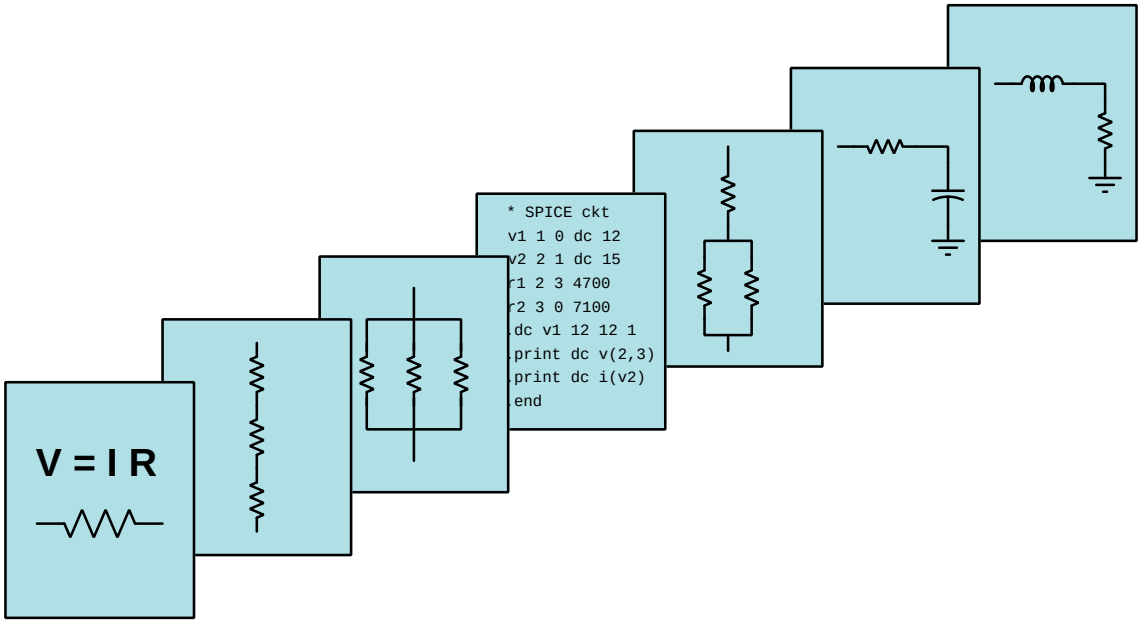
Modular Electronics Learning (ModEL) project



Data Acquisition Circuits

© 2020-2026 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 13 May 2026

This is a copyrighted work, but licensed under the Creative Commons Attribution 4.0 International Public License. A copy of this license is found in the last Appendix of this document. Alternatively, you may visit `http://creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons: 171 Second Street, Suite 300, San Francisco, California, 94105, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.

Contents

1 Introduction **3**
- 1.1 Recommendations for students . . . 3
- 1.2 Challenging concepts related to data acquisition . . . 5
- 1.3 Recommendations for instructors . . . 6

2 Case Tutorial **7**
- 2.1 Example: using Python to control a LabJack model U3 DAQ . . . 8

3 Tutorial **17**
- 3.1 Digital data acquisition . . . 17
- 3.2 Data acquisition units . . . 18
- 3.3 Practical measurement examples . . . 20
- 3.4 DAQ input bias currents . . . 27

4 Derivations and Technical References **31**
- 4.1 Ground in electric and electronic circuits . . . 32
- 4.2 Common-mode voltages and currents . . . 37
- 4.3 Ground-referenced signal sources . . . 44
- 4.4 Floating signal sources . . . 45
- 4.5 Elevated signal sources . . . 46
- 4.6 General-purpose opamp recommendations . . . 47
  - 4.6.1 TLV27x series . . . 47
  - 4.6.2 TLE214x series . . . 48
  - 4.6.3 TL07x / TL08x series . . . 48
  - 4.6.4 MCP600x series . . . 49
  - 4.6.5 LM158/258/358 . . . 49
  - 4.6.6 LM324 . . . 49
  - 4.6.7 LM741 and LM1458 . . . 50
- 4.7 Operational amplifier imperfections . . . 51
  - 4.7.1 Output voltage limits . . . 52
  - 4.7.2 Input voltage limits and latch-up . . . 53
  - 4.7.3 Input bias current . . . 55
  - 4.7.4 Input offset voltage . . . 59
  - 4.7.5 Frequency compensation . . . 60

4.7.6 Gain-bandwidth product . . . 60
4.7.7 Slew rate . . . 61
4.7.8 Common-mode rejection . . . 62
4.8 Protecting amplifier inputs from over-voltage . . . 63

5 Questions **69**
5.1 Conceptual reasoning . . . 73
5.1.1 Reading outline and reflections . . . 74
5.1.2 Foundational concepts . . . 75
5.1.3 Connecting strain gauges to a DAQ . . . 77
5.1.4 Connecting strain gauges to another DAQ . . . 78
5.1.5 Connecting photocells to a DAQ . . . 79
5.1.6 Connecting solar cells to a DAQ . . . 80
5.1.7 DC circuit monitor . . . 81
5.1.8 Measuring light and temperature with a DAQ . . . 82
5.2 Quantitative reasoning . . . 83
5.2.1 Miscellaneous physical constants . . . 84
5.2.2 Introduction to spreadsheets . . . 85
5.2.3 Measuring speed with a DAQ . . . 88
5.2.4 DDC light measurement . . . 89
5.3 Diagnostic reasoning . . . 90
5.3.1 Faulty temperature measurement system . . . 91
5.3.2 Networked DAQ modules . . . 93

A Problem-Solving Strategies **95**

B Instructional philosophy **97**
B.1 First principles of learning . . . 98
B.2 Proven strategies for instructors . . . 99
B.3 Proven strategies for students . . . 101
B.4 Design of these learning modules . . . 102

C Tools used **105**

D Creative Commons License **109**

E References **117**

F Version history **119**

Index **120**

Chapter 1

Introduction

1.1 Recommendations for students

Data acquisition units, or DAQs, are essentially analog-digital converter circuits packaged with digital interfacing and rudimentary analog signal conditioning circuitry to make a complete unit suitable for immediate use measuring and/or controlling analog signals using a computer. They are manufactured in a wide range of styles and capabilities.

This module focuses mostly on the analog signal connections between a DAQ and the external analog circuitry, the are which seems to be most problematic for end-users of DAQs.

Important concepts related to data acquisition units include **analog-digital conversion**, **sample speed**, **resolution**, digital communication **networks**, **common-mode voltage**, **Ohm's Law**, **single-ended** voltage signal, **differential** voltage signal, **unipolar** versus **bipolar** ranges, **rheometric** sensors, **noise** voltage, **ground loops**, cable **shields**, **bonding** conductors, **operational amplifiers**, and input **bias current**.

Here are some good questions to ask of yourself while studying this subject:

- How might an experiment be designed and conducted to test whether a voltage signal source was floating or not? What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?
- How might an experiment be designed and conducted to explore the concept of aliasing? What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?
- How does the number of *bits* in a digital *word* affect the number of possible values for that word?
- What does *resolution* mean for a converter circuit and why is it important?
- What does *speed* mean for a converter circuit and why is it important?
- What can a "differential" input DAQ do that a "single-ended" DAQ cannot?

