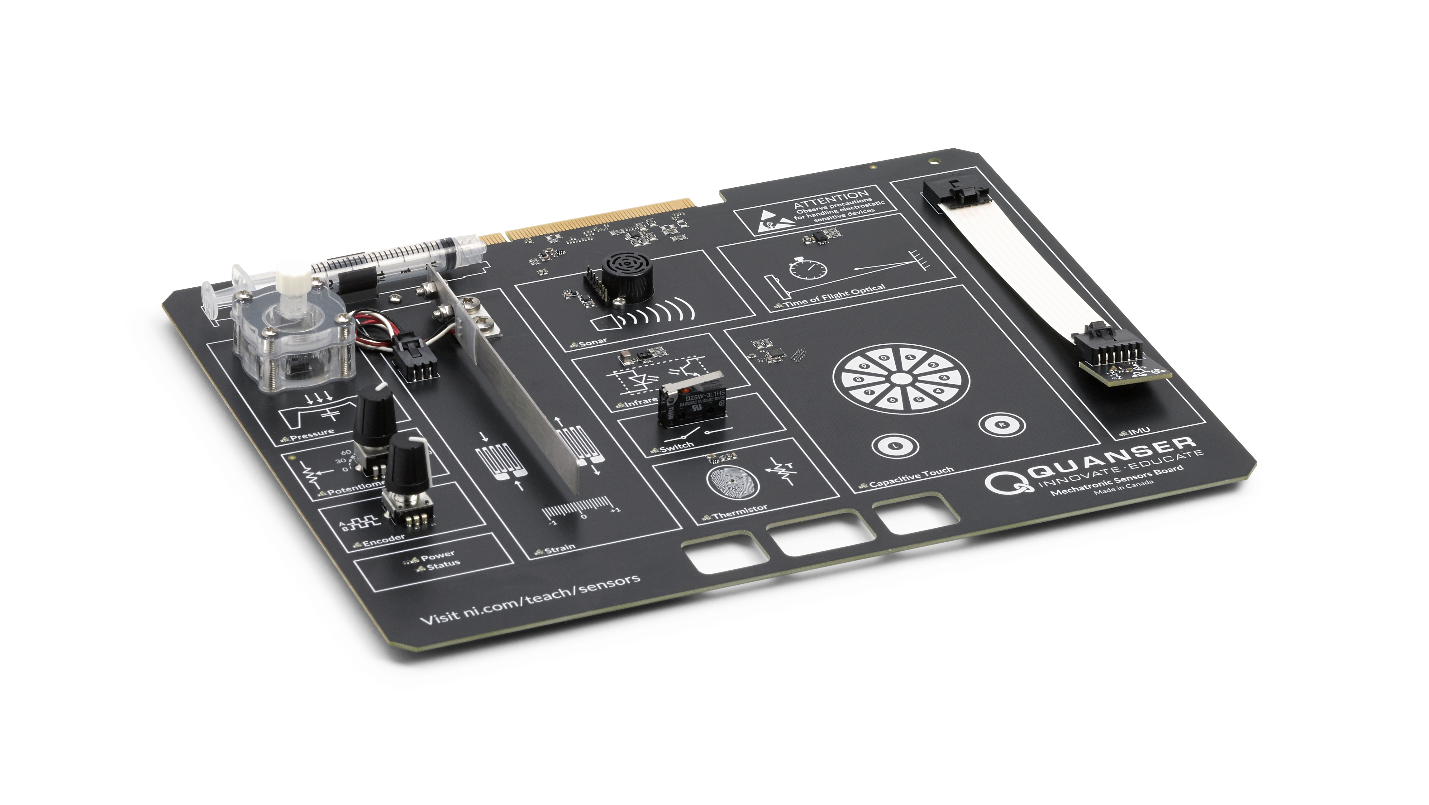


Lab Manual: Fundamentals of Mechatronic Sensors

Using the Quanser Mechatronic Sensors Board for NI ELVIS III



© 2018 Quanser Inc., All Rights Reserved

Printed in Markham, Ontario.

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. Quanser Inc. grants the following rights: a) The right to reproduce the work, to incorporate the work into one or more collections, and to reproduce the work as incorporated in the collections, b) to create and reproduce adaptations provided reasonable steps are taken to clearly identify the changes that were made to the original work, c) to distribute and publically perform the work including as incorporated in collections, and d) to distribute and publicly perform adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. These rights are granted subject to and limited by the following restrictions: a) You may not exercise any of the rights granted to You in above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation, and b) You must keep intact all copyright notices for the Work and provide the name Quanser Inc. for attribution. These restrictions may not be waved without express prior written permission of Quanser Inc.

LabVIEW and National Instruments are trademarks of National Instruments.

All other trademarks or product names are the property of their respective owners.

**Additional Disclaimers:** The reader assumes all risk of use of this resource and of all information, theories, and programs contained or described in it. This resource may contain technical inaccuracies, typographical errors, other errors and omissions, and out-of-date information. Neither the author nor the publisher assumes any responsibility or liability for any errors or omissions of any kind, to update any information, or for any infringement of any patent or other intellectual property right.

Neither the author nor the publisher makes any warranties of any kind, including without limitation any warranty as to the sufficiency of the resource or of any information, theories, or programs contained or described in it, and any warranty that use of any information, theories, or programs contained or described in the resource will not infringe any patent or other intellectual property right. THIS RESOURCE IS PROVIDED “AS IS.” ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY AND ALL IMPLIEDWARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, ARE DISCLAIMED.

No right or license is granted by publisher or author under any patent or other intellectual property right, expressly, or by implication or estoppel.

IN NO EVENT SHALL THE PUBLISHER OR THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, COVER, ECONOMIC, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS RESOURCE OR ANY INFORMATION, THEORIES, OR PROGRAMS CONTAINED OR DESCRIBED IN IT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND EVEN IF CAUSED OR CONTRIBUTED TO BY THE NEGLIGENCE OF THE PUBLISHER, THE AUTHOR, OR OTHERS. Applicable law may not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Table of Contents

[Introduction 7](#_Toc516839756)

[Learning Objectives 7](#_Toc516839757)

[Prerequisites 7](#_Toc516839758)

[Completed Courses 8](#_Toc516839759)

[Hardware, Software, and Tool Knowledge 8](#_Toc516839760)

[Organization of the Lab Manual 8](#_Toc516839761)

[Lab 1: Angular Displacement 8](#_Toc516839762)

[Lab 2: Distance and Proximity 8](#_Toc516839763)

[Lab 3: Temperature 8](#_Toc516839764)

[Lab 4: Strain 9](#_Toc516839765)

[Lab 5: Pressure 9](#_Toc516839766)

[Lab 6: Contact 9](#_Toc516839767)

[Lab 7: Inertial Measurement 9](#_Toc516839768)

[Lab Tools and Technology 10](#_Toc516839769)

[Platform: NI ELVIS III 10](#_Toc516839770)

[Hardware: Quanser Mechatronic Sensors Board 11](#_Toc516839771)

[Software: LabVIEW 12](#_Toc516839772)

[Lab 1: Angular Displacement 13](#_Toc516839773)

[Learning Objectives 13](#_Toc516839774)

[Required Tools and Technology 14](#_Toc516839775)

[Expected Deliverables 15](#_Toc516839776)

[Section 1: Measuring Angular Displacement using a Potentiometer 16](#_Toc516839777)

[1.1 Theory and Background 16](#_Toc516839778)

[1.2 Implement 19](#_Toc516839779)

[1.3 Analyze 21](#_Toc516839780)

[Section 2: Measuring Angular Displacement using an Encoder 23](#_Toc516839781)

[2.1 Theory and Background 23](#_Toc516839782)

[2.2 Implement 29](#_Toc516839783)

[2.3 Analyze 31](#_Toc516839784)

[Section 3: Design Considerations 33](#_Toc516839785)

[Lab 2: Distance and Proximity 35](#_Toc516839786)

[Learning Objectives 35](#_Toc516839787)

[Required Tools and Technology 36](#_Toc516839788)

[Expected Deliverables 37](#_Toc516839789)

[Section 1: Measuring Long-range Distance using a Sonar 38](#_Toc516839790)

[1.1 Theory and Background 38](#_Toc516839791)

[1.2 Implement 41](#_Toc516839792)

[1.3 Analyze 44](#_Toc516839793)

[Section 2: Measuring Short to Mid-range Distance using a Time-of-Flight Sensor 45](#_Toc516839794)

[2.1 Theory and Background 45](#_Toc516839795)

[2.2 Implement 47](#_Toc516839796)

[2.3 Analyze 49](#_Toc516839797)

[Section 3: Measuring Proximity using an Infrared Proximity Sensor 51](#_Toc516839798)

[3.1 Theory and Background 51](#_Toc516839799)

[3.2 Implement 53](#_Toc516839800)

[3.3 Analyze 55](#_Toc516839801)

[Section 4: Design Considerations 56](#_Toc516839802)

[Lab 3: Temperature 57](#_Toc516839803)

[Learning Objectives 57](#_Toc516839804)

[Required Tools and Technology 58](#_Toc516839805)

[Expected Deliverables 59](#_Toc516839806)

[Section 1: Measuring Temperature using a Thermistor 60](#_Toc516839807)

[1.1 Theory and Background 60](#_Toc516839808)

[1.2 Implement 65](#_Toc516839809)

[1.3 Analyze 67](#_Toc516839810)

[Section 2: Design Considerations 68](#_Toc516839811)

[Lab 4: Strain Gage 71](#_Toc516839812)

[Learning Objectives 71](#_Toc516839813)

[Required Tools and Technology 73](#_Toc516839814)

[Expected Deliverables 74](#_Toc516839815)

[Section 1: Measuring Strain 75](#_Toc516839816)

[1.1 Theory and Background 75](#_Toc516839817)

[1.2 Implement 84](#_Toc516839818)

[1.3 Analyze 87](#_Toc516839819)

[Section 2: Design Considerations 88](#_Toc516839820)

[Lab 5: Pressure 91](#_Toc516839821)

[Learning Objectives 91](#_Toc516839822)

[Required Tools and Technology 92](#_Toc516839823)

[Expected Deliverables 93](#_Toc516839824)

[Section 1: Pressure Measurement and Calibration 94](#_Toc516839825)

[1.1 Theory and Background 94](#_Toc516839826)

[1.2 Implement 100](#_Toc516839827)

[1.3 Analyze 104](#_Toc516839828)

[Section 2: Design Considerations 106](#_Toc516839829)

[Lab 6: Contact 108](#_Toc516839830)

[Learning Objectives 108](#_Toc516839831)

[Required Tools and Technology 109](#_Toc516839832)

[Expected Deliverables 110](#_Toc516839833)

[Section 1: Switch Debouncing 111](#_Toc516839834)

[1.1 Theory and Background 111](#_Toc516839835)

[1.2 Implement 116](#_Toc516839836)

[1.3 Analyze 118](#_Toc516839837)

[Section 2: Capacitive Touch Sensor 119](#_Toc516839838)

[2.1 Theory and Background 119](#_Toc516839839)

[2.2 Implement 123](#_Toc516839840)

[2.3 Analyze 124](#_Toc516839841)

[Section 3: Design Considerations 125](#_Toc516839842)

[Lab 7: Inertial Measurement 126](#_Toc516839843)

[Learning Objectives 126](#_Toc516839844)

[Required Tools and Technology 128](#_Toc516839845)

[Expected Deliverables 129](#_Toc516839846)

[Section 1: Measuring Pose and Magnetic Field using an IMU 130](#_Toc516839847)

[1.1 Theory and Background 130](#_Toc516839848)

[1.2 Implement 136](#_Toc516839849)

[1.3 Analyze 139](#_Toc516839850)

[Section 2: Design Considerations 140](#_Toc516839851)

# Lab 1: Angular Displacement



Figure 0: Accurate measurement of angular displacement is essential in many robotic applications

This lab explores angular displacement measurements using a potentiometer and an incremental encoder. Both sensors will be calibrated prior to measuring angular displacements. Different decoding algorithms for the encoders will be examined.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

1. Understand the voltage dividing properties of a potentiometer
2. Calibrate the output of a potentiometer in terms of angular displacement
3. Determine an encoder’s pulses-per-revolution
4. Examine non-quadrature, X2, and X4 decoding
5. Calibrate the output of an encoder in terms of angular displacement
6. Observe the effect of low sampling on the accuracy of measured data

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Record raw potentiometer output
* Screenshot of the potentiometer calibration graph showing fitted calibration curve
* Record potentiometer calibration coefficients
* Calculate potentiometer sensitivity in mV/deg
* Observe encoder signal edge counts using non-quadrature, X2, and X4 decoding
* Calculate encoder PPR
* Calculate encoder angular resolution
* Screenshot of encoder signal response

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Angular Displacement using a Potentiometer

### 1.1 Theory and Background

#### What is a Potentiometer?

A rotary potentiometer, or pot, is a manually controlled variable resistor. As shown in Figure 1-1, it typically consists of an exposed shaft, three terminals (A, W, and B), an encased internal resistive element shaped in a circular pattern, and a sliding contact known as a wiper. By rotating the shaft, the internal wiper makes contact with the resistive element at different positions, causing a change in resistance when measured between the center terminal (W) and either of the side terminals (A or B). The total resistance of the potentiometer can be measured by clamping a multi-meter to terminals A and B.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg |

Figure 1-1: A typical rotary potentiometer

Two other common types of potentiometers are shown in Figure 1-2, which include slide pots as well as trimmer pots.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg  (a) trimmer pot | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Potentiometer - Concept Review.pdf - Adobe Acrobat Pro.jpg  (b) slide pot |

Figure 1-2: Other common types of potentiometers (source: DigiKey)

A schematic diagram of the voltage dividing characteristic of a potentiometer is illustrated in Figure 1-3. By applying a known voltage between terminals A and B (VAB), voltage is divided between terminals AW and WB such that:

Equation 1-1

When connected to an external shaft, a rotary potentiometer can measure absolute angular displacement. In this application, by applying a known voltage to the outside terminals of the pot, we can determine the position of the sensor based on the output voltage VAW or VWB which will be directly proportional to the position of the shaft. One of the advantages of using a potentiometer as an absolute position sensor is that after power loss, position information is retained since the resistance of the pot remains unchanged.

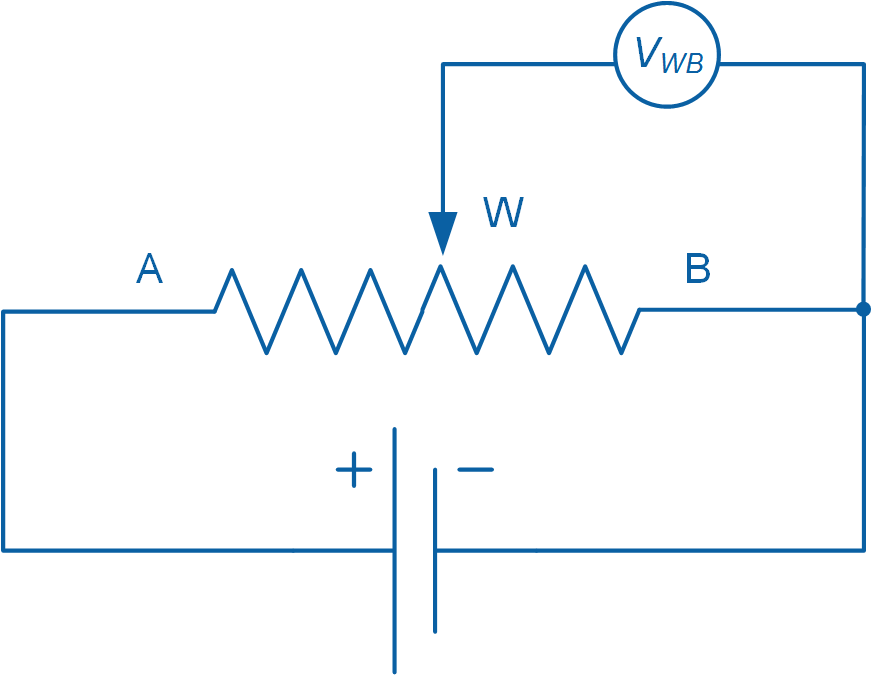


Figure 1-3: Schematic diagram of the voltage dividing characteristic of a rotary potentiometer

Depending on their construction, some pots have physical stops which prevent the shaft from a full 360° rotation. To counter this limitation, pots that do not have physical stops are used when continuous rotational movement is desired. Another limitation is the presence of a dead-band, which occurs when the pot shaft is turned but the resistance stays the same. In continuous rotation potentiometers, dead-band occurs at the limits of the pot and results in a discontinuity in the output when used for angular displacement measurement.

The typical lifespan of a pot is a few thousand rotations. This is because the wiper makes physical contact with the resistive element inside the pot, eventually wearing it out. This physical contact and the presence of dust also causes both mechanical and electrical noise. While the mechanical noise is rather inaudible in modern pots, electrical noise causes variations in the measured output. In audio applications where pots are used for volume and tone control, the electrical noise manifests itself in the form of audible pops and crackles.

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the potentiometer is shown in Figure 1-4.

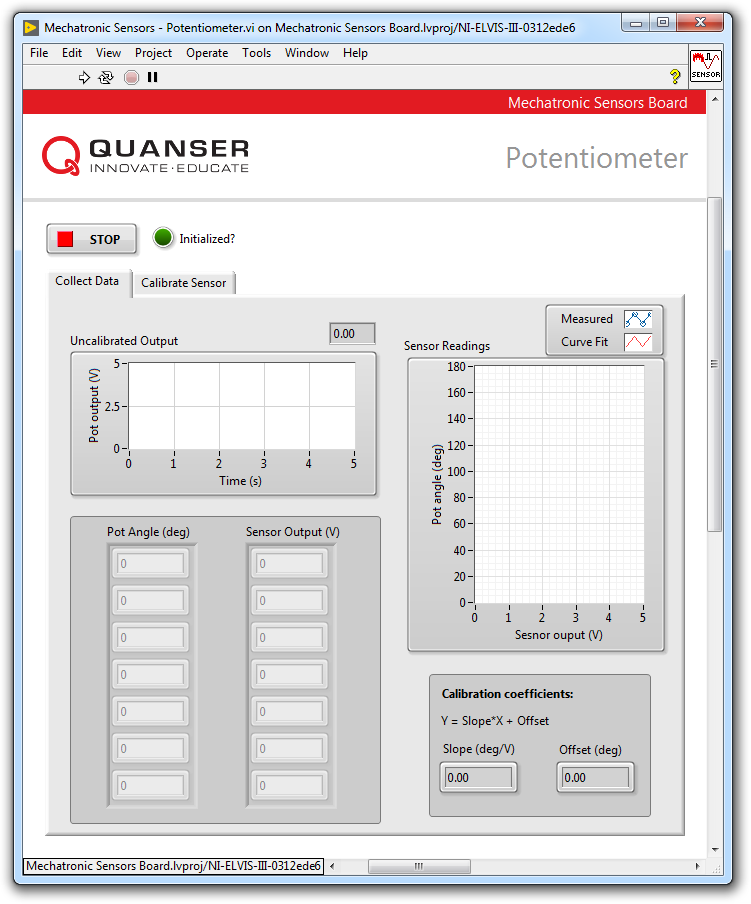


Figure 1-4: VI for collecting data from the potentiometer

#### Collect Data

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors - Potentiometer.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. Set the potentiometer knob to the 0° mark.
7. Enter 0 in the **Pot Angle (deg)** array.
8. Using the **Uncalibrated Output** waveform chart, read the corresponding sensor output and enter the value in the **Sensor Output (V)** array.
9. Continue taking measurements by rotating the potentiometer at 30° intervals. Enter the angular position and measured sensor outputs in the **Pot Angle (deg)** and **Sensor Output (V)** arrays respectively. Take a screenshot of your results.  
     
   Note: Once all of the measured data have been entered, a linear curve is automatically generated to fit the data. The curve is shown in the **Sensor Readings** waveform graph. This curve represents the calibration curve of the sensor.
10. The slope and offset of the calibration curve are automatically calculated by the VI and displayed in the **Slope (deg/V)** and **Offset (deg)** indicators. Make a note of these values.
11. Record the collected data in Table 1-1.
12. Take a screenshot of the **Sensor Readings** graph.
13. Continue to the next section.

Table 1-1: Recorded potentiometer measurements

|  |  |
| --- | --- |
| Angle (deg) | Output (V) |
| 0° |  |
| 30° |  |
| 60° |  |
| 90° |  |
| 120° |  |
| 150° |  |
| 180° |  |

#### Calibrate the Potentiometer

1. Click on the **Calibrate Sensor** tab to calibrate the output of the potentiometer in terms of angular position (in degrees).
2. Use the **Slope (deg/V)** and **Offset (deg)** numeric controls to enter the slope and offset values you obtained during the data collection step.
3. Test the accuracy of your calibration. To do this, set the potentiometer knob to different angles and verify that the correct angular position is displayed in the **Calibrated Output** waveform chart as well as the **Pot Angle (deg)** meter indicator.
4. Press the **Stop** button.

### 1.3 Analyze

1-1 Present the results you recorded in Table 1-1.

1-2 Attach a screenshot of the *Sensor Readings* waveform graph that shows the fitted calibration curve from step 12.

1-3 What was the calibration equation that you obtained?

1-4 What is the sensitivity of the sensor in mV/deg?

## Section 2: Measuring Angular Displacement using an Encoder

### 2.1 Theory and Background

#### What is an Encoder?

An incremental optical encoder, shown in Figure 2-1, is a relative angular displacement sensor which measures angular displacement relative to a previously known position. Unlike an absolute encoder, an incremental encoder does not retain its position information upon power loss. An incremental encoder outputs a series of pulses which correlate to the relative change in angular position. Encoders are commonly used to measure angular displacement of rotating load shafts. Information extracted from an incremental encoder can also be used to derive instantaneous rotational velocities.



Figure 2-1: An incremental rotary encoder manufactured by US Digital

An incremental optical encoder typically consists of a coded disk, an Infrared (IR) LED, and two photosensors. The disk is coded with an alternating light and dark radial pattern causing it to act as a shutter. As shown schematically in Figure 2-2, the light emitted by the IR LED is interrupted by the coding as the disk rotates around its axis.

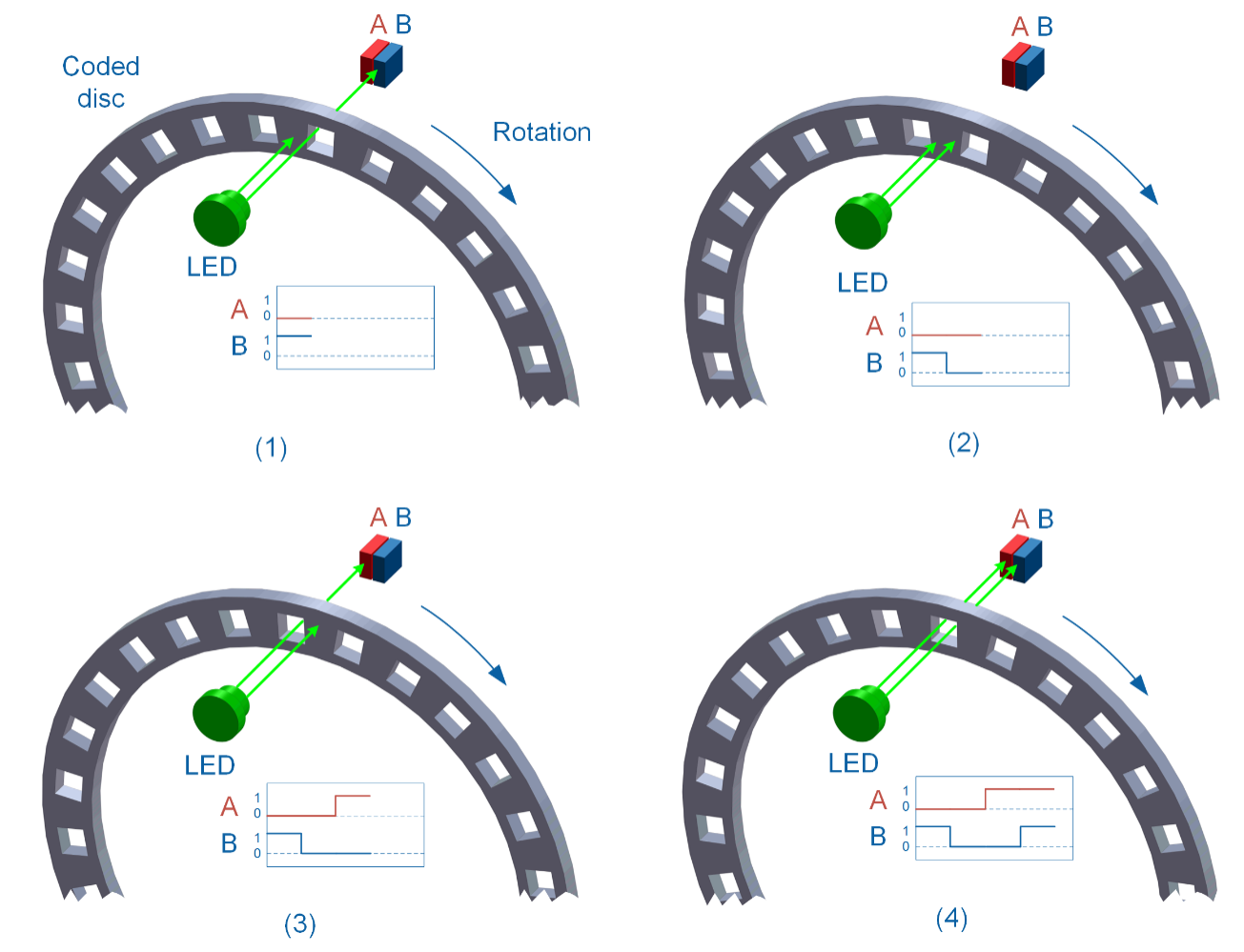


Figure 2-2: Output of an incremental encoder showing signals A and B when rotating in a clockwise manner

The two photosensors (A and B) positioned behind the coded disk sense the infrared light emitted by the IR LED, which results in A and B signals/pulses, in four distinct states as outlined in Table 2-1:

Table 2-1: Different quadrature states

|  |  |  |
| --- | --- | --- |
| State | Signal A | Signal B |
| 1 | OFF | ON |
| 2 | OFF | OFF |
| 3 | ON | OFF |
| 4 | ON | ON |

Encoders which output A and B signals are often referred to as quadrature encoders since the signals are separated in phase by 90° and result in four distinct states. Non-quadrature encoders have only one output signal and thus are unable to detect direction. The resolution of an encoder is determined using the number of light or dark patterns on the disk, a measure that is given in terms of *pulses per revolution*, or PPR.

Some encoders utilize an index pulse (Z channel), which is triggered once for every full rotation of the disk (see Figure 2-3). The index pulse can be used for calibration or so-called homing of a system, as well as a revolution counter. Depending on the encoder, the width of the index plus may be aligned with any of the four quadrature states. For example, the index pulse may have a width that spans a full cycle (4 states), a half cycle (2 states), or a quarter cycle (1 state). In the example shown in Figure 2-3, the width of the index pulse is aligned with a full cycle of the B signal.

There are two methods that an encoder registers the index pulse: (a) using pre-defined states of signals A and B, or (b) using user-defined states of signals A and B, in which case the user must select a combination of A and B states that occurs only once during the width of the index pulse.

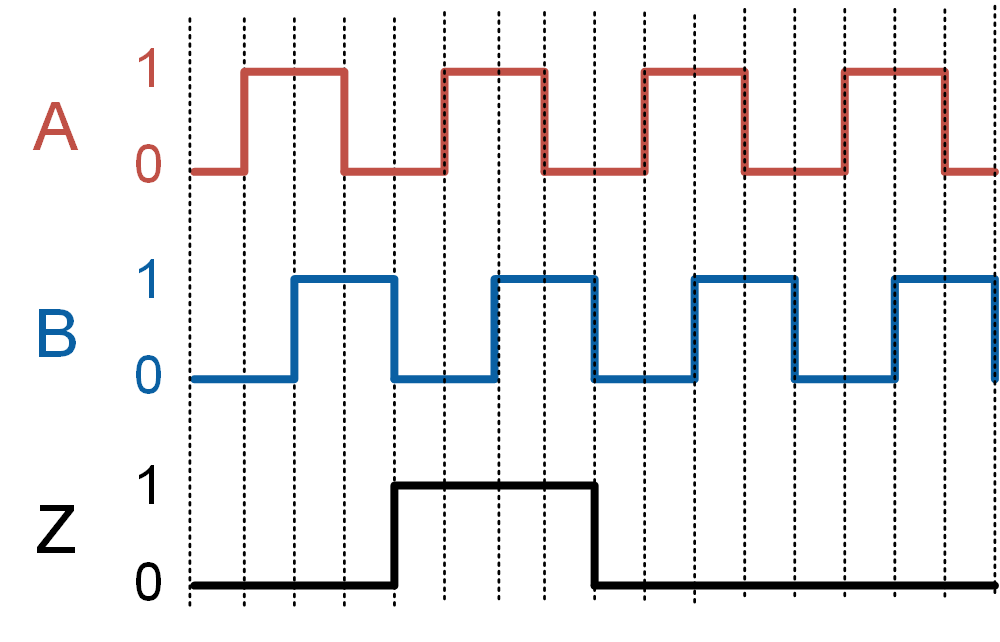


Figure 2-3: Output of a quadrature encoder with an index pulse

#### Encoder Decoding

In order to make encoder measurements, you need to connect the encoder outputs to a counter. A decoding algorithm is then used to determine the number of counts and possibly the direction of rotation.

Four common decoding algorithms are used: non-quadrature, X1, X2, and X4.

##### Non-quadrature

When a non-quadrature decoder is used, only the rising edge of signal A is counted as the shaft rotates. The counter is incremented on the rising edge of signal A. Because signal B is not used, the encoder cannot detect the direction of rotation. For example, using a non-quadrature decoder, a 9 PPR encoder will result in a total of 9 counts for every rotation of the encoder shaft. The count will increase regardless of which direction the shaft is rotated.

