**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Semester:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Section: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

**EE-383**-**Instrumentation and Measurements**

**Experiment # 10**

**Switch Debouncing/Strain Gage/Thermistor**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4/**  **CLO3** | | **PLO4/ CLO4** | **PLO8/ CLO5** | **PLO9/ CLO6** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Teamwork** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Switch Debouncing**

# Why Use 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. Because of their construction, switches suffer from a phenomenon called “switch bouncing”.

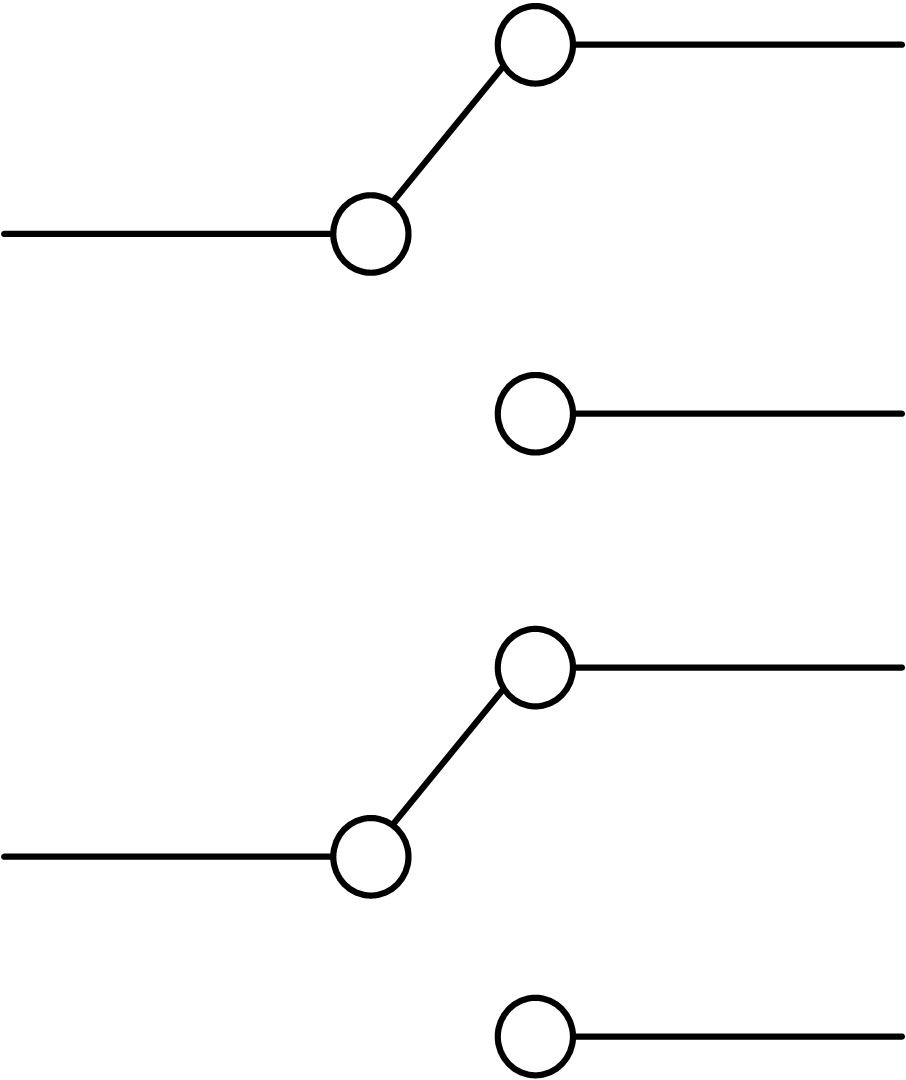
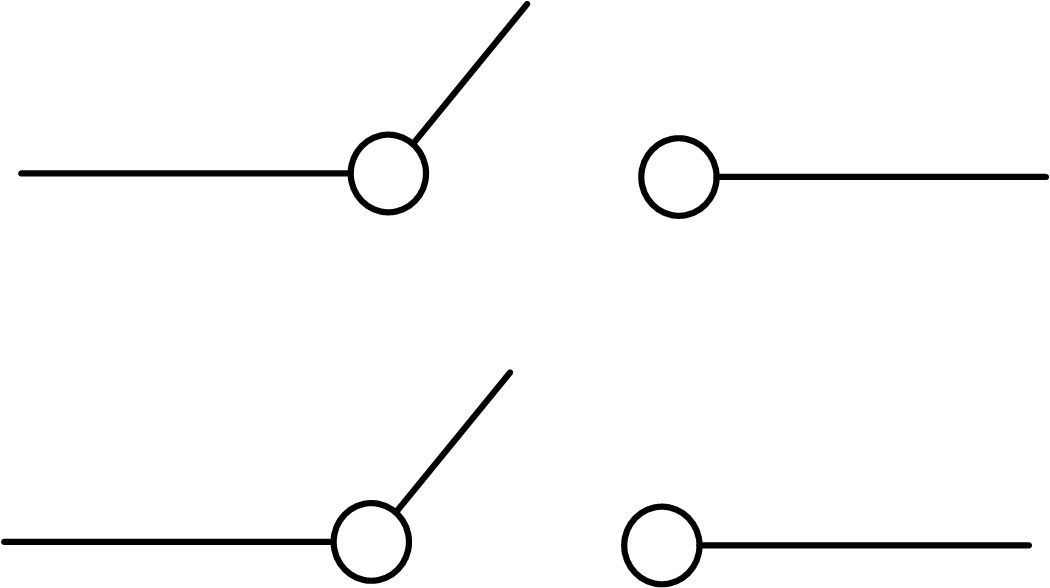
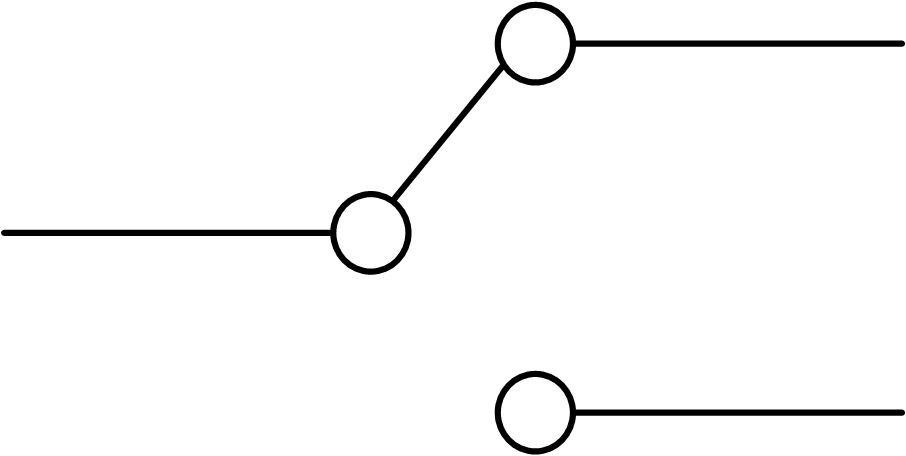
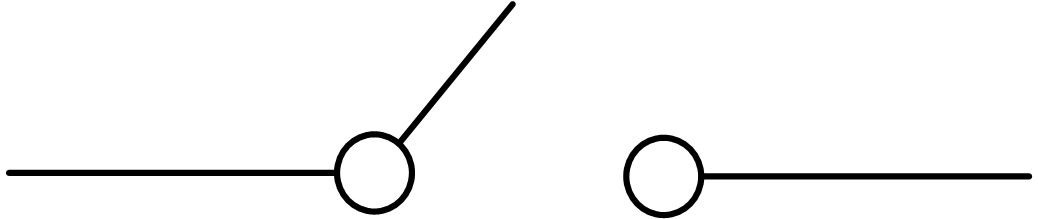
# Types of Switch

Switches can be classified into different types based on several factors such as method of actuation, type of operation, and construction. Figure 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.

1. snap action (b) slide (c) rocker (d) pushbutton (e) toggle Figure 1: Common types of mechanical switches (source: DigiKey).

As illustrated in Figure 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.

Figure 2: Common switch configurations.



* 1. single-pole, single-throw (SPST)
  2. single-pole, double-throw (SPDT)
  3. double-pole, single-throw (DPST)
  4. double-pole, double-throw (DPDT)

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 door bell switch; while the contacts of a normally closed switch are engaged, e.g. an emergency stop switch.

Figure 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.

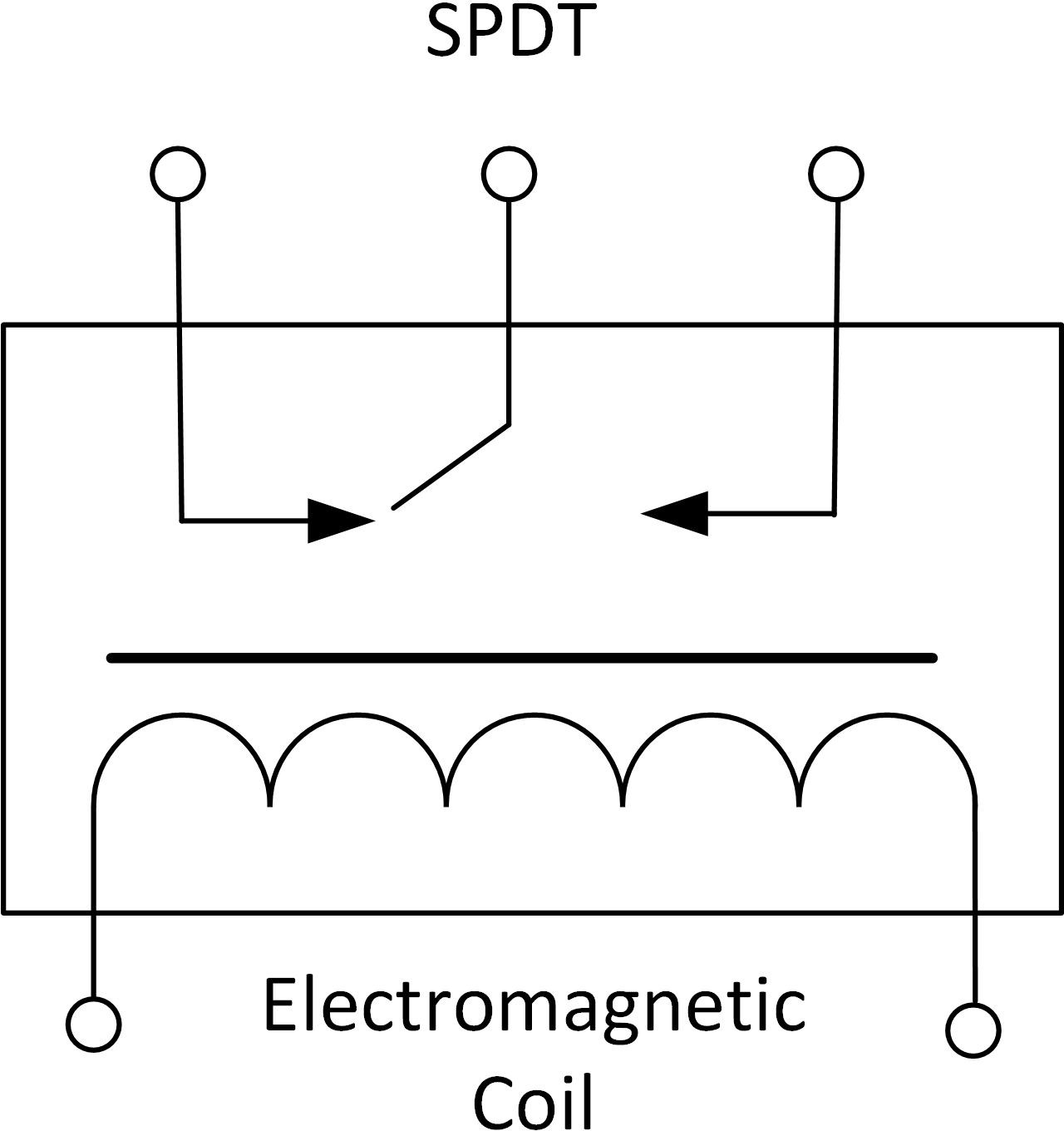


Figure 3: Schematic of a SPDT relay.

Unlike mechanical and electromechanical switches, optical switch does not contain moving components. They are classified as being either reflective or transmissive. As shown schematically in Figure 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.