- How can we tell whether or not a differential-input DAQ will be necessary for a given application?
- What is a "common-mode" voltage in a circuit?
- What is a "ground loop" and why it is undesirable?
- What purpose does a bridge network serve in an analog sensor application?
- How may two single-ended voltage inputs be used as a pseudo-differential voltage input?
- Why do bias currents exist at the input terminals of a DAQ?
- What is meant by the phrase "all grounds are not created equal"?

1.2 Challenging concepts related to data acquisition

The following list cites concepts related to this module's topic that are easily misunderstood, along with suggestions for properly understanding them:

- **Ground** – unfortunately the word "ground" has more than one accepted meaning in the field of electricity and electronics, and so one must be careful to define what is meant by "ground" before freely using this word. In this module the word "ground" primarily means an arbitrary point of reference for measuring electrical potentials, analogous to "sea level" as an arbitrary elevation from which to specify the height of a mountain peak or any other elevation we might wish to state.
- **Differential versus Ground-referenced signals** – voltage can only exist between two points, and so any voltage-based signal may either be referenced to ground or expressed between two ungrounded points (differential).
- **Common-mode voltage** – "common-mode" voltage is the mathematical average of ground-referenced voltages among multiple points. Literally, it is the amount of voltage, measured in reference to ground, that is shared in "common" with all of those terminals. Operational amplifiers, being differential-input devices, ideally ignore common-mode voltage and only respond to differential voltage.

1.3 Recommendations for instructors

This section lists realistic student learning outcomes supported by the content of the module as well as suggested means of assessing (measuring) student learning. The outcomes state what learners should be able to do, and the assessments are specific challenges to prove students have learned.

- **Outcome** – Demonstrate effective technical reading and writing

  Assessment – Students present their outlines of this module's instructional chapters (e.g. Case Tutorial, Tutorial, Historical References, etc.) ideally as an entry to a larger Journal document chronicling their learning. These outlines should exhibit good-faith effort at summarizing major concepts explained in the text.

  Assessment – Students present their "Conceptual Toolbox" Theory Journal entries listing important concepts applied in the Tutorial text(s). A "Conceptual Toolbox" serves as inspiration when solving problems, each concept suggesting a potential problem-solving action. Challenge students to collaboratively generate a Conceptual Toolbox for the day's topic during the class session and then use it when solving problems together, thus fostering the habit of reasoning from principles rather than recalling procedures!

- **Outcome** – Design a data acquisition circuit using a packaged DAQ

  Assessment – Sketch a schematic diagram of a circuit employing a DAQ to measure specified voltages in a power circuit; e.g. pose problems in the form of the "Connecting solar cells to a DAQ" Conceptual Reasoning question.

  Assessment – Sketch a schematic diagram of a circuit employing a DAQ to measure physical variables via sensors; e.g. pose problems in the form of the "Connecting strain gauges to a DAQ" and "Connecting photocells to a DAQ" Conceptual Reasoning questions.

  Assessment – Sketch a schematic diagram interfacing a DAQ to a sensor with proper mathematical scaling; e.g. pose problems in the form of the "Measuring speed with a DAQ" Quantitative Reasoning question.

- **Outcome** – Diagnose a faulted DAQ-based system

  Assessment – Determine the probability of various component faults in a measurement circuit given symptoms and measured values; e.g. pose problems in the form of the "Faulty temperature measurement system" Diagnostic Reasoning question.

- **Outcome** – Independent research

  Assessment – Locate DAQ unit datasheets and properly interpret some of the information contained in those documents including analog input ranges and resolutions, analog output ranges and resolutions, differential versus single-ended signal capability, discrete (digital) I/O count, sampling rate, etc.

Chapter 2

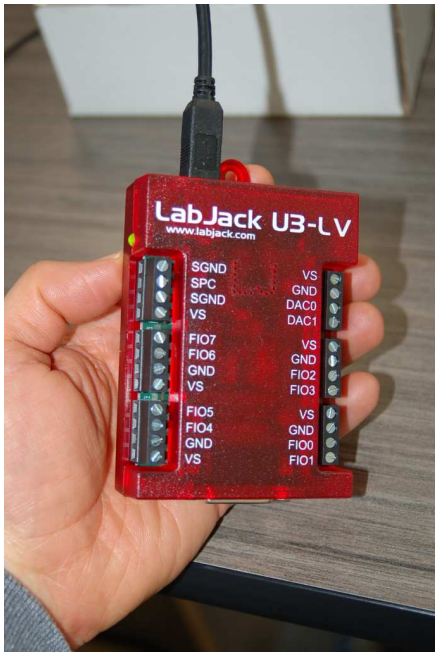
Case Tutorial

The idea behind a *Case Tutorial* is to explore new concepts by way of example. In this chapter you will read less presentation of theory compared to other Tutorial chapters, but by close observation and comparison of the given examples be able to discern patterns and principles much the same way as a scientific experimenter. Hopefully you will find these cases illuminating, and a good supplement to text-based tutorials.

These examples also serve well as challenges following your reading of the other Tutorial(s) in this module – can you explain *why* the circuits behave as they do?

2.1 Example: using Python to control a LabJack model U3 DAQ

A popular manufacturer of low-cost data acquisition (DAQ) hardware is *LabJack*, with their model U3 DAQ being a good entry-level device. This particular model interfaces with a personal computer via a USB cable and is also powered by the computer's USB port 5 Volt DC source:



Several "Flexible I/O" ports (FIO0 through FIO7) are provided which may be configured for either discrete ("digital") or analog input usage. A Python package called `LabJackPython` provides a cross-platform Python programming language library with built-in functions and methods enabling low-level control of LabJack U3, U6, UE9, and U12 DAQ devices. A free software package provided by LabJack called `UD Library Installer` provides multiple applications[1] and software drivers for quick and easy configuration and control of LabJack devices with little or no programming required. The USB drivers contained in LabJack's `UD Library` are necessary in order to use the `LabJackPython` Python package, so you will need to perform *two* software installations: the `UD Library Installer` from LabJack, and the `LabJackPython` using the Python `pip` installation utility operated from the computer's command line.

Follow the instructions from LabJack on how to install this software on your computer before attempting to run the Python examples shown in this section. The Windows command-line instruction I used to install `LabJackPython` package on my computer for these demos is as follows:

```
py -m pip install LabJackPython
```

[1] Among these applications are `Kipling`, `LJControlPanel`, and `LJStreamUD`.

From a Python interpreter shell, you may enter the following commands to control the LED visible on the outside of the U3 unit:

```
>>> import u3
>>> d = u3.U3()
>>> d.toggleLED()
```

The first line (`import u3`) instructs Python to import the U3 device library previously installed on your personal computer. The next line constructs a new object named `d` tied to the first U3 device the computer finds plugged in to its USB port. Note that this simple discovery technique only works when you have *one* LabJack device plugged in at a time! The third and final instruction toggles the binary state of the DAQ's single external LED. Executing that line more than once continues to toggle the LED, first off, then on, then off again, etc. This is a simple and effective test to check that your computer is actually able to communicate with the DAQ via Python commands.