##### X1 Decoder

When an X1 decoder is used, only the rising edge of signal A is counted as the shaft rotates. When a rising edge of signal A occurs, the algorithm looks at the current state of signal B. If signal B is low, the counter is incremented. Otherwise, when signal B is high, the counter is decremented. Using an X1 decoder, a 9 PPR encoder will result in a total of 9 counts for every rotation of the encoder shaft.

##### X2 Decoder

When an X2 decoder is used, both the rising and falling edges of signal A are counted as the shaft rotates. When a rising edge of signal A occurs, the algorithm looks at the current state of signal B. If signal B is low, the counter is incremented. Otherwise, when signal B is high, the counter is decremented. When a falling edge of signal A occurs, if signal B is high the counter is incremented, otherwise when signal B is low the counter is decremented. Using an X2 decoder, a 9 PPR encoder will generate a total of 18 counts for every rotation of the encoder shaft.

##### X4 Decoder

When an X4 algorithm is used, both the rising and falling edges of both signals A and B are counted. Using a state machine diagram, Figure 2-4 illustrates how the counter is incremented or decremented depending on the state of signals A and B. An X4 decoder generates four times the number of counts generated by an X1 decoder resulting in the highest resolution among the three types of decoders. Using an X4 decoder, a 9 PPR encoder will generate a total of 36 counts for every rotation of the encoder shaft.

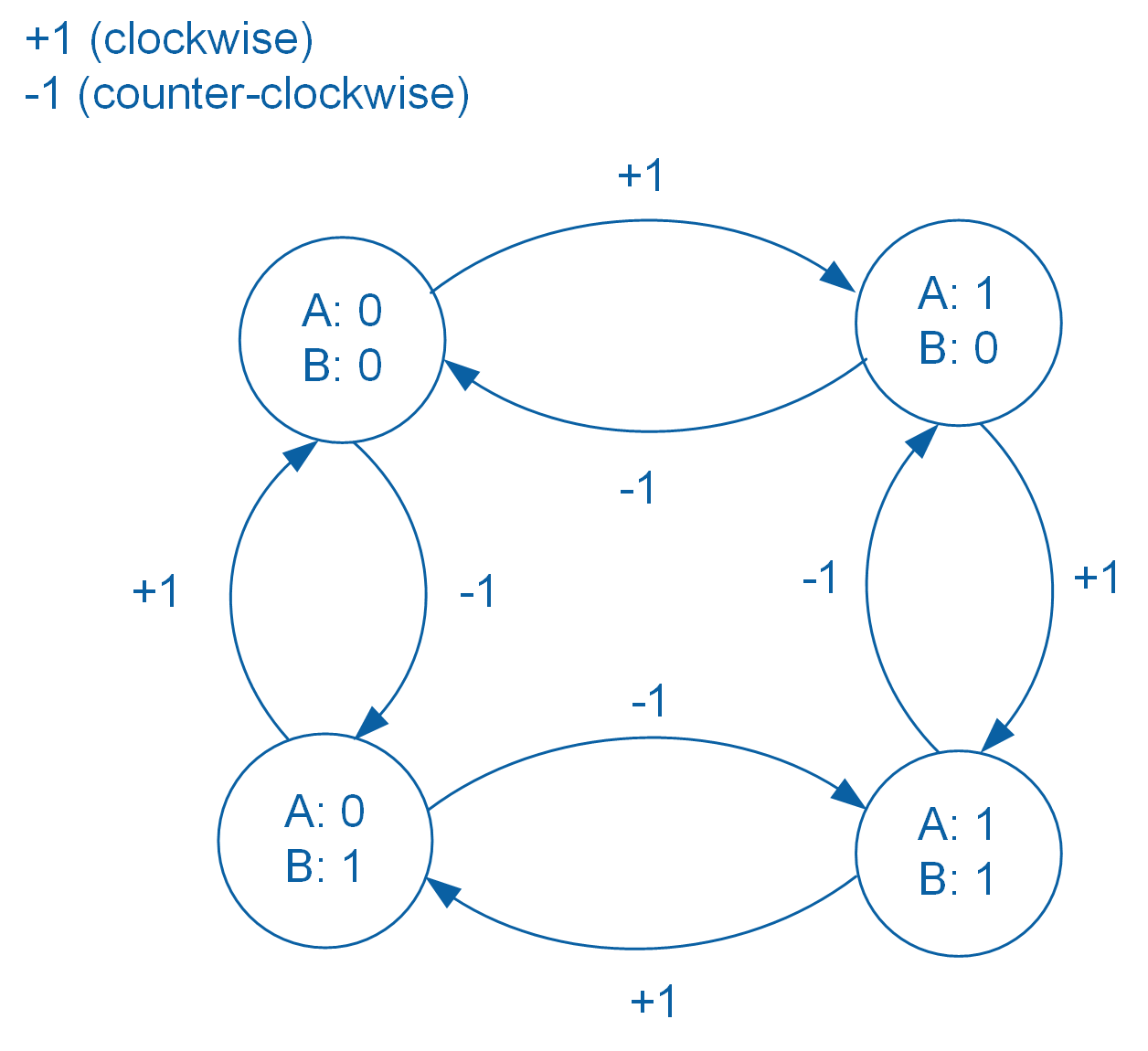


Figure 2-4: State machine representation of the X4 decoding algorithm

#### Measuring Angular Displacement

Pulses generated by an encoder can be converted to angular position (in degrees) using Equation 2-1:

Equation 2-1

where *Counts* is the number of acquired edge counts, *N* = 1, 2, or 4 corresponds to non-quadrature/X1, X2, and X4 decoders respectively, and PPR is the encoder’s PPR value.

The angular resolution of an encoder (not to be confused with encoder resolution, or PPR) depends on the encoder’s *PPR* and the decoding algorithm used, and can be calculated using Equation 2-2:

Equation 2-2

Figure 2-5 compares the number of counts generated by each of the non-quadrature, X1, X2, and X4 decoders.

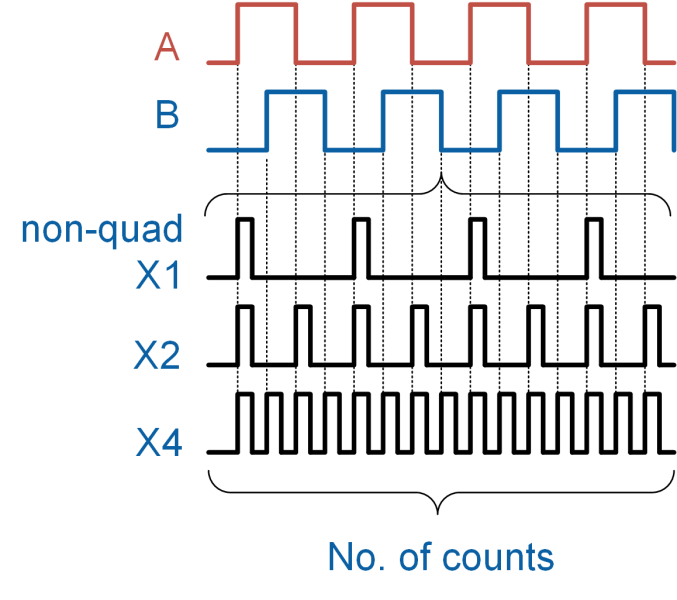


Figure 2-5: Comparison of the number of counts generated by different decoding algorithms

### 2.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the encoder is shown in Figure 2-6.

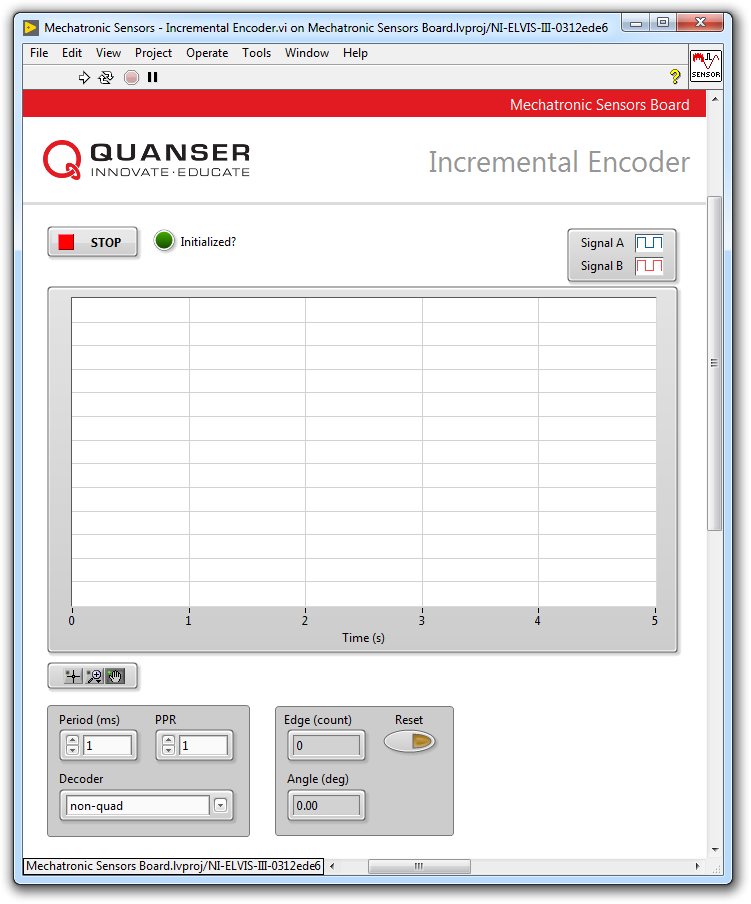


Figure 2-6: VI for collecting data from the encoder

#### Non-quadrature Decoding

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors - Incremental Encoder.vi**
3. From the **Decoder** drop-down menu, select **non-quad**.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. In non-quadrature decoding only signal A is used. Rotate the encoder knob in the clockwise direction. How does the **Edge (count)** numeric display change?
7. Rotate the knob in the counter-clockwise directions. How does the **Edge (count)** numeric display change?

*Note:* At any time you can press the *Reset* button to reset the counter. This will rest the **Edge (count)** and **Angle (deg)** numeric displays to zero.

1. Using the **Edge (count)** numeric display, determine the number of pulses the encoder generates per each full revolution (PPR).

*Note:* PPR is determined in non-quadrature mode. It refers to the total number of pulses generated by *Signal A* when the encoder makes one full revolution. The value of PPR will be used to calibrate the encoder pulses in terms of angular displacement in degrees.

1. Continue to the next section.

#### Calibrate the Encoder

1. Calibrate the pulses of the encoder in terms of angular displacement. To do this, enter the PPR value which was calculated in the previous section in the **PPR** numeric control and press the **Enter** key.
2. Verify the accuracy of your calibration. To do this, first press the **Reset** button then rotate the encoder knob and verify that the correct angular position is displayed in the **Angle (deg)** numeric indicator.
3. Continue to the next section.

#### X2 Decoding

1. From the **Decoder** drop-down menu, select **X2**.
2. Press the **Reset** button.
3. In X2 decoding both signals A and B are used. Rotate the encoder knob in the clockwise direction. How do the **Edge (count)** and **Angle (deg)** numeric displays change?  
     
   *Note:* An encoder will have a fixed PPR value regardless of the decoding algorithm that is used.
4. Rotate the knob in the counter-clockwise direction. How do the **Edge (count)** and **Angle (deg)** numeric displays change?
5. Examine the behavior of signal A and signal B.
6. What is the resolution of the measured angular displacement?
7. Continue to the next section.

#### X4 Decoding

1. From the **Decoder** drop-down menu, select **X4**.
2. Press the **Reset** button.
3. Rotate the knob in the clockwise and counter-clockwise directions. How do the **Edge (counts)** and **Angle (deg)** numeric displays change?
4. What is the resolution of the measured angular displacement?
5. Examine the behavior of signal A and signal B as you slowly rotate the encoder knob in the clockwise direction. In particular, compare the behavior of signals A and B and you rotate the encoder in the clockwise direction with the state machine diagram shown in Figure 2-4. Take a screenshot of your results.
6. Press the **Stop** button.

### 2.3 Analyze

2-1 How did the Edge (count) numeric display change when the knob was rotated in the clockwise direction in step 6?

2-2 How did the Edge (count) numeric display change when the knob was rotated in the counter-clockwise direction in step 7? Explain the observed behavior.

2-3 What is the PPR of the sensor that you calculated in step 8?

2-4 When using non-quadrature decoding, as the encoder knob was rotated in step11, did the Angle (deg) numeric indicator display the correct angular position?

2-5 When using X2 decoding, as you rotated the encoder knob in step 15, how did the Edge (count) and Angle (deg) numeric displays change?

2-6 As you rotated the encoder knob in the counter-clockwise directions in step 16, how did the Edge (count) and Angle (deg) numeric displays change? Explain the observed behavior.

2-7 What is the resolution of the measured angular displacement that you calculated in step 18? Does it match the expected angular resolution using Equation 2-2?

2-8 When using X4 decoding, as you rotated the encoder knob in the clockwise and counter-clockwise directions in step 22, how did the Edge (counts) and Angle (deg) numeric displays change?

2-9 What is the resolution of the measured angular displacement that you calculated in step 23? Does it match the expected angular resolution using Equation 2-2?

2-10 Compare the angular resolutions of the X2 and X4 decoding.

2-11 Using the screenshot you captured in step 24, compare the behavior of signals A and B and you rotate the encoder in the clockwise direction with the state machine diagram shown in Figure 2-4. Do your results match the state machine sequence?

## Section 3: Design Considerations

3-1 You are tasked to select an incremental encoder for measuring the position of a DC motor. You require a minimum resolution of 0.01 radians. Furthermore, your data acquisition system (DAQ) and software can only perform X2 decoding. What is the minimum PPR required to meet this design constraint?

3-2 As noted in the background section, a counter is used to count the pulses generated by an encoder. Counters are specified using a bit size. For example, a 24-bit counter is capable of counting up to 224 = 16,777,216 counts. When a counter exceeds its counting ability, it will wrap or overflow.

Assume that you have a 1,000 PPR encoder and a 24-bit counter at your disposal. You plan to use the encoder to monitor the rotational velocity of a shaft that rotates at a maximum speed of 500 rpm. Your DAQ performs X4 decoding. Calculate the minimum duration that the counter can monitor the shaft’s speed before it overflows.

3-3 Sampling data at an appropriate rate is required for making accurate measurements. Sample rate or sample frequency is the rate at which a DAQ reads or digitizes an input signal. The unit of sample rate is hertz, or samples per second. The sample rate is often defined by the user, although each DAQ will have limitations on how fast it can acquire data. One can think of DAQ sampling as snapshots, similar to the frames of a movie. The faster a DAQ samples a real signal, the greater the resolution and detail that can be seen in the digitized signal.

Figure 2-7 shows the effect of different sampling rates when acquiring a sinusoidal signal. As illustrated by the figure, sampling at different rates will result in different digitized waveforms. As sampling rate is increased, the resulting digitized sample more accurately resembles the true shape of the original signal.

Examine the effect of a low sampling rate on angular displacement measurement using the encoder. To do this, stop your VI using the **Stop** button. The default sampling period in the VI is set to 1 ms (equivalent to a sampling rate of 1,000 Hz). Change the sampling period to **20 ms** (equivalent to a sampling rate of 50 Hz). Select **X4** decoding and set the PPR numeric control to the value you determined earlier in this lab. Re-run the VI. Press the **Reset** button. Now slowly rotate the encoder shaft 360 degrees in the clockwise direction. The **Angle (deg)** numeric display should indicate an angular displacement of approximately 360 degrees. Reset the encoder again but this time, very quickly, rotate the encoder shaft 360 degrees in the clockwise direction. What value does the **Angle (deg)** numeric display indicate? Explain your results.

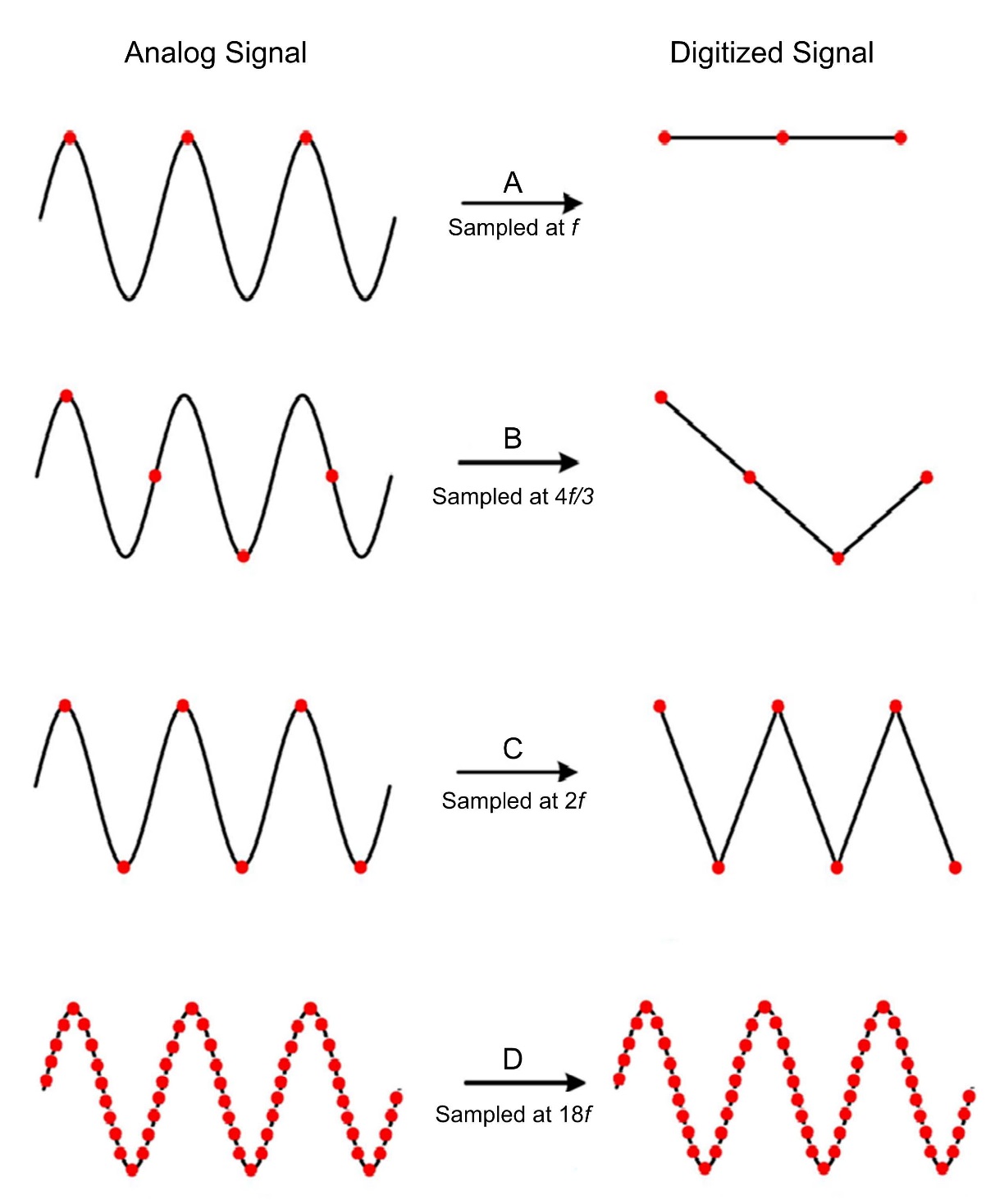


Figure 2-7: Effect of sample rate on an acquired sinusoidal signal

# Thí nghiệm 2: Khoảng cách và tiệm cận



Figure 0: Distance measurement is an integral part of mobile robotic applications

Bài thí nghiệm này khám phá phép đo khoảng cách tầm xa sử sụng cảm biến siêu âm (sonar), đo khoảng cách tầm trung đến tầm gần sử dụng cảm biến Time-of-Flight (ToF) và đo tiệm cận với cảm biến tiệm cận hồng ngoại (infrared proximity sensor). Hiệu chỉnh cảm biến, phân tán phép đo cũng như hệ số phản xạ mục tiêu sẽ được kiểm tra.

## Mục tiêu thí nghiệm

Sau khi hoàn thành bài thí nghiệm, anh/chị có thể nắm được cách:

* Hiệu chỉnh giá trị khoảng cách đầu ra của cảm biến sonar
* Đặc trưng hóa sự phân tán của cảm biến Time-of-Flight
* Xác định độ phân giải của cảm biến
* Đo khoảng cách của mục tiêu sử dụng cảm biến tiệm cận hồng ngoại
* Kiểm tra ảnh hưởng của hệ số phản xạ mục tiêu tại đầu ra cảm biến

## Công cụ và công nghệ yêu cầu

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * Xem User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * Xem User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Trước khi tải xuống và cài đặt, liên hệ giáo viên hướng dẫn hoặc quản lý phòng thí nghiệm để biết thêm thông tin về cơ sở hạ tầng và giấy phép phần mềm của phòng thí nghiệm. * Tải xuống và cài đặt NI ELVIS III   <http://www.ni.com/academic/download>   * Hướng dẫn:   http://www.ni.com/academic/students/learn-labview/ |
| Accessories:  * Bìa cứng 12x12 inch (trắng) * Bìa cứng 12x12 inch (đen) | |

## Yêu cầu sản phẩm

Trong bài thí nghiệm, anh/chị sẽ thu thập các sản phẩm sau:

* Ghi lại dữ liệu cảm biến sonar đã được hiệu chỉnh
* Ảnh chụp màn hình đồ thị hiệu chỉnh sóng siêu âm
* Ghi lại hệ số hiệu chỉnh cảm biến sonar
* Tính toán độ phân giải cảm biến sonar
* Tính toán độ nhạy cảm biến sonar
* Ghi lại dữ liệu phân tán cảm biến ToF
* Ảnh chụp màn hình hành vi phân tán ToF
* Tính toán độ lệch chuẩn của dữ liệu phân tán cảm biến ToF
* Ghi lại dữ liệu ngưỡng của cảm biến tiệm cận
* Vẽ đồ thị mối liên hệ giữ số xung của cảm biến tiệm cận với ngưỡng cảm biến tiệm cận
* Kiểm tra ảnh hưởng của hệ số phản xạ mục tiêu lên ngưỡng tiệm cận
* Kế hoạch thử nghiệm để so sánh các cảm biến sonar, ToF và tiệm cận để đo khoảng cách

Giáo viên hướng dẫn của anh/chị có thể yêu cầu nộp báo cáo thí nghiệm. Liên hệ giáo viên hướng dẫn để biết thêm thông tin về các yêu cầu hoặc mẫu báo cáo cụ thể.

## Phần 1: Đo khoảng cách tầm xa sử dụng sóng siêu âm

### 1.1 Cơ sở lý thuyết

#### Cảm biến Sonar là gì?

Sonar là loại cảm biến sử dụng sự lan truyền của sóng siêu âm để phát hiện các đối tượng. Hình 1-1 shows an cảm biến khoảng cách sonar tích hợp. Cảm biến bao gồm một bộ thu (transmitter), một bộ phát (receiver) và một mạch điều hòa tín hiệu.



Figure 1-1: Cảm biến sonar MaxBotix sử dụng trên Quanser Mechatronic Sensors Board

Như minh họa trong Hình 1-2, transmitter tạo ra một xung siêu âm hình nón, xung này sẽ phản xạ lại khi gặp vật thể và được thu bởi receiver. Mạch điều hòa tín hiệu đo thời gian từ khi sóng siêu âm lan truyền đến vật thể, dội ngược lại và được bắt bởi receiver. Vì sóng âm truyền với tốc độ cố định (343 m/s trong không khí tại 20 °C), tốc độ truyền đi và dội ngược lại là giống nhau, do đó khoảng cách *d* từ cảm biến tới vật thể có thể được tính như công thức 1-1:

Equation 1-1

Với *d* là khoảng cách giữa cảm biến và vật thể, *c* là vận tốc âm thanh, và *t* là khoảng thời gian xung siêu âm truyền từ transmitter về the receiver. Cảm biến sonar thường tạo nhiều xung đồng bộ mỗi giây.

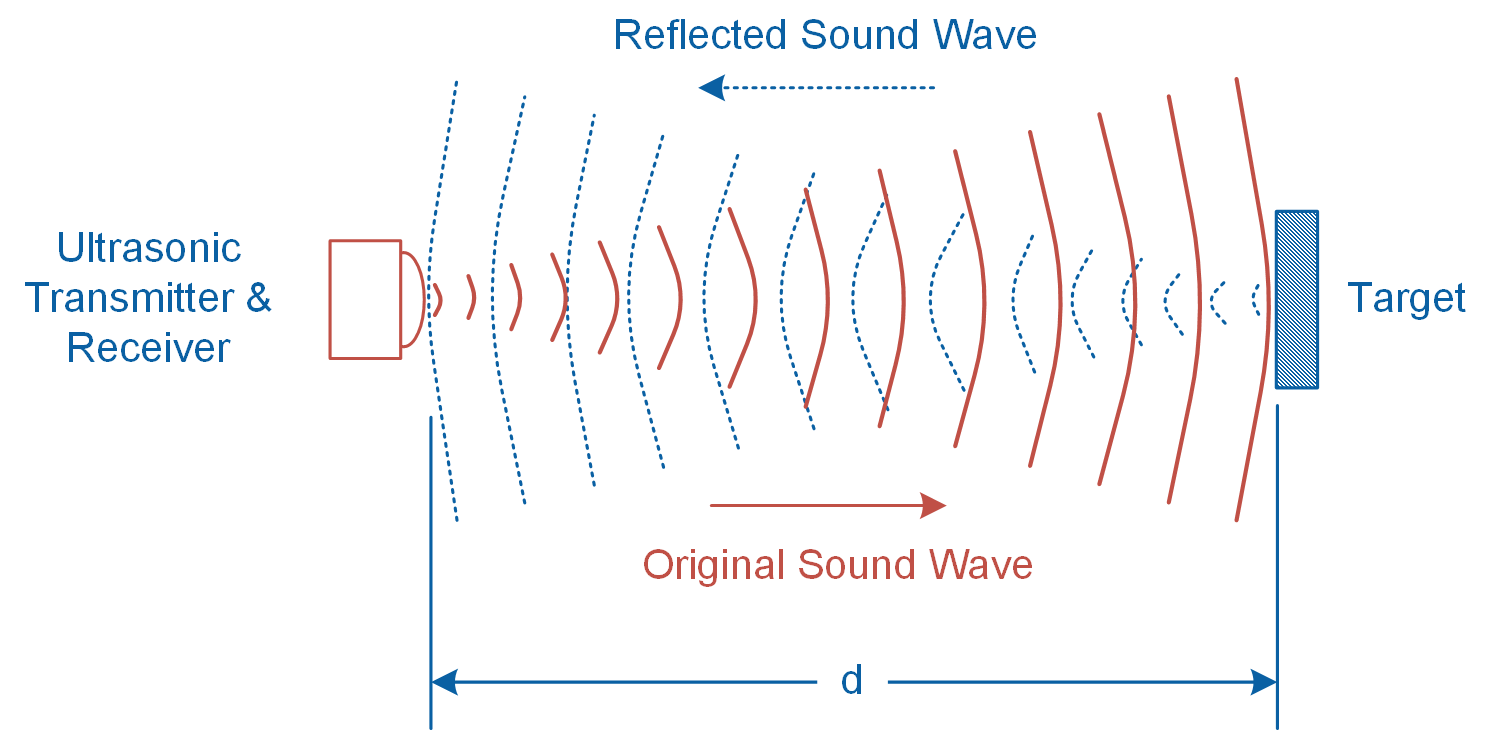


Figure 1-2: Sonar technique transmits and receives ultrasonic pings

Sonar là cảm biến đo khoảng cách lý tưởng để đo khoảng cách tầm xa cũng như để phát hiện các vật thể lớ. Cảm biến gắn trên bo mạch cảm biến cơ điện tử Quanser (Quanser Mechatronic Sensors Board) có thể đo vật thể trong khảng từ 6 đến 254 inches với độ phân giải ±1 inch. Trước khi được đo bởi hệ thống thu thập dữ liệu, tín hiệu tạo ra bởi transmitter được khuếch đại sử dụng bộ khuếch đại không đảo như minh họa trong Hình 1-3. Liên hệ giữa điện áp đầu vào ra đầu ra của bộ khuếch đại được tính toán sử dụng công thức 1-2:

Equation 1-2

Với *Vin* là tín hiệu tạo bởi cảm biến, *VO* là tín hiệu đã khuếch đại, điện trở *R1* = 10 k và *R2* = 10 k xác định hệ số khuếch đại theo công thức: *G* = 1 + *R2*/ *R1.*

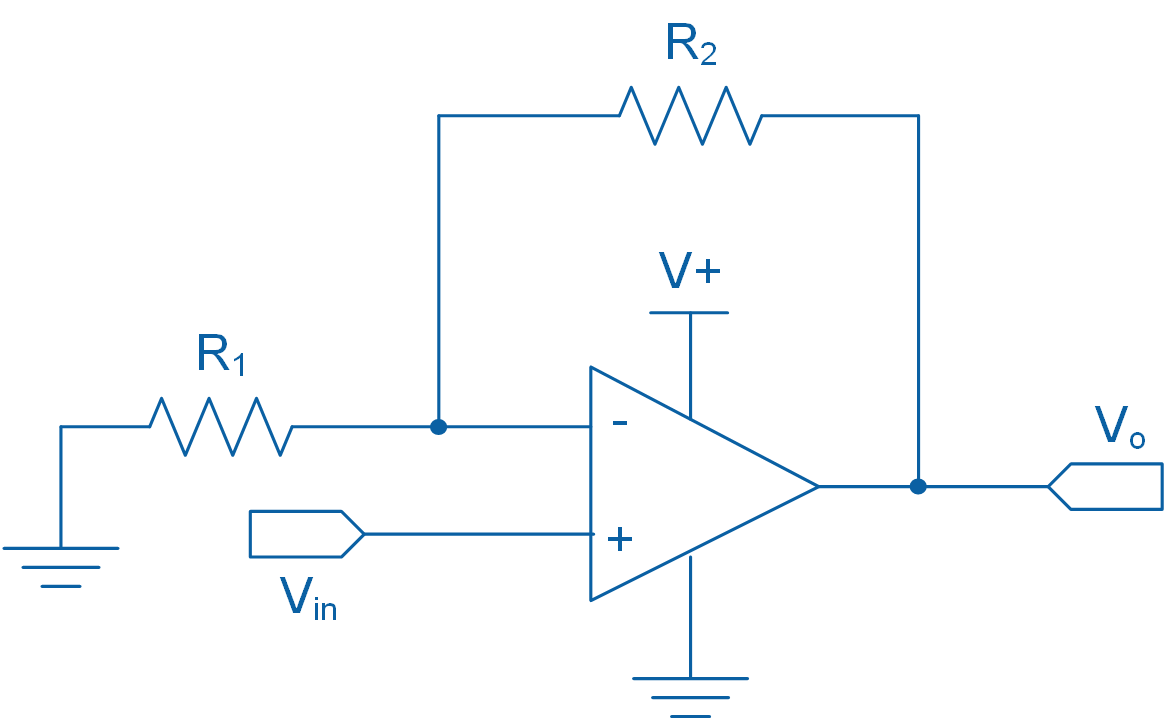


Figure 1-3: Non-inverting amplifier circuit diagram

Cảm biến sonar không thích hợp cho phép đo cự ly gần, độ phân giải của chúng tương đối lớn. Hơn nữa, chúng không thể phát hiện các vật mềm như quần áo, chăn và các vật liệu xốp. Vì kĩ thuật sonar dựa vào việc đo sóng phản xạ nên sonar không hoạt động trong không gian.