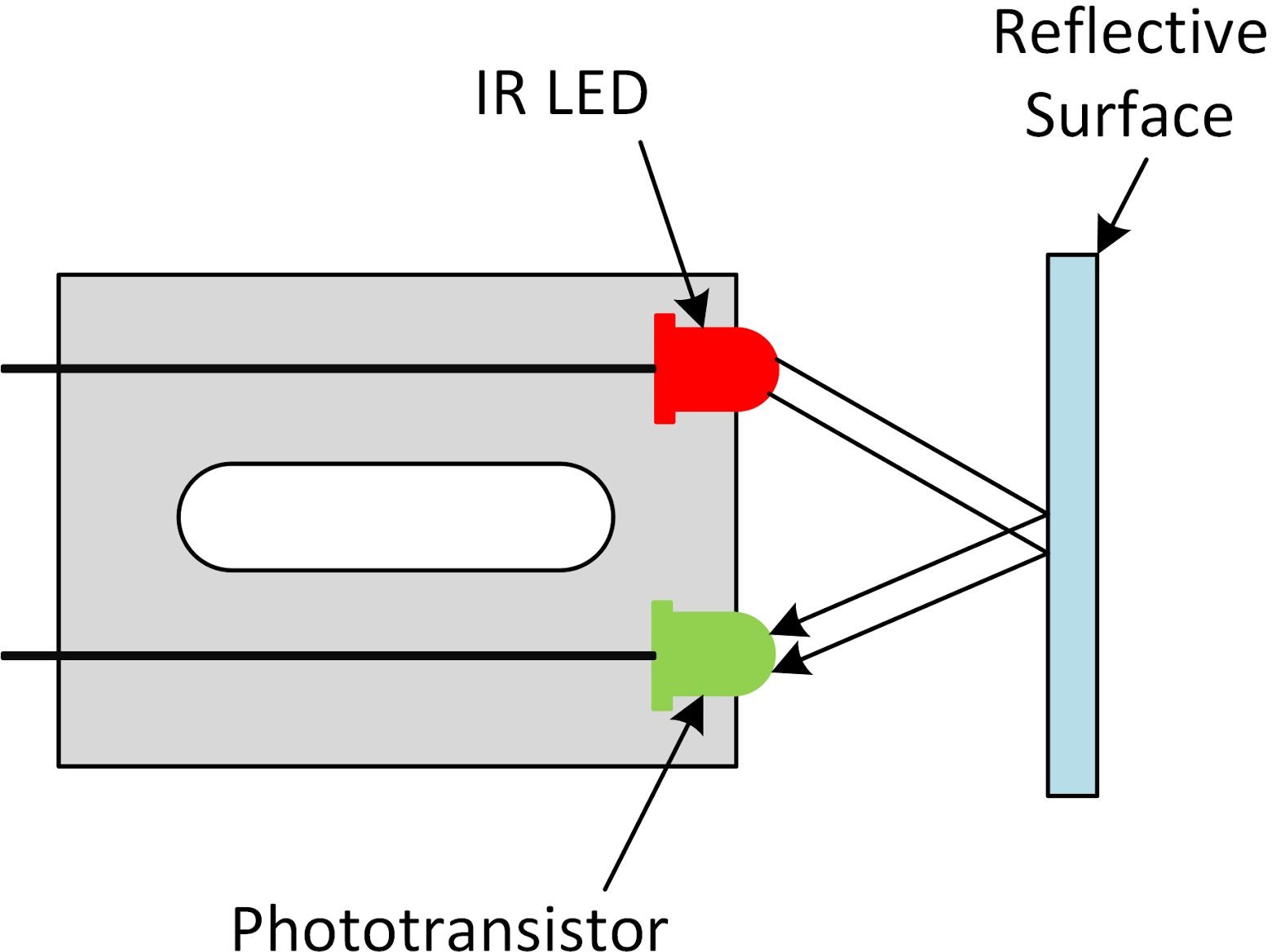
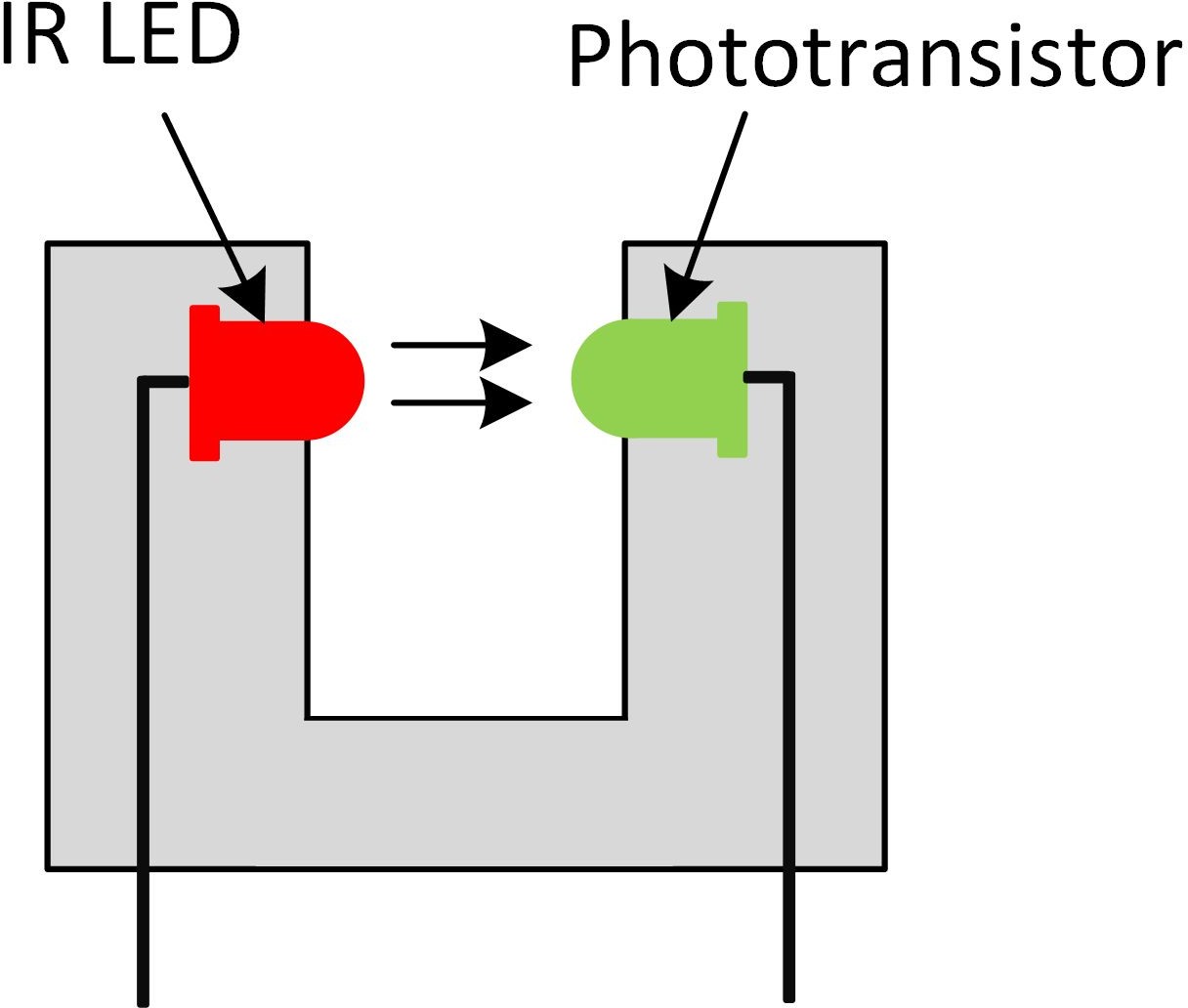


Figure 4: Schematic of a transmissive (left) and reflective (right) photo switch,

# Switch Debouncing

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 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 2000 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 time. Such spurious on/off actions may cause other processes to trigger unwantedly. To counter bouncing, a signal conditioning technique, called switch debouncing, is implemented.

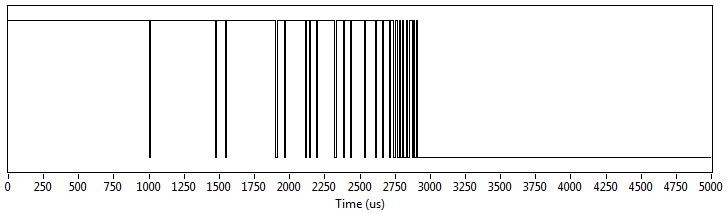


Figure 5: Switch bouncing when pressing a snap action switch.

# Debouncing Techniques

Switch debouncing can be implemented using both hardware or software. Figure 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 7. This circuit, known as an RC debouncer, uses a capacitor and resistor to filter out rapid changes in the switch output.

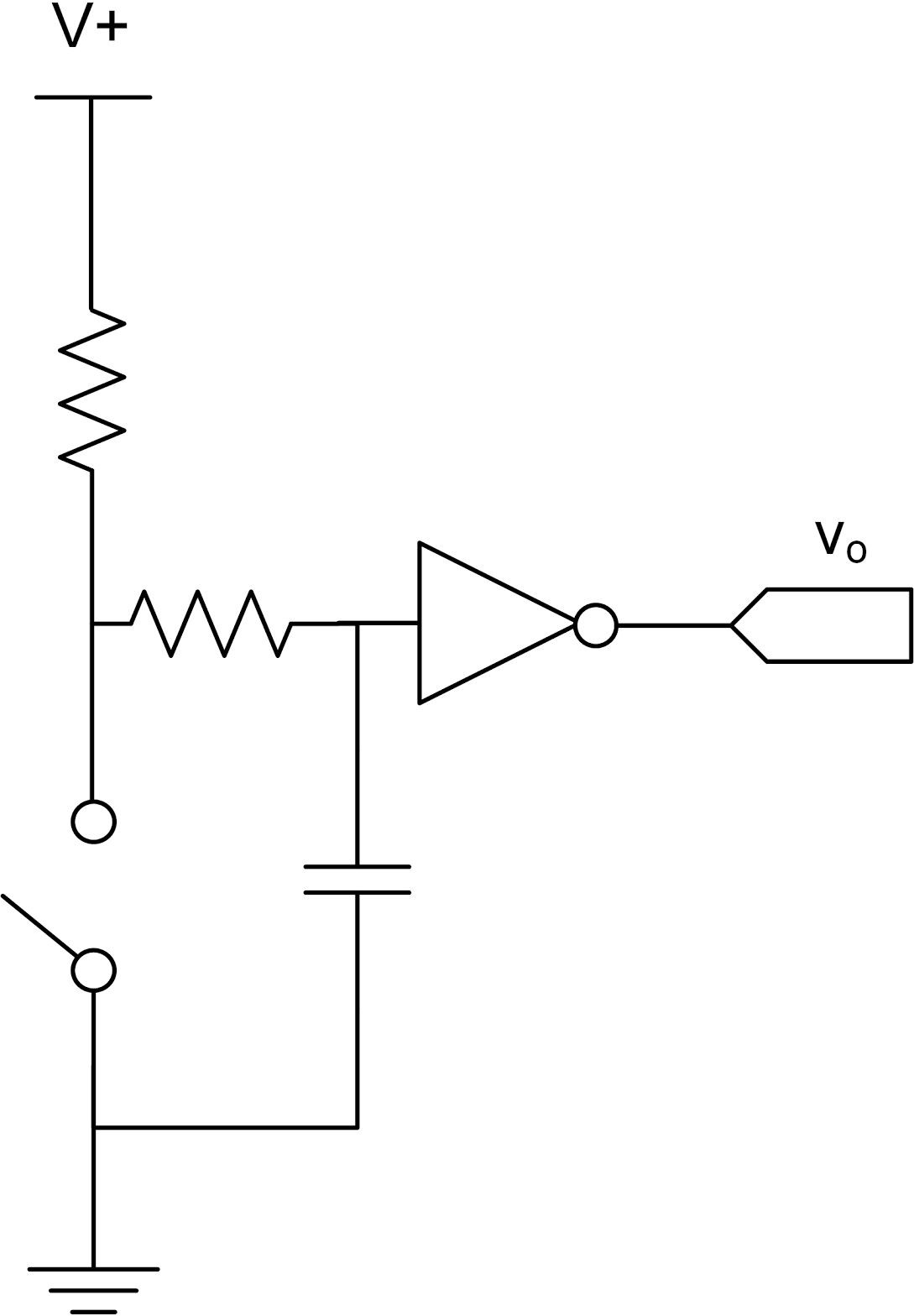
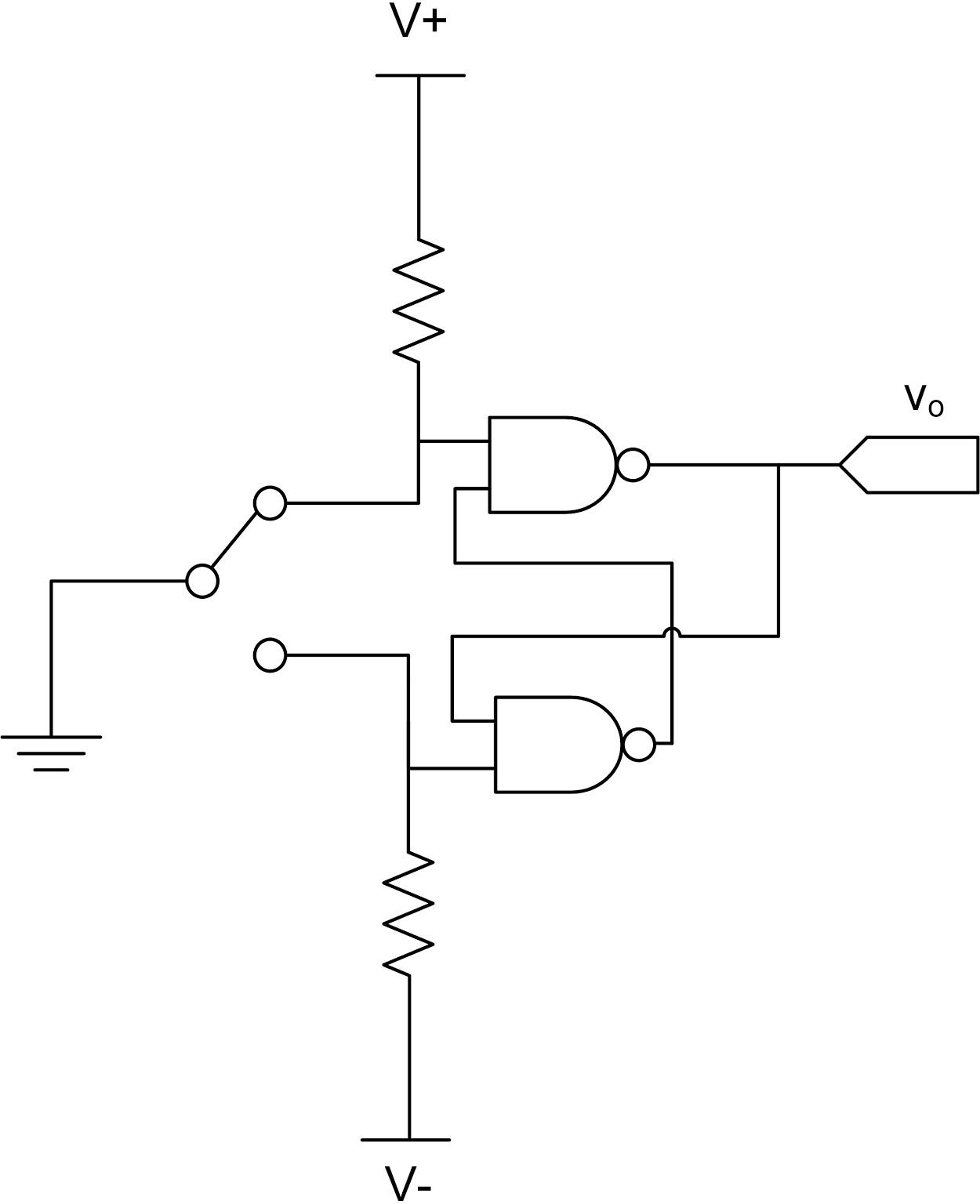


Figure 6: SR Debouncer. Figure 7: RC Debouncer.