This next test uses the Python interpreter to manually read the analog voltage applied to input FIO0:

```
>>> import u3
>>> d = u3.U3()
>>> d.configIO(FIOAnalog = 0x01)
>>> d.getAIN(0)
```

The `configIO()` method[2] sets the functionality of the FIO inputs using a hexadecimal value whose eight bits relate to FIO0 through FIO7, respectively. In this case we are configuring FIO0 to be an analog input and the rest digital, since `0x01` is equal to a binary value of `0b 0000 0001`. If we had wished to configure the first three FIO inputs to be analog, we would have used the hexadecimal value `0x07` which is binary `0b 0000 0111`. It is important we specify the analog nature of any FIOs before we attempt to read their analog voltage values, otherwise the next instruction will return an error message!

The `getAIN(0)` method then reads the analog voltage applied to FIO0 and returns a floating-point value scaled in actual Volts according to the calibration data stored within the U3 DAQ's memory. Incidentally, if you wish to view this calibration data, you may do so using the following command:

```
>>> d.getCalibrationData()
```

[2]In object-oriented programming languages such as Python, a *method* is a function attached to an object. In this case, the object is `d` which is the particular U3 DAQ we're communicating with, and `configIO()` is one of the methods associated with that object.

An extremely simple Python program using the `toggleLED()` method is shown below, also using Python's `time` library to create a two-second pause in the `while` loop's execution:

```
import u3
import time
d = u3.U3()
while (1):
  d.toggleLED()
  time.sleep(2)
```

Writing this short program using a text editor program, then saving it to a filename ending with the extension `.py` makes it immediately recognizable to the computer's operating system as a Python source code file. Invoking that source file, either by double-clicking on its icon or by specifying it by name at a command-line interface (e.g. `py myfile.py`), will cause Python to execute these instructions.

A simple Python program reading the first three FIO analog voltage inputs (and re-reading them with every press of the "Enter" key on the controlling computer) is shown here ready to save to a file ending in `.py` and executed by double-clicking on the Windows icon for that file:

```
import u3
d = u3.U3()
d.configIO(FIOAnalog = 0x07) # Sets first three FIOs to analog mode

while (1):
  a = d.getAIN(0)  # Reads FIO0 as analog input and stores in "a"
  b = d.getAIN(1)  # Reads FIO0 as analog input and stores in "b"
  c = d.getAIN(2)  # Reads FIO0 as analog input and stores in "c"
  print("FIO0 =", a, "Volts")
  print("FIO1 =", b, "Volts")
  print("FIO2 =", c, "Volts")
  input()  # Pauses until user presses Enter
```

An example of this program's output when run is shown here, pressing the "Enter" key three times to get three readings of the analog inputs:

```
FIO0 = 0.438432256 Volts
FIO1 = 0.37528848000000004 Volts
FIO2 = 0.35026924800000003 Volts

FIO0 = 0.41698720000000006 Volts
FIO1 = 0.36397025600000005 Volts
FIO2 = 0.34490798400000006 Volts

FIO0 = 0.41758289600000004 Volts
FIO1 = 0.36397025600000005 Volts
FIO2 = 0.344312288 Volts
```

A more detailed and sophisticated way to read analog inputs on the model U3 DAQ makes use of the `getFeedback()` method. This method reads the raw analogto-digital converter (ADC) count value as an integer number.

```
import u3
d = u3.U3()
d.configIO(FIOAnalog = 0x07) # Sets first three FIOs to analog mode

while (1):
  a_raw, = d.getFeedback(u3.AIN(0))  # Reads FIO0 counts and stores in "a_raw"
  b_raw, = d.getFeedback(u3.AIN(1))  # Reads FIO1 counts and stores in "b_raw"
  c_raw, = d.getFeedback(u3.AIN(2))  # Reads FIO2 counts and stores in "c_raw"
  print("FIO0 =", a_raw, "counts")
  print("FIO1 =", b_raw, "counts")
  print("FIO2 =", c_raw, "counts")
  input()  # Pauses until user presses Enter
```

Note the comma symbols prior to the = assignment used to send the results of the `getFeedback()` method to the variables `a_raw`, `b_raw`, and `c_raw`. This is because the `getFeedback()` method returns a *list*[3] rather than a single variable. The list in this case happens to contain only one entry, and placing a comma before the "equals" assignment operator extracts that one entry's value as a regular integer value. If not for the commas, variables `a_raw`, `b_raw`, and `c_raw` would all become lists themselves.

An example of this program's output when run is shown here, pressing the "Enter" key three times to get three readings of the analog inputs:

```
FIO0 = 12048 counts
FIO1 = 10064 counts
FIO2 = 9520 counts

FIO0 = 11584 counts
FIO1 = 9808 counts
FIO2 = 9408 counts

FIO0 = 11568 counts
FIO1 = 9904 counts
FIO2 = 9440 counts
```

[3]Lists in Python are similar to arrays or structures in C or C++, namely collections of variables under a single name for easy access.

`LabJackPython` also supports discrete, or digital, I/O control for any FIO channels in digital mode rather than analog. For example, we may run the following commands manually from a Python interpreter to read the status of FIO6:

```
>>> import u3
>>> d = u3.U3()
>>> d.configIO(FIOAnalog = 0x01)
>>> d.getFeedback(u3.BitDirWrite(6,0))   # Sets the direction of FIO6 as input
>>> d.getFIOState(6)   # Reads the digital logic state of FIO6
```

In this example, the method `configIO(FIOAnalog = 0x01)` setting FIO0 to analog input mode is really just ensuring all the other FIO channels are set to digital. Next we have the `getFeedback(u3.BitDirWrite(6,0))` instruction setting (writing) the direction bit to a 0 value for FIO6 which configures that FIO to be an input rather than an output. The `getFIOState()` method reads the logical state of input FIO6 and displays it as either a 1 or a 0 depending on what the FIO6 input terminal is connected to.

Next, we will examine the Python instructions needed to configure FIO6 as a digital output rather than a digital input, and then setting its output value to be 1 and 0 in turn:

```
>>> import u3
>>> d = u3.U3()
>>> d.configIO(FIOAnalog = 0x01)
>>> d.getFeedback(u3.BitDirWrite(6,1))   # Sets the direction of FIO6 as output
>>> d.setFIOState(6, 1)   # Writes the digital logic state of FIO6 to a 1 value
>>> d.setFIOState(6, 0)   # Writes the digital logic state of FIO6 to a 0 value
```

Note that in all these examples it is not necessary to re-execute the `import u3` or `d = u3.U3()` or `d.configIO(FIOAnalog = 0x01)` instructions so long as they have already been executed during the same Python interpreter session. They are shown in these test examples assuming a fresh start of Python each time.