### 1.2 Thực hiện

Virtual Instrument (VI) được dùng để thu thập dữ liệu và hiệu chỉnh cảm biến (Hình 1-4).

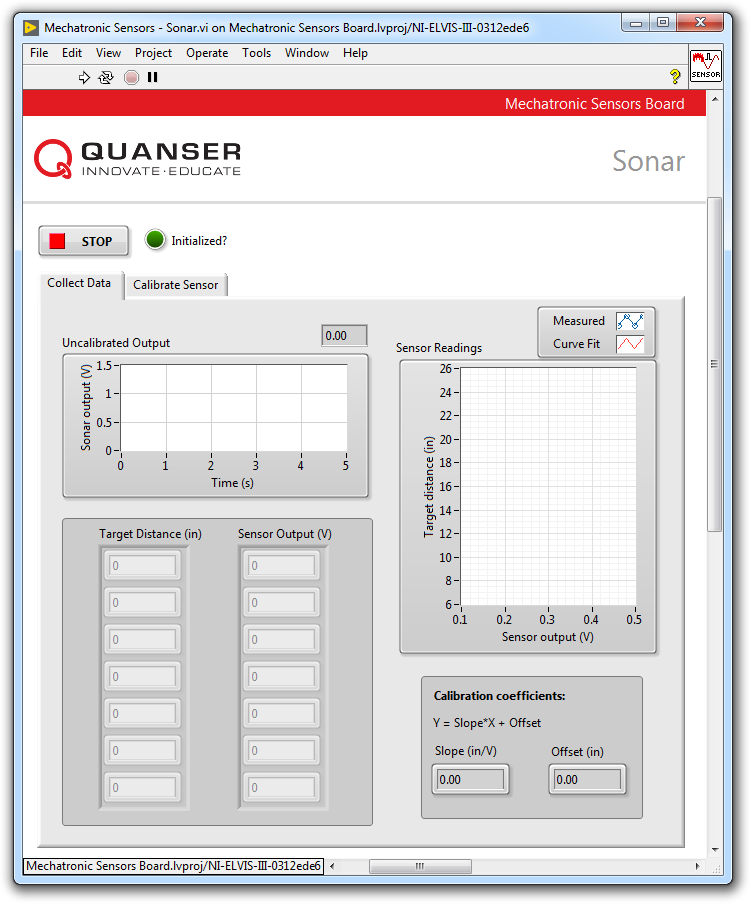


Figure 1-4 VI for collecting data from the sonar

#### Thu thập dữ liệu

1. Trước khi thực hiện bài thí nghiệm, sử dụng **Application Board Power** switch để bật nguồn Quanser Mechatronic Sensors Board. Cảm biến sonar MaxBotix thực hiện tự hiệu chỉnh khi bật nguồn để bù cho nhiệt độ và độ ẩm xung quanh. Đảm bảo rằng không có vật thể nào ở gần cảm biến trong khi bật nguồn. Độ nhạy tốt nhất đạt được khi vùng phát hiện của cảm biến không có bất kì vật thể nào trong vòng 14inch khi khởi động.
2. Mở **Mechatronic Sensors Board.lvproj**
3. Từ cửa sổ **Project Explorer**, chọn mở **Mechatronic Sensors - Sonar.vi**
4. Click vào tab **Collect Data**.
5. Chạy VI.
6. Đợi đèn báo LED **Initialized?** bật.
7. Nhập **7** vào mảng **Target Distance (in)**.
8. Giữ tấm bìa cứng 12x12in ở khoảng cách 7 inch so với bề mặt của bảng ứng dụng. Sử dụng đồ thị dạng sóng **Uncalibrated Output**, đọc đầu ra cảm biến tương ứng và nhập giá trị vào mảng **Sensor Output (V)**.  
     
   *Lưu ý:* Vì cảm biến tạo ra một xung hình nón rộng, không nên đứng quá gần cảm biến trong khi thực hiện phép đo. Nếu bộ phận cơ thể ở trong khoảng đo sẽ gây trở ngại cho phép đo. Khi đo mục tiêu ở khoảng cách xa hơn, anh/chị cần phải lùi lại và duỗi thẳng cánh tay của mình trong khi giữ mục tiêu để có được phép đo hợp lệ.
9. Tiếp tục thực hiện phép đo bằng cách di chuyển mục tiêu ra xa cảm biến một khoảng 3inch mỗi lần. mỗi phép đo, nhập khoảng cách mục tiêu và đầu ra cảm biến đo được lần lượt vào mảng **Target Distance (in)** và **Sensor Output (V)**.  
     
   *Lưu ý*: Sau khi nhập các số liệu đo được, một đường cong tuyến tính sẽ tự động được tạo, được hiển thị trong biểu đồ **Sensor Reading**. Đường này đại điện cho đường hiệu chỉnh của cảm biến.
10. Ghi lại dữ liệu đo được vào Bảng 1-1.
11. Độ dốc và độ lệch của đường chuẩn được VI tự động tính toán và hiển thị trong phần **Slope (in/V)** và **Offset (in)**. Ghi lại những giá trị này vào Bảng 1-2.
12. Chụp ảnh màn hình đồ thị **Sensor Reading**.
13. Tiếp tục thực hiện phần tiếp theo.

Table 1-1 Recorded sonar measurements

|  |  |
| --- | --- |
| Target Distance (in) | Sensor Output (V) |
| 7 |  |
| 10 |  |
| 13 |  |
| 16 |  |
| 19 |  |
| 22 |  |
| 25 |  |

Table1-2: Calibration coefficients

|  |  |
| --- | --- |
| Slope (in/V) | Offset (in) |
|  |  |

#### Hiệu chỉnh cảm biến Sonar

1. Click vào cửa sổ **Calibrate Sensor** để hiệu chỉnh đầu ra của cảm biến về độ dịch chuyển tuyến tính của mục tiêu (đơn vị inch).
2. Sử dụng điều khiển số (numeric controls) **Slope (in/V)** và **Offset (in)** để nhập giá trị độ dốc và độ lệch thu được trong quá trình thu thập dữ liệu.
3. Kiểm tra độ chính xác của hiệu chỉnh. Để kiểm tra, đặt mục tiêu ở các vị trí đã biết khác nhau trong phạm vi đã hiệu chuẩn và xác nhận rằng khoảng cách chính xác được hiển thị trong biểu đồ dạng sóng **Calibrated Output** cũng như chỉ báo thanh trượt **Distance (in).**
4. Sử dụng biều đồ dạng sóng **Calibrated Output**, ước tính độ phân giải của cảm biến (đơn vị inches). Bắt đầu với giữ mục tiêu cách cảm biến 7 inch. Với tốc độ ổn định, từ từ di chuyển mục tiêu ra xa cảm biến và quan sát đáp ứng của nó. Đầu ra được hiệu chỉnh của cảm biến sẽ có đáp ứng gần giống step response (step-like response). Thay đổi khoảng cách nhỏ nhất mà cảm biến có thể phát hiện là bao nhiêu? Chụp ảnh màn hình đáp ứng của cảm biếnSo sánh kết quả với độ phân giải của cảm biến được cho trong Phần 1.1.
5. Bấm nút **Stop**.

### 1.3 Phân tích

1-1 Trình bày kết quả anh/chị đo được trong Bảng 1-1.

1-2 Đính kèm ảnh chụp màn hình của đồ thị dạng sóng Sensor Readings hiển thị đường cong hiệu chỉnh phù hợp (fitted calibration curve) từ bước 12.

1-3 Trình bày hệ số cân bằng đo được trong Table 1-2.

1-5 Độ phân giải của cảm biến anh/chị tính được trong Bước 17? So sánh với độ phân giải được cho trong Phần 1.1. Đính kèm ảnh chụp màn hình step-like response của cảm biến.

1-6 Như lưu ý trong Phần 1.1, Bo mạch cảm biến cơ điện tử Quanser thực hiện mạch khuếch đại không đảo như trong Hình 1-3. Sử dụng công thức 1-2 và các hệ số hiệu chuẩn ghi trong bảng 1-2, hãy tính độ nhạy của cảm biến theo V/in? Giả sử R1 = R2 = 10k.

## Phần 2: Đo khoảng cách tầm ngắn đến tầm trung sử dụng cảm biến Time of Flight

### 2.1 Cơ sở lý thuyết

#### Cảm biến Time of Flight là gì?

Cảm biến Time-of-Flight (ToF), như hình Figure 2-1, sử dụng nguyên lý Time-of-Flight để đo khoảng cách tầm gần đến tầm trung. Time-of-Flight là một kĩ thuật đo được sử dụng để đo khoảng cách giữa cảm biến và mục tiêu, dựa trên chênh lệch thời gian giữa thời gian phát tín hiệu và khi tín hiệu quay lại do được phản xạ bởi mục tiêu. Các cảm biến ToF thường có footprint nhỏ (vài millimeters) và đo được khoảng cách từ 100 mm đến 2,000 mm. Hầu hết cảm biến ToF là cảm biến số (digital), điều này có nghĩa các giao thức SPI hoặc I2C sẽ được sử dụng để giao tiếp với cảm biến ToF.



Figure 2-1: A ToF sensor manufactured by RF Digital Corporation

Như minh họa trong Hình 2-2, cảm biến ToF bao gồm một LED hồng ngoại hoặc nguồn laser công suất thấp. trong khoảng thời gian đều đặn, cảm biến phát xung một nguồn sáng. Mỗi khi nguồn sáng bật lại từ một vật thể, bộ thu (receiver) tích hợp sẽ đo "thời gian bay" hoặc khoảng thời gian chùm tia truyền đến mục tiêu và bật trở lại receiver. Khoảng cách đến mục tiêu có thể tính theo Equation 2-1:

Equation 2-1

với *d* là khoảng cách đến mục tiêu tính theo mét, *td* là thời gian để receiver phát hiện được chùm tia phản xạ tính bằng giây, và *c* là tốc độ ánh sáng tính bằng m/s. Ví dụ, giả sử c = 300.000.000m/s, đối tượng nằm cách cảm biến 2 mét, cần khoảng 13 nano giây để chùm xung truyền từ nguồn đến receiver.

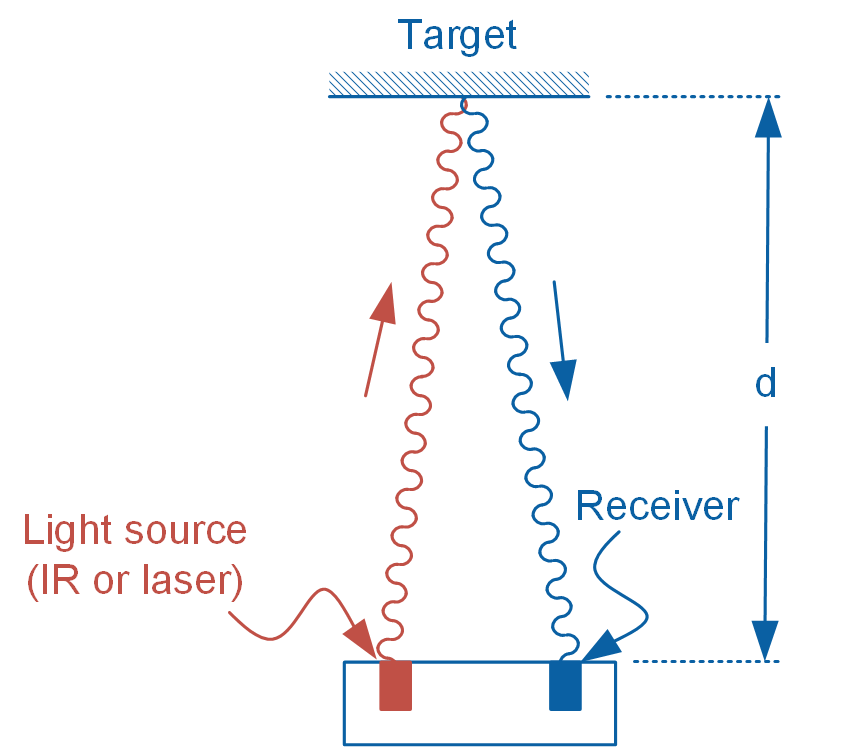


Figure 2-2: The principle of ToF

Vì công nghệ ToF sử dụng ánh sáng thay vì xung âm thanh để phát hiện đối tượng, không như cảm biến sonar, ToF có thể hoạt động trong chân không. Cảm biến ToF được sử dụng rộng tãi trong ngành công nghiệp robot và hàng điện tử tiêu dùng. Trong ứng dụng về robot, cảm biến ToF được sử dụng để phát hiện và tránh vật cản. Trong khi đó ứng dụng của ToF trong hàng tiêu dùng bao gồm phát hiện bàn tay ở các máy xà phòng và vòi nước tự đông. Các ứng dụng nâng cao hơn bao gồm nhận dạng cử chỉ, phát hiện chuyển động theo hướng và điều khiển âm lượng hoặc độ cao.

Khi thiết kế hệ thống đo lường ToF, điều quan trọng là phải xem xét các điều kiện môi trường có thể ảnh hưởng đến phép đo như thế nào. Ví dụ: vì cảm biến ToF thường chứa nguồn sáng hồng ngoại, sự hiện diện của các nguồn sáng khác có chứa thành phần hồng ngoại phong phú, chẳng hạn như bóng đèn halogen, có thể khiến cảm biến xuất ra các giá trị sai.

### 2.2 Thực hiện

Phần mềm Virtual Instrument (VI) được sử dụng để thu thập dữ liệu từ cảm biến ToF (Hình 2-3).

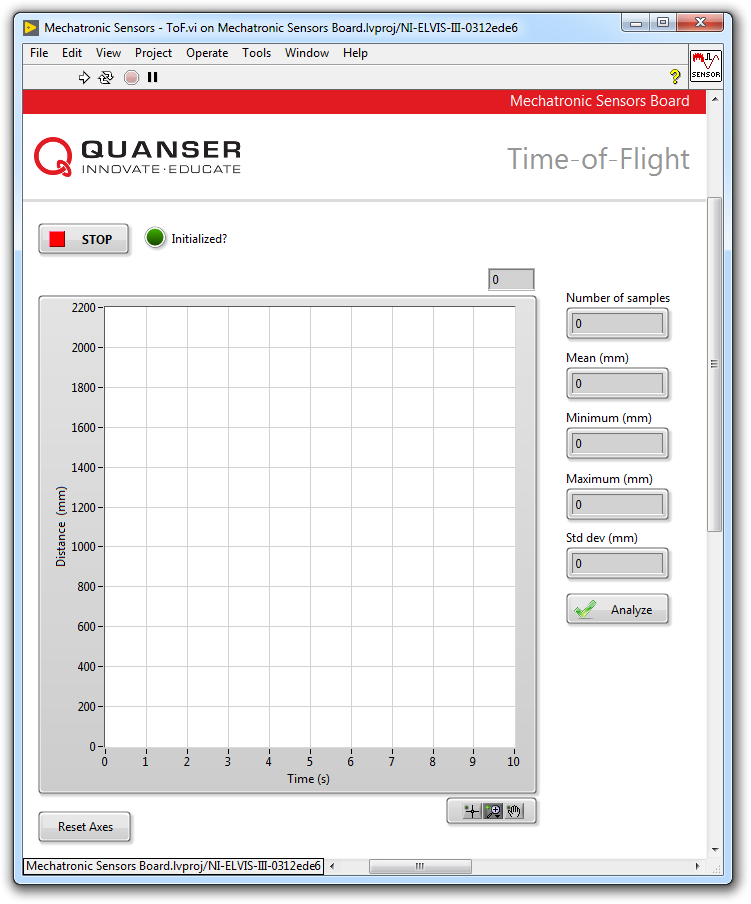


Figure 2-3: VI for collecting data from the ToF sensor

#### Observe Measurement Scatter

1. Mở file **Mechatronic Sensors Board.lvproj**
2. Từ của sổ **Project Explorer**, chọn file **Mechatronic Sensors - ToF.vi**
3. Chạy VI.
4. Đợi đèn báo LED **Initialized?** bật.

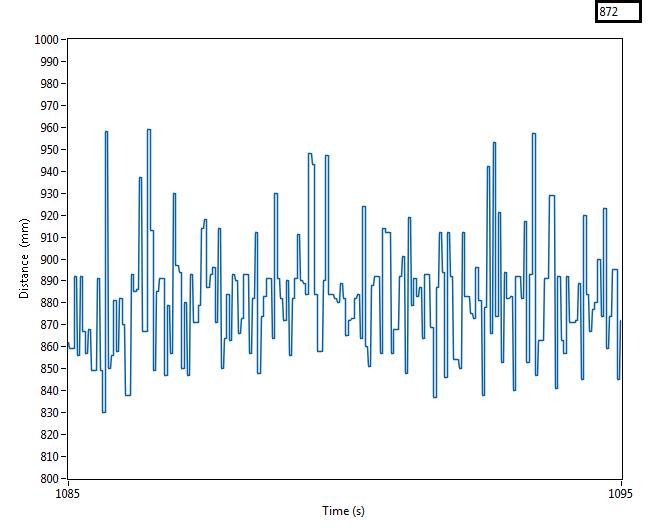
Cảm biến ToF số của Quanser Mechatronic Sensors Board xuất ra một giá trị 11-bit trong khoảng từ 0 and 2048, tương ứng trực tiếp với khoảng cách đến mục tiêu tính bằng millimeters. Do đó, không cần phải hiệu chỉnh khoảng cách đo được của cảm biến.  
  
Tuy nhiên, đầu ra của cảm biến ToF thể hiện sự thay đổi ngẫu nhiên, hoặc phân tán, như trong Hình 2-4. Quan sát sự phân tán này bằng cách giữ một mục tiêu bằng bìa cứng có kích thước 12x12 inch ở các vị trí ngẫu nhiên khác nhau trong khoảng từ 100 mm đến 1000 mm. Chụp ảnh màn hình các quan sát. Mức độ phân tán có thay đổi theo khoảng cách mục tiêu không?  


Figure 2-4 Measurement scatter associated with the output of the ToF sensor

#### Quantify Measurement Scatter

1. Định lượng mức độ phân tán của phép đo. Để thực hiện, giữ cố định mục tiêu ở khoảng cách xấp xỉ 100 mm từ cảm biến ToF. Chờ kết quả ổn định trong ít nhất **3s,** sau đó chọn nút **Analyze**. VI được lập trình để thu thập 300 điểm dữ liệu với tốc độ 100 Hz. Khi nút *Analyze* được bấm, VI sẽ tính toán và hiển thị giá trị trung bình, tối thiểu, tối đa và độ lệch chuẩn của dữ liệu thu được. Ghi lại các giá trị trong lần thử đầu tiên vào Bảng 2-1. Lặp lại phép đo hai lần nữa và ghi dữ liệu vào Bảng 2-1.
2. Lặp lại bước trước nhưng giữ cố định mục tiêu ở khoảng cách xấp xỉ 1000 mm. Ghi kết quả vào Table 2-1.
3. Bấm **Stop**.

Table 2-1: Scatter data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Approx. distance (mm) | Trial | Mean  (mm) | Max  (mm) | Min  (mm) | Std Dev  (mm) |
| 100 | 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 1000 | 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

### 2.3 Phân tích

2-1 Xem chi tiết các quan sát về đầu ra của cảm biến ToF khi mục tiêu được đặt ở các khoảng cách khác nhau (Bước 5). Các kết quả có xu hướng gì khi mục tiêu được đặt xa cảm biến hơn? Đính kèm ảnh chụp các kết quả của anh/chị.

2-2 Trình bày kết quả đo được trong Table 2-1.

2-3 Độ lệch chuẩn là một thước đo thống kê về lượng biến thiên, hoặc sự phân tán, trong một tập hợp các điểm dữ liệu được đo. Độ lệch chuẩn lớn hơn có nghĩa là sự phân tán lớn hơn trong dữ liệu thu được. Trong bài thí nghiệm này, vì các phép đo cho một khoảng cách cho trước được lặp lại nhiều lần, anh/chị phải áp dụng thống kê tổng hợp (pooled statistics) để cung cấp một ước tính thống kê tốt nhất về dữ liệu được đo.

Đối với mỗi khoảng các, hãy tính toán độ lệch chuẩn tổng hợp (*pooled standard deviation*) bằng cách sử dụng dữ liệu được ghi trong Bảng 2-1 và công thức bên dưới:

với Spooled là độ lệch chuẩn tổng hợp cho khoảng cách cho trước, N là số lần đo, và Si là độ lệch chuẩn mỗi lần đo. So sánh độ lệch chuẩn tổng hợp giữa 2 khoảng cách khác nhau. Kết quả của anh/chị cho biết gì về sự phân tán khi mục tiêu được đặt xa cảm biến hơn?

## Phần 3: Đo tiệm cận sử dụng cảm biến tiệm cận hồng ngoại

### 3.1 Cơ sở lý thuyết

#### Cảm biến tiệm cận hồng ngoại là gì?

Một cảm biến quang họ phản xạ điển hình được minh họa trong Hình 3-1. Nó bao gồm một điốt phát tia hồng ngoại (IRED) và một photodiode (detector). Điốt phát quang và detector được gắn cạnh nhau trên các trục song song.



Figure 3-1: A digital proximity sensor manufactured by Broadcom Limited

Khi một bề mặt được đặt ở vị trí gần của cảm biến, bề mặt đó sẽ phản xạ ánh sáng hồng ngoại do IRED phát ra vào photodiode. Bề mặt được đặt càng xa cảm biến, thì càng ít ánh sáng phản xạ vào photodiode. Cảm biến sẽ xuất ra tín hiệu tương tự tỷ lệ với khoảng cách đo được hoặc sẽ xuất ra tín hiệu kỹ thuật số tương đương với khoảng cách đã đo.



Figure 3-2: Schematic diagram of a digital proximity sensor

Hình 3-2 minh họa sơ đồ hoạt động của cảm biến tiệm cận kỹ thuật số. Khi được cấp nguồn, IRED phát ra ánh sáng hồng ngoại dạng xung, một phần của ánh sáng này được phản xạ bởi mục tiêu phản xạ vào điốt quang. Photodiode (diode quang) là một chất bán dẫn chuyển đổi các photon ánh sáng thành dòng điện. Sau đó, dòng điện cảm ứng được chuyển đổi thành tín hiệu kỹ thuật số (thường là số đếm) bằng bộ chuyển đổi Analog-to-Digital (ADC) tích hợp sẵn. Trong ví dụ minh họa, đầu ra của ADC được đọc bằng giao thức I2C.

Mechatronic Sensors Board sử dụng cảm biến tiệm cận hồng ngoại kĩ thuật số Broadcom APDS-9190. Người dùng có thể cài đặt số xung IRED phát ra (từ 1 đến 255) mỗi chu kì hoạt động. Mỗi xung có chu kì 16.3 s. Số xung cao sẽ tăng độ nhạy của cảm biến. Cảm biến xuất ra giá trị đếm trong khoảng từ 0 đến 1023. Giá trị này phụ thuộc vào khoảng cách của mục tiêu và là thước đo lượng ánh sáng IR phản xạ. Giá trị đếm là 1023 có nghĩa là mục tiêu đã đạt đến ngưỡng phát hiện gần của cảm biến. Khi đầu ra bão hòa ở 1023 số đếm, nó sẽ không tăng ngay cả khi mục tiêu di chuyển gần cảm biến hơn.

Cảm biến tiệm cận thường được sử dụng trong các ứng dụng trường gần. Ví dụ, trong điện thoại di động, cảm biến có thể phát hiện khi người dùng đặt điện thoại gần tai.

### 3.2 Thực hiện

Virtual Instrument (VI) được sử dụng để thu thập dữ liệu từ cảm biến tiệm cận IR (Hình 3-3).

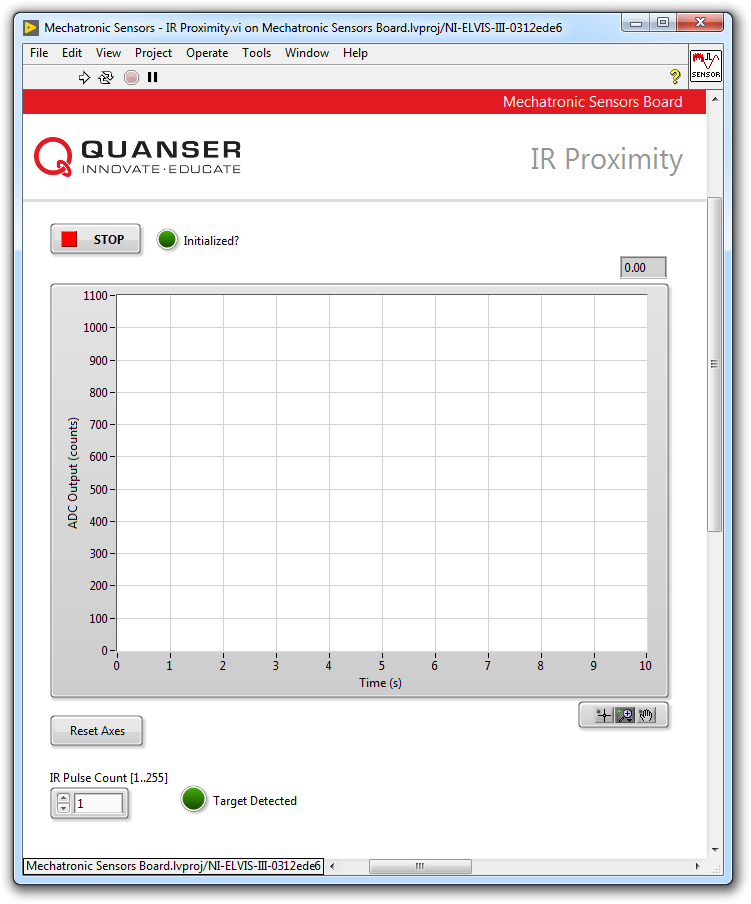


Figure 3-3: VI for collecting data from the IR proximity sensor

#### Observe Sensor Behavior

1. Mở file **Mechatronic Sensors Board.lvproj**
2. Từ của sổ **Project Explorer**, chọn **Mechatronic Sensors – IR Proximity.vi**
3. Chạy VI.
4. Đợi đèn báo LED **Initialized?** bật.
5. Biểu đồ dạng sóng cho thấy số đếm dạng raw (raw count) của đầu ra ADC của cảm biến.
6. Đảm bảo điều khiển số **IR Pulse Count [1…255]** được đặt về **1**. Cài đặt này khiến cảm biến tạo ra 1 xung IRED trong mỗi chu kỳ hoạt động.
7. Giữ tay của anh/chị ở khoảng cách khoảng 30 cm với cảm biến. Từ từ di chuyển tay về phía cảm biến và quan sát đáp ứng của cảm biến; cụ thể là quan sát cách số đếm (count) đầu ra tăng lên cũng như bất kỳ sự phân tán nào trong đầu ra của cảm biến. Khi bàn tay đạt đến ngưỡng gần, số đếm đầu ra sẽ đạt giá trị tối đa là 1023. Sử dụng thước đo, ước tính khoảng cách ngưỡng theo milimét và ghi lại giá trị trong Bảng 3-1. Chụp ảnh màn hình kết quả.

Table 3-1: Sample recorded data

|  |  |
| --- | --- |
| IR Pulse Count | Proximity Threshold (mm) |
| 1 |  |
| 10 |  |
| 50 |  |
| 100 |  |
| 150 |  |
| 255 |  |

1. Dừng VI bằng cách bấm nút **Stop**.
2. Như lưu ý trong mục 3.1, số lượng xung cao hơn dẫn đến độ nhạy cảm biến lớn hơn cũng như giới hạn ngưỡng tiệm cận lớn hơn. Để quan sát, đặt số **IR Pulse Count [1...255]** thành **10** và chạy lại VI, lặp lại bước 7. Ghi kết quả vào Bảng 3-1.
3. Lặp lại bước 8 và 9 cho các giá trị còn lại của **IR Pulse Count** trong Bảng 3-1.  
     
   *Lưu ý:* Trước khi nhập giá trị đếm xung mới, phải dừng VI. Sau đó nhập giá trị mới và chạy lại VI.
4. Bấm nút **Stop**.

### 3.3 Phân tích

3-1 Trình bày kết quả anh/chị đã ghi trong Bảng 3-1.

3-2 Xem chi tiết các quan sát của anh/chị về đầu ra của cảm biến tiệm cận IR khi bạn di chuyển tay đến gần cảm biến hơn để tìm các giá trị đếm xung khác nhau. Đính kèm ảnh chụp màn hình kết quả.

3-3 Nhận xét về mức độ phân tán mà anh/chị quan sát được đối với các giá trị đếm xung IR khác nhau. Đính kèm ảnh chụp màn hình kết quả.