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 8, where the output of a switch is sampled at 10

ms intervals and when 2 consecutive stable reading are encountered the debounced switch turns from low to high.

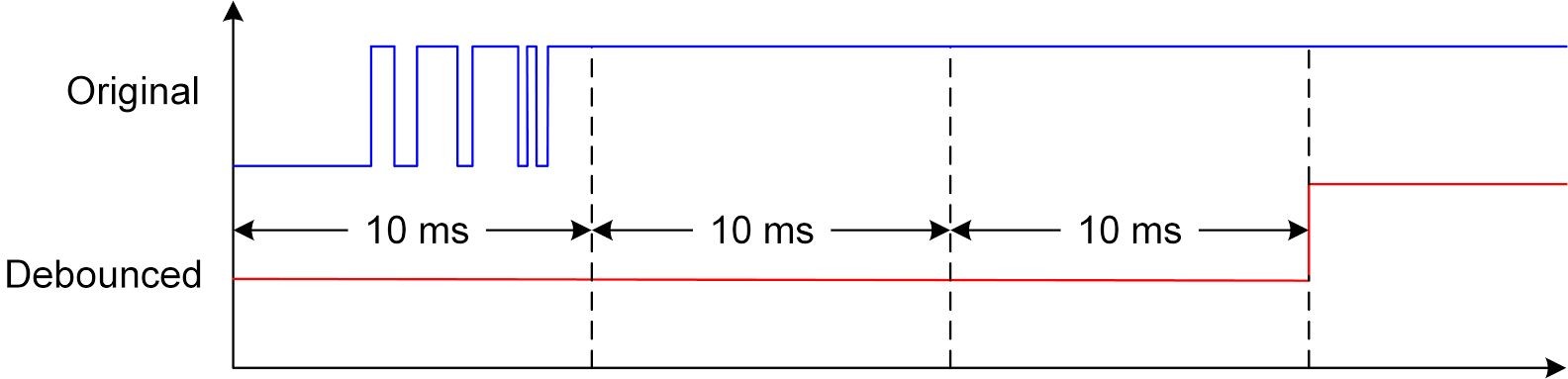
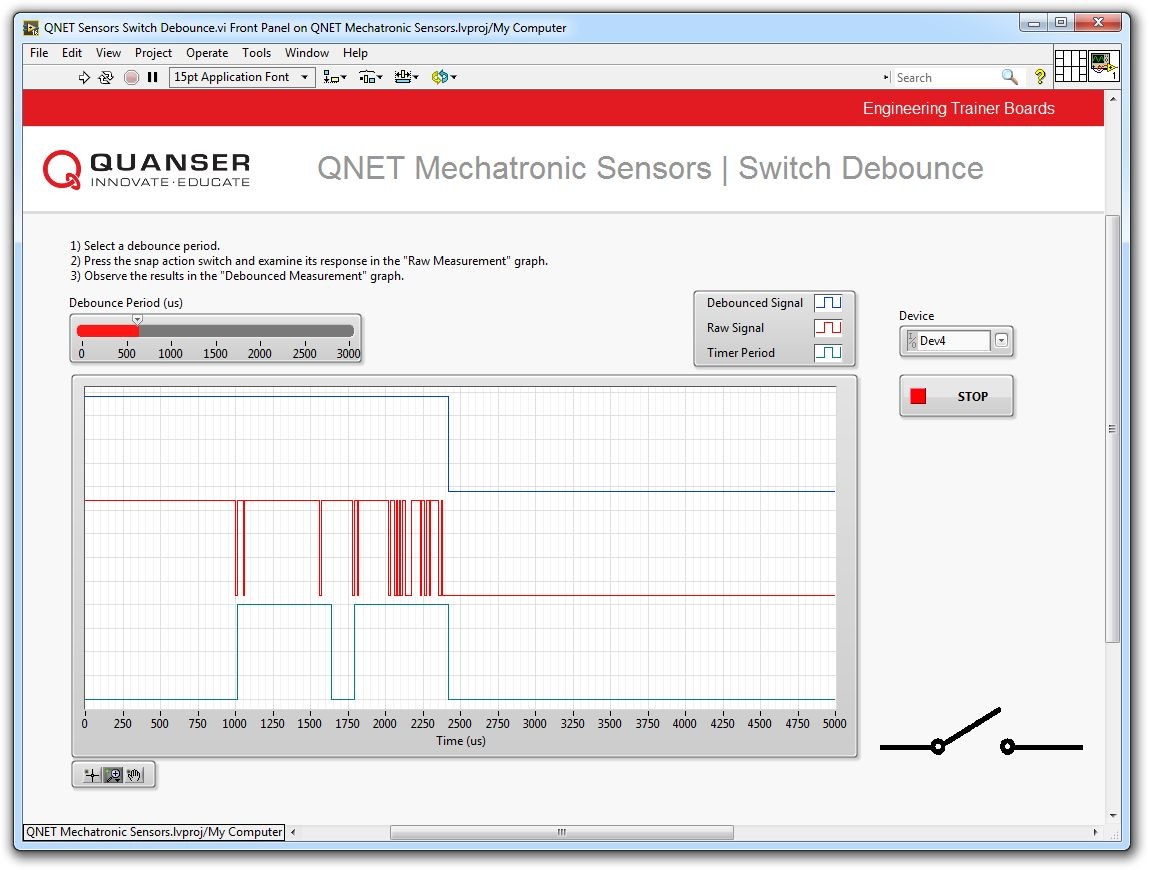


Figure 8: Software switch debouncing

**Lab Procedure**

Switch Debouncing

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



# Collect Data

Figure 1: VI used to debounce the snap action switch.

1. Open QNET Mechatronic Sensors.lvproj.
2. From the *Project Explorer* window, open QNET Sensors Switch Debounce.vi.
3. From the *Device* drop-down menu, select the device name.
4. Run the VI.
5. Click the snap action switch once. By default, the debounce period is set to 0 microseconds, which means the output of the switch will not be debounced. Therefore, the graph will display the same results for the *Raw Signal* and *Debounced Signal*.
6. Export a copy of the graph. To do this, right-click on the graph and select *Export* | *Export Simplified Image.*
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?
8. Continue to the next section.

# Switch Debounce

1. Set the *Debounce Period (us)* slider to 2000.
2. Click the snap action switch once. Examine the response of the timer, raw signal, and the debounced signal.

*Note:* The QNET Mechatronic Sensors board uses software debouncing. The VI uses 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.e. the signal value remains high), the timer is started again and the raw signal is sampled accordingly until the debounced signal is generated.

1. Set the *Debounce Period (us)* slider to 1000, 500, and 100. Examine the response of the timer, raw signal, and the debounced signal.
2. How do different debounce period values affect the debounced signal?
3. If no further experiments are required, press the *Stop* button and power down the experiment.

# Concept Review

**Strain Gage**

# Why Measure Strain?

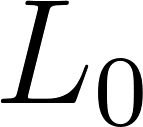
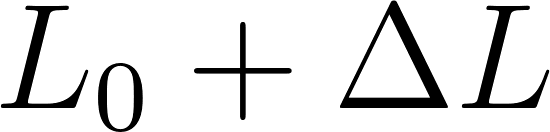
Strain is the amount of deformation of a body due to an applied force. Strain measurement is an important step in engineering design, as well as testing and maintenance practices, since it allows for determing the safe level of stress in structures and materials. Strain is commonly measured using a strain gage sensor. Strain gages are fundamental sensing devices that are used as the building blocks of many other types of transducers, including pressure, load, and torque, and are used extensively in structural test and monitoring applications. Even though strain gages are common, they are one of the most difficult types of sensors to install and acquire reliable data.

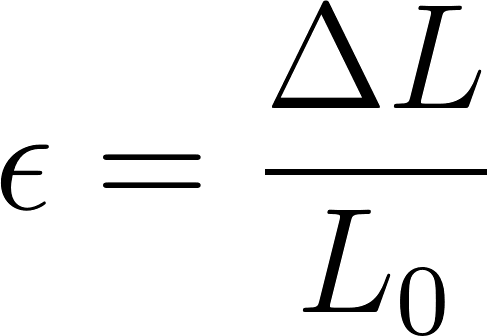
# What is Strain?

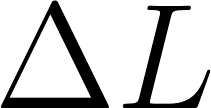
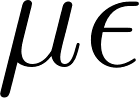
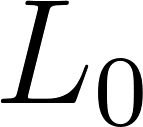
[](https://www.codecogs.com/eqnedit.php?latex=%20\sigma%20)Strain is a measure of deformation in a solid body due to an applied force. Figure 1 illustrates a rectangular bar being subjected to an axial tensile stress ( ).



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

This stress causes a change in the original length of the bar from [](https://www.codecogs.com/eqnedit.php?latex=%20L_0%20) to  [.](https://www.codecogs.com/eqnedit.php?latex=%20L_0%20%2B%5CDelta%20L%20) We define strain ( ) using Equation 1:

 (1)

[](https://www.codecogs.com/eqnedit.php?latex=%20\Delta%20L%20)[](https://www.codecogs.com/eqnedit.php?latex=%20\mu%20\epsilon%20)where [](https://www.codecogs.com/eqnedit.php?latex=%20L_0%20) is the original length and 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 microstrain ( ) by multiplying strain by 106.

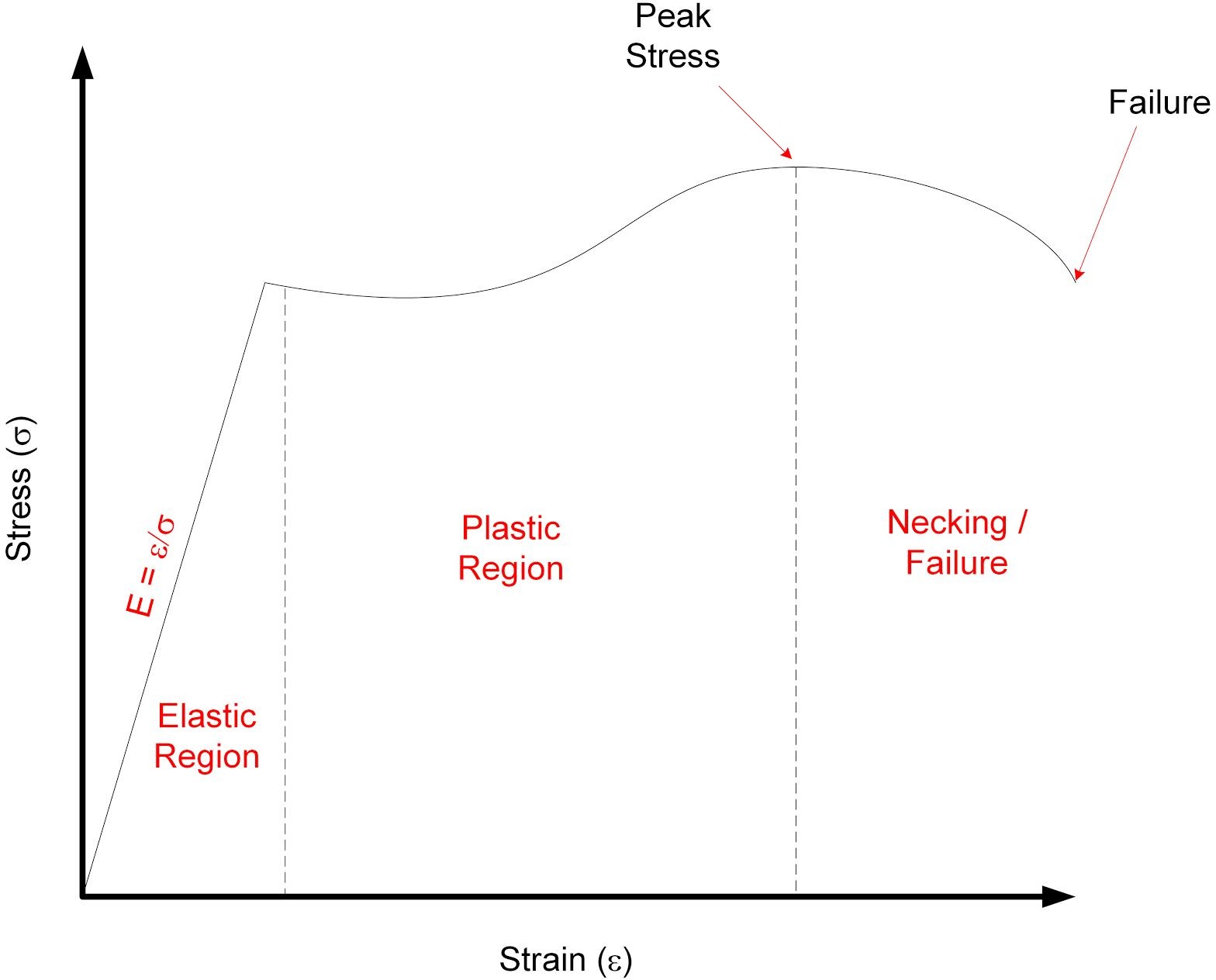
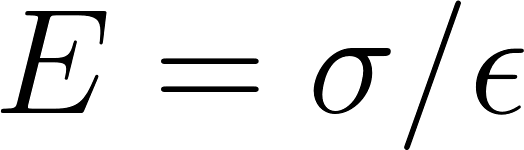


Figure 2: Stress-strain curve.