LabJack model U3 DAQs also contain 32-bit hardware counters which may be read via Python. These counters may be configured to read the digital (high/low) status of any FIO input terminal, incrementing by one every time the logic level transitions from a "high" state to a "low" state (i.e. a *negative-edge* clocked counter). The following Python program demonstrates how to read a counter inside of a model U3 DAQ:

```
import u3
d = u3.U3()

# Enables Counter 0 and sets FIO0-FIO3 to analog mode (0x0F = 0b 0000 1111)
d.configIO(EnableCounter0 = True, FIOAnalog = 0x0F)

while (1):
  count, = d.getFeedback(u3.Counter0(Reset = False))
  print("Counter value =", count)
  count, = d.getFeedback(u3.Counter0(Reset = True))
  input()  # Pauses until user presses Enter
```

Note how the first `getFeedback()` method does not reset the counter, but the second one does. This causes the counter's value to be re-set to zero just prior to the program waiting for the user's "Enter" keystroke. If no negative-edge pulses are detected before pressing "Enter", the next displayed counter value will be zero. However, if the input terminal receives some pulses prior to the "Enter" keystroke then the next value printed to the console after pressing "Enter" will register the number of pulses accumulated.

An example of this program's output when run is shown here, pressing the "Enter" key three times to get display four readings of the counter:

```
Counter value = 0

Counter value = 318

Counter value = 0

Counter value = 771
```

While it would be possible to write a Python program to repeatedly read the digital status of any FIO terminal and increment a Python variable to count how many times that digital signal has pulsed, this strategy would miss pulses whose frequency exceeded the Python program's loop execution rate. The beauty of a hardware counter is that the DAQ keeps its own count of pulses regardless of whether that counter value is read by a computer or not. Thus, when the Python program reads the counter's value, it is sure to read the true value of pulses captured in real time.

The model U3's counter is assigned to any FIO terminal other than FIO0 through FIO3 by means of an *offset* value which defaults to four (4). So, by default, FIO4 is the terminal driving the enabled counter. However, if we wish to assign the counter to sense logic states at a different terminal, all we have to do is change the offset value. The following program shows how this is done within the `configIO()` method, in this particular case setting that offset value to five (5) which makes FIO5 the terminal where any negative pulse edges will increment the counter:

```
import u3
d = u3.U3()

# Enables Counter 0 and sets FIO0-FIO3 to analog mode (0x0F = 0b 0000 1111)
d.configIO(EnableCounter0 = True, TimerCounterPinOffset = 5, FIOAnalog = 0x0F)

while (1):
  count, = d.getFeedback(u3.Counter0(Reset = False))
  print("Counter value =", count)
  count, = d.getFeedback(u3.Counter0(Reset = True))
  input()  # Pauses until user presses Enter
```

LabJack model U3 DAQs support up to *two* hardware counters, called counter 0 and counter 1. In the following program[4] we see both counters being read, with the offset value set to five (5) which means counter 0 is reading the digital status of FIO5 and counter 1 is reading the digital status of FIO6:

```
import u3
d = u3.U3()

# Enables Counters 0 & 1 and sets FIO0-FIO3 to analog mode (0x0F = 0b 0000 1111)
d.configIO(EnableCounter0 = True, EnableCounter1 = True,
           TimerCounterPinOffset = 5, FIOAnalog = 0x0F)

while (1):
  count0, = d.getFeedback(u3.Counter0(Reset = False))
  count1, = d.getFeedback(u3.Counter1(Reset = False))
  print("Counter 0 value =", count0, "  Counter 1 value =", count1)
  count0, = d.getFeedback(u3.Counter0(Reset = True))
  input()  # Pauses until user presses Enter
```

This program resets the value of counter 0 before every "Enter" keystroke from the user, but allows counter 1 to continue accumulating.

[4] Here the `configIO()` method's line is broken into two simply for readability on the printed page of this document.

Chapter 3

Tutorial

3.1 Digital data acquisition

Digital computing technology is ubiquitous in our modern world: from mobile telephones to personal computers to digital entertainment systems, most readers of this text are surrounded by digital computers. If we wish to use this digital computing power to measure, archive, and/or analyze data from the physical world, we need some way for digital computers to *interface* with sensors capable of sensing physical variables. Such interface units are often referred to as *data acquisition units*, or *DAQ*s.

DAQs provide signal conditioning necessary to make sensor signals compatible with the digital logic circuitry of a computer. This includes analog-to-digital converters (ADCs) to convert analog electrical signals into digital number values, digital-to-analog converters (DACs) to convert digital number values into analog electrical signals, and often optical isolators in order to provide a safe measure of electrical isolation between the sensitive circuitry of the digital computer and the sometimes hazardous electrical nature of physical sensors. DAQs also provide some means of exchanging digital data with the computer, usually some for of serial network such as USB or Ethernet.

DAQs typically offer multiple "channels" of input and/or output. For example, a DAQ offering four channels of analog input would be able to sense analog electrical signals from as many as four different sensors, using either four ADCs or one ADC multiplexed to the four inputs to digitize those analog signals one at a time. A DAQ with two analog output channels would have either two DACs or a single DAC multiplexed to those two channels to generate analog electrical signals at the command of the digital computer.

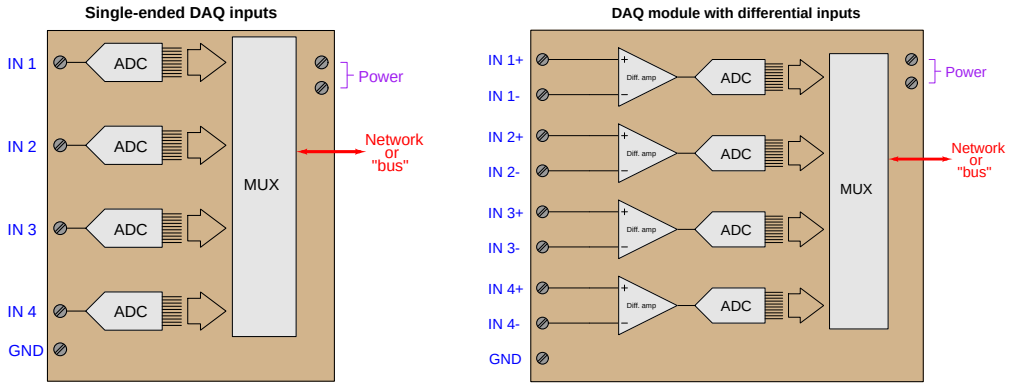
DAQs are often used in laboratory and test environment to temporarily interface digital computers to analog devices for the purpose of taking measurements of voltage and/or current signals over time. Research and development (R&D) laboratories, for example, make extensive use of DAQs in their work to develop and improve system designs, the DAQ hardware being the "eyes and ears" of digital computers to interact with the analog world.