3-4 Sử dụng dữ liệu được ghi lại trong Bảng 3-1, vẽ một đường đồ thị liên quan đến số lượng xung IRED của cảm biến với ngưỡng tiệm cận của nó. Loại đồ thị nào phù hợp nhất với dữ liệu? Sử dụng đồ thị này ước tính số lượng xung cần thiết để dẫn đến ngưỡng tiệm cận là 70 mm. Chạy VI và xác nhận kết quả của anh/chị.

## Section 4: Cân nhắc thiết kế

4-1 Anh/chị được yêu cầu thiết kế một bộ khuếch đại không đảo cho cảm biến sonar tương tự như thiết kế trong Hình 1-3. Cảm biến có độ nhạy 0.01 V/in và có thể đo khoảng cách tối đa là 254 in. Bạn có thể tùy ý sử dụng DAQ có khả năng đo tín hiệu tương tự từ 0 V đến 10 V. Xác định hệ số khuếch đại tối đa cho phép của mạch trong khi đáp ứng các ràng buộc thiết kế. Đề xuất các giá trị thích hợp cho các điện trở khuếch đại R1 và R2.

4-2 Trong trường họp lý tưởng, cảm biến tiệm cận phải mang lại kết quả có thể tái tạo lại bất kể màu sắc mục tiêu, kết cấu bề mặt và hệ số phản xạ bề mặt. Tuy nhiên, trên thực tế, cảm biến tiệm cận IR phụ thuộc nhiều vào khả năng phản xạ ánh sáng IR của mục tiêu. Kiểm tra và nhận xét về hoạt động của cảm biến tiệm cận IR mà bạn đã sử dụng trong Phần 2 bằng cách sử dụng các mục tiêu có hệ số phản xạ khác nhau. Đặc biệt, kiểm tra ảnh hưởng của hệ số phản xạ bề mặt đối với ngưỡng tiệm cận. Sử dụng tông màu trắng và tông màu đen lần lượt là phản xạ IR tốt và kém. Trong quá trình thử nghiệm, hãy đặt **IR Pulse Count [1..255]** thành **255**.

4-3 Lập kế hoạch thử nghiệm để so sánh hiệu suất và tính phù hợp của cảm biến sonar, ToF và cảm biến tiệm cận làm cảm biến khoảng cách. Một số thông số chính cần kiểm tra cho từng cảm biến bao gồm:

* Tầm đo
* Độ phân giải
* Độ nhạy
* Loại giao tiếp (digital hay analog)
* Khả năng tương thích mục tiêu
  + Hoàn thiện bề mặt
  + Hình dáng
  + Màu sắc
* Điều kiện môi trường hoạt động
* Độ tuyến tính

# Lab 3: Nhiệt độ



Figure 0: Many industries rely on the accurate measurement of temperature

Bài thí nghiệm này khám phá các đặc tính của một điện trở nhiệt. Hiệu chuẩn hai điểm sẽ được áp dụng cho cảm biến và hằng số thời gian nhiệt của nó sẽ được xác định bằng cách sử dụng đầu vào đơn vị (step input).

## Mục tiêu thí nghiệm

Sau khi hoàn thành bài thí nghiệm, anh/chị có thể nắm được cách:

* Áp dụng hiệu chuẩn 2 điểm cho thermistor
* Xác định hằng số βcủa một thermistor
* Áp dụng bộ lọc xung thấp cho đầu ra thermistor
* Xác định hằng số thời gian nhiệt của thermistor bằng đầu vào đơn vị (step input)

## Công cụ và công nghệ yêu cầu

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * Xem User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * Xem User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 trở lên  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Trước khi tải xuống và cài đặt, liên hệ giáo viên hướng dẫn hoặc quản lý phòng thí nghiệm để biết thêm thông tin về cơ sở hạ tầng và giấy phép phần mềm của phòng thí nghiệm. * Tải xuống và cài đặt NI ELVIS III   <http://www.ni.com/academic/download>   * Xem hướng dẫn   http://www.ni.com/academic/students/learn-labview/ |
| Accessories:  * Điện trở nhiệt (thermistor) * Mảng băng keo nhỏ |  |

## Yêu cầu sản phẩm

Trong bài thí nghiệm này, anh/chị sẽ thu thập các sản phẩm sau:

* Nhiệt độ đầu ngón tay sử dụng nhiệt kế
* Đầu ra thermistor ở trạng thái ổn định khi đo nhiệt độ đầu ngón tay
* Tính điện trở của thermistor khi đo nhiệt độ ngón tay
* Xác định tham số βcủa thermistor
* Xác định hằng số thời gian nhiệt của thermistor trong các điều kiện khác nhau

Giáo viên hướng dẫn của anh/chị có thể yêu cầu nộp báo cáo thí nghiệm. Liên hệ giáo viên hướng dẫn để biết thêm thông tin về các yêu cầu hoặc mẫu báo cáo cụ thể.

## Phần 1: Đo nhiệt độ sử dụng thermistor

### 1.1 Cơ sở lý thuyết

#### Thermistor là gì?

Có nhiều loại bộ chuyển đổi (transducer) khác nhau để đo nhiệt độ: cặp nhiệt điện, cảm biến nhiệt độ (RTD), điện trở nhiệt (thermistor) và loại mạch tích hợp (integrated circuit). Mỗi loại đều có những ưu và nhược điểm riêng. Cảm biến loại mạch tích hợp có chi phí thấp và có đầu ra tuyến tính, nhưng vì chúng được gắn trên PCB nên chúng phụ thuộc vào bố cục bo mạch. Cặp nhiệt điện (thermocouple) có dải nhiệt độ rộng, tương đối rẻ, dễ sử dụng nhưng kém ổn định và nhạy cảm nhất. Mặt khác, RTD ổn định và chính xác nhất trong số các cảm biến nhưng chậm, cấu trúc mỏng manh và tương đối đắt tiền nhất. Một thermistor phản hồi rất nhanh và có độ nhạy cao nhất nhưng có một phạm vi nhiệt độ hạn chế. Hình 1-1 so sánh độ nhạy điển hình của ba cảm biến nhiệt độ.

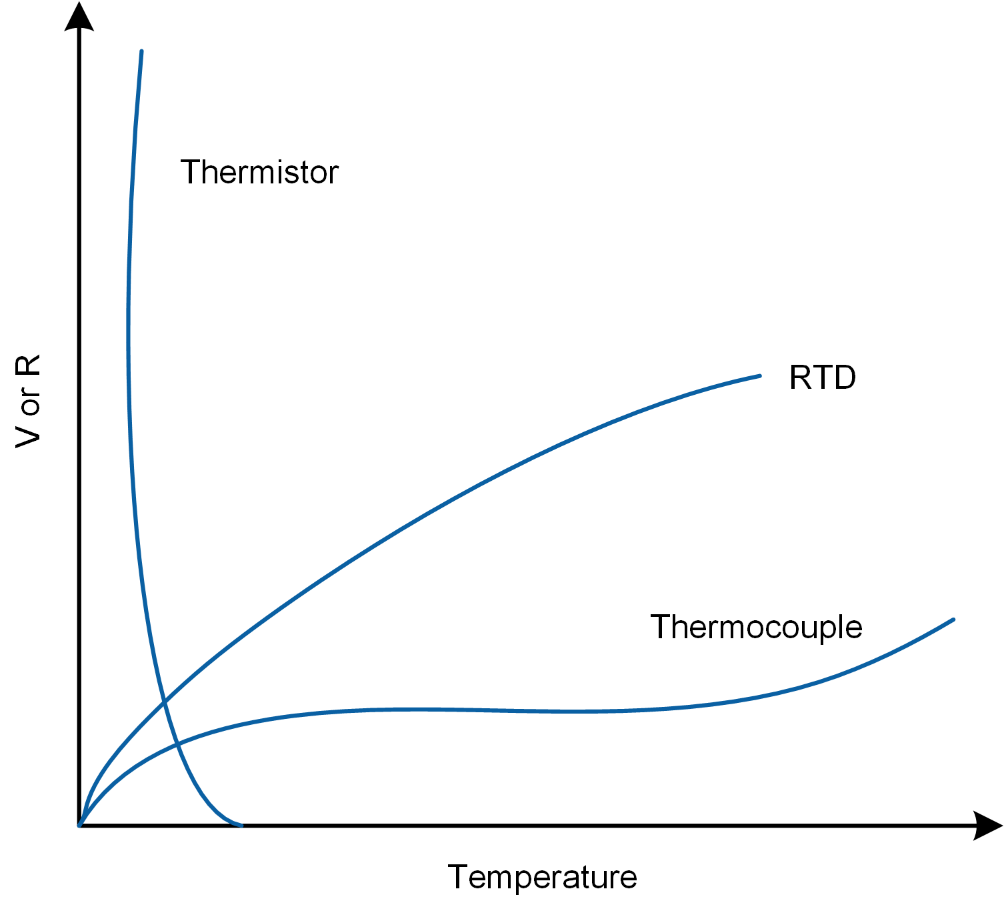


Figure 1-1: Comparison between the sensitivity of thermistors, RTD, and thermocouples

thermistor khác với cặp nhiệt điện ở chỗ chúng là cảm biến điện trở với cảm biến phía sau là cảm biến tạo điện áp. Vì một dòng điện phải được chạy qua một thermistor, nó rất nhạy cảm với nhiệt độ. Sự tự phát nhiệt này biểu hiện dưới dạng sai số đo. Hình 1-2, minh họa các hình dạng khác nhau của các thermistor thông dụng.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Thermistor - Concept Review.jpg  (a) board-mount type | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Thermistor - Concept Review.jpg  (b) string type |

Figure 1-2: Different types of common thermistors manufactured by Murata

Vì là một cảm biến điện trở, điện trở của một thermistor phụ thuộc vào nhiệt độ. Mối quan hệ giữa điện trở của thermistor, *RT*, và nhiệt độ, *T*, có thể được mô tả sử dụng phương trình tham số parameter:

Equation 1-1

với *R0* là điện trở danh định của cảm biến ở nhiệt độ *T0* tính bằng Kelvin, và  là thông số phụ thuộc vào vật liệu, nhiệt độ và cấu tạo của thermistor và thường nằm trong khoảng từ 3500 - 4700 K. Đối với thermistor trên Mechatronic Sensors board, điện trở cảm biến danh định là:

*R0* = 47,000 ohm

khi nhiệt độ ở 25 độ C hoặc:

*T0 =* 298.15 K

Phương trình tham số β cung cấp một ước tính chấp nhận được của nhiệt độ đo được trong các ứng dụng mà có một khoảng nhiệt độ hẹp (thường là dưới 20°C). Các tham số β có thể được thiết lập sử dụng hai điểm tham chiếu nhiệt độ.

Một nhiệt độ ước lượng chính xác hơn (± 0,01°C so với khoảng 100°C) có thể được thực hiện bằng cách sử dụng phương trình Steinhart-Hart như được đưa ra trong phương trình 1- 2:

Equation 1-2

trong đó *A*, *B*và *C*là các tham số Steinhart-Hart được xác định bằng quy trình hiệu chuẩn 3 điểm và *RT* là điện trở của thermistor ở nhiệt độ *T* tínhbằng Kelvin.

#### NTC và PTC Thermistors

Có hai loại thermistor: Hệ số nhiệt âm (NTC) và Hệ số nhiệt dương (PTC). Điện trở của thermistor NTC giảm khi nhiệt độ tăng. Ngược lại, điện trở của một thermistor PTC tăng lên khi các nhiệt độ tăng lên. thermistor NTC nhạy hơn so với thermistor PTC, và do đó thể hiện sự thay đổi điện trở cao hơn nhiều khi tiếp xúc với cùng mức nhiệt độ.

Đo đầu ra của thermistor

Tương tự như hầu hết các cảm biến điện trở, đầu ra của một thermistor được đo bằng cách sử dụng một mạch phân áp. Hình 1.3 minh họa các mạch điện áp phân chia sử dụng trong Mechatronic Sensor board, với *RT* là giá trị thermistor.

Sử dụng quy tắc phân áp, điện áp đầu ra của mạch điện trong hình 1-3 bằng phương trình 1-3:

Equation 1-3

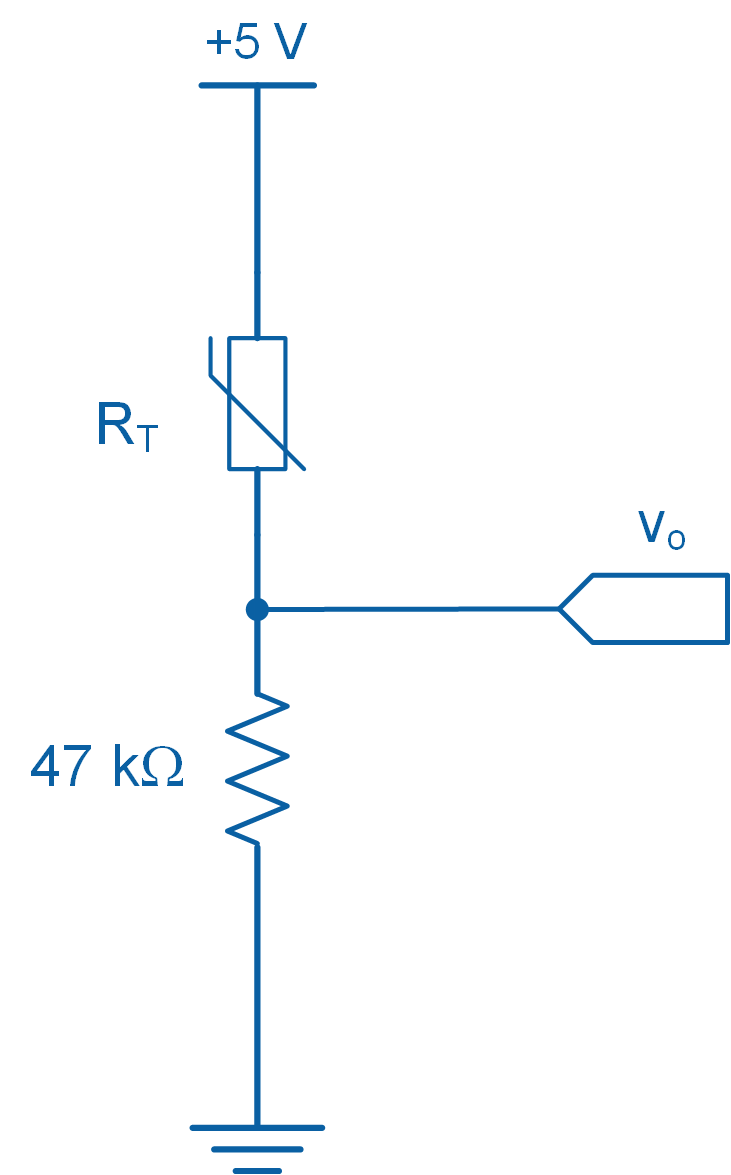


Figure 1-3: Thermistor circuit used in the Mechatronic Sensors board

Thời gian đáp ứng của thermistor

Cảm biến nhiệt độ phản ứng khác nhau với những thay đổi trong đầu vào của chúng. Ví dụ, một nhiệt kế thuỷ ngân khi được mang ra ngoài trời vào một ngày mùa đông lạnh giá sẽ phản ứng chậm với sự thay đổi của nhiệt độ. Nói cách khác, nhiệt kế thuỷ ngân không phản ứng tức thời với sự thay đổi nhiệt độ. Thời gian phản hồi là thước đo thời gian cần thiết để cảm biến phản hồi lại sự thay đổi. Thông thường, các cảm biến có thời gian phản hồi nhanh là tốt nhất.

#### Hằng số thời gian nhiệt

Hằng số thời gian nhiệt (τ) là thước đo phản ứng của cảm biến đối với sự thay đổi và được định nghĩa là thời gian để đầu ra của cảm biến đạt 63,2% (= 1-1/e *)*của điều kiện trạng thái ổn định từ điều kiện ban đầu.

Hình 1 - 4, minh họa một đường cong đáp ứng thời gian điển hình. Nó cho thấy đầu ra của cảm biến hoạt động như thế nào khi nó đạt đến trạng thái ổn định (steady-state). Như thể hiện trong Hình 1-4, sau một thời gian không đổi () đầu ra của hệ thống đạt 63,2% giá trị trạng thái ổn định, sau 2 hằng số thời gian (2) hệ thống đạt 86,5% giá trị trạng thái ổn định và 95 % đạt được sau 3 hằng số thời gian (3). Về mặt lý thuyết, trạng thái ổn định đạt được sau thời gian vô hạn, tuy nhiên, trong thực tế, người ta đợi cho đến khi đầu ra của hệ thống hoặc cảm biến nằm trong một biên độ sai số chấp nhận được (thường là 3 hằng số thời gian). Trong các ứng dụng nhiệt, hằng số thời gian thường lớn. Nhiệt điện trở thường có hằng số thời gian là 0,5-4s.

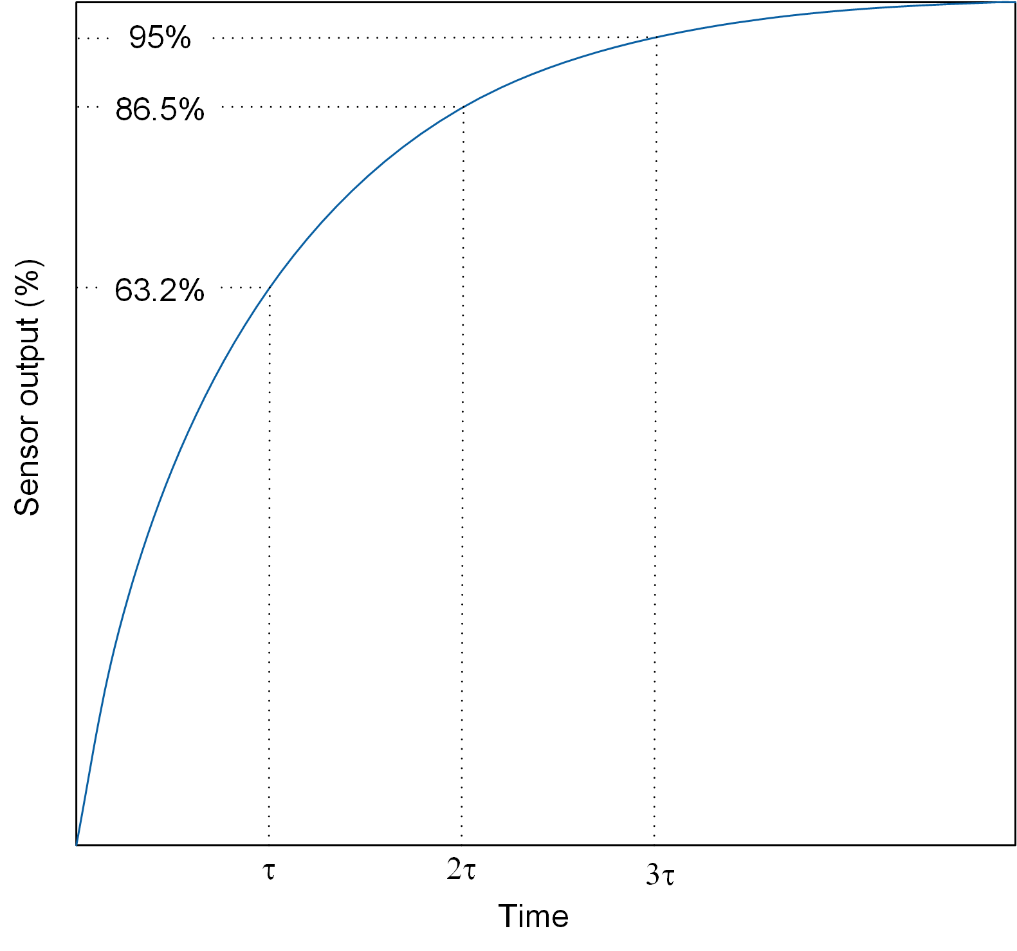


Figure 1-4: Time constant curve

### 1.2 Thực hiện

Virtual Instrument (VI) được sử dụng để thu thập dữ liệu và hiệu chuẩn thermistor (Hình 1-5).

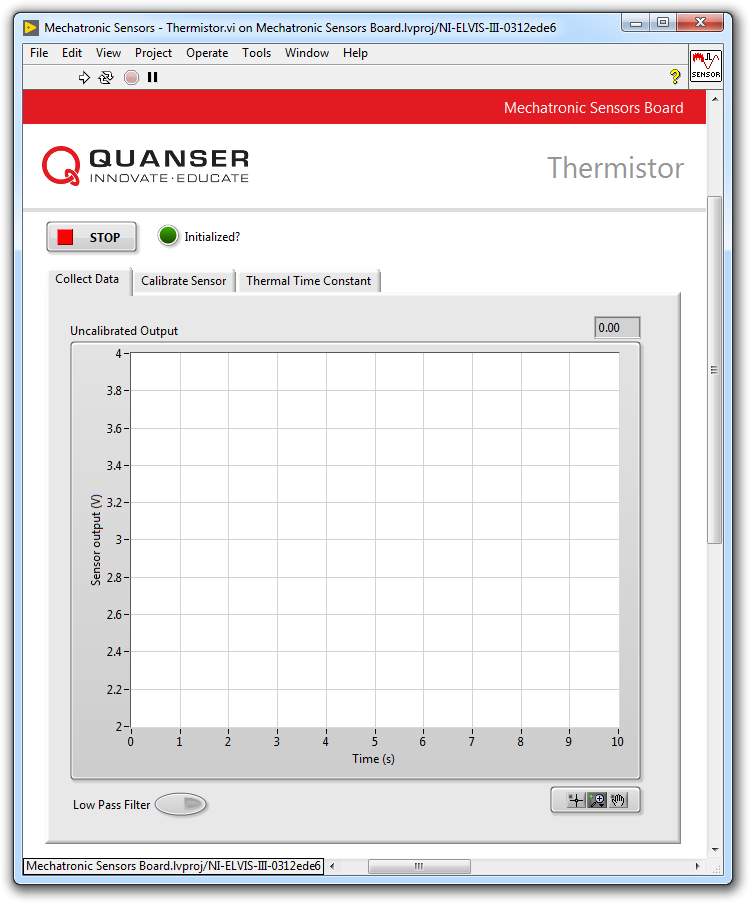


Figure1-5: VI for collecting data from the thermistor

#### Thu thập dữ liệu

1. Trước khi bắt đầu thử nghiệm này, hãy đo nhiệt độ của ngón tay cái của anh/chị bằng nhiệt kế và ghi lại giá trị trong Bảng 1-1.  
     
   Lưu ý: Để tiến hành thành công thí nghiệm, nhiệt độ đầu ngón tay của anh/chị phải trên 30°C.
2. Mở file **Mechatronic Sensors Board.lvproj**
3. Từ cửa sổ **Project Explore**, chọn **Mechatronic Sensors - Thermistor.vi**
4. Click vào tab **Collect Data**.
5. Chạy VI.
6. Đợi đèn báo LED **Initialized?** bật.
7. Nhẹ nhàng đặt đầu ngón tay của bạn lên nhiệt điện trở và kiểm tra và chụp ảnh màn hình của phản ứng.  
     
   Lưu ý: Nếu đầu ra của cảm biến bị nhiễu, hãy bật nút **Low Pass Filter**. Quá trình lọc làm giảm các thành phần không mong muốn khỏi đầu ra cảm biến, dẫn đến tín hiệu đầu ra "mượt mà hơn".
8. Sử dụng biểu đồ dạng sóng **Uncalibrated Output**, quan sát đầu ra cảm biến tương ứng khi nó đạt đến trạng thái ổn định (sau khoảng 15 s). Ghi lại giá trị trong Bảng 1-1
9. Tiếp tục thực hiện phần tiếp theo.

Table 1-1: Recorded fingertip temperature and steady-state sensor output

|  |  |
| --- | --- |
| Fingertip Temperature (°C) | Sensor Output (V) |
|  |  |

#### Hiệu chỉnh nhiệt điện trở

1. Sử dụng đầu ra trạng thái ổn định của cảm biến từ phần thu thập dữ liệu và Công thức 1-3, xác định điện trở của thermistor. Ghi lại giá trị này trong Bảng 1-2.
2. Sử dụng nhiệt độ của đầu ngón tay mà bạn đã xác định trong bước thu thập dữ liệu và phương trình 1-1, xác định thông số  của thermistor. Ghi lại giá trị này trong Bảng 1-2. Điện trở danh định của cảm biến là R0 = 47,000 ohm tại 25 °C.

Table 1-2: Sensor resistance and -parameter

|  |  |
| --- | --- |
| Sensor Resistance (ohm) | Calculated -parameter |
|  |  |

1. Trong VI, nhấp vào tab *Calibrate Sensor* để hiệu chỉnh đầu ra của thermistor theo nhiệt độ (tính bằng °C).
2. Nhập thông số  anh/chị đã tính toán cho thermistor bằng cách sử dụng điều khiển số *B* numeric control.
3. Kiểm tra xem nhiệt độ đã hiệu chuẩn có trùng khớp với nhiệt độ đầu ngón tay mà bạn đã đo trước đó không. Để thực hiện việc này, hãy nhẹ nhàng đặt đầu ngón tay của bạn lên cảm biến và xác minh rằng nhiệt độ đầu ngón tay chính xác được hiển thị trong biểu đồ dạng sóng *Calibrated Output* cũng như chỉ báp nhiệt kế *Temperature (C)*.
4. Bấm *Stop*.

### 1.3 Phân tích

1-1 Nhiệt độ đầu ngón tay mà anh/chị ghi trong Bảng 1-1 là bao nhiêu?

1-2 Điện áp đầu ra ở trạng thái ổn định của thermistor mà anh/chị ghi trong Bảng 1-1 là bao nhiêu? Tín hiệu có bị nhiễu không? Đính kèm ảnh chụp màn hình kết quả.

1-3 Điện trở của cảm biến mà anh/chị ghi trong Bảng 1-2 là bao nhiêu? Trình bày tính toán của anh/chị.

1-4 Thông số βcủa thermistor mà anh/chị ghi trong Bảng 1-2 là bao nhiêu? Trình bày các tính toán của anh/chị.

1-5 Trong bước 14, nhiệt độ đầu ngón tay đã hiệu chỉnh gần với nhiệt độ bạn ghi trong Bảng 1-1 như thế nào?

## Phần 2: Cân nhắc thiết kế

2-1 Giả sử rằng anh/chị được giao nhiệm vụ chọn một thermistor đo nhiệt độ của không khí bên trong ống dẫn. Một hạn chế thiết kế (design constraint) yêu cầu đầu ra của cảm biến đạt trạng thái ổn định trong vòng chưa đầy 5 giây để van an toàn có thể được vận hành để tránh hệ thống quá nóng. Xác định nếu thermistor sử dụng trong Mechatronic Sensors Board thỏa mãn điều kiện thiết kế. Giả sử trạng thái steady state đặt được sau 3 lần hằng số thời gian.

*Gợi ý:* Để kiểm tra tính phù hợp của cảm biến, anh/chị phải xác định hằng số thời gian nhiệt của nó bằng cách làm theo các bước sau:

* Chạy VI và đảm bảo rằng:
  + Nếu đầu ra cảm biến bị nhiễu, hãy nhấp vào tab *Collect Data* để kích hoạt low-pass filter.
  + Nhấp vào tab *Calibrate Sensor* and nhập tham số  mà anh chị đã xác định trước đó trong *B* numeric control.
* Nhấp vào tab *Thermal Time Constant*.
* Nhẹ nhàng đặt đầu ngón tay của bạn lên thermistor and và kiểm tra phản ứng bằng cách sử dụng biểu đồ dạng sóng *Calibrated Output*.
* Khi đầu ra cảm biến đã đạt đến trạng thái ổn định, hãy nhấp vào nút *Plot*. Đáp ứng của cảm biến sẽ được ghi lại trong đồ thị dạng sóng *Time Constant* như Hình 2-1.
* Sử dụng công cụ *Cursor* và thông tin được cung cấp trong phần Cơ sở lý thuyết, xác định hằng số thời gian nhiệt của thermistor.

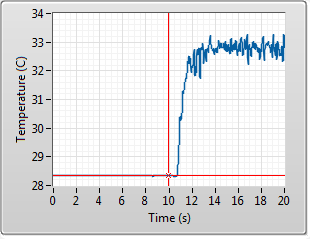


Figure 2-1: Typical thermistor response

2-2 Khi tiến hành phòng thí nghiệm này, anh/chị có thể đã quan sát thấy nhiễu trong đầu ra của cảm biến. Để khắc phục sự cố này, VI triển khai một bộ lọc thông thấp dựa trên phần mềm. Bộ lọc thông thấp cho phép các tần số thấp hơn mức cắt (cut-off) được xác định trước để “vượt qua (pass)”, trong khi “chặn (block)” hoặc làm giảm thông tin tần số còn lại. Điều này dẫn đến tín hiệu đầu ra "mượt mà hơn". Nhược điểm của bộ lọc là nó làm chậm phản ứng của các cảm biến. Trong thí nghiệm này, khi anh/chị đặt ngón tay trên thermistor, cơ thể của anh/chị hoạt động như một ăng-ten, thu hút nhiễu điện từ (EMI) không mong muốn từ môi trước. Nguồn gốc phổ biến của EMI thường là các đường dây nguồn AC.

Một phương pháp thay thế để làm giảm nhiễu trên thermistor là cách điện với cảm biến khi chạm vào. Điều này có thể đạt được bằng cách che thermistor bằng một miếng băng dính nhỏ hoặc màng bọc thực phẩm. Đặt tấm chắn như vậy sẽ cách điện cảm biến khỏi đầu ngón tay của bạn, đồng thời cho phép trao đổi nhiệt với cảm biến.

Figure 2-2 mình họa đầu ra thô (chưa lọc) của thermistor với các trường hợp sau: khi LPF được bật, và khi một băng dính trong được đặt trên cảm biến. Kết quả chỉ ra rằng việc đặt một miếng băng keo trong trên cảm biến giải quyết hợp lý trong việc lọc nhiễu không mong muốn!