Figure 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 (  ) 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 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.

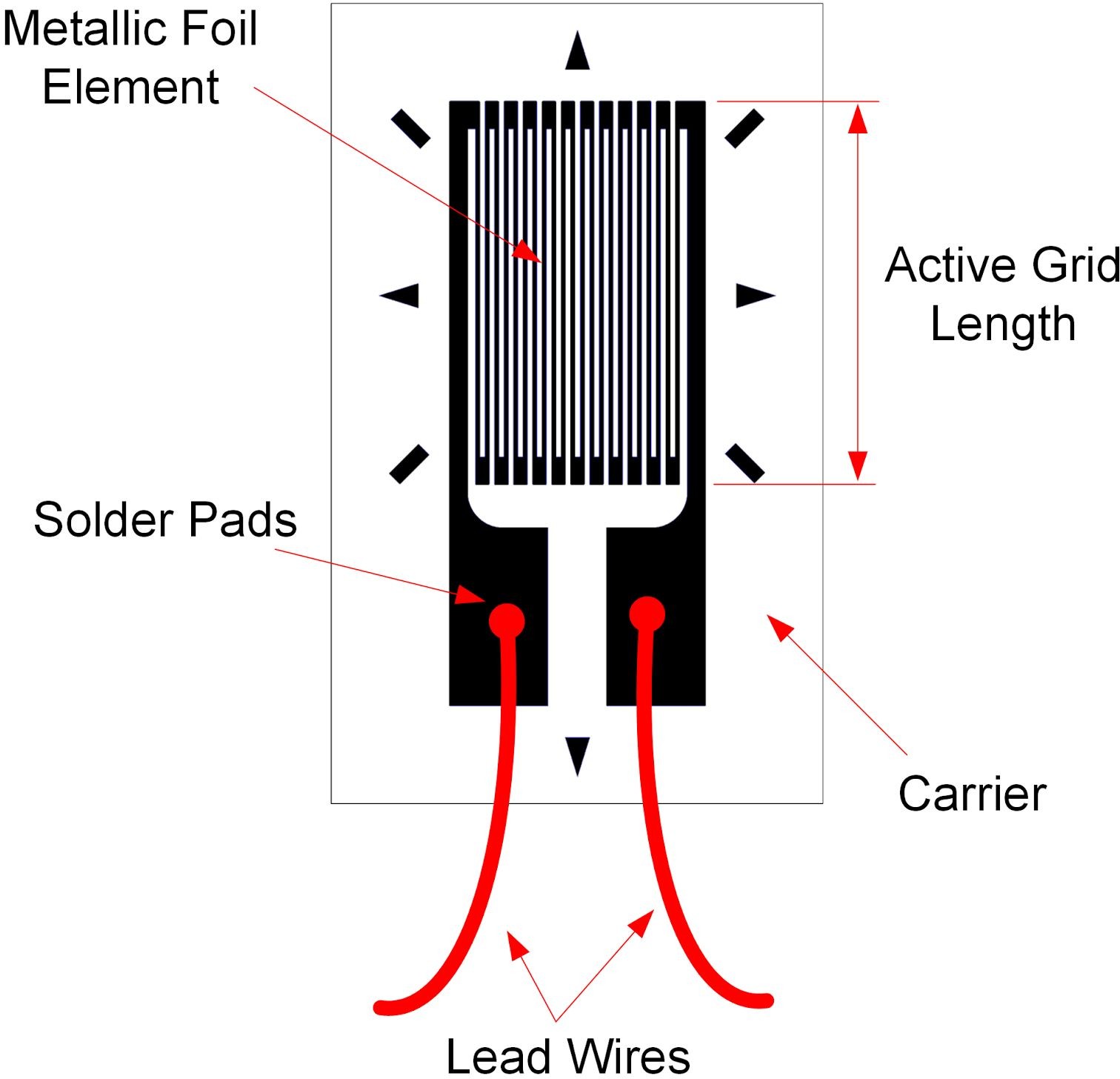
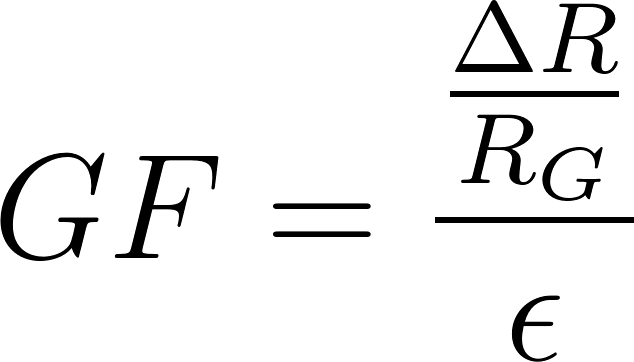


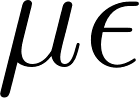
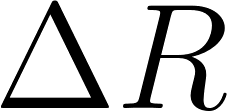
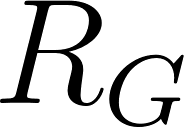
Figure 3. Schematic diagram of a strain gage (source: Wikipedia)

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 ohms. A higher nominal resistance and lower excitation voltage is 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 2:

(2)

[](https://www.codecogs.com/eqnedit.php?latex=\mu%20\epsilon%20)where [](https://www.codecogs.com/eqnedit.php?latex=%20\Delta%20R%20) is the change in resistance when the strain gage is deformed, [](https://www.codecogs.com/eqnedit.php?latex=%20R_G%20) is the nominal resistance of the gage, and ε 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. 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 are very low (typically in the range of microvolts), removing a large fixed voltage drop allows for the signal to be amplified using an amplifier.

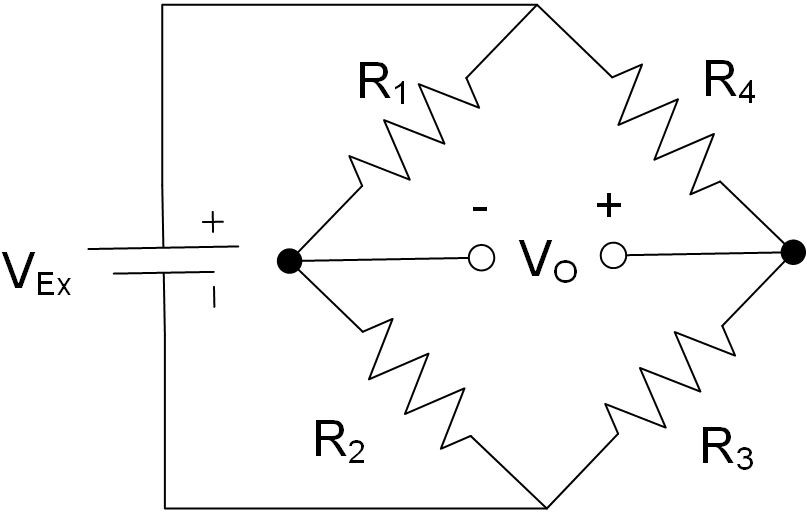
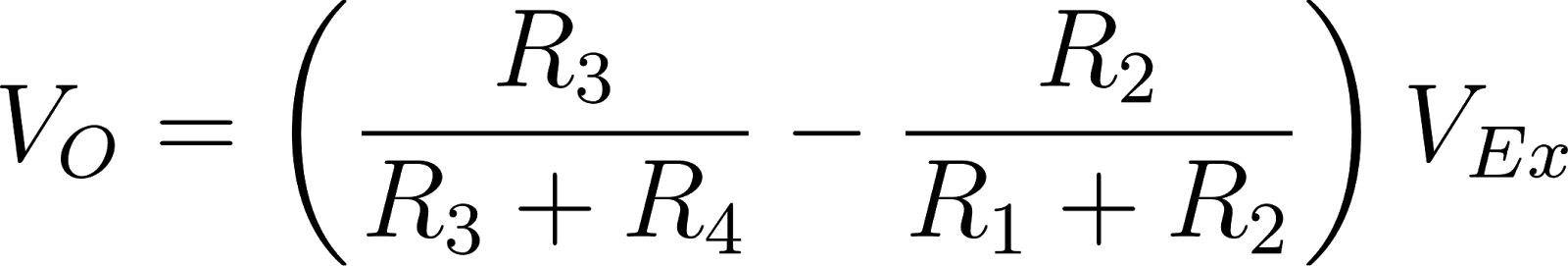


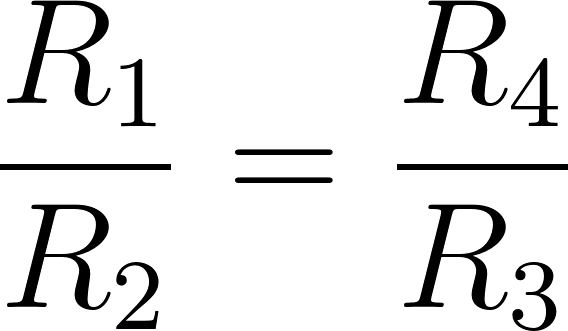
Figure 4: A Wheatstone bridge circuit for measuring the output of resistive sensors.

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

(3)

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

*balanced*:

(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 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 ohms when unstrained and the fixed resistors each equal 120 or 350 ohms, respectively.

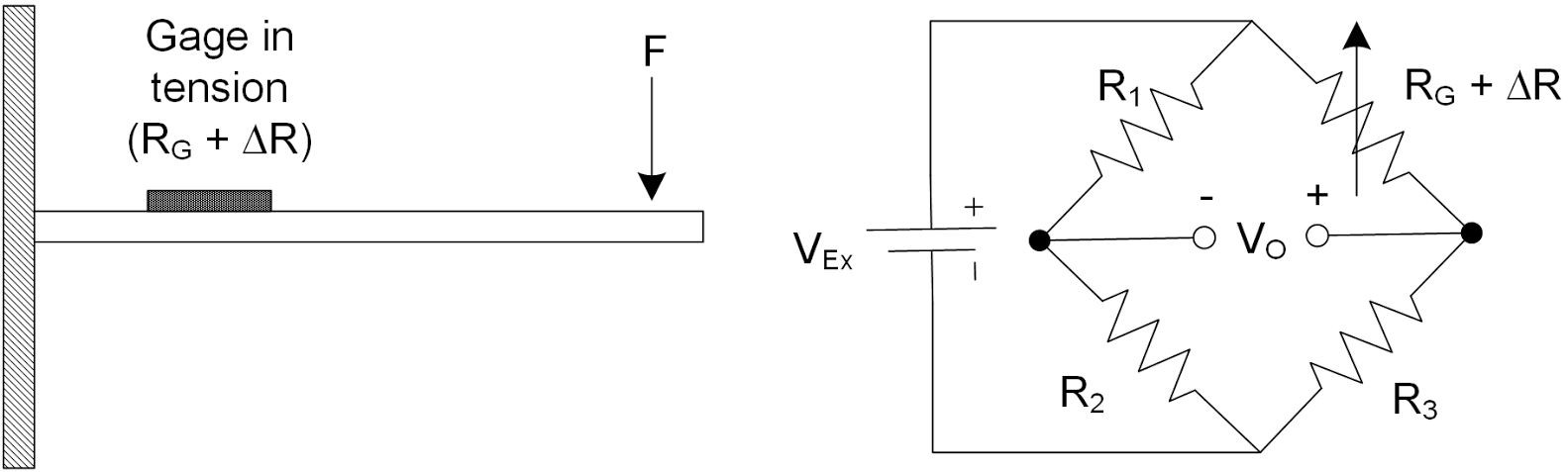


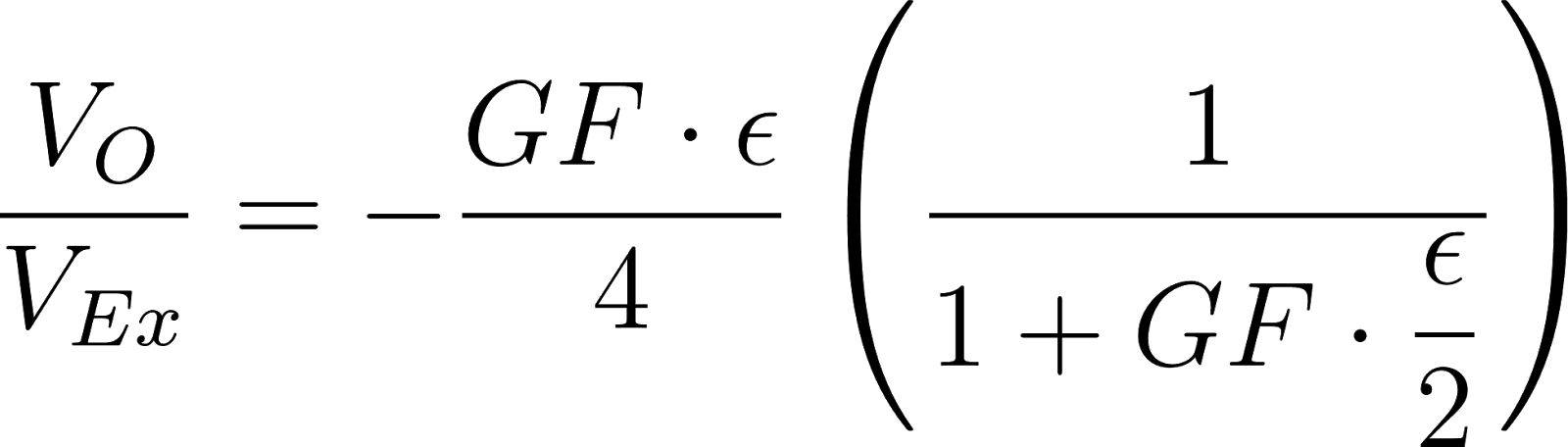
Figure 5: Quarter-bridge configuration for measuring strain in a cantilever beam.