3.2 Data acquisition units

When an operational amplifier front-end is combined with an analog-digital converter as well as digital interface circuitry necessary to network with a computer, the result is called a *data acquisition unit*, or *DAQ*. These are typically multi-channel instruments designed to sample more than one analog voltage signal (and/or current signal) simultaneously. Some DAQs have analog *output* capability as well, although this is less common. DAQs vary widely in design and in options:

- **Sample speed** – This is the frequency at which the analog-digital converter(s) sample new measurements. Rather than use the unit of Hertz which is common for frequency, sample rate is typically specified in the unit of *samples per second*. Modern DAQs offer extremely fast sampling speeds, and so you will often find this specified as kSa/s (kilo-samples per second), MSa/s (mega-samples per second), or GSa/s (giga-samples per second). The sampling rate of any ADC must be significantly faster than the highest frequency of interest in order to capture sufficient details of that wave. Failure to sample fast enough will result in *aliasing*.
- **Resolution** – Any form of analog-digital conversion resolves the analog signal range into a discrete number of *counts* defined by the number of binary bits in the digital word. For example, a 14-bit DAQ channel divides the analog signal range into 2^{14} discrete steps (i.e. a count value ranging between `0x0000` and `0x3FFF`, or 0 to 16383 decimal, inclusive). Resolution may be specified as the number of bits, or alternatively as the smallest resolvable analog signal increment. For example, a 14-bit DAQ channel representing an analog signal range of 0 to 4 Volts DC would have a resolution of $\frac{4}{2^{14}-1}$ or 0.24416 milliVolts.
- **Network types** – The digital network connection between the DAQ and a computer may be serial (e.g. EIA/TIA-232), USB, Ethernet, a parallel-style computer bus standard such as PCI, or even wireless (radio).
- **Analog ranges** – Analog-digital conversion circuits are necessarily limited in the range of analog signals they may convert, and so matching the DAQ's range to the analog signal application is critically important. The DAQ must obviously have sufficient range, but it is also important that the DAQ's analog range not be too large or else it will be a waste of resolution. For example, a DAQ range of 0 to 10 Volts for an application only requiring a 0 to 0.5 Volt signal would only use $\frac{1}{20}$ of the DAQ's analog-digital conversion range, which means the practical resolution would not be as good as what the number of conversion bits for the DAQ might suggest. 14 bits of resolution over a 10 Volt range is 0.6104 milliVolts per count, but if used on a 0.5 Volt range this means only 819 counts from end to end. A 10-bit DAQ with a range of 0 to 0.5 Volts would actually perform better!
- **Unipolar versus Bipolar** – If the analog signal range begins at zero and ends at a positive value, the range is said to be *unipolar*. A *bipolar* range, by contrast, has a mid-scale count value at zero (analog) signal, with the low end of the range being a negative analog quantity and the high end being a positive quantity.
- **Signal types** – DAQ inputs (and outputs, for those with that capability) come in two basic signal types: *single-ended* and *differential*. Single-ended is where the analog voltage in question is measured with reference to ground, while differential is where the signal exists between two dedicated terminals on the DAQ.

That last point about input types merits further elaboration. Sample diagrams of four-channel analog input DAQs appear below to contrast these two input signal types:



Variations on these two themes exist as well. For example, some DAQs offer the feature of a separate analog "sense" terminal electrically distinct from ground that serves as the common reference for all channels. Thus, it is useful when interfacing with multiple signals sharing the same "common-mode" voltage.

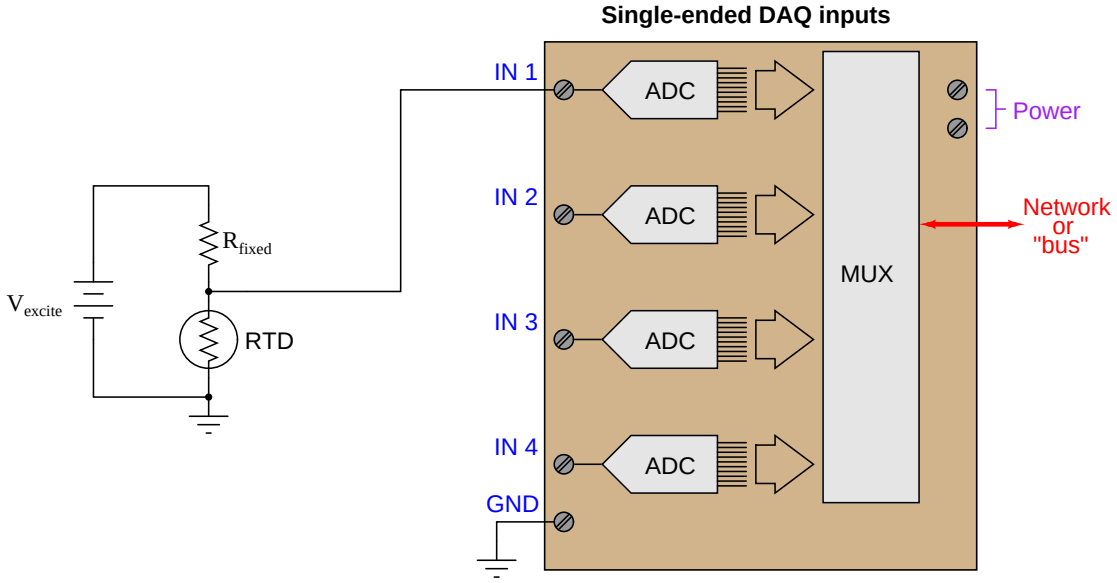
It is often helpful to think of DAQ input terminals as being the "red" and "black" test leads of a multimeter. For the single-ended DAQ each of the "IN" terminals is a "red" test lead while the "GND" terminal is the "black" lead (i.e. as though multiple meters share the same common black test lead but each have their own dedicated red lead); for the differential DAQ the "red" and "black" correspond to the "+" and "−" terminal pairs of each input channel. When deciding to connect a DAQ to any circuit under test, first ask yourself the question *"how would I connect a multimeter to measure the same?"* and then connect the DAQ in the same manner.

As one would expect, each of these DAQ options comes at a price. High sampling speed, high resolution, fast networking, unusually large or small signal ranges, and differential signal capability all equate to more complexity and higher cost.

3.3 Practical measurement examples

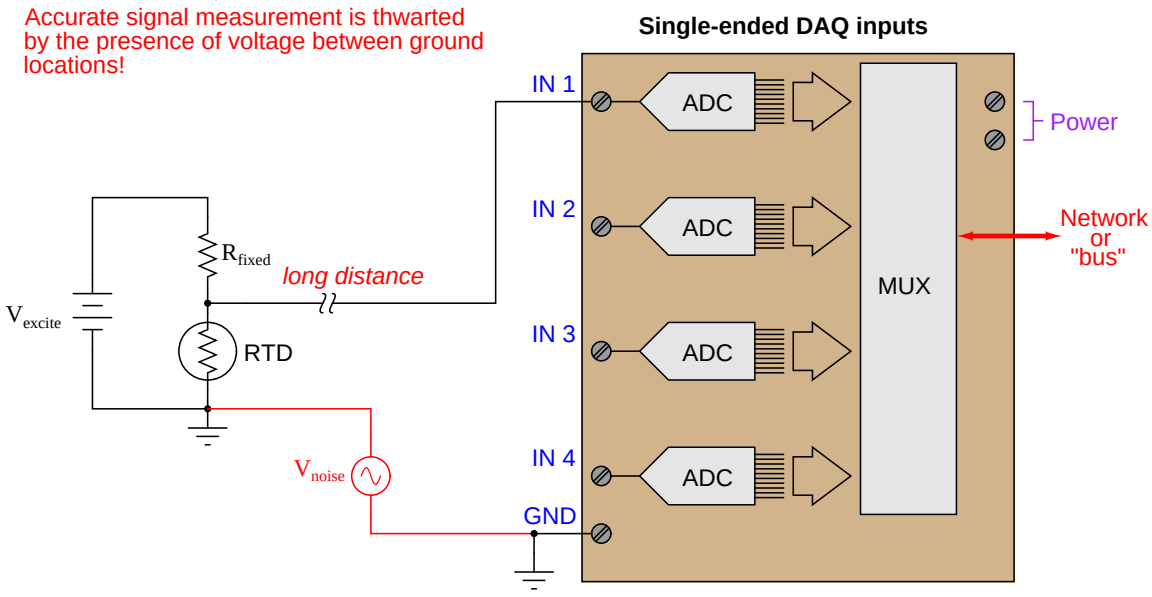
Perhaps the greatest area of misunderstanding with DAQs revolves around how to connect the unit to the signal source. This may sound like a trivial task, but given the existence of ground-referenced, floating, and elevated signal sources it definitely is not. This section will explore some of the necessary design decisions for proper source/DAQ interfacing.