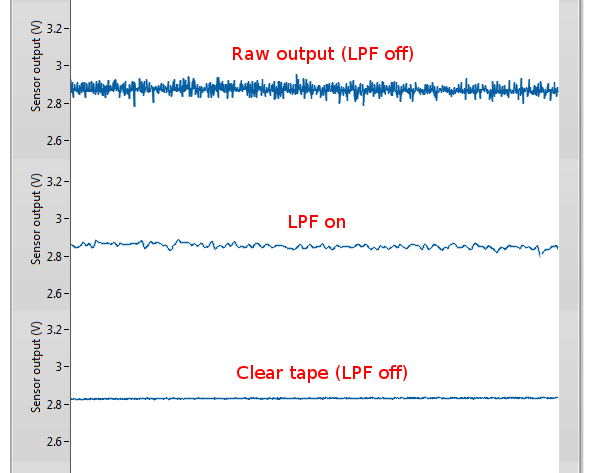


Figure 2-2: Thermistor raw and filtered responses

Nhiệm vụ của anh/chị là quan sát mức độ hiệu quả của việc sử dụng một miếng băng keo trong suốt hoặc màng bọc thực phẩm để làm giảm nhiễu của thermistor. Lưu lại ảnh chụp màn hình tương tự như trong Hình 2-2. Thêm vào đó, xác định hằng số thời gian nhiệt của thermistor khi băng dính trong được đặt trên cảm biến (tắt LPF trong khi kiểm tra). Kết quả có nói lên rằng thermistor phản hồi chậm hơn không?

# Lab 4: Strain Gage



Figure 0: Strain measurement is crucial in engineering design, as well as testing and maintenance practices

This lab explores the concept of strain measurement using a strain gage. The output of a strain gage mounted on a cantilever beam, placed in a quarter-bridge Wheatstone configuration, will be calibrated in terms of beam tip displacement. Furthermore, the natural frequency of the beam assembly will be measured by applying a fast Fourier transform to the response of the beam due to an impulse.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

* Understand the difference between quarter, half, and full Wheatstone bridge circuits
* Calibrate the output of a strain gage in terms of beam displacement
* Determine the natural frequency of a cantilever beam using a strain gage by applying a fast Fourier transform

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |
| AccessoriesFew small paper clips |  |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Record the zero offset of the bridge circuit
* Record the 5-point calibration data used to calibrate the strain gage
* Record the obtained calibration curve coefficients
* Screenshot of the calibration data showing the fitted curve
* Determine sensor sensitivity
* Determine the natural frequency of beam assembly
* Observe the relationship between mass and natural frequency

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Strain

### 1.1 Theory and Background

#### What is Strain?

Strain is a measure of deformation in a solid body due to an applied force. Figure 1-1 illustrates a rectangular bar being subjected to an axial tensile stress ().

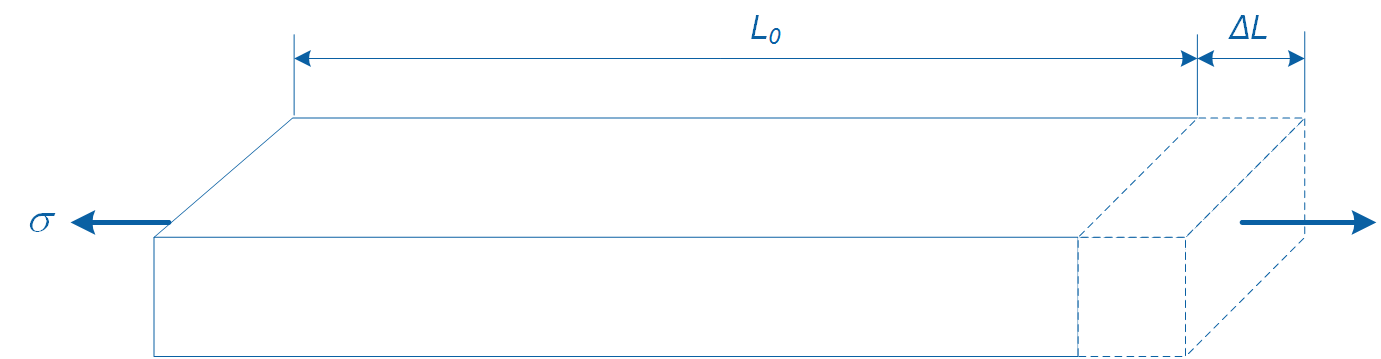


Figure 1-1: Change in length due to an axial tensile force

This stress causes a change in the original length of the bar from *L0* to *L0* + *L*. We define strain (**) using Equation 1-1:

Equation 1-1

where *L0* is the original length and *L* is the change in length due to the applied force. Strain is dimensionless and expressed as a percentage (%) or in mm/mm. However, since strain values are typically very small, strain is expressed in micro-strain () by multiplying strain by 106.

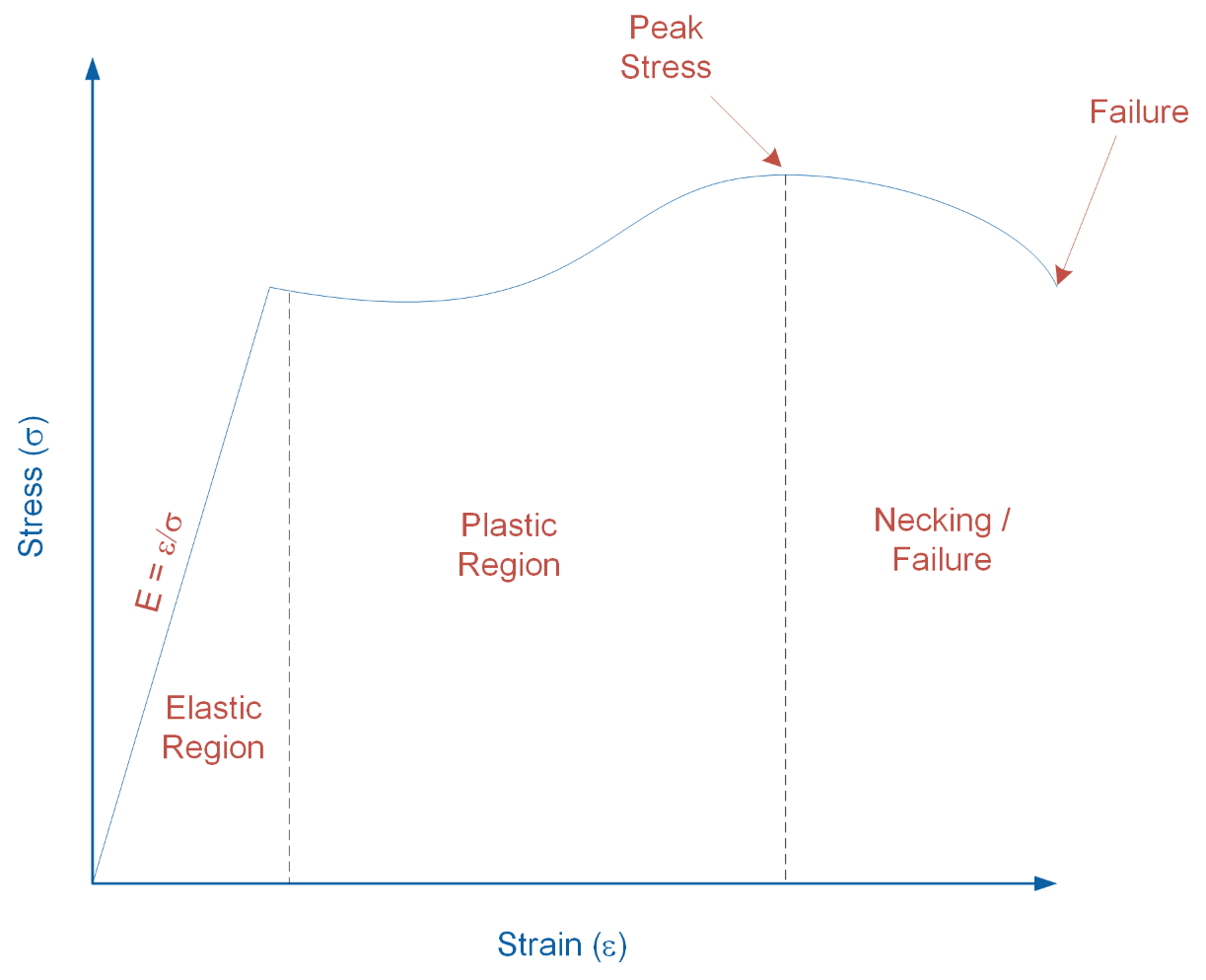


Figure 1-2: Stress-strain curve

Figure 1-2 illustrates the relationship between stress and strain when applied to a solid body. As shown in the graph, as stress increases, the solid body undergoes various deformation stages. In the elastic region, the body does not experience permanent physical change. In this region, the stress-strain relationship exhibits a linear relationship and we define the slope of the relationship as the modulus of elasticity (*E*=**) of the body. In the plastic region, the body permanently deforms, followed by the necking region where necking occurs prior to fracture.

#### Strain Gage

Strain gage is a sensor used for measuring strain in solid bodies. As shown in Figure 1-3, it is constructed from a fine metallic foil element formed into a grid pattern and mounted on a thin backing called a carrier. Strain gages are commonly bonded to test specimens using cyanoacrylate based adhesives or two-part epoxies. When an appropriate bond between the gage and specimen is established, any deformation in the specimen is transferred to the gage. This causes the resistance of the strain gage to change. When a strain gage is under tension its resistance increases, while under compression its resistance decreases.

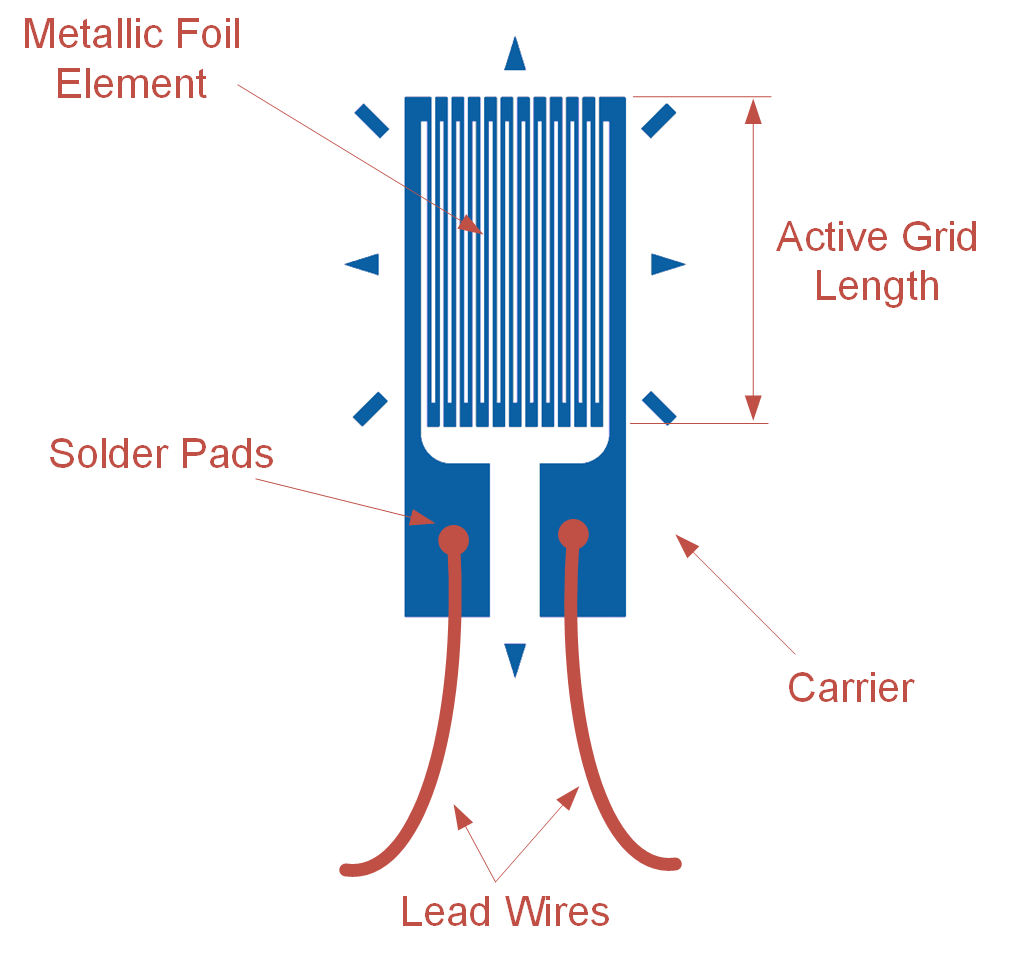


Figure 1-3: Schematic diagram of a strain gage

Strain gages vary in shape, orientation, and number depending on the type of strain being measured. The output of a strain gage is measured by connecting its lead wires to a dedicated signal conditioning circuit and DAQ, or dedicated strain measuring instrument.

Most strain gages have nominal resistances of 120 or 350 ohm. A higher nominal resistance and lower excitation voltage are desirable since that decreases measurement error due to lower Ohmic/self-heating effects.

The sensitivity to strain of a strain gage is called Gage Factor (GF) and is defined using Equation 1-2:

Equation 1-2

where *R* is the change in resistance when the strain gage is deformed, *RG* is the nominal resistance of the gage, and  is the induced strain. Typical gage factor values are approximately 2. For example, *GF* = 2 means if 1% strain is induced in a specimen, the gage’s relative resistance will change by 2%. Strain gages typically measure strains of up to 5% or 50,000 .

#### Wheatstone Bridge

The output of a strain gage is not directly measured; rather the voltage drop due to the change in the sensor’s resistance is measured using a Wheatstone bridge circuit (Figure 1-4). It offers several advantages over voltage dividing circuits, which are typically used for measuring the output of resistive sensors. One benefit is that a Wheatstone bridge allows for higher measurement sensitivity and lower measurement error. Another advantage is that it removes large fixed voltage drops which are present in a typical voltage dividing circuit. Since the output of Wheatstone bridge circuits is very low (typically in the range of microvolts), removing a large fixed voltage drop allows for the signal to be amplified using an amplifier.

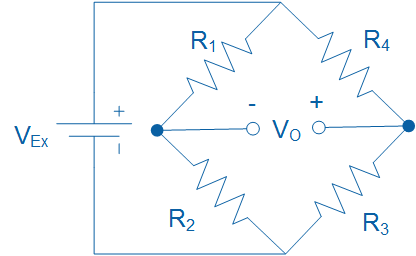


Figure 1-4: A Wheatstone bridge circuit for measuring the output of resistive sensors

The relationship between the resistors (*R1*, *R2*, *R3*, and *R4*), excitation voltage (*VEx*) and output voltage (*VO*) is governed by Equation 1-3:

Equation 1-3

For *VO* to be zero, the following relationship must hold true, in which case the bridge is said to be balanced:

Equation 1-4

However, when the resistance of one of the resistors changes value, the bridge circuit generates an output voltage and is said to become unbalanced. Generally, the following three distinct Wheatstone bridge configurations are used for measuring the output of strain gages: (a) quarter-bridge, (b) half-bridge, and (c) full-bridge configuration.

##### Quarter-bridge Configuration

Figure 1-5 illustrates a quarter-bridge Wheatstone bridge configuration. It consists of a single active strain gage (*RG*) and three fixed external precision resistors. It is the simplest strain measurement configuration and offers the lowest measurement sensitivity. Typically *RG* = 120 or 350 ohm when unstrained and the fixed resistors each equal 120 or 350 ohm, respectively.

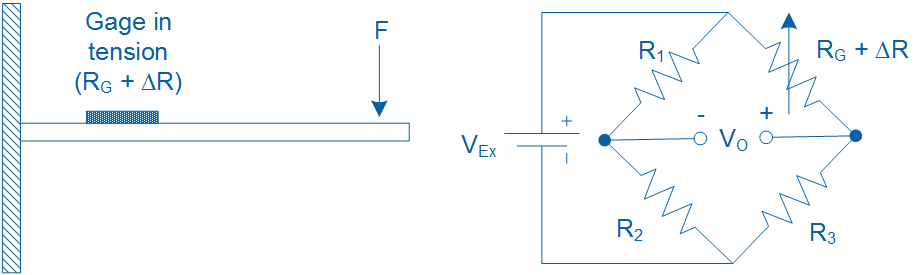


Figure 1-5: Quarter-bridge configuration for measuring strain in a cantilever beam

When a bending force is applied to the beam, it causes the beam along with the strain gage to deform. As a result, the resistance of the gage changes and a voltage output (*VO*) is generated which can be measured using a DAQ. The voltage output is proportional to the strain induced in the beam.

For example, assume that a bending force causes *RG*, which has a nominal resistance of 350 ohm, to increase by 0.0085 ohm. If the bridge is excited at 5 V, using Equation 1-3, the output voltage of the bridge circuit will be *V0* = -30 microvolts.

In practice, the output voltage of a quarter-bridge configuration is very minute and will require amplification to increase measurement resolution before being measured using a DAQ. Typical strain gage circuits or DAQs have built-in amplifiers to increase the signal levels to 10 mV/V (10 mV per each volt of excitation).

Assuming *R1* = *R2* = *R3* = *RG*, and substituting Eq. 1-2 in Eq. 1-3, the output voltage (*VO*) of the quarter-bridge circuit can be expressed in terms of *VEx*, *GF*, and the measured strain () as shown in Equation 1-5:

Equation 1-5

Note that the presence of the term 1/(1+GF∙/2) indicates non-linearity in the output of a quarter-bridge configuration with respect to strain.

##### Half-bridge Configuration

A half-bridge configuration uses two active strain gages and two fixed external resistors. Depending on the type of strain being measured (e.g. bending, torsion, tension, etc.), strain gages in a half-bridge configuration are mounted differently on a specimen. It offers the benefit of twice the sensitivity of a quarter-bridge configuration. Figure 1-6 illustrates a half-bridge configuration for measuring bending strain in a cantilever beam with two active gages mounted on opposite sides of the beam. When a bending force is applied to the beam, it causes tension in one of the gages while the other gage compresses.

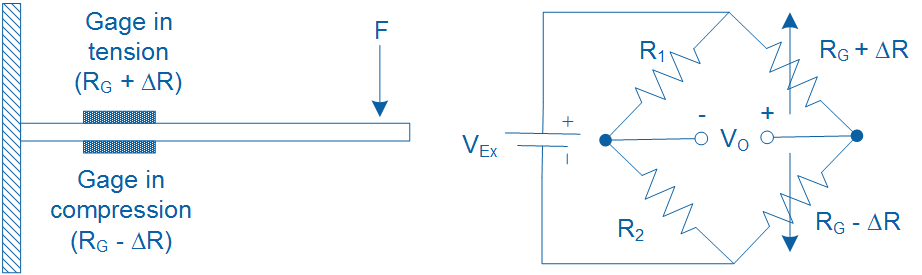


Figure 1-6: Half-bridge configuration for measuring strain in a cantilever beam

Assuming *R1* = *R2* = *RG*, and substituting Eq. 1-2 in Eq. 1-3, the output voltage (*VO*) of the half-bridge circuit can be expressed in terms of *VEx*, *GF*, and the measured strain () as shown in Equation 1-6:

Equation 1-6

##### Full-bridge Configuration

As illustrated in Figure 1-7, a full-bridge configuration uses 4 active strain gages of equal resistance (*RG*) and thus does not employ any external fixed resistors to complete the bridge circuit. Substituting Eq. 1-2 in Eq. 1-3, the output voltage of the full-bridge circuit (*VO*) can be expressed in terms of *VEx*, *GF*, and strain () as shown in Equation 1-7:

Equation 1-7

A full-bridge configuration produces twice the sensitivity of a half-bridge configuration and four times the sensitivity of a quarter-bridge configuration.

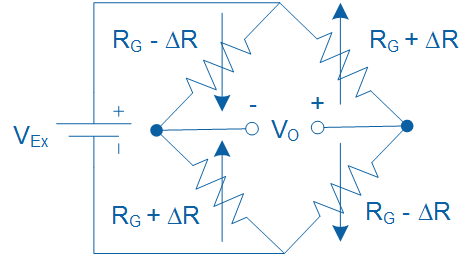


Figure 1-7: Full-bridge configuration for measuring strain

#### Effect of Temperature

In practice, change in temperature has a noticeable effect on the resistance of a strain gage, resulting in temperature induced strains. Such strains are caused by either self-heating of a strain gage, or due to differential thermal expansion between the strain gage and the specimen on which it is mounted.

Several practical methods exist to compensate for temperature-induced strain. One configuration, called quarter-bridge type II, uses one active gage and one dummy gage. This configuration is illustrated in Figure 1-8. The dummy gage is either mounted onto an identical unstrained secondary specimen which is placed in the proximity of the strained specimen, or it is mounted on the same specimen but in the transverse direction. Assuming *R1* = *R2* = *RG*, the output voltage (*VO*) of the quarter-bridge type II circuit can be expressed in terms of *VEx*, *GF*, and the measured strain () using Equation 1-5.

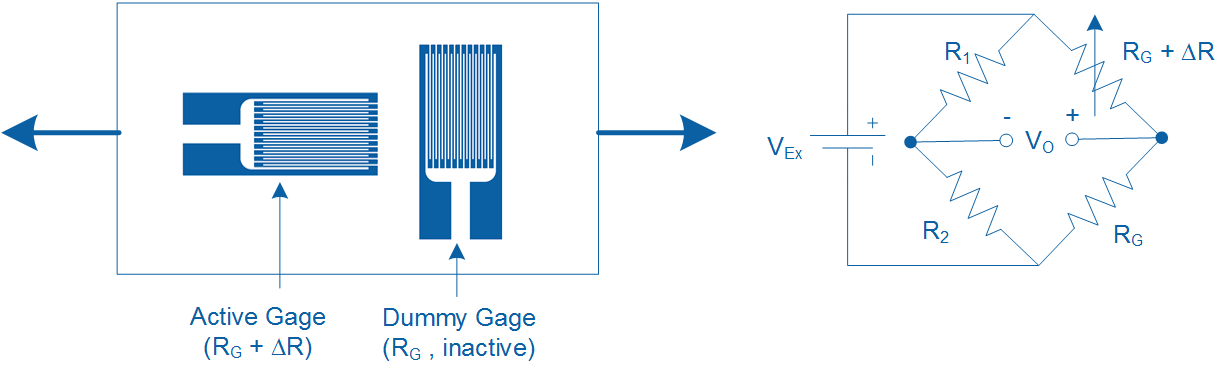


Figure 1-8: Quarter-bridge type II temperature compensating configuration

In this configuration, both the active and dummy gages experience the same fluctuations in temperature, with the resulting temperature induced strain canceling each other in the bridge configuration. Thus, any measured strain is due to active gage experiencing load-induced strain.

A more practical method of compensating for temperature-induced strain involves using a 3-wire self-temperature compensating strain gage. Such gages are made of alloys whose change in resistance due to temperature counters the change in resistance due to the differential thermal expansion between the gage and specimen. One of the limitations of such gages is that they should be mounted only on certain types of specimen.

#### Strain Gage Calibration

Strain gage calibration is the process of determining the mathematical relationship between the output of the Wheatstone bridge circuit versus the physical quantity being measured. Depending on the application the output of the bridge circuit may be calibrated to indicate strain (), deflection (mm), or mass (kg) by applying a series of known forces, displacements, or masses respectively. As part of the calibration process, the user must first adjust the *zero offset* and *full-scale span*.

Zero offsetting is the process of adjusting a Wheatstone bridge output to zero under no-load conditions. Such zero offsets exist because of normal tolerances in strain gage assemblies. This process, which establishes a reference point for measurement, is also referred to as null offsetting. Null offsetting is done using external resistors or via potentiometers built into the amplifier circuit of the strain measurement DAQ. Alternatively, zero offsetting can be achieved by means of software compensation rather than removing the offset off the bridge. However, if the offset is large enough, software compensation will limit the dynamic range of the measurement.

Full-scale span is the output range of a Wheatstone bridge circuit when the gage is subjected to maximum and minimum deflection. In practice, setting full-scale span requires the user to deflect the beam/gage assembly to its maximum or minimum positions and adjust the Wheatstone bridge output to a desired level using a gain potentiometer. Full-scale span is sometimes referred to as full-scale output (FSO) in sensor documentation.

Once zero offset and span have been adjusted, the user must apply three to five known inputs (e.g. deflection or load) to the specimen/strain gage assembly, and record the corresponding output of the bridge circuit. A calibration equation is then obtained by fitting a line to the measured points. Once the calibration equation is determined, it can be used to calculate the calibrated physical quantity for any given output of the bridge circuit.

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the strain gage is shown in Figure 1-9.

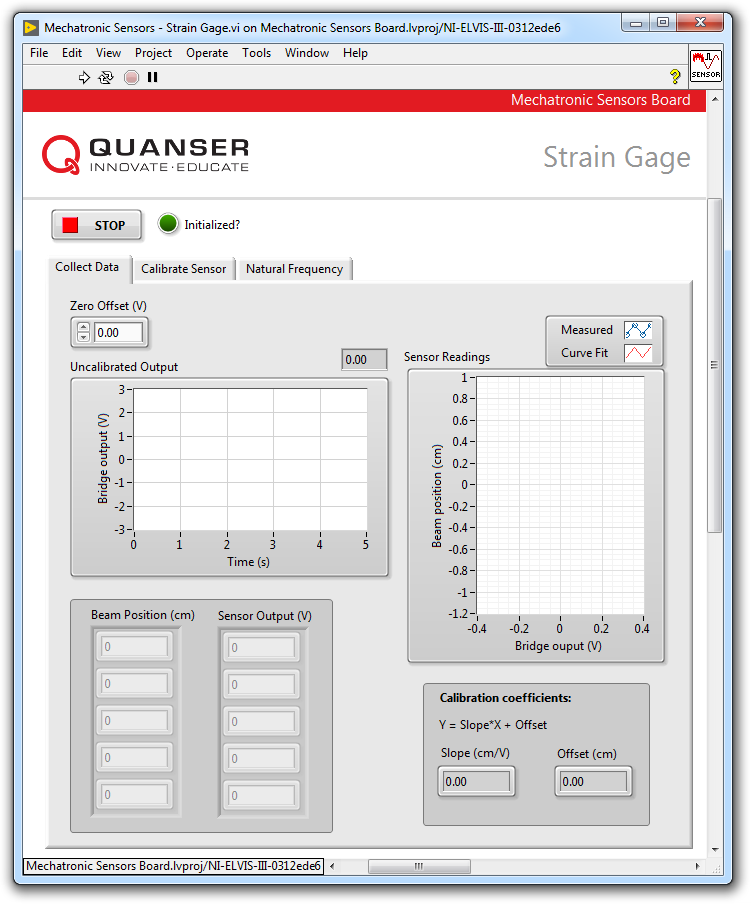


Figure1-9: VI for collecting data from the strain gage

#### Collect Data

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Strain Gage.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. Using the **Uncalibrated Output** waveform chart, read the initial strain gage bridge output.
7. Balance the output of the strain gage bridge. To do this, make sure position and hold the tip of the cantilever beam at the 0 cm mark. Adjust the **Zero Offset (V)** numeric control such that the uncalibrated output of the bridge circuit is as close as possible to 0.00 V. Record the zero offset value in Table 1-1.
8. Enter -1 in the **Beam Position (cm)** array.
9. Flex the tip of the cantilever beam to the -1 cm mark.
10. Read the corresponding strain gage output and enter the value in the **Sensor Output (V)** array.
11. Repeat the process by moving the tip of the beam to the following positions: -0.5 cm, 0 cm, +0.5 cm and +1 cm. Each time, enter the beam position and measured sensor outputs in the **Beam Position (cm)** and **Sensor Output (V)** arrays respectively.
12. Once the measured readings are entered, a linear curve is automatically generated to fit the data. The curve is shown in the **Sensor Readings** waveform graph. This curve represents the calibration curve of the sensor. Take a screenshot of the graph.
13. The slope and offset of the calibration curve are automatically calculated by the VI and displayed in the **Slope (cm/V)** and **Offset (cm)** indicators. Make a note of these values in Table 1-2.
14. Record the collected data in Table 1-3.
15. Take a screenshot of the **Sensor Readings** graph.
16. Continue to the next section.

Table 1-1: Recorded bridge zero offset

|  |  |
| --- | --- |
| Zero offset (V) |  |

Table1-2: Calibration coefficients

|  |  |
| --- | --- |
| Slope (cm/V) | Offset (cm) |
|  |  |

Table 1-3: Recorded bridge output

|  |  |
| --- | --- |
| Beam Position (cm) | Bridge Output (V) |
| -1.0 |  |
| -0.5 |  |
| 0.0 |  |
| +0.5 |  |
| +1.0 |  |

#### Calibrate the Strain Gage

1. Click on the **Calibrate Sensor** tab to calibrate the output of the strain gage bridge circuit in terms of linear displacement of the tip of the cantilever beam (in cm).
2. Use the **Slope (cm/V)** and **Offset (cm)** numeric controls to enter the slope and offset values you obtained during the data collection step.
3. Test the accuracy of your calibration. To do this, flex the cantilever beam to different positions and verify that the correct position is displayed in the **Calibrated Output** waveform chart as well as the **Beam Position (cm)** slider indicator.
4. Press the **Stop** button.

### 1.3 Analyze

1-1 What is the initial bridge zero offset that you recorded in Table 1-1?

1-2 Present the calibration coefficients that you recorded in Table 1-2.

1-3 Present the calibration data you recorded in Table 1-3.

1-4 Attach a screenshot of the Sensor Readings waveform graph showing the fitted calibration curve from step 12.

1-5 What calibration equation did you obtain?

1-6 What is the sensitivity of the amplified bridge circuit in V/cm?

1-7 How well did your calibrated output match the actual beam tip position in step 19?

1-8 Based on the data you collected in Table 1-1, use Equation 1-5 in to determine the maximum and minimum strains induced in the cantilever beam as it’s flexed between -1 cm and +1 cm. For each case, determine if the strain gage is under tension (or compression). Assume a gage factor of GF = 2, a bridge excitation voltage of VEx = +5 V, and an amplification gain of 100.

*Hint:* The output of the bridge circuit is amplified prior to being displayed in the VI. All calculations done using Equation 1-5 must be done using actual (i.e. pre-amplified) bridge output values.