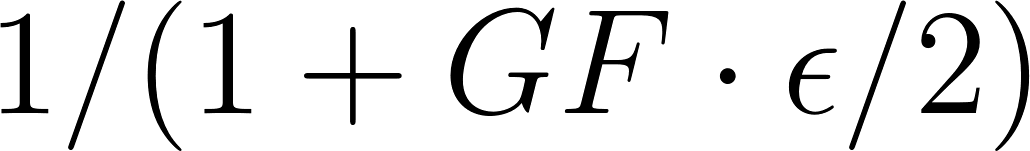
When a bending force is applied to the beam, it causes the beam along with the strain gage to deform. As a result, the resistance of the gage changes and a voltage output (VO) is generated which can be measured using a DAQ. The voltage output is proportional to the strain induced in the beam.

For example, assume that a bending force causes RG, which has a nominal resistance of 350 ohms, to increase by 0.0085 ohms. If the bridge is excited at 5 V, using Equation 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. 2 in Eq. 3, the output voltage (VO) of the quarter-bridge circuit can be expressed in terms of VEx, GF, and the measured strain ( [)](https://www.codecogs.com/eqnedit.php?latex=%20%5Cepsilon%20) as shown in Equation 5:

(5)

Note that the presence of the term  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 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.

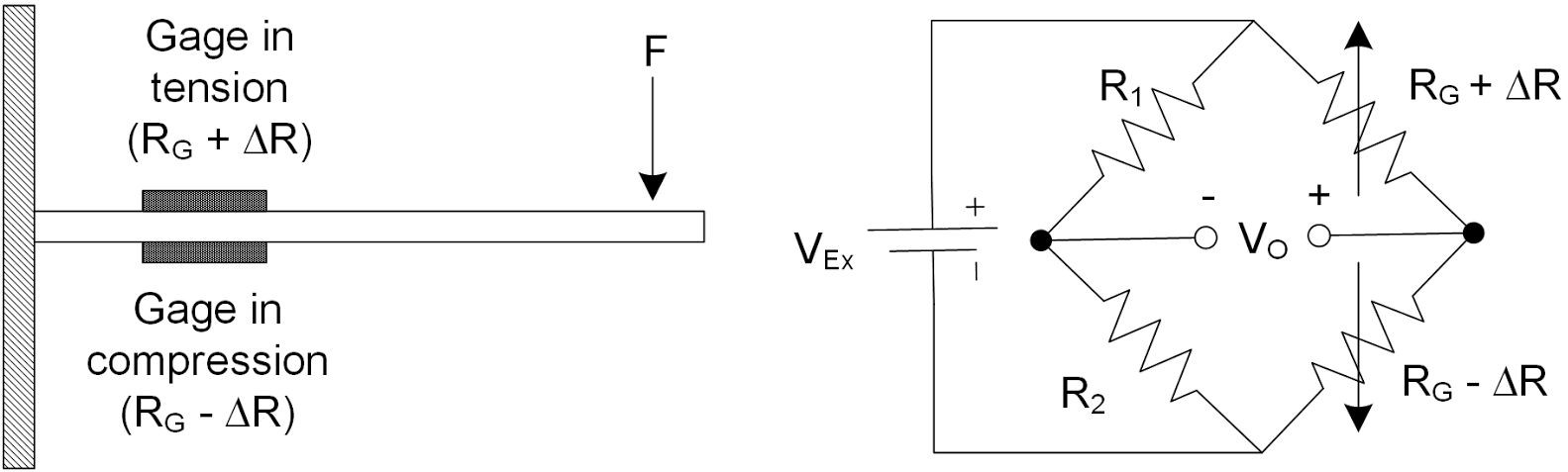
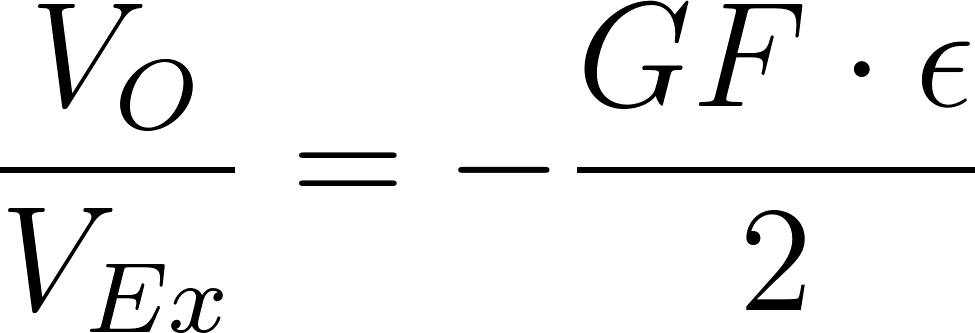


Figure 6: Half-bridge configuration for measuring strain in a cantilever beam.

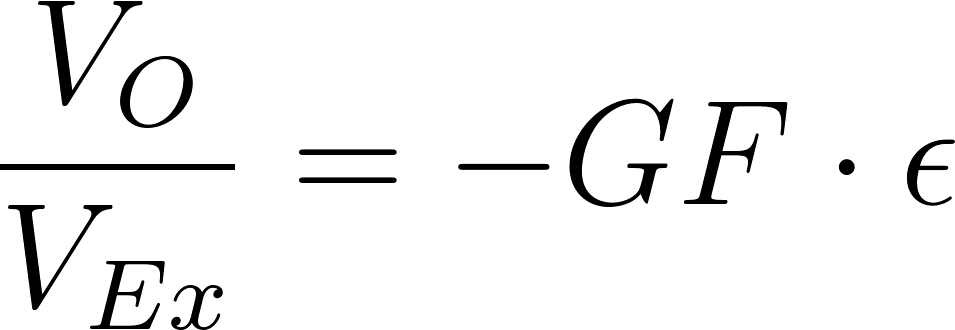
Assuming R1 = R2 = RG , and substituting Eq. 2 in Eq. 3, the output voltage (VO) of the half-bridge circuit can be expressed in terms of VEx, GF, and the measured strain ( ) as shown in Equation 6:

(6)

## Full-bridge Configuration

As illustrated in Figure 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 Equation 2 in

Equation 3, the output voltage of the full-bridge circuit (VO) can be expressed in terms of VEx, GF, and strain ( ) as shown in Equation 7:

(7)

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

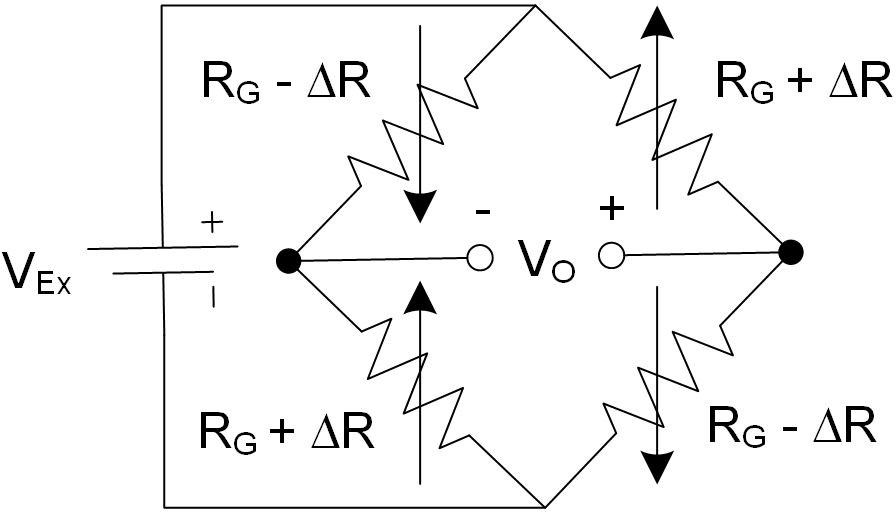


Figure 7: Full-bridge configuration for measuring strain.

# 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 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 5.

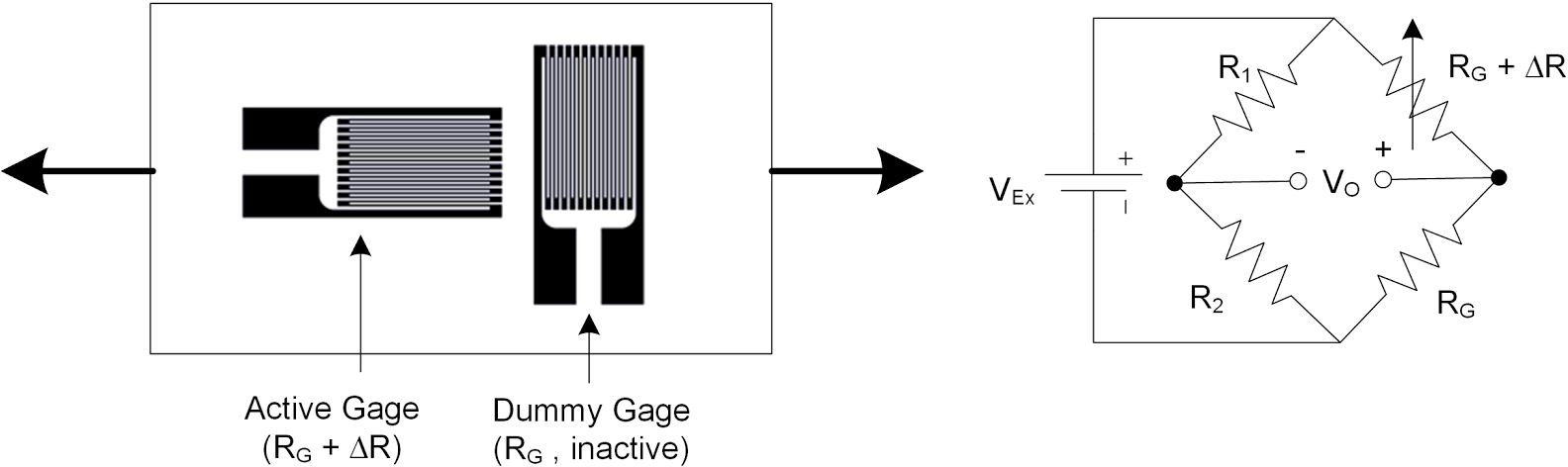


Figure 8: Quarter-bridge type II temperature compensating configuration.

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 ( [)](https://www.codecogs.com/eqnedit.php?latex=%20%5Cepsilon%20), 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 exists 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.