Let's begin with a ground-referenced signal source: the simple voltage divider circuit using a *Resistive Temperature Detector* or *RTD* as the sensing element. An RTD is similar to a thermistor with a positive temperature coefficient, except that it is much more linear and accurate than a thermistor. If the divider is located close to the data acquisition (DAQ) analog input device, a single wire will suffice for connecting the two:



Each analog-to-digital converter (ADC) inside the DAQ unit is built to digitize a voltage with reference to ground, which is exactly the type of signal generated by our simple RTD voltage divider circuit. This type of DAQ analog input is called *single-ended*, and it is generally the default configuration on inexpensive DAQ units.

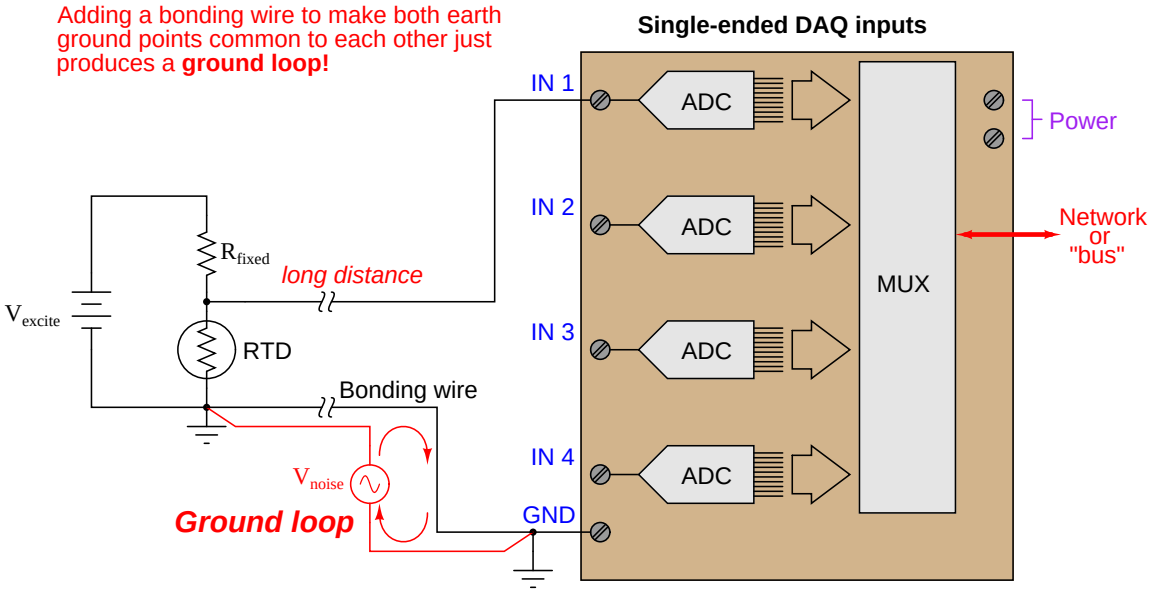
We cannot simply connect a single-ended DAQ input to a ground-referenced signal source using a single wire, however, if the two are located far apart from each other:



The problem here is that "all grounds are not created equal" over significant distances. If the ground path is literally through the earth (soil), there will be a myriad of noise sources adding spurious voltage to the measured signal: lightning strikes, ground leakage currents from AC power devices, and other sources of "noise" potential will become a part of the signal loop. Even continuous-metal ground paths over long distances can incur voltage drops significant enough to corrupt precision signal measurements. The grounding conductors used in AC power systems, for example, while continuous (no interruptions) between all points of use in the power system still drop enough millivoltage to significantly compromise instrumentation signals.

In essence, what appears to be a ground-referenced voltage signal source is actually an *elevated* voltage signal source, with the common-mode voltage being "noise" present between the two different ground locations.

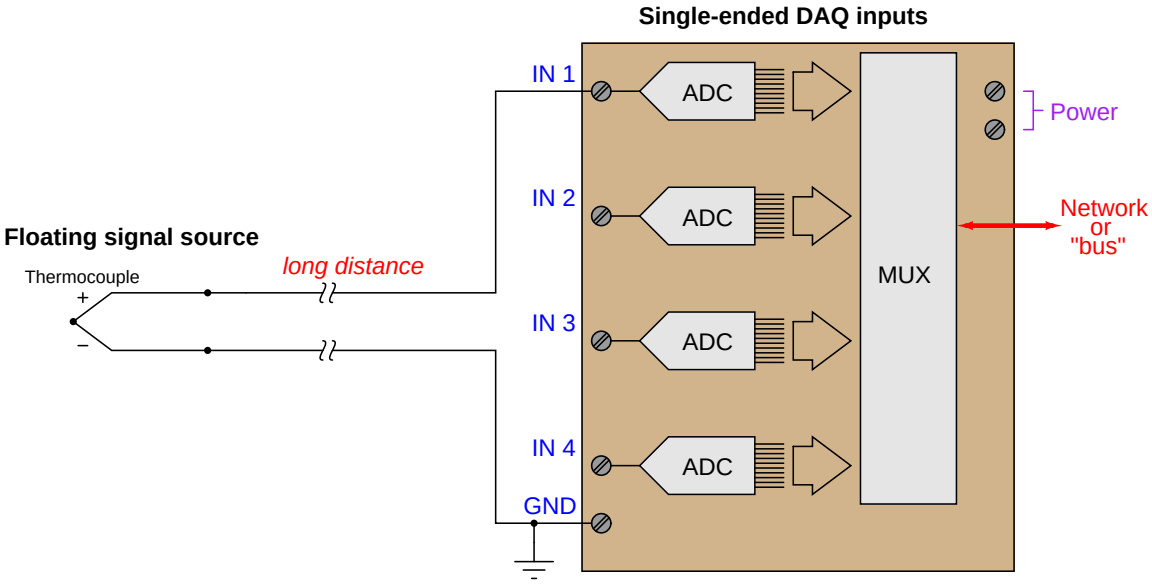
At first, it would seem that connecting the two grounds together with a dedicated length of wire would solve the differential ground problem. Unfortunately, it does not. The noise sources intercepted through the earth are often of significant power, which just means any conductor stretched between two different earth grounds may end up carrying substantial amounts of current, and subsequently dropping voltage along the way due to wire resistance by Ohm's Law ($V_{noise} = I_{ground}R_{wire}$).