## Section 2: Design Considerations

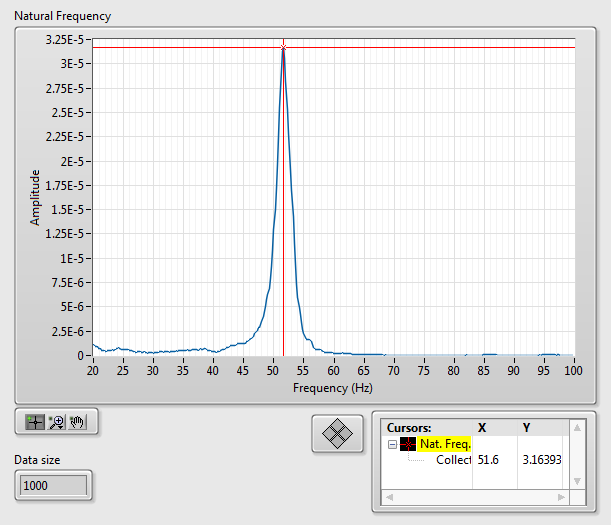
2-1 A strain gage can be used to indirectly measure other physical quantities such as vibration. Accurate measurement of vibration is imperative in ensuring the health of electro-mechanical systems. For example, excessive vibration can cause fractures in the fuselage of an airplane, or cause solder joints to break free in electronic circuit boards. Strain-based vibration switches are commonly used as a simple protection device that senses vibration and triggers an alarm or shuts down a machine if vibration exceeds a certain threshold.

The strain gage mounted on the cantilever beam in the Mechatronic Sensors Board can be used to determine the natural frequency of the beam assembly. Natural frequency is a property of an object that quantifies the frequency at which it “wants” to naturally vibrate when subjected to a disturbance. If a system has a natural frequency that matches normal environmental vibration, then the system will vibrate more violently and may prematurely fail.

  
A structure may fail if its natural frequency matches environmental vibration

Determine the natural frequency of the beam assembly by following the steps below:

* Run the VI.
* Click on the **Natural Frequency** tab.
* Ensure the beam is at rest (i.e. not vibrating).
* With one finger gently bend and then release the tip of the beam.
* Wait for a couple of seconds for the beam to stop vibrating then promptly press the **Stop** button.
* The VI will apply a fast Fourier transform to the captured data and display the result in the **Power Spectrum** waveform graph. The response should look similar to the figure below.
* Using the **Cursor** tool, measure the peak frequency.



Sample natural frequency response

2-2 Natural frequency of a cantilever beam can be calculated using the following equation:

where *fn* is natural frequency in Hz, *k* is stiffness of the beam in N/m, and *m* is the mass of the beam in kg. The equation indicates that natural frequency has an inverse relationship with the square root of mass. Validate this relationship by attaching a small paper clip to the tip of the cantilever beam and measuring the natural frequency of the modified beam. Does increasing the beam mass decrease its natural frequency?

# Lab 5: Pressure



Figure 0: Pressure measurement is an essential part of industrial automation

This lab explores pressure measurement using a capacitive pressure transducer. Reference points based on Boyle’s law will be established prior to calibrating the transducer. Both gage and absolute pressure will be measured.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

* Establishing calibration reference points based on Boyle’s law
* Conducting a 5-point sequential calibration of a pressure transducer
* Comparing upscale and downscale calibration curves
* Measuring pressure using both gage and absolute scales

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Determine local atmospheric pressure adjusted for elevation
* Calculate and record pressure reference points using the syringe by applying Boyle's law
* Record calibration data (up-scale and down-scale directions)
* Screenshot of calibration graphs showing fitted curves
* Record calibration coefficients (up-scale and down-scale directions)
* Record maximum and minimum generated absolute and gage pressures
* Calculate sensor sensitivity in V/kPa
* Calculate theoretical capacitance for given sensor specifications
* Compare obtained sensor calibration with manufacturer's calibration

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Pressure Measurement and Calibration

### 1.1 Theory and Background

#### What is a Pressure?

Pressure is the amount of force (F) acting per unit area (A) as defined in Equation 1-1:

Equation 1-1

Various units are used to express pressure. The SI unit of pressure is pascal (Pa), which is newtons per meter squared (N/m2). Other common units include pounds per square inch (psi), bar, and atmosphere (atm). Pressure at sea level under standard atmospheric conditions is defined as *patm* = 1 atm and equals 101.32 kPa.

Pressure is measured using two different scales: gage or absolute. Gage pressure (*pgage*) is pressure relative to local atmospheric pressure (*p0*), while absolute pressure (*pabs*) is pressure relative to perfect vacuum. The relationship between absolute and relative pressure is given in Equation 1-2 and illustrated in Figure 1-1.

Equation 1-2

Note that the value of *p0* depends on atmospheric and geographical conditions and may or may not be higher than standard atmospheric pressure.



Figure1-1: Comparison between different pressure scales

#### Boyle’s Law

Boyle’s law states that the product of volume and pressure of a confined gas is constant. It is mathematically expressed as:

Equation 1-3

where *p* is the pressure of the gas, *V* is the volume of the gas, and *k* is a constant. An alternative form of Boyle’s law is commonly used to compare a gas under two different conditions, and expressed as:

Equation 1-4

where *p1* and *V1* are the pressure and volume of the gas under condition 1, and *p2* and *V2* are the pressure and volume of the gas under condition 2. Figure 1-2 illustrates Boyle’s law using a pressure chamber. In this example, as the plunger is pushed into the chamber, the pressure of the gas increases from its initial state of *p1* to *p2*, while the volume of the gas decreases from *V1* to *V2*.

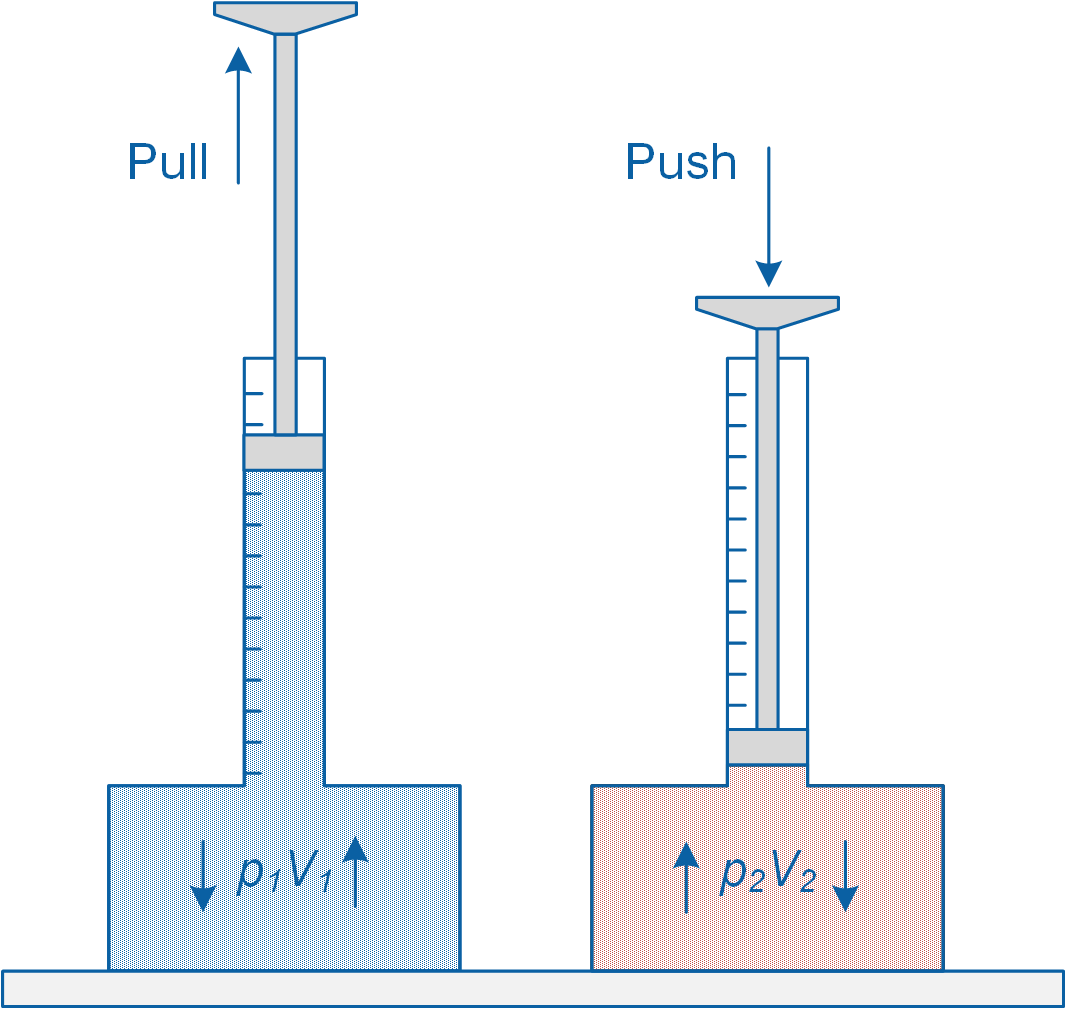
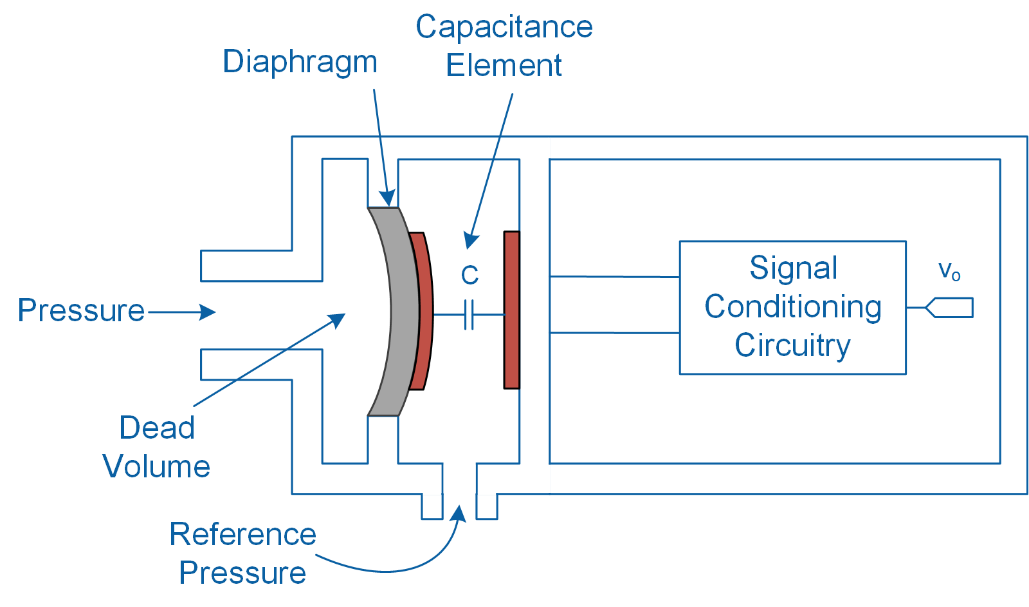


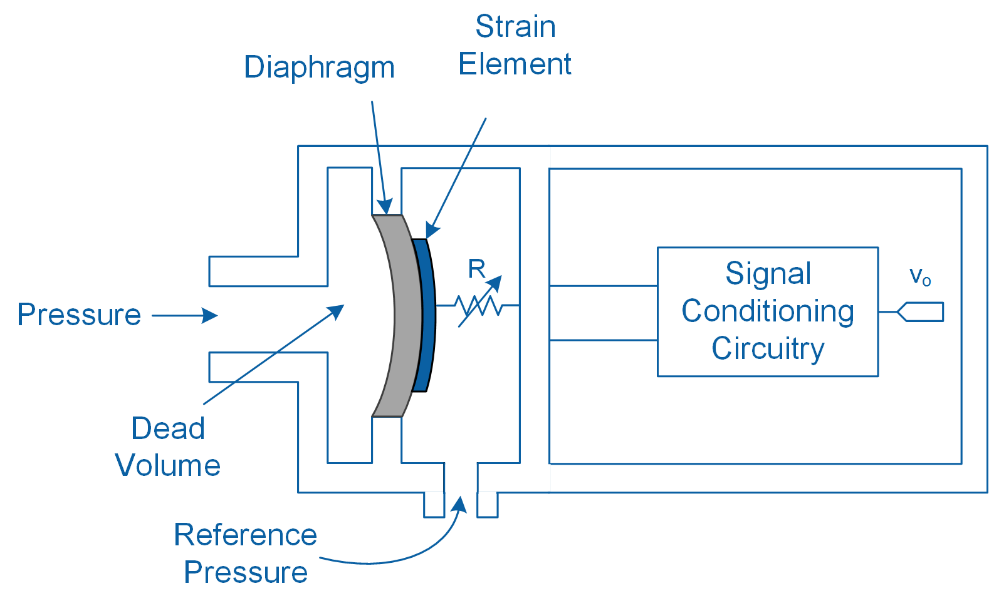
Figure1-2: Demonstration of Boyle’s law

#### Pressure Transducer

Pressure transducers are used in modern mechatronic applications to measure gage, absolute, and differential pressure. They are electromechanical devices that output an electrical signal proportional to the sensed pressure. A pressure transducer has an internal diaphragm or membrane that flexes due to an applied pressure.



(a) Capacitive type



(b) Resistive type

Figure1-3: Schematic illustration of two common types of pressure transducers

As schematically illustrated in Figure 1-3, a pressure transducer typically consists of either a strain/piezoresistive element or a parallel plate capacitor that is attached to a diaphragm. Depending on the type of sensing element used, any deformation in the diaphragm due to an applied pressure causes a change in the sensing element’s resistance or capacitance. Changes in the sensing element are then converted into a detectable electrical signal using signal conditioning circuitry.

Pressure transducers typically have a fast response behavior and allow for high precision measurements. However, because of their built-in electronics, they require a power source for operation.

As illustrated in Figure 1-3, a pressure transducer incorporates a built-in reference pressure port. When measuring gage pressure, the port is exposed to atmospheric pressure. However, when measuring absolute pressure, the port is sealed so that measurements are referenced against perfect vacuum. In differential pressure measurement the main and reference pressure ports are exposed to two different sources of pressure.

#### Pressure Transducer Calibration

A convenient way to convert the output of a pressure transducer to a unit of pressure is to use the manufacturer’s published sensitivity and offset values for the transducer. The sensitivity of a pressure transducer, which relates the output of the sensor to the measured pressure, is typically given in terms of volts per unit of pressure.

Alternatively, a sequential calibration test can be performed, which involves applying a sequential series of known pressures (known as *standards*) to the transducer and recording the corresponding output. The calibration can be done by increasing the input pressure (upscale direction), or by decreasing the input pressure (downscale direction).

The relationship between the input and output of the sensor is mathematically expressed in the form of a first order polynomial equation that is fitted to the recorded data. Using the calibration equation, unknown pressures can be determined for any given output signal.

Table 1-1: Results of a 5-point calibration

|  |  |
| --- | --- |
| Pressure (psi) | Output (V) |
| 0.0 | 0.100 |
| 5.0 | 1.255 |
| 10.0 | 2.550 |
| 15.0 | 3.505 |
| 20.0 | 4.950 |

As an example, Table 1-1 shows results of a 5-point calibration in the upscale direction. The sensors’ calibration equation is established by fitting a line to the points as shown in Figure 1-4.

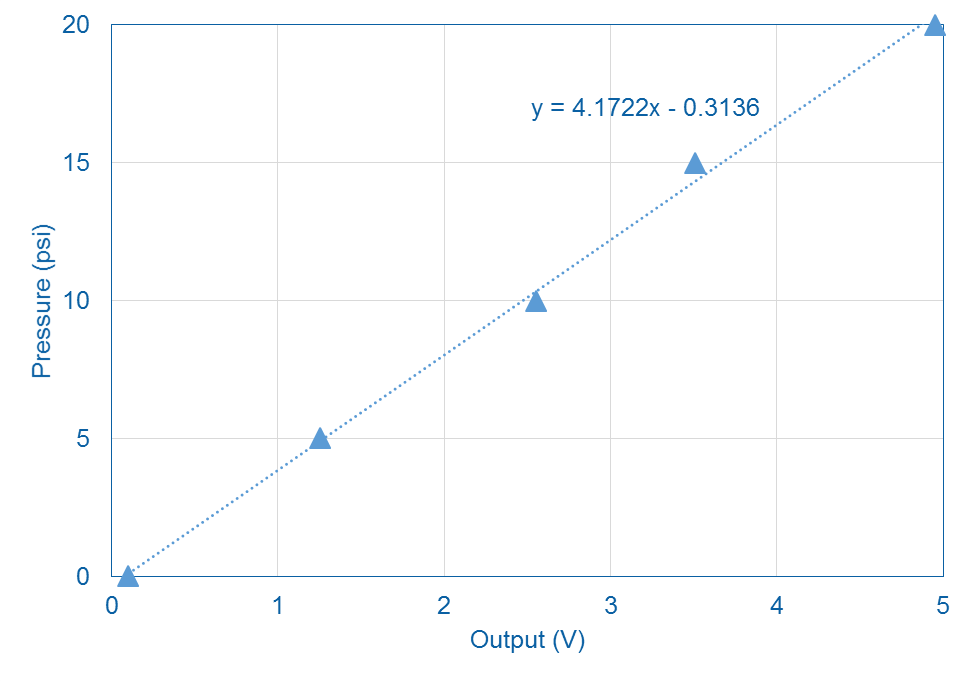


Figure 1-4: Line fitted to the recorded data

### 1.2 Implement

The Virtual Instrument (VI) used to collect data from and calibrate the pressure transducer is shown in Figure 1-5.

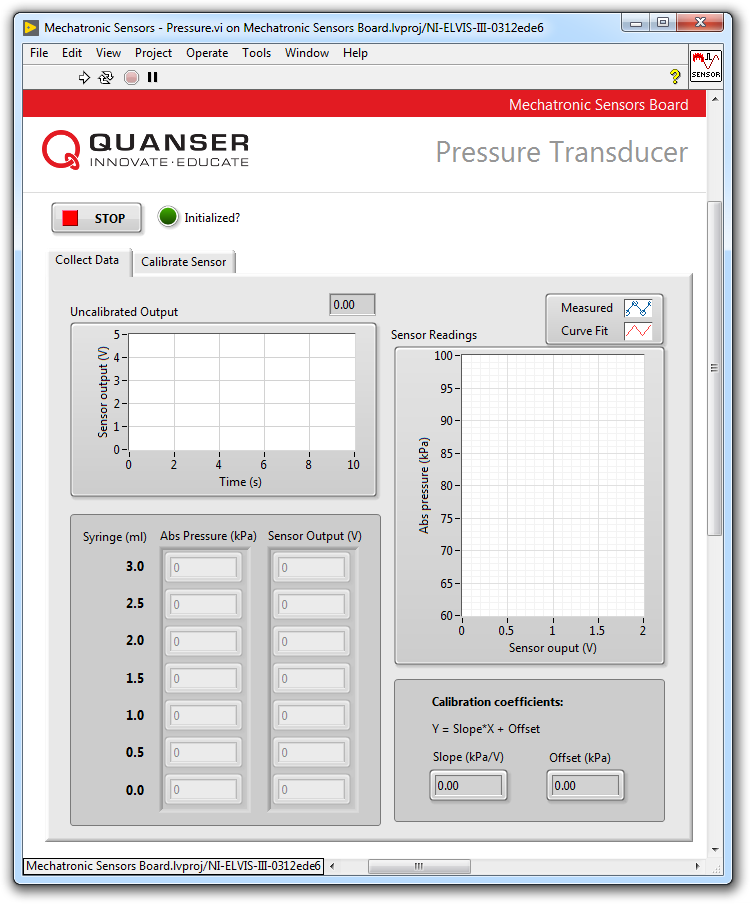


Figure1-5: VI for collecting data from the pressure transducer

#### Establish Reference Points

1. Determine your local atmospheric pressure (in kPa) using a barometer or look up the atmospheric pressure at your closest weather station (e.g. using www.weather.com). If you do not have access to this data, assume a local atmospheric pressure of 101.3 kPa.  
     
   *Note:* Weather stations typically report local atmospheric pressure corrected to sea-level conditions (0 m altitude and 15 °C ambient temperature). In such case, correct the reported sea-level pressure to your local atmospheric pressure by taking into consideration your altitude and ambient temperature using Equation 1-5:

Equation 1-5

where *plocal* is the local atmospheric pressure adjusted for temperature and altitude in kPa, *psea* is the sea-level pressure reported by the local weather station in kPa, *h* is your local altitude in meters, and *T* is ambient temperature in Celsius.

1. In the next section, you will be applying a series of known absolute pressures to the transducer using the syringe. Initially, you will start at atmospheric pressure with the syringe plunger aligned with the 3 ml mark. You will then increase the applied pressure by depressing the syringe plunger.  
     
   Using the adjusted local atmospheric pressure that you determined in the previous step and Equations 1-3 and 1-4, calculate the pressures that can be generated when the syringe plunger is positioned at the following volumetric markings: 3.0, 2.5, 2.0, 1.5, 1.0, 0.5, and 0.0 ml. Note that the pressure chamber has a dead volume of 5 ml. Record your calculations in Table 1-2.  
     
   *Note:* The dead volume refers to the volume of the pressure chamber. It excludes the volume of the attached syringe.
2. Continue to the next section.

Table1-2: Reference pressures using the syringe

|  |  |
| --- | --- |
| Syringe (ml) | Absolute Pressure (kPa) |
| 3.0 |  |
| 2.5 |  |
| 2.0 |  |
| 1.5 |  |
| 1.0 |  |
| 0.5 |  |
| 0.0 |  |

#### Collect Data

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Pressure.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. If attached, unscrew the syringe from the pressure chamber.
7. Position the syringe plunger at the 3 ml mark.
8. If present, remove the protective cap from the pressure sensor chamber.
9. Attach the syringe to the Luer lock connector on the pressure sensor chamber by gently twisting the syringe in the clockwise direction. Ensure a secure connection is made. The applied pressure inside the chamber is now equal to atmospheric pressure.  
     
   *Note:* Attach the syringe such that the graduation markings are facing forward.
10. Enter the first value from Table 1-1into the **Abs Pressure (kPa)** array located in the VI front panel. Now using the **Uncalibrated Output** waveform chart, read the corresponding sensor output and enter the value in the **Sensor Output (V)** array.
11. Continue taking measurements by depressing the syringe plunger at 0.5 ml intervals. Enter the applied pressures (using the values from Table 1-1) and the corresponding sensor outputs in the **Abs Pressure (kPa)** and **Sensor Output (V)** arrays respectively.  
      
    *Note:* Once all of the measured readings have been entered, a linear curve is automatically generated to fit the data. The curve is shown in the **Sensor Readings** waveform graph. This curve represents the calibration curve of the sensor.
12. Record the collected data in Table 1-3.
13. The slope and offset of the calibration curve are automatically calculated by the VI and displayed in the **Slope (kPa/V)** and **Offset (kPa)** indicators. Make a note of these values in Table 1-4.
14. Take a screenshot of the **Sensor Readings** graph showing the fitted curve.
15. Continue to the next section.

Table1-3: Recorded calibration data

|  |  |  |
| --- | --- | --- |
| Syringe (ml) | Absolute Pressure (kPa) | Output (V) |
| 3.0 |  |  |
| 2.5 |  |  |
| 2.0 |  |  |
| 1.5 |  |  |
| 1.0 |  |  |
| 0.5 |  |  |
| 0.0 |  |  |

Table1-4: Calibration coefficients

|  |  |
| --- | --- |
| Slope (kPa/V) | Offset (kPa) |
|  |  |

#### Calibrate the Pressure Transducer

1. Click on the **Calibrate Sensor** tab to calibrate the output of the transducer in terms of pressure in kPa.
2. Use the **Atmospheric Pressure (kPa)** numeric control to enter the adjusted atmospheric pressure you determined in step 1.
3. Use the **Slope (kPa/V)** and **Offset (kPa)** numeric controls to enter the slope and offset values you obtained during the data collection steps.
4. Move the plunger to different positions and observe the displayed pressures in the **Calibrated Output** waveform chart as well as the **Absolute** and **Gage** gauge indicators. What is the difference between the absolute and gage readings?
5. What is the maximum absolute and gage pressures (in kPa) that you can generate?
6. Can you generate negative gage pressure using the syringe? What is the lowest absolute and gage pressure (in kPa) that you can generate?
7. Press the **Stop** button.

### 1.3 Analyze

1-1 What is the local atmospheric pressure (in kPa), adjusted for altitude and temperature, that you determined in step 1? Show your calculations.

1-2 Present the reference points that recorded in Table 1-2. Show sample calculations.

1-3 Present the calibration data that you recorded in Table 1-3.

1-4 Present the calibration coefficients that you recorded in Table 1-4.

1-5 Attach a screenshot of the *Sensor Readings* waveform graph showing the fitted calibration curve.

1-6 What calibration equation did you obtain?

1-7 How did you generate negative gage pressure using the syringe?

1-8 What is the maximum and minimum absolute as well as gage pressures (in kPa) that you were able to generate in steps 23 and 24?

1-9 What is the sensitivity of the sensor in V/kPa?

## Section 2: Design Considerations

2-1 In this lab you obtained a calibration curve for the pressure transducer by means of a sequential test where pressure was increased in the upscale direction. When designing a mechatronic system, the following question may arise: can one use the same calibration equation to infer pressures outside of the applied lower and upper pressure limits? Put differently, can one use the same calibration equation to extrapolate beyond the calibration range?  
  
Partially examine this question by repeating this lab but this time calibrating the transducer using negative gage pressures (downscale direction). How does the downscale calibration curve compare with the upscale curve originally obtained in this lab? Can the upscale curve be used to extrapolate negative pressures?

2-2 The Mechatronic Sensors Board uses an Infineon KP236N6165 pressure sensor. The manufacturer provides the following calibration equation relating the sensor’s output voltage (Vout) to the measured pressure (P):

where VDD = 5 V, a = 0.00876 1/kPa, and b = -0.48571. The calibration equation applies to an operating range of 60 to 165 kPa. How does the manufacturer’s calibration equation compare with the equation that you obtained during this lab? Measure atmospheric pressure using the calibration equation that you experimentally obtained as well as using the manufacturer’s calibration equation. How do they compare? Comment on potential sources of error. The manufacturer reports a sensor accuracy of +/-1.0 kPa when the temperature is between 0 and 85 degrees Celsius.

2-3 As noted in the Theory and Background section, a capacitive pressure sensor contains a parallel plate capacitor that is attached to the sensor’s diaphragm. When the sensor is subjected to different pressures, it causes the sensors’ diaphragm to flex, and as a result the capacitance of the parallel plate capacitor changes. The sensors’ capacitance, C, is determined by the following equation:

where *d* is the distance between the plates, *A* is the overlapping area of the two plates,  is the proportionality constant (= 8.85 x 10-15 F/mm when the area is measured in mm2 and distance is measured in mm), and *c* is the dielectric constant of the material between the capacitor plates.

Assume you are designing a new capacitive pressure transducer. Estimate the theoretical capacitance of the sensors if the plate area is A = 1 mm2, plate distance is d = 500 microns, and the dielectric material between the plates is air c = 1.

# Lab 6: Contact



Figure 0: Touch pads and keyboards are examples of commonly used contact sensors

This lab explores contact sensors including a snap action switch and a capacitive touch sensor. Mechanical switch debouncing, as well as scroll and single-button touch action using a capacitive touch sensor, are examined.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

1. Apply software-based debouncing to a snap action switch
2. Use capacitive touch sensing to simulate single-button touch and scroll actions

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Screenshots of switch debouncing for different debounce periods
* Screenshot of the response of the touchpad due to contact
* Estimate of the distances at which the virtual buttons indicate contact for different threshold values
* Pseudo code describing the virtual button counter implemented in the VI
* Pseudo code for determining scroll direction

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Switch Debouncing

### 1.1 Theory and Background

#### Types of Switches

Switches are devices that create or break electrical contact by means of mechanical actuation, photo interruption, or magnetic actuation. Switches are found in most electronic applications. Depending on the type of switch used, it can create or break one or more electrical contacts. Switches can be classified into different types based on several factors such as method of actuation, type of operation, and construction. Figure 1-1 shows several common types of mechanical switches that are used in mechatronic applications. They include snap action, slide, rocker, pushbutton, and toggle switches. These switches typically contain internal metallic contacts and springs, which require physical actuation to operate.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg(a) snap action | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg(b) slide | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg (c) rocker | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg(d) push button | C:\Users\amolki\Desktop\QNET Mechatronic Sensors - Switch Debouncing - Concept Review.jpg (e) toggle |

Figure 1-1: Common types of mechanical switches (source: DigiKey)

As illustrated in Figure 1-2, switches can be classified based on their switching action. In this type of classification, pole refers to the number of circuits that can be controlled when the switch is actuated, and throw refers to the number of possible contacts for each of the poles.

|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\amolki\Desktop\1.png | C:\Users\amolki\Desktop\2.png | C:\Users\amolki\Desktop\3.png | C:\Users\amolki\Desktop\4.png |
| (a) single-pole, single-throw (SPST) | (b) single-pole, double-throw (SPDT) | (c) double-pole, single-throw (DPST) | (d) double-pole, double-throw (DPDT) |

Figure 1-2: Common switch configurations

Another type of classification of switches is whether they are normally open (NO) or normally closed, (NC). They refer to the state of the switch prior to being actuated. The contacts of a normally open switch are not engaged before being activated, e.g. a doorbell switch; while the contacts of a normally closed switch are engaged, e.g. an emergency stop switch.

Figure 1-3, schematically illustrates another popular type of mechatronic switch called a relay. A relay is an electromechanical switch which contains a coil. When the coil is energized it generates a magnetic force which causes an internal switch to latch. Relays are supplied in various configurations such as SPST, SPDT, and DPST.