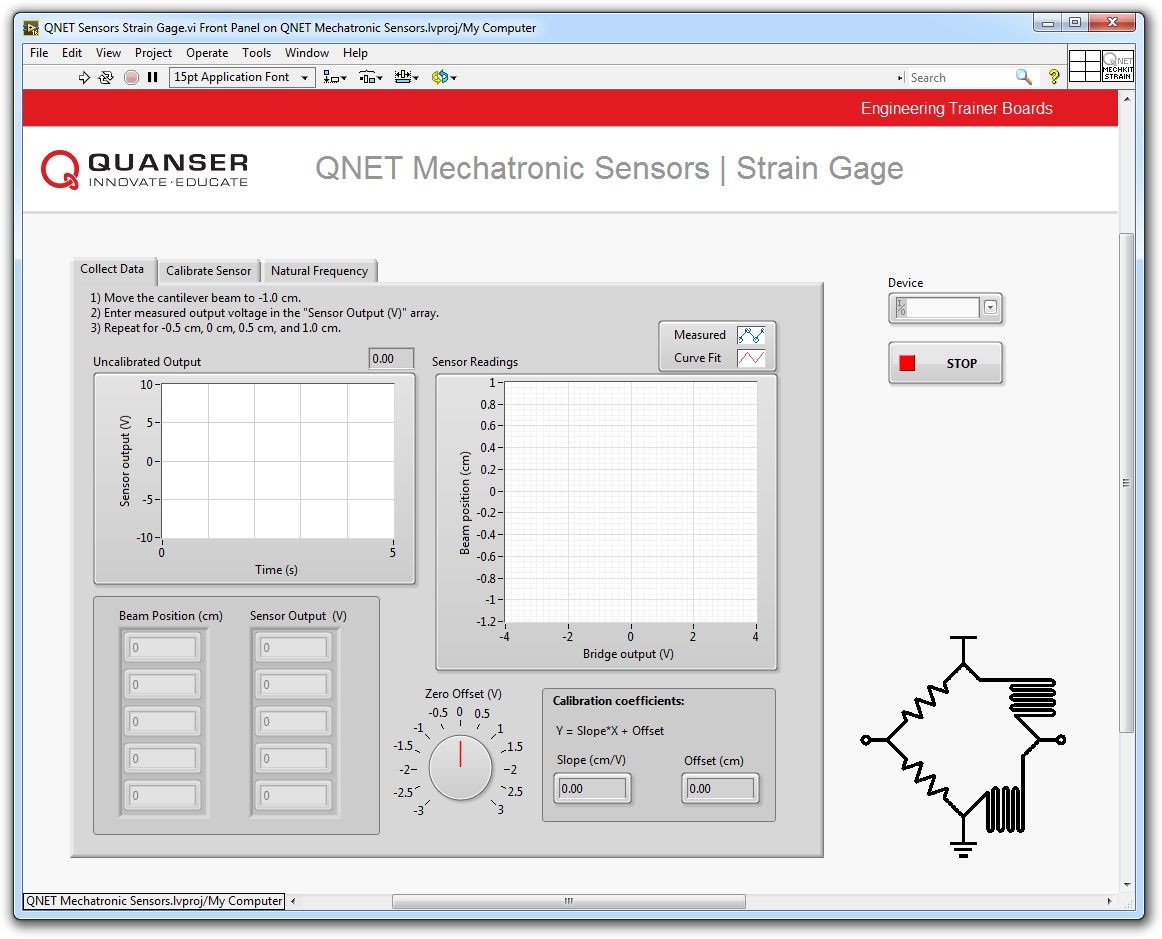
Full-scale span is the output range of a Wheatstone bridge circuit when the gage is subjected to maximum and minimum deflection. In practice, setting full-scale span requires the user to deflect the beam/gage assembly to its maximum or minimum positions and adjust the Wheatstone bridge output to a desired 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.

**Lab Procedure**

Strain Gage

The Virtual Instrument (VI) used to collect data from, calibrate, and determine the natural frequency of the strain gage is shown in Figure 1.



# Collect Data

Figure 1: VI used to collect data from the strain gage.

1. Open QNET Mechatronic Sensors.lvproj.
2. From the *Project Explorer* window, open QNET Sensors Strain Gage.vi.
3. Click on the *Collect Data* tab.
4. From the *Device* drop-down menu, select the device name.
5. Run the VI.
6. Using the *Uncalibrated Output* waveform chart, read the initial strain gage bridge output.
7. Balance the strain gage bridge by adjusting the *Zero Offset (V)* kbob such that the uncalibrated output of the bridge circuit is as close as possible to 0.00 V

*Note:* Balance the output of the bridge when the strain gage is at rest (i.e. the beam is not flexed).

1. Enter -1 in the *Beam Position (cm)* array.
2. Flex the tip of the cantilever beam to the -1 cm mark.
3. Read the corresponding strain gage output and enter the value in the *Sensor Output (V)* array.
4. 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. Enter the beam position and measured sensor outputs in the *Beam Position (cm)* and *Sensor Output (V)* arrays respectively.
5. As 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.
6. 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.
7. Record the collected data in Table 1.
8. Continue to the next section.

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

Table 1: Recorded sensor output.

# 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. Continue to the next section.

# Determine Natural Frequency

1. To determine the natural frequency of the cantilever beam, click on the *Natural Frequency* tab.
2. Ensure the Beam is at rest (i.e. not vibrating).
3. With one finger manually perturb its tip of the beam.
4. Wait for a couple of seconds for the beam to stop vibrating and immediately press the *Plot* button.
5. The VI captures the resulting response in the *Power Spectrum* waveform graph. The response should look similar to Figure 2.

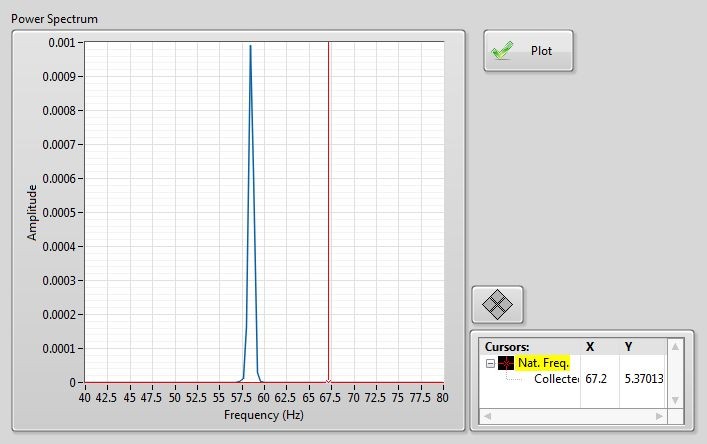


Figure 2: Sample natural frequency response.

1. Using the *Cursor* tool, measure the peak frequency and record it in Table 2.

|  |  |
| --- | --- |
| **Trial** | **Natural Frequency (Hz)** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

Table 2: Recorded natural frequency measurements.

1. Repeat the measurement several times to determine the average natural frequency of the beam. Record your measurements in Table 2.

*Note:* To ensure data is properly collected at each measurement attempt, wait at least 10 seconds between consecutive measurements.

1. If no further experiments are required, press the *Stop* button and power down the experiment.

# Concept Review

**Thermistor**

# Why Measure Temperature?

Temperature is perhaps the most commonly measured physical quantity. Temperature is the degree of hotness or coldness of an object. The measurement and control of temperature has many applications ranging from healthcare to chemical to the food processing industries. Common temperature measurement sensors include voltage-generating sensors such as thermocouples, and resistive-type sensors such as thermistors and resistance temperature detectors (RTD). The choice of temperature sensor determines the range and accuracy at which temperature can be measured. Therefore the proper selection of a temperature sensor is of utmost importance.

# What is a Thermistor?

There are several different types of transducers available to measure temperature: thermocouple, RTD, thermistor, and integrated circuit type. Each have their own advantages and disadvantages. Integrated circuit type sensors are low cost and have a linear output, but because they are mounted on a PCB they are board layout dependent. Thermocouples have a wide temperature range, are relatively cheap, and are easy to use but are the least stable and sensitive. An RTD, on the other hand, is most stable and accurate of the sensors but is slow, has a fragile construction, and is relatively the most expensive. A thermistor responds very quickly, and has the highest sensitivity but has a limited temperature range. Figure 1 compares the typical sensitivity of the three temperature sensors.

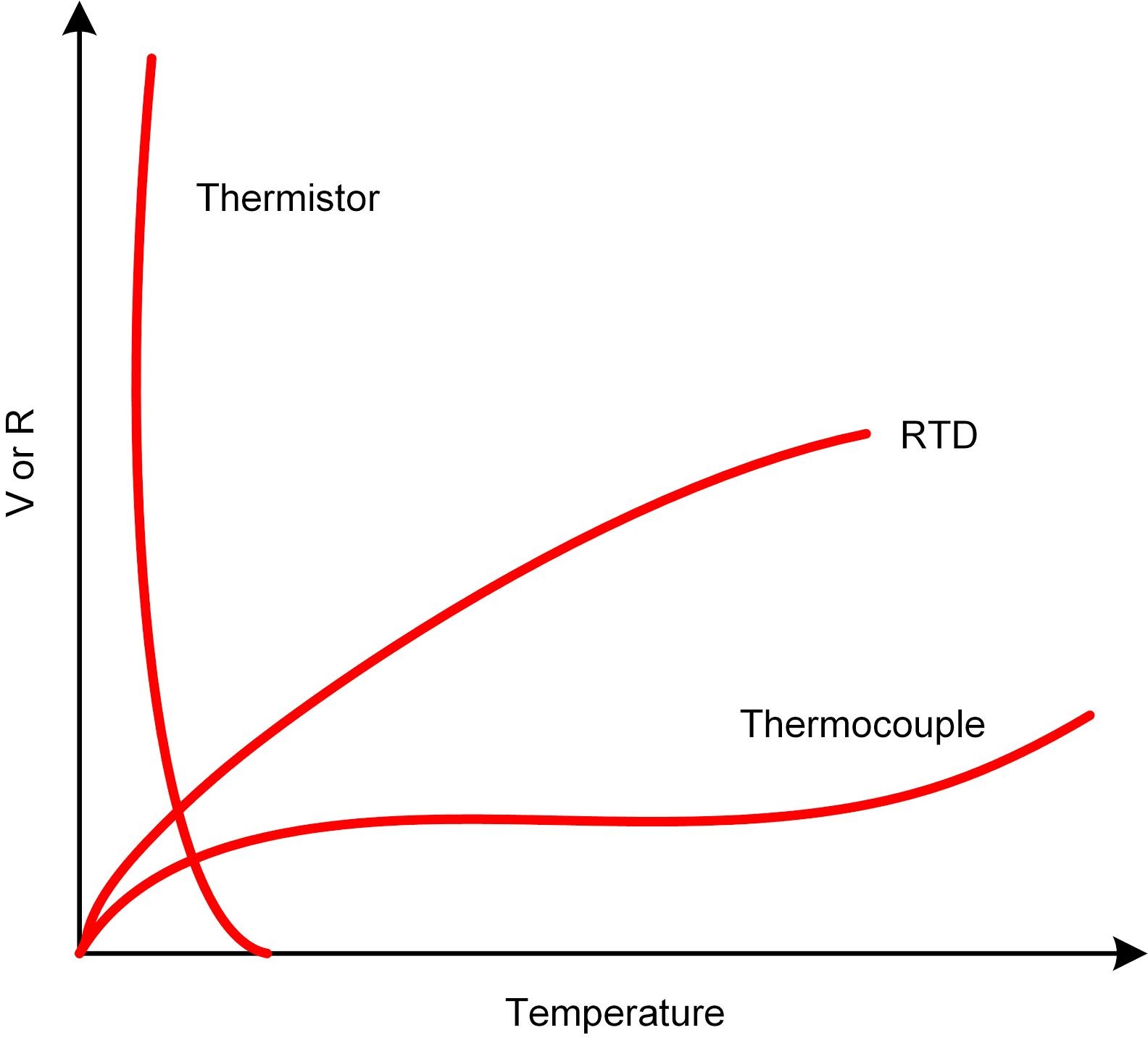
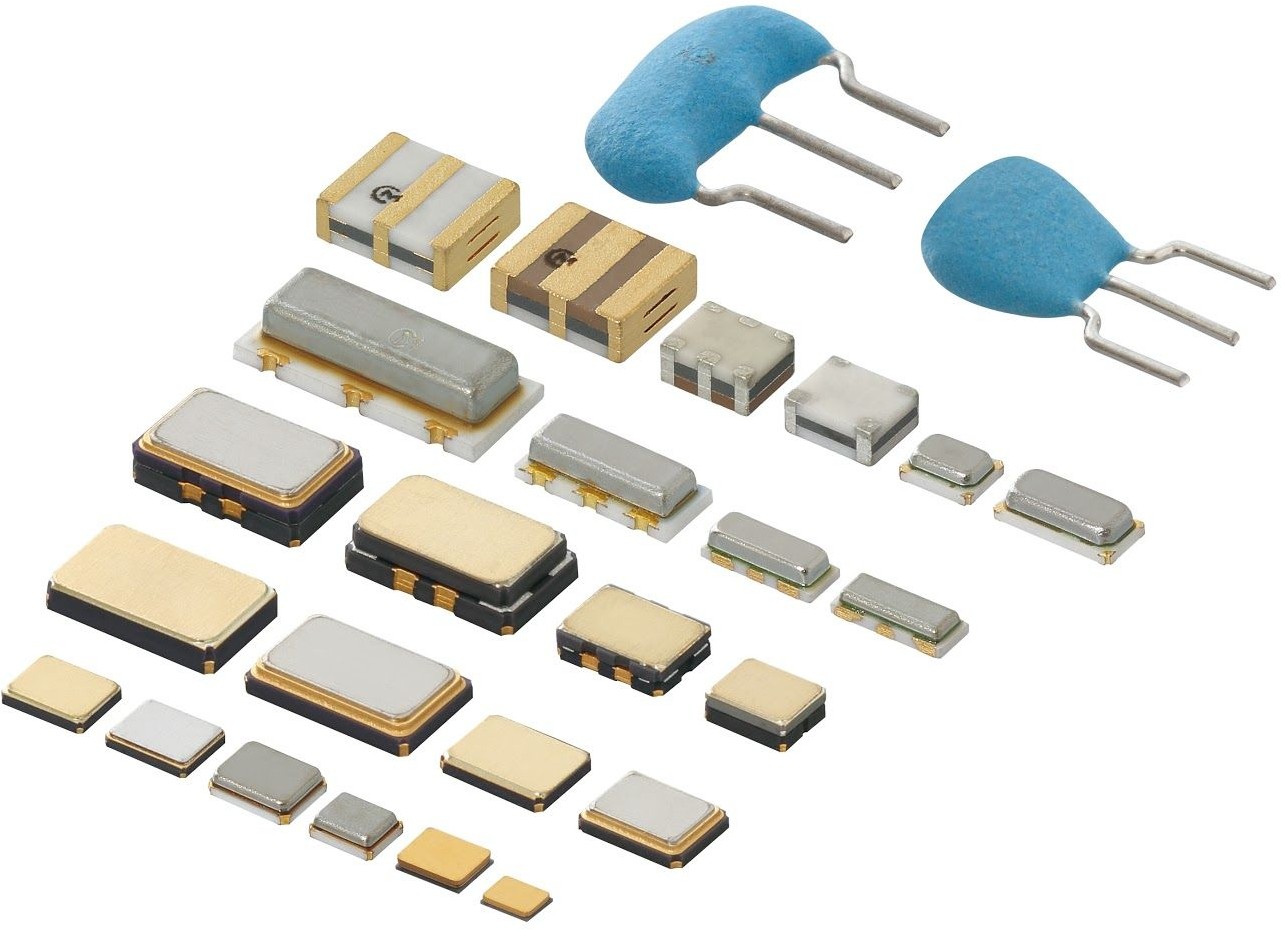
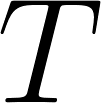
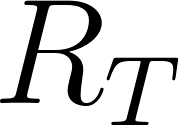
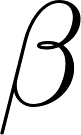