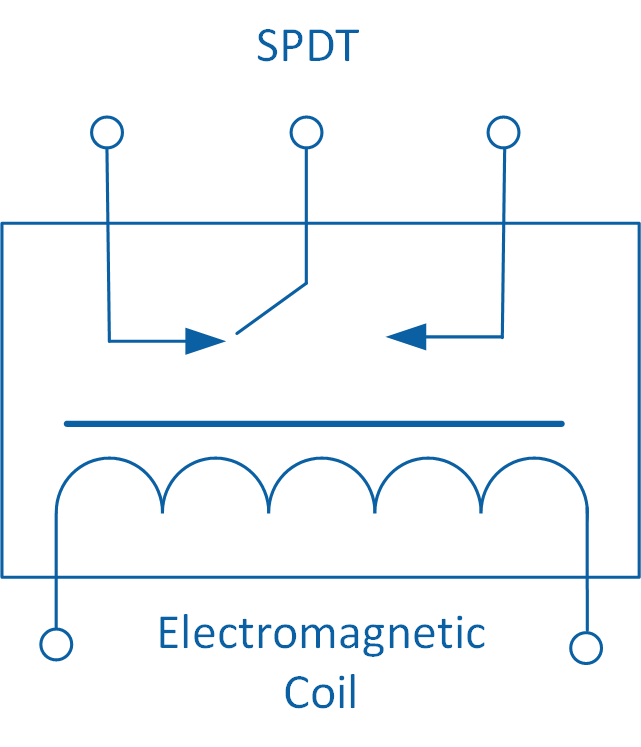


Figure 1-3: Schematic of an SPDT relay

Unlike mechanical and electromechanical switches, the optical switch does not contain moving components. They are classified as being either reflective or transmissive. As shown schematically in Figure 1-4, a reflective optical switch uses a reflective surface to bounce an Infrared LED beam onto its phototransistor. In contrast, in a transmissive optical switch, the Infrared LED shines directly onto the phototransistor. In both types of switches, once the Infrared light is interrupted, switching action is initiated which stops the flow of current.

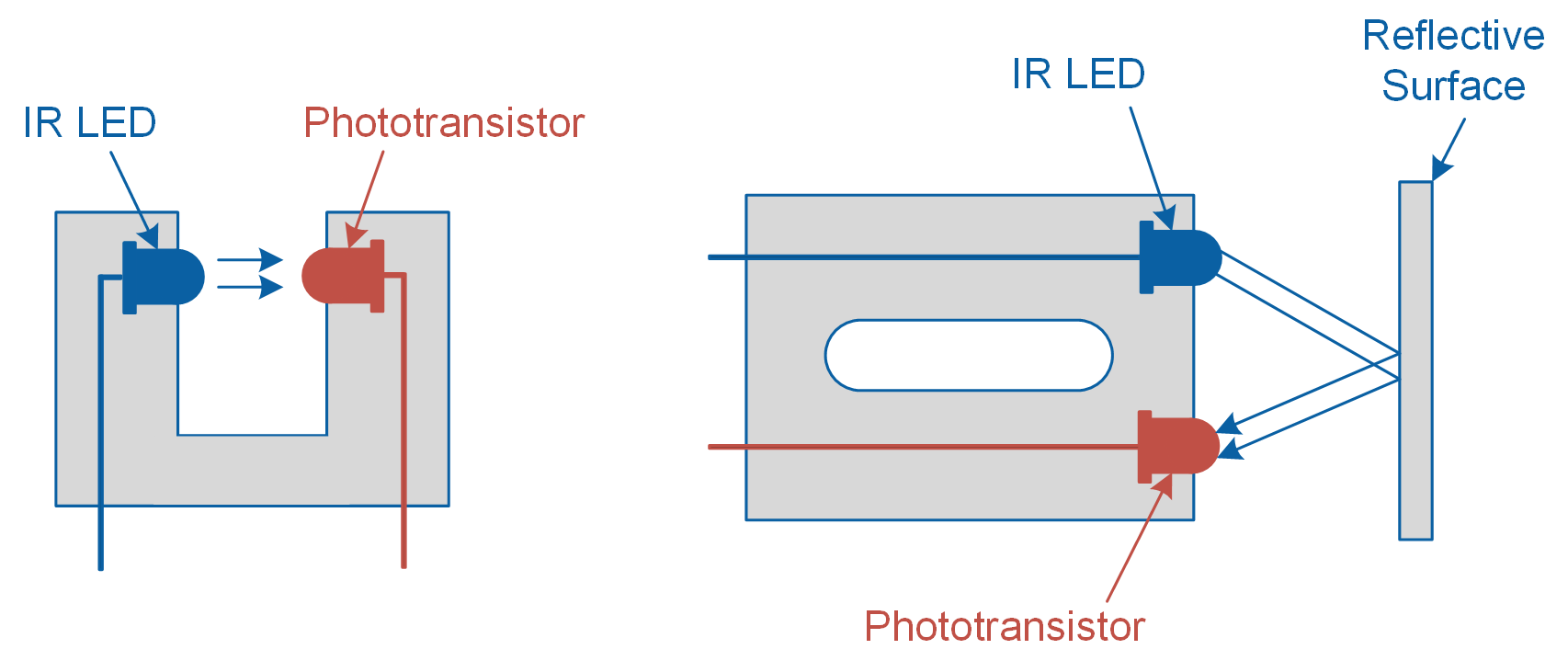


Figure 1-4: Schematic of a transmissive (left) and reflective (right) photo switch

#### Switch Bouncing

All switches suffer from a phenomenon called “switch bouncing”, which refers to the back and forth bouncing between ON and OFF states prior to settling on a final state. Bouncing in mechanical switches occurs because of the flexible components used inside the switch which physically bounce until a secure mechanical contact is made. Figure 1-5 shows the result of switch bouncing in a snap action switch. In this example, the switch is initially *high*. When the switch is depressed, the output bounces between *high* and *low* states for approximately 2,000 microseconds before it settles on a low state. Switch bouncing is an undesired characteristic, since a data acquisition system may think that you’ve actuated the switch button multiple times. Such spurious on/off actions may cause other processes to trigger unwantedly. To counter bouncing, a signal conditioning technique, called switch debouncing, is implemented.

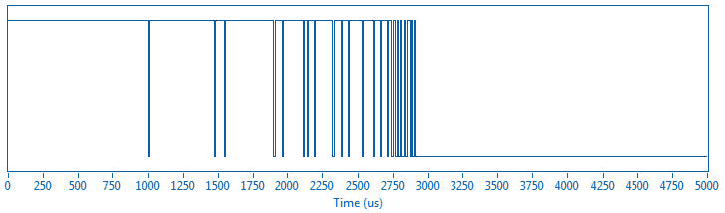


Figure 1-5: Switch bouncing when pressing a snap action switch

#### Debouncing Techniques

Switch debouncing can be implemented using both hardware and software. Figure 1-6 shows a digital debouncing circuit for a single-pole, double-throw switch. This circuit uses two cross-coupled NAND gates that create a simple Set-Reset (SR) latch. An alternative (analog) debouncing circuit is shown in Figure 1-7. This circuit, known as an RC debouncer, uses a capacitor and resistor to filter out rapid changes in the switch output.

|  |  |
| --- | --- |
| C:\Users\amolki\Desktop\2.png  Figure 6: SR debouncer | C:\Users\amolki\Desktop\1.png  Figure 7: RC debouncer |

Alternatively, debouncing can be done using software. A simple method involves using a timer to sample output of a switch and look for n sequential stable readings. If the switch signal consistently remains low (or high) during each of the sequential samples, then the switch is considered to be stable. If a bounce is detected in one of the samples, the timer is reset, and the sequential samples are retaken until n stable samples are read. This technique is illustrated in Figure 1-8, where the output of a switch is sampled at 10 ms intervals and when 2 consecutive stable reading is encountered the debounced switch turns from low to high.

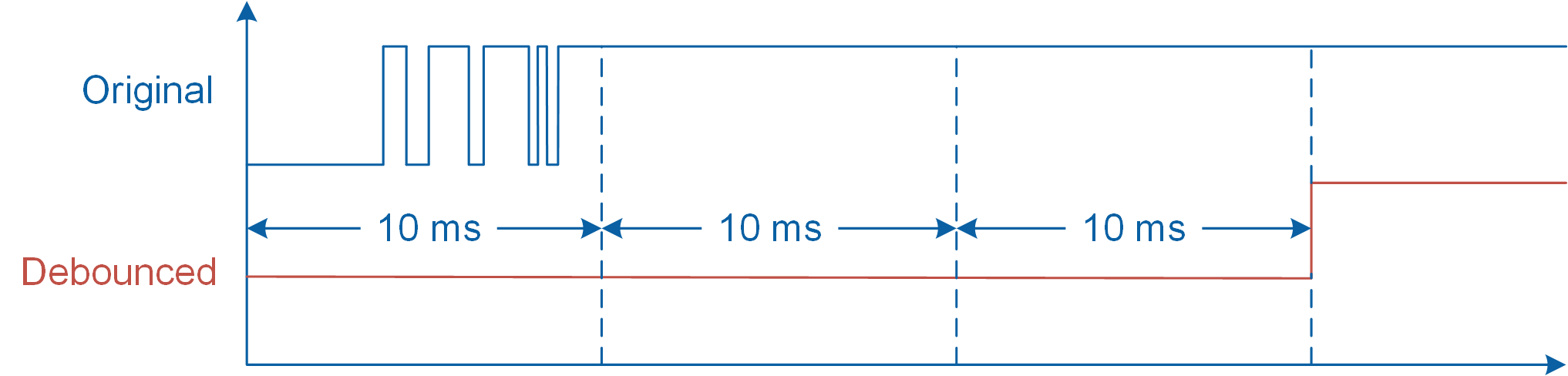


Figure 1-8: Software switch debouncing

### 1.2 Implement

The Virtual Instrument (VI) used to examine and debounce the snap action switch is shown in Figure 1-9.

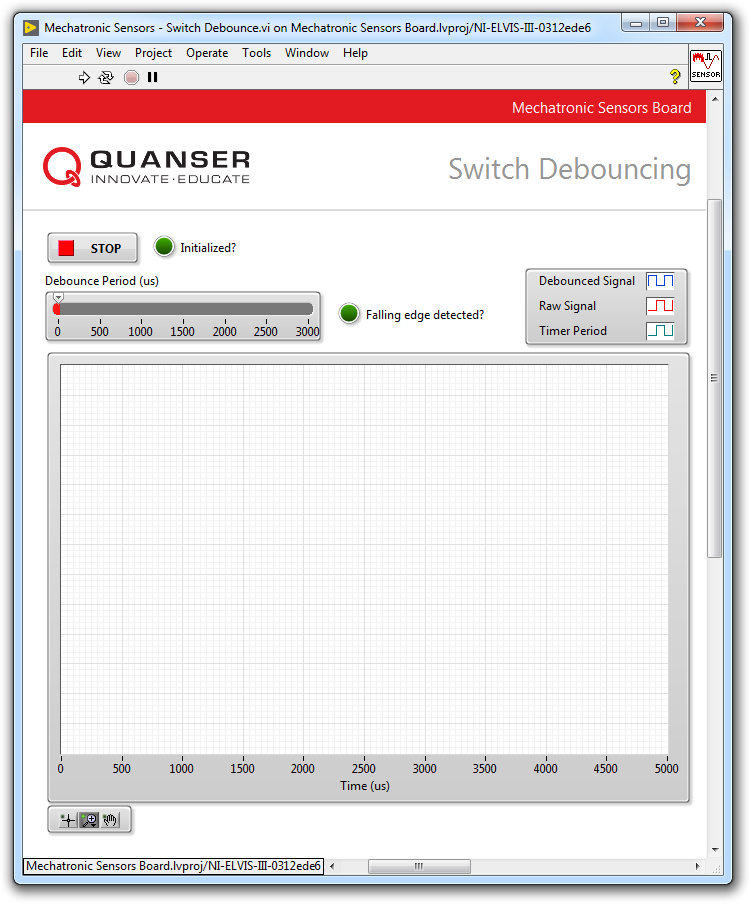


Figure 1-9: VI for examining and debouncing the snap action switch

#### Observe Switch Bouncing

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Switch Debounce.vi**
3. Run the VI.
4. Wait for the **Initialized?** LED indicator to turn on.
5. Click the snap action switch until the **Falling edge detected?** LED flashes momentarily. This indicates that a sample data containing a falling edge was successfully acquired. Once acquired, the results will be displayed in the waveform graph. Since by default the debounce period is set to 0 microseconds, the acquired switch data will not be debounced by the VI. Therefore, the graph will display the same results for **Raw Signal** and **Debounced Signal**.
6. Take screenshots of your results.
7. With the debounce period still set to 0 microseconds, press the snap action switch several times and observe switch bouncing. Is switch bouncing different each time you press the switch? Take screenshots of your results.
8. Continue to the next section.

#### Switch Debouncing

1. Set the **Debounce Period (us)** slider to **2000**.
2. Click the snap action switch until the **Falling edge detected?** LED flashes momentarily. Examine the response of the timer, raw signal, and the debounced signal. Take a screenshot of your results.  
     
   *Note:* The Mechatronic Sensors Board implements software debouncing. The VI implements a timer which has a period equal to the *Debounce Period (us)* slider value. When an edge change is detected in the raw signal, the software starts the timer and samples the raw signal again when the timer expires. If the measured raw signal value is different from the signal value prior to starting the timer (e.g. the signal value changed from high to low), the VI assumes that the switch has stopped bouncing and the debounced signal is generated. Otherwise, if the output of the switch value is still the same as prior to starting the timer (e.g. the signal value remains high), the timer is started again and the raw signal is sampled accordingly until the debounced signal is generated.
3. Set the **Debounce Period (us)** slider to **1000**, **500**, and **100**. Examine the response of the timer, raw signal, and the debounced signal. Take screenshots of your results.
4. Press the **Stop** button.

### 1.3 Analyze

1-1 Attach screenshots of the switch bouncing effect that you observed when the debounce period was set to 0 microseconds. Is the switch bouncing behavior different each time you pressed the switch?

1-2 Was the VI successful in debouncing the output of the switch when the debounce period was set to 2000 microseconds? Attach a screenshot of your results. Comment on your results.

1-3 Was the VI successful in debouncing the output of the switch when the debounce period was set to 1000, 500, and 100 microseconds? How do different debounce period values affect the debounced signal? Attach a screenshot of your results.

## Section 2: Capacitive Touch Sensor

### 2.1 Theory and Background

#### What is a Capacitive Touch Sensor?

A capacitance touch sensor measures changes in electrical capacitance to detect the presence of a finger that is either placed directly on the sensor or in its proximity. It is used in numerous applications ranging from consumer electronic touchpads to fluid-level monitoring systems in the automotive industry.

#### Self-Capacitance Sensing

A variety of methods are used to measure the capacitance between two points. A common method is called self-capacitance sensing. A capacitive touch sensor that implements this method, incorporates a single pin and measures the capacitance between that pin and a ground reference. This type of sensor is primarily used when single-touch (e.g. button) or sliding action is required.

Figure 2-1 illustrates the operating principle of a capacitive touch sensor that uses self-capacitance sensing. The sensor assembly typically consists of a conductive sensor pad, a ground hatch, and an overlay, all of which are mounted on a printed circuit board (PCB). The overlay is a protective dielectric layer that covers the ground hatch and sensor pad and prevents direct finger contact. Overlay material with a dielectric constant ranging between r = 2.0 and 8.0 are recommended, which include acrylic, PET film, and ceramic.

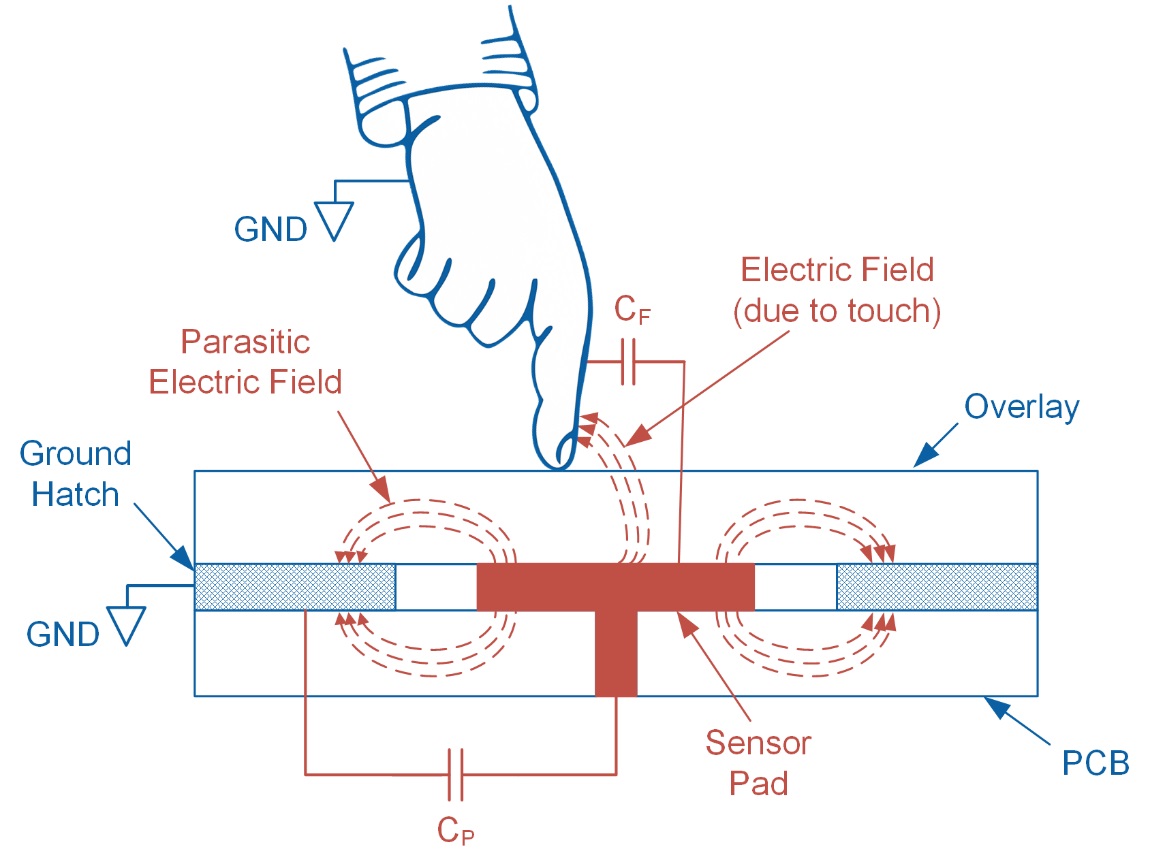


Figure 2-1: The operating principle of a self-capacitance touch sensor

In the absence of physical contact with the sensor, the parasitic electric field between the sensor pad and ground hatch results in the parasitic capacitance *CP*. This parasitic capacitance is a simplification of the distributed capacitance that includes the effects of the sensor pad, the overlay, and the PCB. In this case, the overall measured sensor capacitance, *CS*, is equal to *CP*. However, when a finger makes contact with the sensor pad, the two form an equivalent parallel plate capacitor. The resulting finger capacitance (*CF)* causes the overall sensor capacitance to become:

Equation 2-1

*CF* can be calculated using Equation 2-2:

Equation 2-2

where *0* is the free space permittivity, *r* is the dielectric constant of the overlay, *A* is the contact area between the finger and the sensor pad, and *D* is the overlay thickness.

#### Measuring Capacitance

The Quanser Mechatronic Sensors Board uses a Cypress CY8CMBR3116-LQXI capacitive sensor with an I2C interface. The sensor integrates a circular touch pad that is divided into ten sectors labeled **0** to **9,** as well as two individual buttons labeled **L** and **R**. The sensor converts capacitance into a digital count ranging between 0 and 255. An output of 0 indicates zero finger capacitance, while an output of 255 indicates the maximum measurable capacitance typically induced when a finger makes contact with the touchpad. The sensitivity of the sensor is set to 50 counts/0.1 pF.

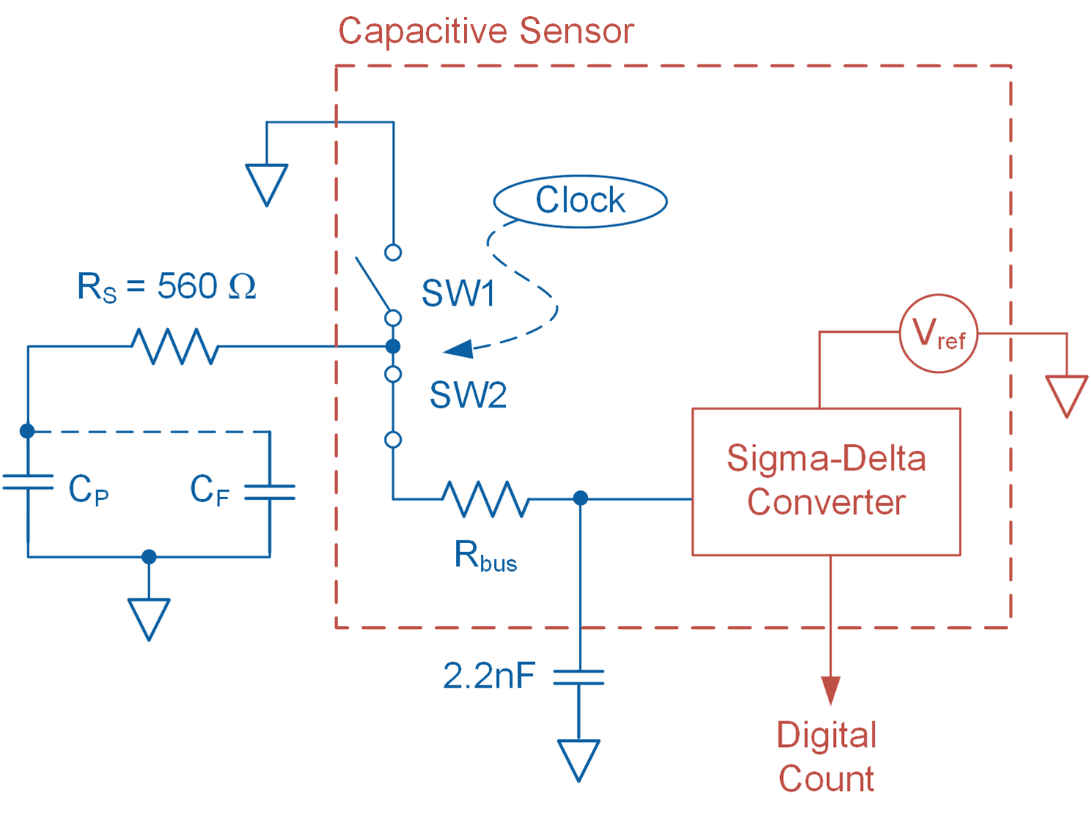


Figure 2-2: Schematic diagram of the capacitive sensor

A schematic diagram of the sensor and signal conditioning circuit is shown in Figure 2-2. The external resistor *RS* and internal resistor *Rbus* help reduce noise. Internally, a high-speed clock continuously switches *SW1* and *SW2*. This switching action results in the formation of an equivalent resistance (*Req*). The sensor then uses a Sigma-Delta modulator to convert the equivalent current (*ICS*) measured through *Req* into a digital count. Details of the equivalent resistor model are shown in Figure 2-3. In practice, when a finger is placed on the sensor pad, the presence of *CF* causes the value of *CS* to increase. This action causes *Req* to decrease, therefore increasing *ICS*, and resulting in a higher output count.

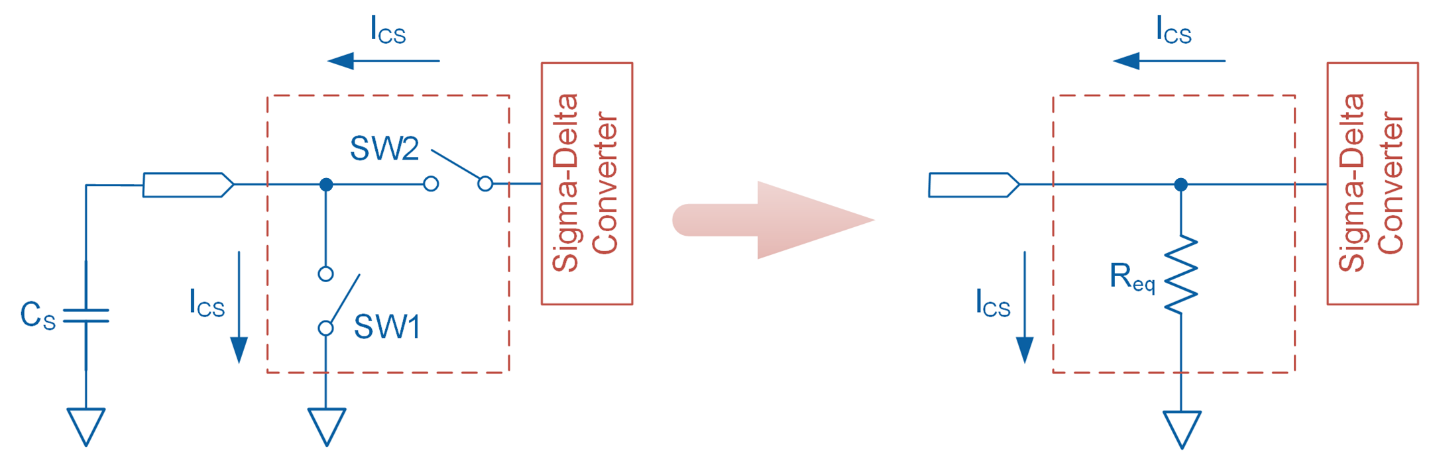


Figure 2-3: Equivalent resistor model

#### Advantages of Capacitive Touch Sensors

Capacitive touch sensors have several advantages over mechanical switches. Some of these advantages include:

* Since touch sensors lack moving components, they are less prone wear over time and therefore do not exhibit the type of mechanical failure that is common in mechanical switches.
* Most mechanical switches have cases that are not sealed, therefore they are more prone to environmental degradation due to moisture and dust build-up.
* Dust build-up over time may cause mechanical switches to become sticky, requiring a larger actuation force to operate.
* The absence of mechanically switching components means that capacitive touch sensors do not suffer from switch bouncing.

### 2.2 Implement

The Virtual Instrument (VI) used to examine the behavior of the capacitive touch sensor is shown in Figure 2-4.

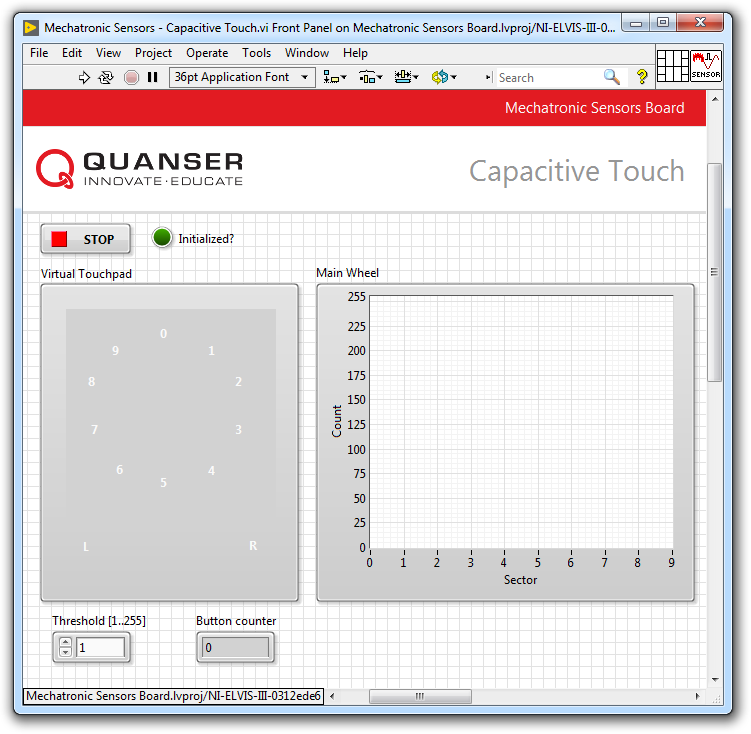


Figure 2-4: VI for examining the behavior of the capacitive touch sensor

#### Observe Sensor Behavior

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Capacitive Touch.vi**
3. Run the VI.
4. Wait for the **Initialized?** LED indicator to turn on.
5. The **Main Wheel** waveform graph displays the output of each of the ten sectors of the circular touch pad. As noted earlier, the sensor outputs a digital count that ranges between 0 and 255.  
     
   Slowly move your finger toward sector **0** and observe the response of the sensor using the **Main Wheel** waveform graph. The sensor will initially output a count value of 0, but will gradually increase to a value of 255 as your finger almost makes full contact with the touch pad. Take screenshots of your results.
6. Use the **Threshold [1..255]** numeric control to control the behavior of the virtual touch pad in the VI. The VI is programmed to compare the output of the capacitive touch sensor with the threshold: if the output of the sensor is equal to or greater than the threshold, corresponding individual sectors and touch buttons turn **red** indicating contact. Otherwise, they remain **black** indicating no contact.  
     
   With the threshold set to its default value of 1, slowly move one finger toward the **right** touch button and examine the behavior of the **virtual** touch button. Estimate the physical distance at which the virtual button turns **red**.
7. Change the threshold value to 255 and again estimate the physical distance at which the virtual button turns **red**.
8. Press the **Stop** button.

### Analyze

2-1 Describe the behavior of the sensor as you moved your finger close to the touch pad in Step 5. Attach screenshots of your results.

2-2 With the threshold set to 1 and 255 respectively, at what physical distances did the virtual button turn red (Step 6 and 7)? Describe the effect of the threshold setting on the behavior of the virtual touch pad. Is a higher threshold value desirable?

## Section 3: Design Considerations

3-1 The capacitance sensor implemented in the Quanser Mechatronic Sensors Board can be used to virtually perform single-button and scroll actions. To show this, the VI used in this lab implements a button counter. Each time the right button is tapped, the **Button counter** numeric indicator is incremented by a value of 1 and decremented by the same value if the left button is tapped. Run the VI and examine this functionality.

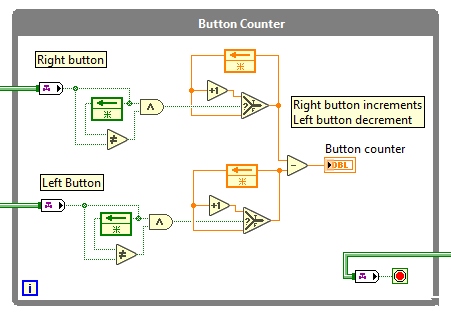


Figure 3-1: LabVIEW code that performs button counting

A screenshot of the block diagram that performs the counting is shown in Figure 3-1. Wire channels are used to pass Boolean values representing the status of the right and left buttons to the While Loop (a value of 0 indicates no contact, and a value of 1 indicates contact). Present the pseudo-code equivalent of this LabVIEW code.

3-2 Present strategies on how you would implement a LabVIEW VI that determines the direction of scrolling based on the output of the touchpad’s individual sectors.

# Lab 7: Inertial Measurement



Figure 0: Precise attitude measurement is instrumental to the control of unmanned aerial vehicles

This lab explores acceleration, rotation, and magnetic field measurements using an Inertial Measurement Unit (IMU) sensor. In particular, roll, pitch, and yaw measurements will be determined using the output of the accelerometer and gyroscope sensors. Also, earth’s magnetic field direction will be approximated using the magnetometer.

## Learning Objectives

After completing this lab, you should be able to complete the following activities:

1. Identify components of an Inertial Measurement Unit
2. Estimate roll and pitch using an accelerometer
3. Estimate yaw by integrating the output of a gyroscope
4. Analyze gyroscopic drift
5. Approximate earth’s magnetic field direction using a magnetometer
6. Determine magnetometer offset
7. Observe Euler geometric singularities when estimating roll and pitch

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Mechatronic Sensors Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-mechatronic-sensors-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III * <http://www.ni.com/academic/download> * View Tutorials * http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Record raw accelerometer data
* Examine and calculate roll and pitch angles using accelerometer output
* Examine yaw angle polarity
* Measure maximum and minimum magnetometer outputs
* Determine magnetic North
* Estimate gyro drift
* Estimate magnetometer offset
* Observe and estimate Euler singularities when determining roll and pitch

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Measuring Pose and Magnetic Field using an IMU

### 1.1 Theory and Background

#### What is an IMU?

An internal measurement unit, or IMU, is a sensor that measures a body’s acceleration, angular velocity, as well as magnetic field. Using an IMU, it is possible to determine the attitude of a body. An attitude of a body comprises of its pitch, roll, and yaw. Figure 1-1 illustrates that attitude of an aircraft. In this example, pitch is the degree of tilt in the aircraft’s nose (rotation about the pitch axis), roll is the degree of rotation about the longitudinal axis of the aircraft’s fuselage (rotation about the roll axis), and yaw refers to the aircraft’s heading (rotation about the yaw axis). It should be noted that different applications may use different conventions for the orientation of the axes and polarity of rotation. IMUs are used also used in a wide range consumer goods such as smartphones, tablets, motion-based game controllers, and many wearable technologies.

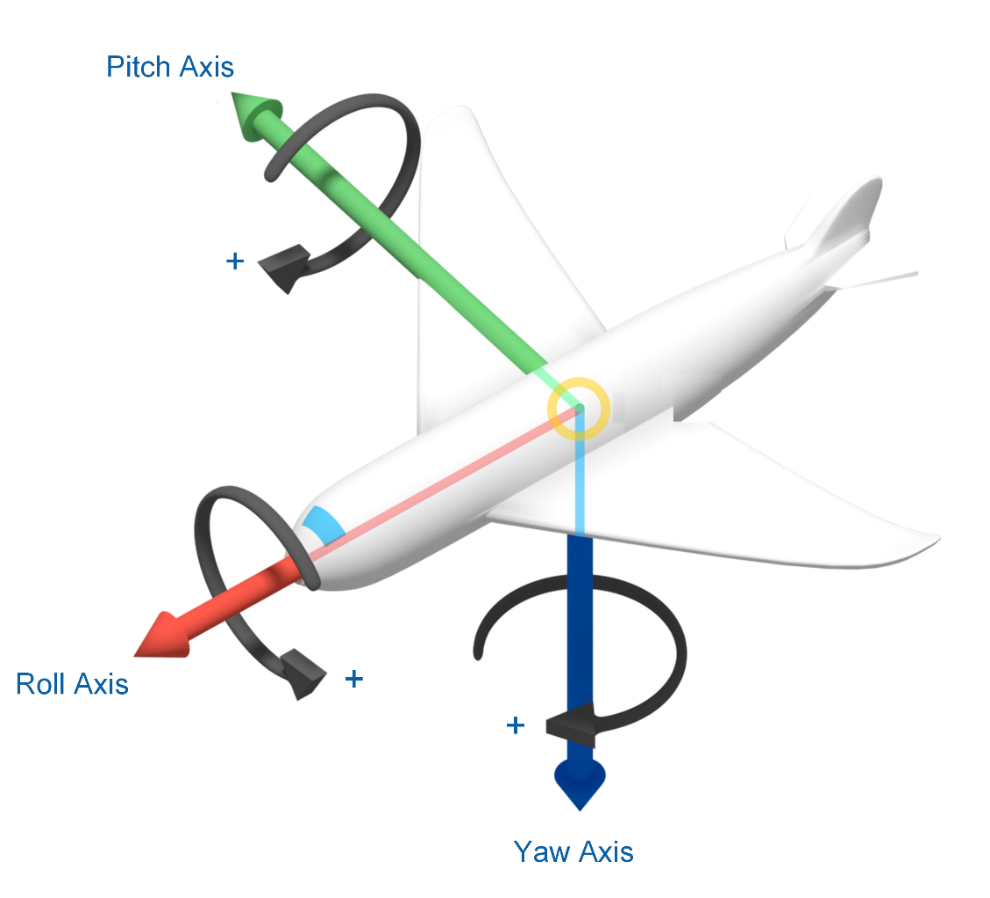


Figure 1-1: An aircraft’s attitude (roll, pitch, and yaw)

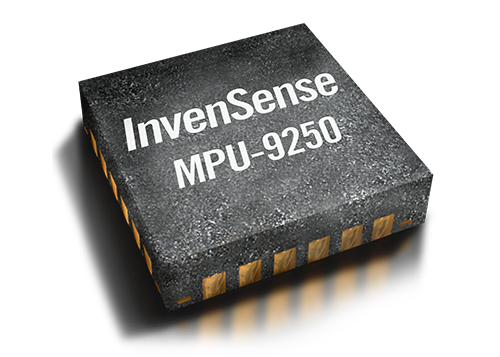


Figure 1-2: Image of a TDK InvenSense MPU-9250

The Quanser Mechatronic Sensors Application Board incorporates a 9-axis TDK InvenSense MPU-9250 IMU sensor (Figure 1-2). The sensor uses three independent vibratory MEMS rate gyroscopes to detect motion about the X-, Y-, and Z-axes. When the gyros are rotated about any of the sensor’s axes, the Coriolis Effect causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. The sensor includes an Analog-to-Digital converter to output digital values of the measured angular velocities up to 2,000 deg/sec.

The sensor also incorporates a 3-axis accelerometer that measures acceleration in units of *g*. The accelerometers use separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. Figure 1-3 illustrates the orientation of axes of sensitivity and polarity of rotation for the accelerometer and gyroscope. The accelerometer outputs a value of +1 g for any axis that is vertically pointing upward, and outputs a value of -1 g when the axis is vertically pointing downward. For any arbitrary inclination, the accelerometer outputs the magnitude of the X, Y, and Z acceleration vectors. For example, when the sensor is placed on a flat horizontal surface with its Z-axis facing upward, the sensor outputs AX = 0 g, AY = 0 g, and AZ= +1 g. Similarly, when the sensor’s X-axis is facing vertically downward, the sensor outputs AX = -1 g, AY = 0 g, and AZ = 0 g.

The sensors also include a 3-axis magnetometer capable of measuring up to ±4,800 T. Communication with the sensor is done using the I2C protocol. Figure 1-4 illustrates the orientation of axes of sensitivity for the magnetometer.

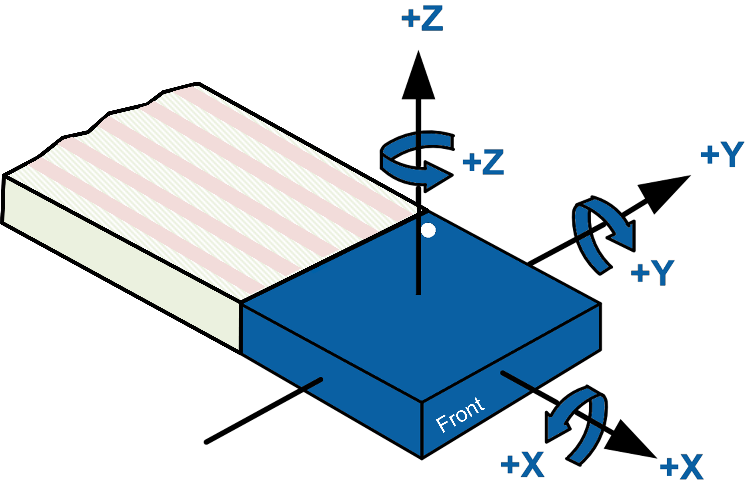


Figure 1-3: Orientation of axes of sensitivity and polarity of rotation for the accelerometer and gyroscope

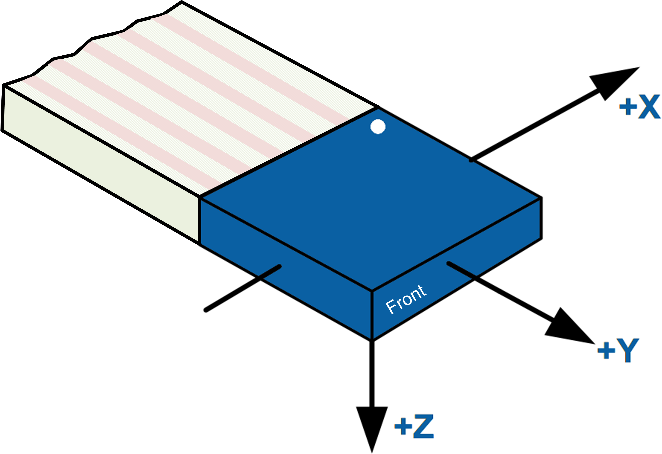


Figure 1-4: Orientation of axes of sensitivity for the magnetometer

#### Estimating Pitch and Roll using an Accelerometer

Measurements from a 3-axis accelerometer can be used to determine the pitch and roll of a body. Figure 1-5 illustrates how to determine the roll angle (ϕ) using the output of an accelerometer and basic trigonometric functions. As shown in the figure, when the accelerometer is held horizontally with its Z-axis pointing upward, the accelerometer outputs a value of AZ = +1 g. In this orientation, the magnitude of the X and Y gravitational acceleration vectors equal AX = AY = 0 g, resulting in a summed magnitude of Asum = +1 g since the sensor is always subjected to gravitational force.

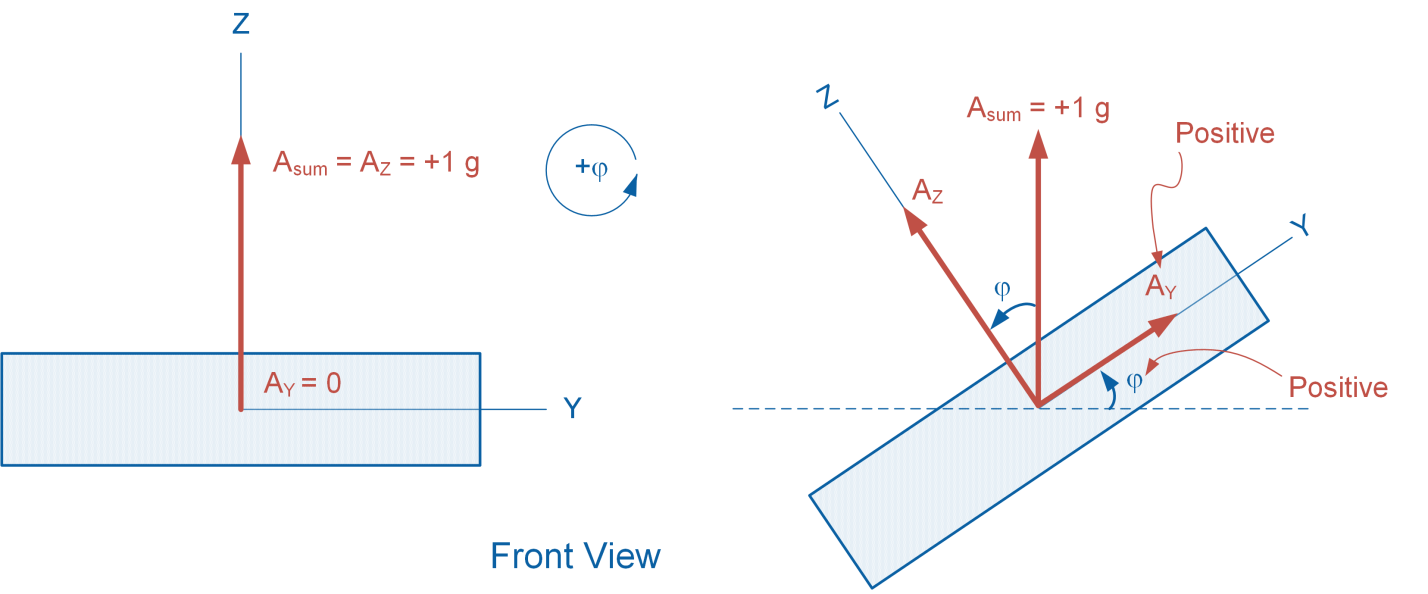


Figure 1-5: Determining roll angle using accelerometer output

However, when the accelerometer is rolled about its X-axis by angle ϕ, the magnitudes of AY and AZ will vary depending on the roll angle, while the magnitude of AX remains 0. Equation 1-1 shows the relationship between the roll angle and AY and AZ acceleration components:

Equation 1-1

*Note:* The *atan2* function computes the arctangent for angles in any of the four quadrants of a 2-dimensional plane, whereas the *atan* function computes the arctangent in only two quadrants. It is defined as follows:

Figure 1-6 illustrates how to determine the pitch (ϕ) angle using the output of an accelerometer and basic trigonometric functions. Similar to the previous example, when the accelerometer is held horizontally with its Z-axis facing upward, the accelerometer outputs a value of AZ = +1 g. In this orientation, the magnitude of the X and Y gravitational acceleration vectors equal AX = AY = 0 g, resulting in a summed magnitude of Asum = +1 g since the sensor is only subjected to gravitational force.

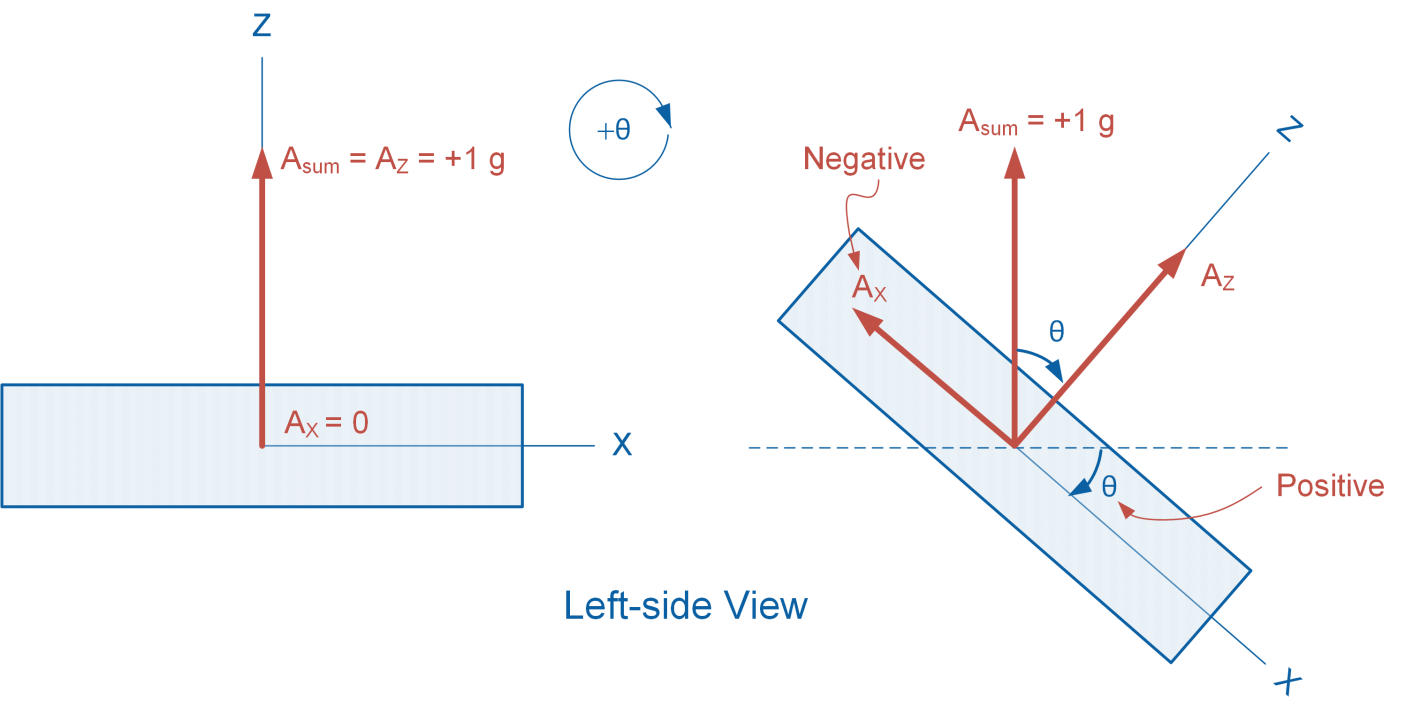


Figure 1-6: Determining pitch angle using accelerometer output

However, when the accelerometer is pitched about its Y-axis by angle θ, the magnitudes of AX and AZ will vary depending on the pitch angle, while the magnitude of AY remains 0. Equation 1-2 shows the relationship between the pitch angle and AX and AZ acceleration components:

Equation 1-2

#### Estimating Yaw using a Gyroscope

Unlike roll and pitch, yaw cannot be deduced from accelerometer measurements. This is because as a body yaws about the Z-axis, an accelerometer does not register changes in AX, AY, AZ. To counter this problem, the yaw of a body can be estimated using a gyroscope. A gyroscope measures the rate of angular change, or angular velocity. As such, in order to determine angular position, or yaw, one must integrate angular velocity about the Z axis with respect to time as shown in Equation 1-3:

Equation 1-3

#### Determining Earth’s Magnetic Heading using a Magnetometer

A magnetometer is a sensor that measures the strength of a magnetic field. The 3-axis magnetometer incorporated in the application board uses highly sensitive Hall sensor technology for detecting terrestrial magnetism along the X-, Y-, and Z-axes. With the advent of MEMS technology, magnetometers have been miniaturized to the extent that they can be incorporated in integrated circuits as micro-size digital compasses. Magnetometers are widely used in military and aerospace applications for positioning and establishing directional references.

It is important to note the existence of a discrepancy between Earth's magnetic north (the direction the north end of a compass needle points, corresponding to the direction of the Earth's magnetic field lines) and Earth's true north (the direction along a meridian towards the geographic North Pole). This discrepancy, called magnetic declination, is defined as the angle on the horizontal plane between magnetic north and true North. Magnetic declination varies both with time and geographical location.

Earth’s magnetic heading can be estimated by using the HX or HY components of a magnetometer output. This can be done by holding the magnetometer horizontally such that its Z-axis is pointed vertically downward. While monitoring the HX output, rotate the sensor on the horizontal plane. When the sensor outputs the maximum value of HX, the X-axis is pointing toward magnetic north.

### 1.2 Implement

The Virtual Instrument (VI) used to examine the behavior of the capacitive touch sensor is shown in Figure 1-7.

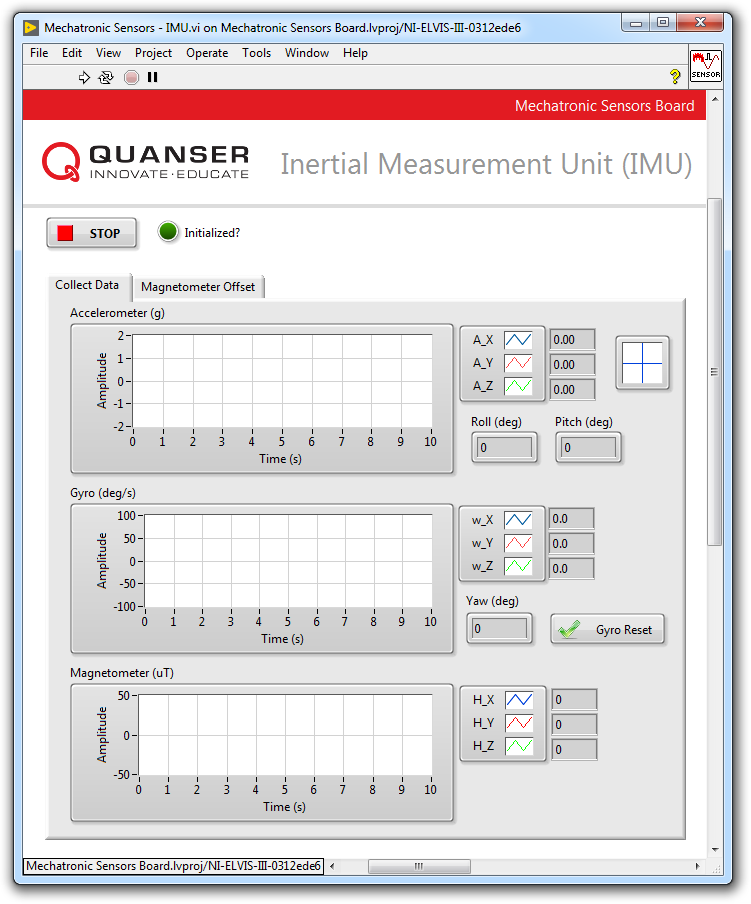


Figure 1-7: VI used for measuring the output of the IMU

#### Raw Accelerometer Output

1. Open **Mechatronic Sensors Board.lvproj**
2. From the **Project Explorer** window, open **Mechatronic Sensors – Pressure.vi**
3. Click on the **Collect Data** tab.
4. Run the VI.
5. Wait for the **Initialized?** LED indicator to turn on.
6. The direction of the axes of the accelerometer is silk screened on the IMU sensor. Hold the IMU sensor such that the accelerometer X-axis is pointing vertically upward. Record the value displayed in the **A\_X** numeric indicator in Table 1-1.
7. Hold the IMU sensor such that the accelerometer X-axis is pointing vertically downward. Record the value displayed in the **A\_X** numeric indicator in Table 1-1.
8. Repeat steps 7 and 8 for the accelerometer Y and Z axes and record your results in Table 1-1.

Table1-1: Raw accelerometer output

|  |  |
| --- | --- |
| Accelerometer Axis Direction | Raw Output (g) |
| X upward |  |
| X downward |  |
| Y upward |  |
| Y downward |  |
| Z upward |  |
| Z downward |  |

#### Examine Roll Angle

1. The VI implements Equation 1-1 to estimate the roll of the sensor. Hold the sensor in the horizontal plane with the accelerometer Z-axis pointing vertically upward. The **Roll (deg)** numeric indicator must output a value of **0**. Slowly roll the sensor sideways (about the accelerometer’s X-axis) and examine changes in the **Roll (deg)** numeric indicator. Does the indicated roll angle polarity match the diagram shown in Figure 1-3?
2. Holding the sensor at a +45 degree roll angle. Make note of the **A\_Y** and **A\_Z** accelerometer outputs and use Equation 1-1 to estimate the roll angle. Does your calculation agree with the roll angle displayed in the VI?

#### Examine Pitch Angle

1. The VI implements Equation 1-2 to estimate the pitch of the sensor. Hold the sensor in the horizontal plane with the accelerometer Z-axis pointing vertically upward. The **Pitch (deg)** numeric indicator must output a value of **0**. Slowly pitch the sensor downward (about the accelerometer’s Y-axis) and examine changes in the **Pitch (deg)** numeric indicator. Does the indicated pitch angle polarity match the diagram shown in Figure 1-3?
2. Pitch the sensor downward holding it at a +45 degree angle. Make note of the **A\_X** and **A\_Z** accelerometer outputs and use Equation 1-2 to estimate the pitch angle. Does your calculation agree with the pitch angle displayed in the VI?

#### Examine Yaw Angle

1. The VI estimates the yaw angle by integrating the Z-axis component of the gyro output. Hold the sensor in the horizontal plane and click the **Gyro Reset** button to zero the yaw angle.
2. While keeping the sensor level in the horizontal plane, slowly yaw it sideways. Examine changes in the **Yaw (deg)** numeric indicator. Does the indicated yaw angle polarity match the diagram shown in Figure 1-3?

#### Find Magnetic North

1. Determine magnetic North by holding the sensor in the horizontal plane with the magnetometer’s Z-axis pointing downward. Slowly rotate the sensor about the Z-axis and examine the **H\_Y** numeric indicator. Determine the maximum and minimum values indicated by the **H\_Y** numeric indicator. The direction of the Y-axis at which maximum **H\_Y** value occurs points toward magnetic North.
2. Validate your finding with a device that indicates magnetic North (e.g. a compass).
3. Stop the VI.

### 1.3 Analyze

1-1 Present the raw accelerometer data you recorded in Table 1-1.

1-2 When examining roll angles in Step 9, did the roll angle polarity match the diagram shown in Figure 1-3?

1-3 What are the A\_Y and A\_Z accelerometer outputs noted in Step 10? Using these values and Equation 1-1, calculate the sensor's roll angle. Show your calculations. Did your calculation agree with the roll angle displayed in the VI?

1-4 When examining pitch angles in Step 11, did the polarity match the diagram shown in Figure 1-3?

1-5 What are the A\_X and A\_Z accelerometer outputs noted in Step 12? Using these values and Equation 1-2, calculate the sensor's pitch angle. Show your calculations. Did your calculation agree with the pitch angle displayed in the VI?

1-6 When examining yaw angles in Step 14, did the polarity match the diagram shown in Figure 1-3?

1-7 What are the maximum and minimum H\_Y values recorded in Step 16? Did you successfully validate your magnetic North finding?

## Section 2: Design Considerations

2-1 As explained earlier in Section 1.1, angular position, or yaw, can be deduced by integrating the output of the gyroscope about the Z axis with respect to time. This process, however, leads to a common problem known as gyroscopic drift. Gyroscopic drift is unavoidable. Even expensive high-end models will have significant drift. The drift occurs because the process of integration not only integrates angular rate, but also integrates any bias and noise that is present in the sensor. Gyroscopic drift is more evident when the sensor is at rest.

Your task is to observe and quantify the drift present in the Sensor board’s gyroscope. To do this, run the VI, hold the sensor in the horizontal plane with the accelerometer’s Z axis pointing in the upward direction. Click the **Gyro Reset** button to zero the output of the **Yaw (deg)** numeric indicator. Using a stopwatch, estimate the average drift that you observe in terms of degrees per minute.

2-2 A sensor that outputs a fixed value when it should be measuring zero is said to exhibit a *zero offset error*. Magnetometers often exhibit zero offset errors along their axes of measurement. Two factors contribute to the error: hard-iron effects and soft-iron effects. Hard iron interferences are due to the sensor’s internal construction, which causes a constant additive value to the measured in addition to earth’s magnetic field. Soft iron interferences are caused by environmental conditions, e.g. the presence of nearby material which influences the measured magnetic field.

In practice, zero offset errors must be determined and corrected for when conducting an experiment. Your task is to quantify the zero offset error present in the magnetometer along its Y and Z axes. To do this, run the VI, click the **Magnetometer Offset** tab, and click the **Plot** button. Gently move the sensor within the magnetometer’s Y-Z plane. As you move the sensor, the waveform graph will plot the Y and Z components of the magnetometer. Once sufficient data has been plotted, you should expect to obtain a circular or elliptical pattern as shown in Figure 2-1. Identify the center of the ellipse and from there estimate the zero offset errors along the Y and Z axes. Take a screenshot of your results.

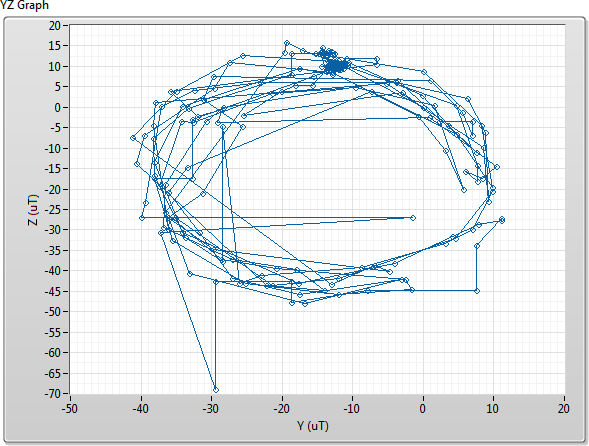


Figure 2-1: Sample magnetometer offset error along the Y and Z axes

2-3 Deducing roll and pitch angles by applying Equations 1-1 and 1-2 to the accelerometer output introduces unwarranted mathematical singularities. Known as Euler singularities, they manifest themselves as incorrect roll and pitch angle measurements when the sensor is posed in certain orientations.

Your task is to observe these singularities. Run the VI and wait for the **Initialized?** LED to turn on. Hold the IMU in the horizontal plane with the accelerometer Z-axis pointing upward such that the indicated roll and pitch angles are close to zero. Slowly pitch the IMU downward, while preventing any roll in the sensor. Observe how the indicated pitch value increases in the VI as the tilt increases. At the same time, closely observe the roll of the sensor. Since you are not rolling the sensor you would expect the VI to indicate a roll angle of close to 0 deg. However, as the pitch angle approaches +90 deg you will notice the roll angle very quickly increases. This unwarranted roll measurement is caused by the Euler singularity problem. Observe a similar singularity in the measured pitch as you roll the sensor sideways.

Using the raw accelerometer outputs (A\_X, A\_Y, and A\_Z) and Equations 1-1 and 1-2, explain how these singularities occur.