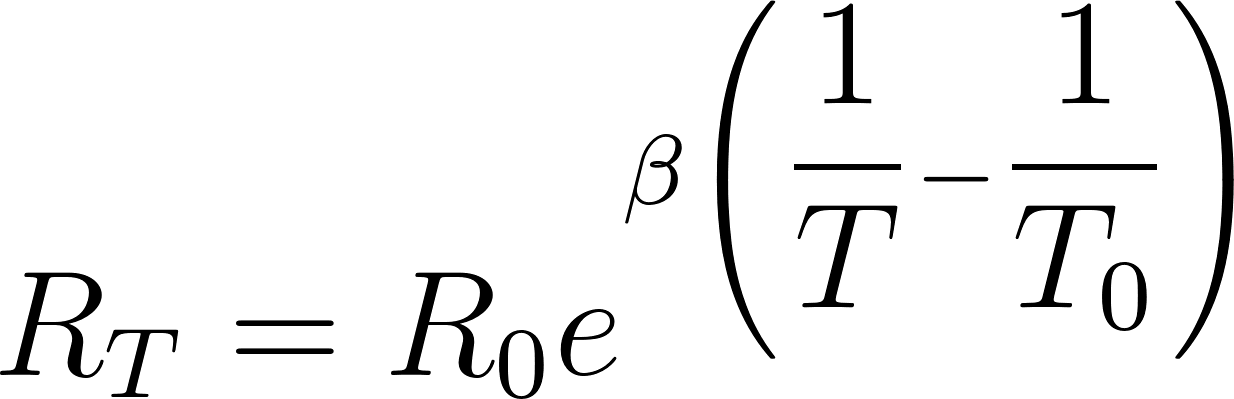
Figure 1: Comparison between the sensitivity of thermistors, RTD, and thermocouples.

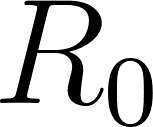
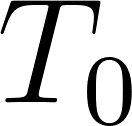
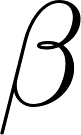
Thermistors differ from thermocouples in that they are resistive sensors with the latter being a voltage generating sensor. Because a current must be passed through a thermistor, it is susceptible to ohmic heating. This self-heating manifests itself in the form of measurement error. Figure 2, illustrates various shapes of common thermistors.

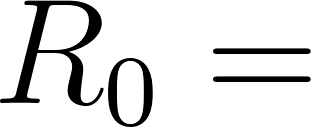
(a) board-mount type (b) string type Figure 2: Different types of common thermistors manufactured by Murata.

[](https://www.codecogs.com/eqnedit.php?latex=%20T%20)As a resistive sensor, the resistance of a thermistor is dependent on temperature. The relationship between the resistance of the thermistor, [](https://www.codecogs.com/eqnedit.php?latex=%20R_T%20), and temperature, , can be described using the 

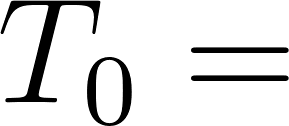
-parameter equation:

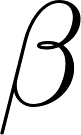
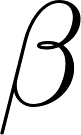
(1)

where [](https://www.codecogs.com/eqnedit.php?latex=%20R_0%20) is the nominal resistance of the sensors at temperature [](https://www.codecogs.com/eqnedit.php?latex=%20T_0%20) in Kelvin, and  is a parameter which depends on the material, temperature, and construction of the thermistor and typically ranges between 3500-4700 K. For the thermistor on the Mechatronic Sensors board, the nominal sensor resistance is:

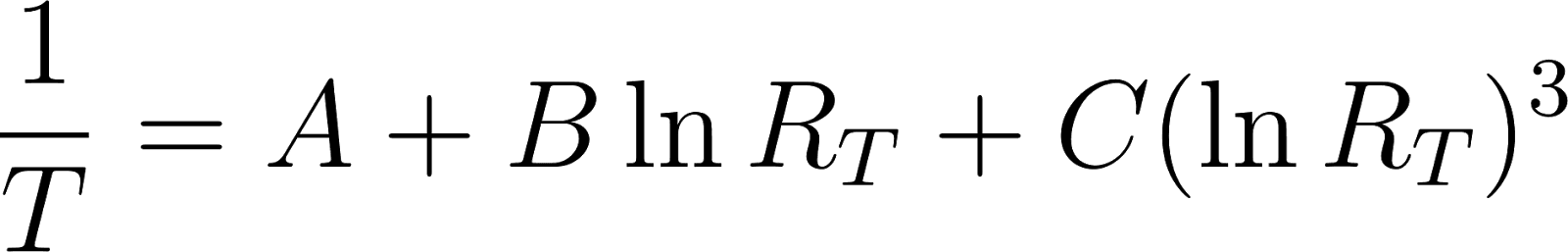
[](https://www.codecogs.com/eqnedit.php?latex=R_0%20=%20) 47000 ohms

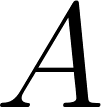
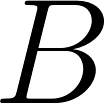
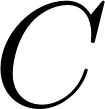
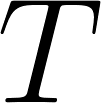
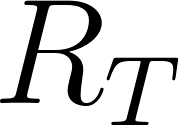
when the temperature is at 25 degrees Celsius or:

[](https://www.codecogs.com/eqnedit.php?latex=%20T_0%20=%20) 298.15 K

The -parameter equation provides an acceptable estimation of the measured temperature in applications that have a narrow temperature span (typically less than 20°C). The  -parameter can be established using two temperature reference points.

A more accurate estimation temperature (0.01°C over a 100°C span) can be made using the Steinhart-Hart equation as given in Equation 2:

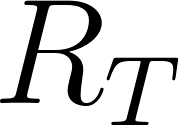
 (2)

[](https://www.codecogs.com/eqnedit.php?latex=A)[](https://www.codecogs.com/eqnedit.php?latex=B)[](https://www.codecogs.com/eqnedit.php?latex=C)[](https://www.codecogs.com/eqnedit.php?latex=T)where , , and are the Steinhart-Hart parameters which are determined by means of a 3-point calibration process, and [](https://www.codecogs.com/eqnedit.php?latex=R_T) is the resistance of the thermistor at temperature in Kelvin.

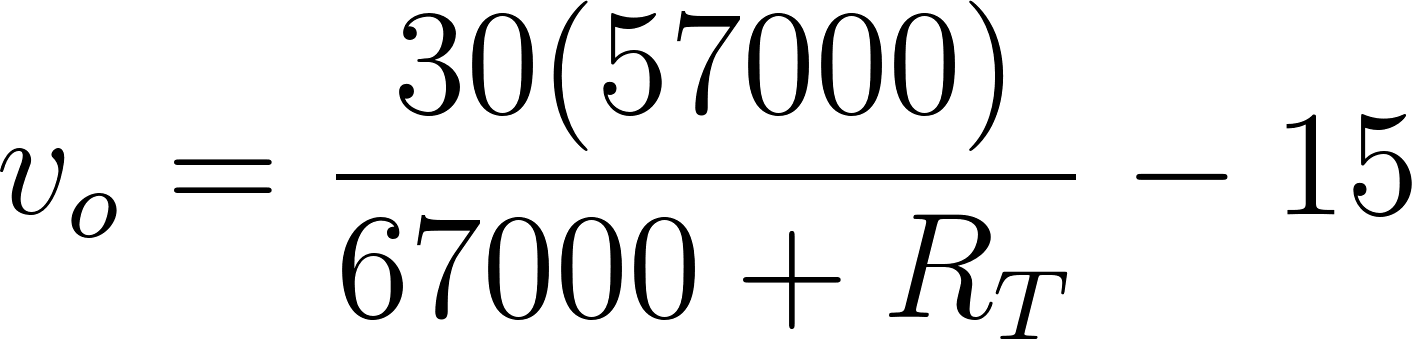
# NTC and PTC Thermistors

Two types of thermistors are available: Negative Thermal Coefficient (NTC) and Positive Thermal Coefficient (PTC). The resistance of an NTC thermistor decreases with an increase in temperature. Conversely, the resistance of a PTC thermistor increases when temperature increases. NTC thermistors are more sensitive compared to PTC thermistors, and therefore exhibit a much higher change in resistance when exposed to the same levels of temperature.

# Measuring the Output of a Thermistor

Similar to most resistive sensors, the output of a thermistor is measured using a voltage dividing circuit. Figure 3 illustrates the voltage dividing circuit used in the Mechatronic Sensor board, where the thermistor is labeled as [](https://www.codecogs.com/eqnedit.php?latex=%20R_T%20).

Using the voltage divider rule, the output voltage of the circuit shown in Figure 3 is by Equation 3:

(3)

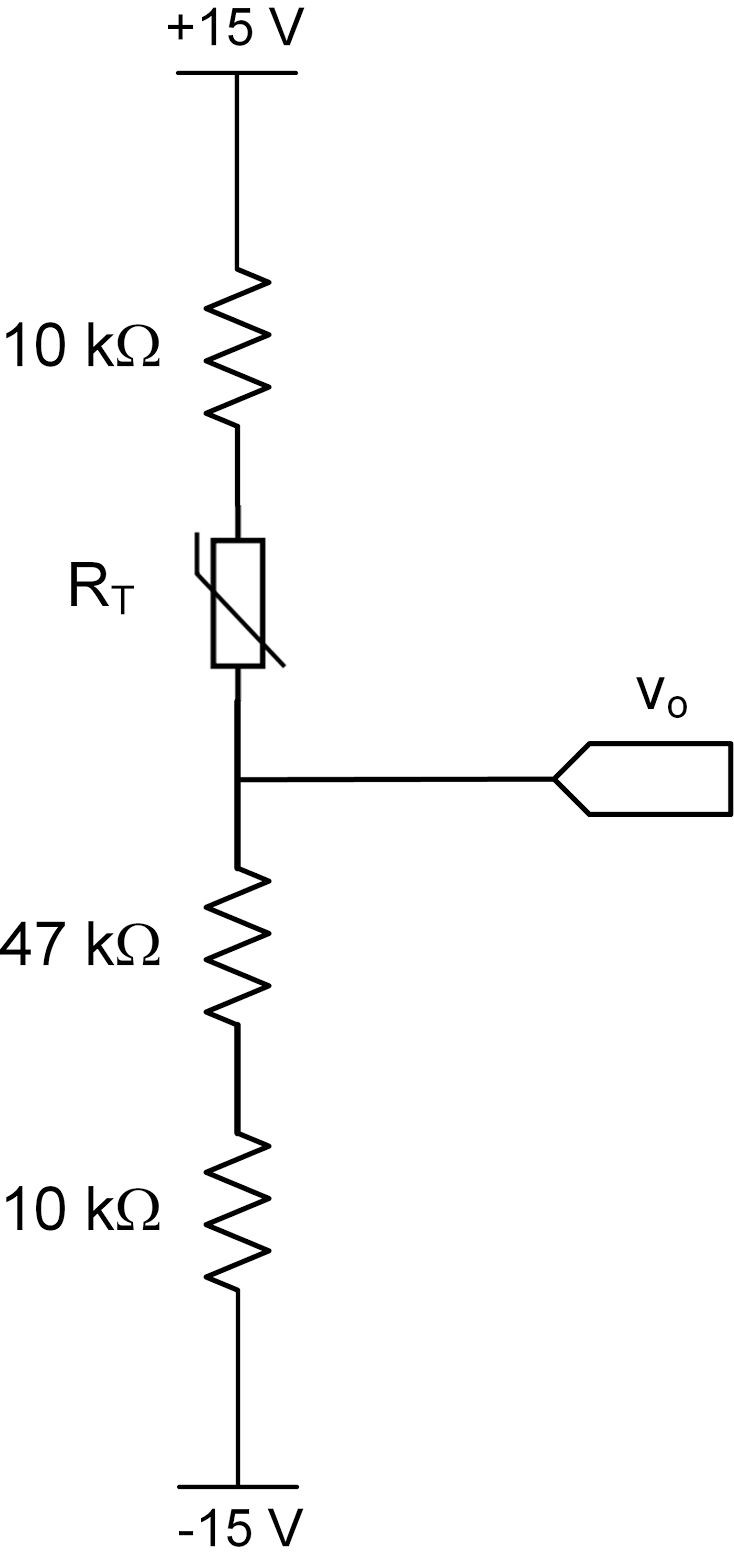
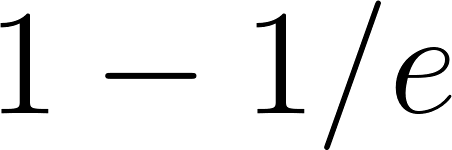


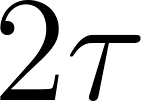
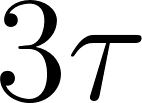
Figure 3: Thermistor circuit used in the Mechatronic Sensors board.

# Time Response of a Thermistor

Temperature sensors respond differently to changes in their input. For example, a liquid-filled bulb thermometer when taken outside on a cold winter day responds slowly to the change in temperature. In other words, the bulb thermometer does not instantaneously respond to a change in temperature. Time response is a measure of the time it takes for a sensor to respond to change. Typically sensors with fast response times are desirable.

Thermal Time Constant

[](https://www.codecogs.com/eqnedit.php?latex=\tau%20)Thermal time constant ( ) is a measure of a sensor’s response to change and is defined as the time it takes for the sensor’s output to reach  or 63.2% of the steady-state condition from an initial condition.

[](https://www.codecogs.com/eqnedit.php?latex=\tau)[](https://www.codecogs.com/eqnedit.php?latex=2\tau)[](https://www.codecogs.com/eqnedit.php?latex=3\tau)Figure 4, illustrates a typical time response curve. It shows how a sensor’s output behaves as it reaches steady state. As shown in Figure 4, after one time constant ( ) the system output reaches 63.2% of its steady state value, after 2 time constants ( ) the system reaches 86.5% of the its steady state value, and 95% is reached after 3 time constants ( ). Theoretically, steady-state is achieved after infinite time, in practice however, one waits until the output of the system or sensor is within an acceptable margin of error (typically 3 time constants). In thermal applications time constants are typically large. Thermistors typically have a time constant of 0.5 -4 seconds.

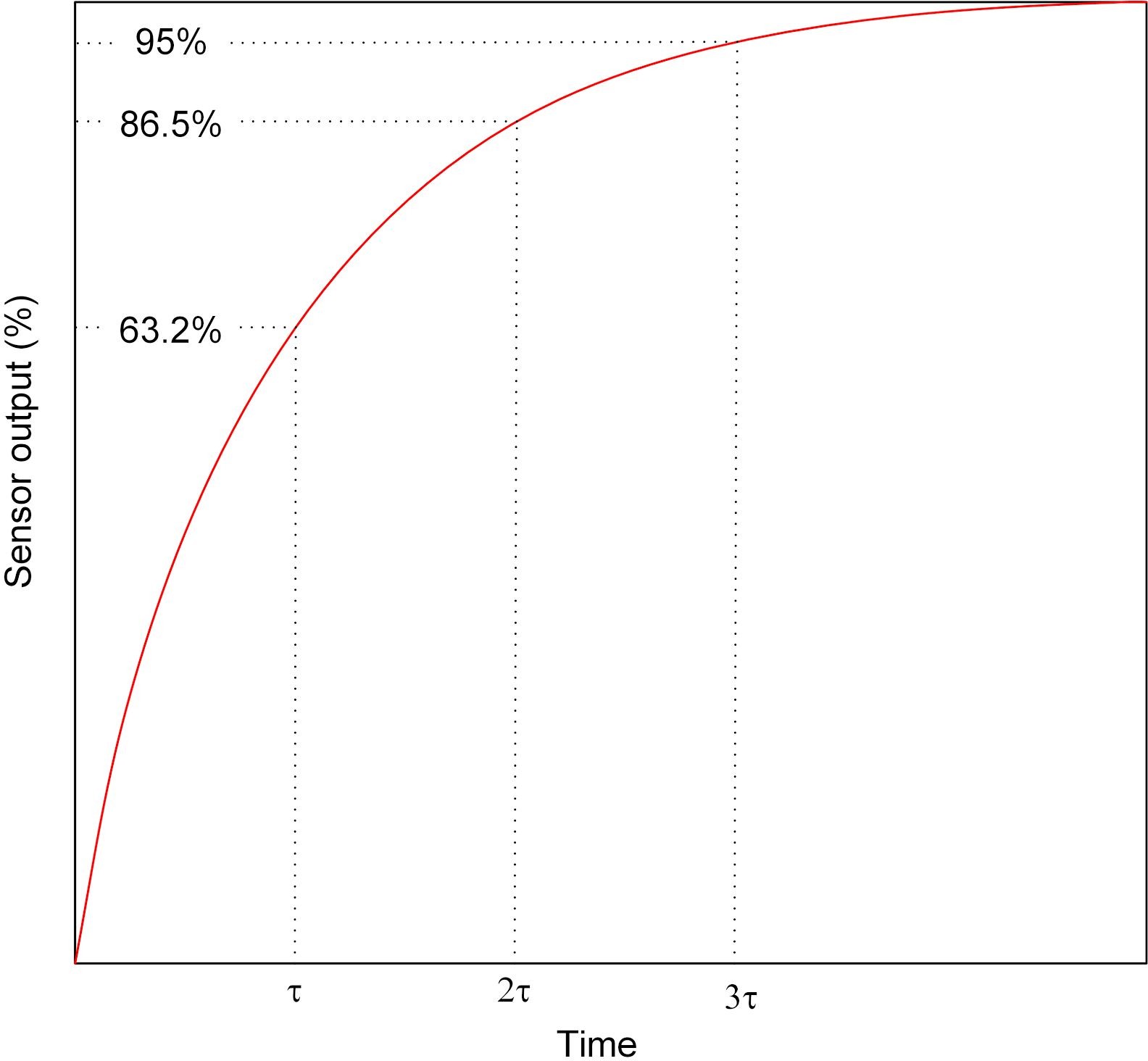


Figure 4: Time constant curve.

**Lab Procedure**

Thermistor

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

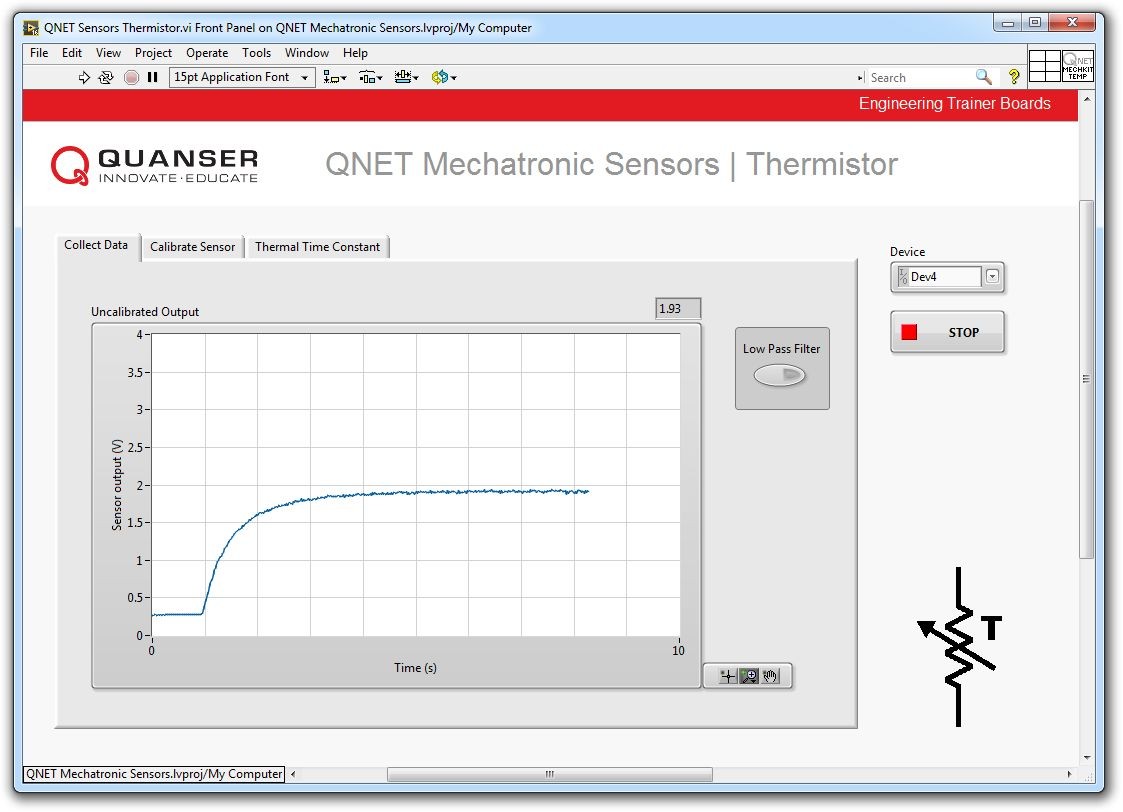


Figure 1: VI used to collect data from and calibrate the thermistor.

# Collect Data

1. For best results, your NI ELVIS II and the QNET Mechatronic Sensors board should be at room temperature before starting this experiment.

*Note:* This is because the NI ELVIS II (and thus the thermistor) heats up over time. To observe the largest temperature differential, it's best to start with the board at room temp.

1. Determine the temperature of your finger tip with a thermometer.

*Note:* For accurate results, it is recommended that you measure the temperature of your fingertip. If you do not have a thermometer, assume a finger tip temperature of 32 °C.

1. Power on the NI ELVIS II using the power switch on the back of the unit. The *READY* and *ACTIVE*

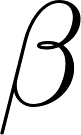
LEDs on the the NI ELVIS II panel should briefly light, and the *READY* LED should stay on.

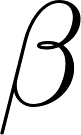
1. Power on the QNET Mechatronic Sensors board by switching on the NI ELVIS II prototyping board power switch. The *POWER* LED next to the switch should light up.
2. Open QNET Mechatronic Sensors.lvproj.
3. From the *Project Explorer* window, open QNET Sensors Thermistor.vi.
4. Click on the *Collect Data* tab.
5. From the *Device* drop-down menu, select the device name.
6. Run the VI.
7. Gently place your fingertip on the thermistor and examine the response.

*Note:* If the output of the sensor is noisy click the *Low Pass Filter* button.

1. Using the *Uncalibrated Output* waveform chart, read the corresponding sensor output when it reaches steady-state (after approximately 15 s).
2. Stop the VI.
3. Switch OFF the QNET Mechatronic Sensors board as well as the NI ELVIS II.
4. Continue to the next section.

# Calibrate the Thermistor

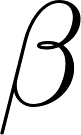
1. Using the steady-state output of the sensor from the data collection step and Equation 3 in the Concept Review, determine the resistance of the thermistor.
2. Using the temperature of your fingertip which you determined in the data collection step and Equation 1 in the Concept Review, determine the  [-](https://www.codecogs.com/eqnedit.php?latex=%20%5Cbeta%20)parameter of the thermistor. The nominal resistance of the sensor is R0 = 47000 ohms at 25 °C.

*Note:* For a fingertip temperature of 32 °C you should expect an output voltage of *vo* ≅ 1.5 V and ≅

3200. If your result is drastically different, identify potential sources of error.

1. For best results, your NI ELVIS II and the QNET Mechatronic Sensors board should be at room temperature before continuing with this experiment.
2. Power on the NI ELVIS II using the power switch on the back of the unit. The *READY* and *ACTIVE*

LEDs on the the NI ELVIS II panel should briefly light, and the *READY* LED should stay on.

1. Power on the QNET Mechatronic Sensors board by switching on the NI ELVIS II prototyping board power switch. The *POWER* LED next to the switch should light up.
2. In the VI, click on the *Calibrate Sensor* tab to calibrate the output of the thermistor in terms of temperature (in °C).
3. From the *Device* drop-down menu, select the device name.
4. Re-run the VI.
5. Enter the -parameter you calculated for the thermistor using the *B* numeric control.
6. Test the accuracy of your calibration. To do this, gently place your fingertip on the sensor and verify that the correct fingertip temperature is displayed in the *Calibrated Output* waveform chart as well as the *Temperature (C)* thermometer indicator.
7. Stop the VI.
8. Switch OFF the QNET Mechatronic Sensors board as well as the NI ELVIS II.
9. Continue to the next section.

# Determine the Thermal Time Constant

1. For best results, your NI ELVIS II and the QNET Mechatronic Sensors board are should be at room temperature before continuing with the experiment.
2. Power on the NI ELVIS II using the power switch on the back of the unit. The *READY* and *ACTIVE*

LEDs on the the NI ELVIS II panel should briefly light, and the *READY* LED should stay on.

1. Power on the QNET Mechatronic Sensors board by switching on the NI ELVIS II prototyping board power switch. The *POWER* LED next to the switch should light up.
2. In the VI, click on the *Thermal Time Constant* tab.
3. From the *Device* drop-down menu, select the device name.
4. Re-run the VI.
5. Gently place your fingertip on the thermistor and examine the response using the *Calibrated Output* waveform chart.
6. After approximately 15 seconds, click the *Plot* button. The response of the sensor will be captured in the *Thermal Time Constant* waveform graph as shown in Figure 2.

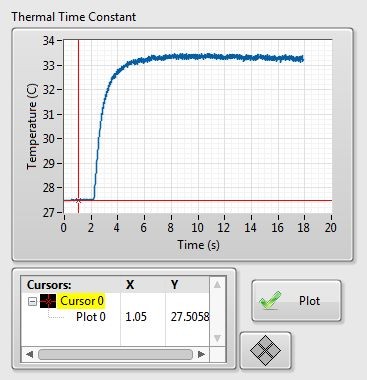


Figure 2: Typical thermistor response.

1. Using the *Cursor* tool and referring to the Concept Review, determine the thermal time constant of the thermistor.
2. If no further experiments are required, press the *Stop* button and power down the experiment.