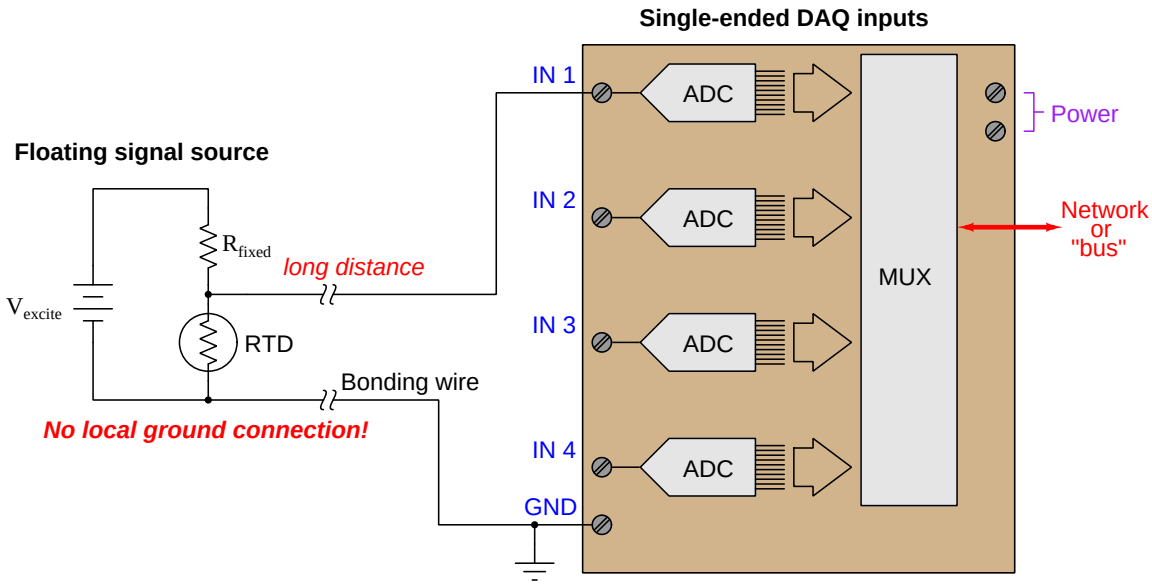
This is called a *ground loop*, and it should be avoided in signal circuits at all cost! Not only may the ground currents still produce significant noise voltage in the measurement circuit, but the ground currents may even become strong enough to damage the bonding wire! Ground loops are often unintentionally formed when the *shield* conductor of a long signal cable is earth-grounded at both ends.

A reasonable question to ask at this point is, "What constitutes a *long* distance when connecting ground-referenced signal sources to DAQ modules?" A simple rule to follow is that one cannot rely on ground points to be electrically common to each other (at least not common enough for precise signal-measurement purposes) if those points lie outside the same metal enclosure. If the signal source and DAQ analog input physically dwell inside the same metal enclosure, you can probably rely on the ground points being truly common to each other. If not, you should use some other means of measuring the signal. When in doubt, a sure test is to actually measure the potential difference between the source and DAQ ground points to see if any exists, being sure to check for AC noise voltage as well as DC.

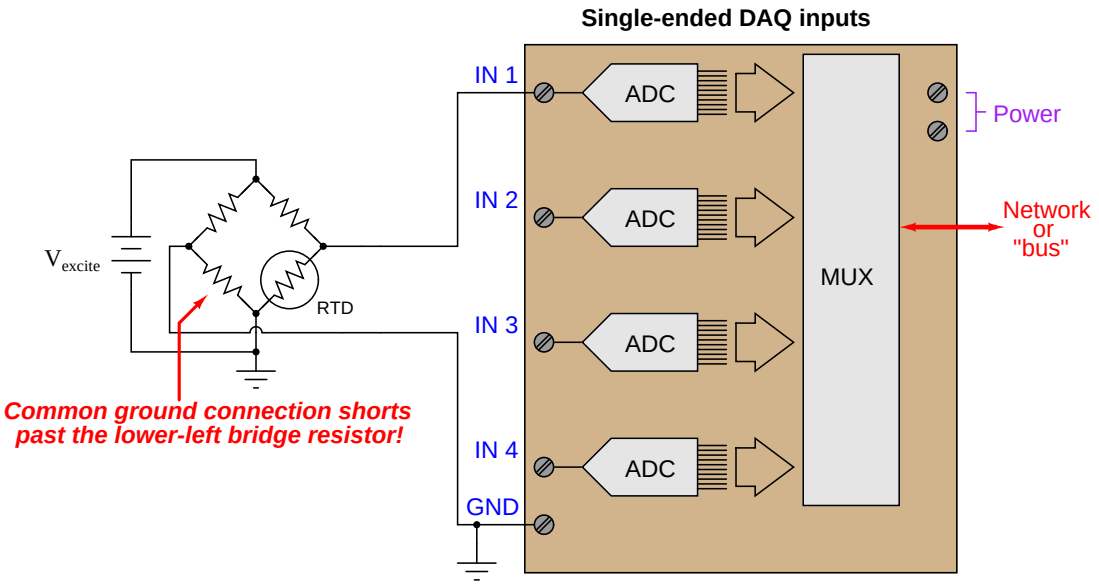
Here is an example of a single-ended DAQ module successfully measuring a voltage signal source located far away:



Since an ungrounded thermocouple has no connection whatsoever to any ground, there will be no ground loop when connected to a single-ended DAQ input. The same is true for battery-powered sensor circuits lacking connection to earth ground:



Single-ended analog DAQ inputs have trouble measuring elevated signal voltages regardless of distance. Here, we see an example where someone has tried to connect a single-ended DAQ input to a grounded-excitation RTD bridge[1], with common-mode voltage equal to one-half the excitation source voltage. The result is disastrous:

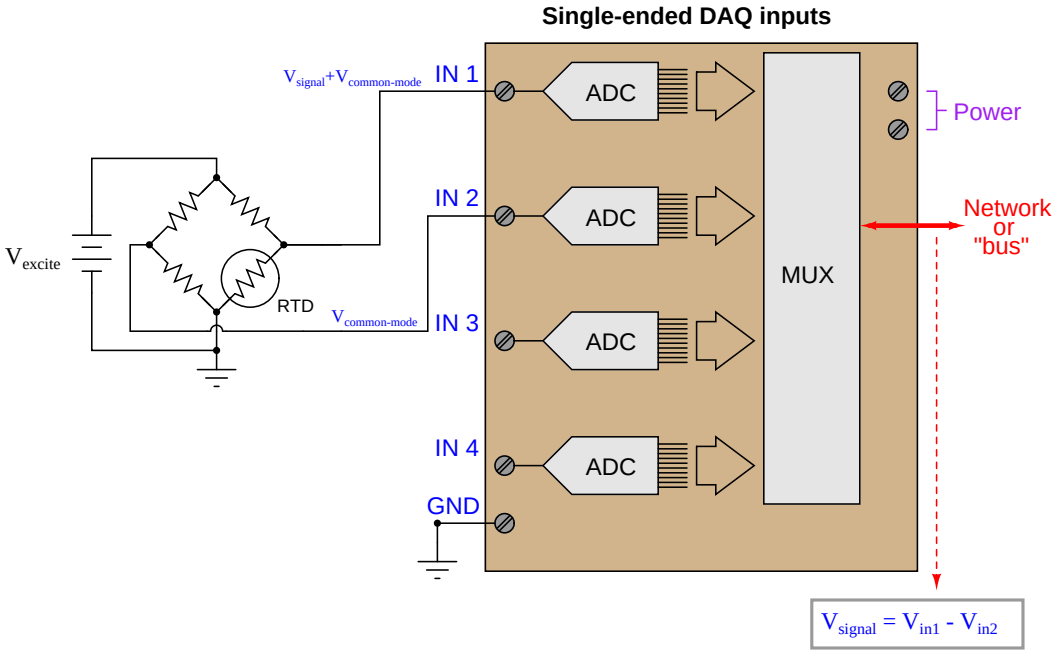


If you follow the bottom wire in this diagram carefully, you will see how it effectively jumpers past the lower-left resistor in conjunction with the two ground connections. Yet, eliminating the wire simply substitutes one problem for another: without the bottom wire in place, the voltage seen by input channel 1 on the DAQ will be $V_{signal} + V_{common-mode}$ rather than V_{signal} all by itself.

[1] A *bridge* network is essentially a pair of voltage dividers energized by the same source, the difference in voltage between the two dividers being zero when both dividers have the same ratio. They are used extensively for rheometric (resistance-based) sensors because they allow the output voltage signal to be zero for a non-zero sensor resistance.

A clever way to solve the problem of measuring elevated signal sources is to use *two* analog DAQ channels: one to measure $V_{signal} + V_{common-mode}$ and the other to measure $V_{common-mode}$, and then digitally subtract one measurement from the other. Thus, two single-ended input channels may function as one *differential* input channel.

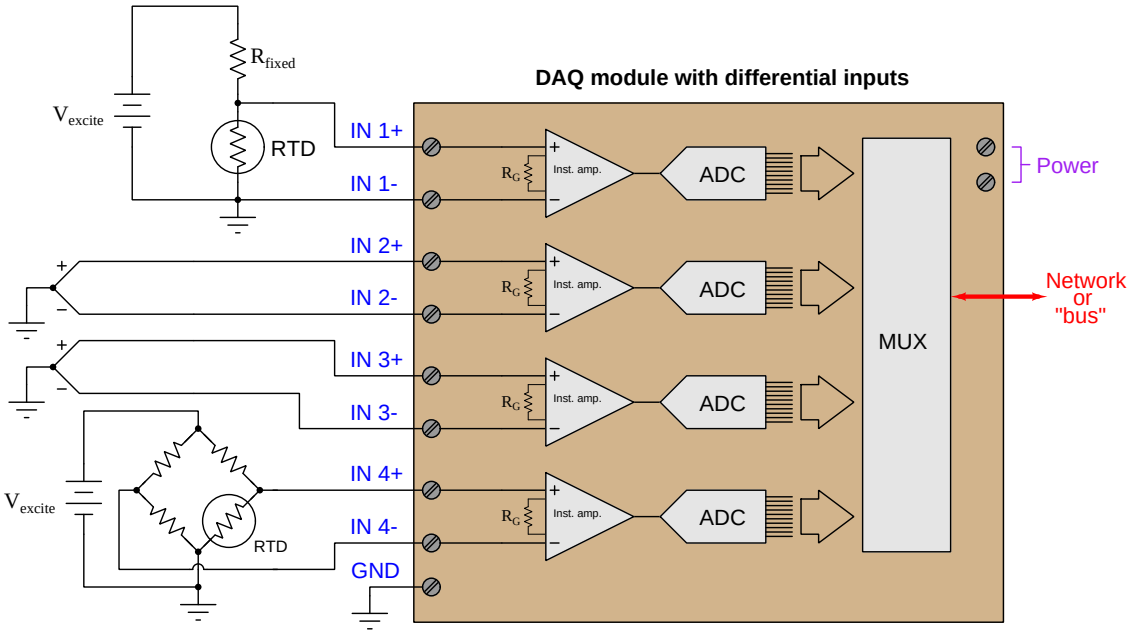
Here, we see an example of this measurement technique applied to the grounded-excitation RTD bridge:



The subtraction of the two channels' digitized values may take place in a controller separate from the DAQ module, or within the DAQ module if it is equipped with enough "intelligence" to perform the necessary calculations.

An added benefit of using this dual-channel method is that any noise voltage existing between the signal ground and the DAQ ground will be common to both channels, and thus should cancel when the two channels' signals are mathematically subtracted.

A more versatile hardware solution for measuring any form of voltage signal is a DAQ equipped with true *differential* input channels. Here, each ADC is equipped with its own instrumentation amplifier, measuring the difference in potential between two ungrounded input terminals:



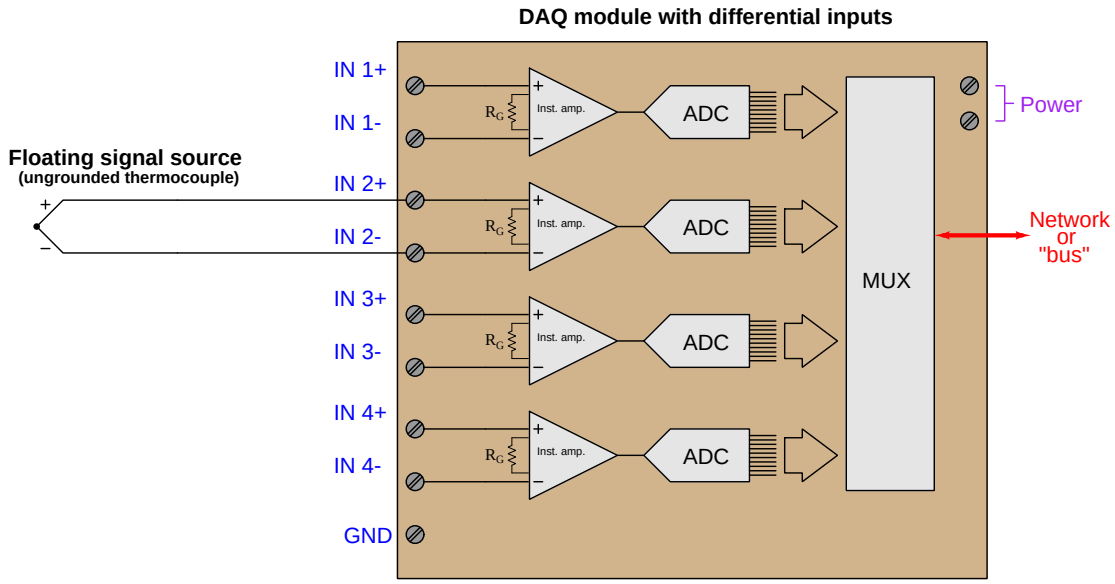
In this example we see one DAQ module measuring four different voltage signal sources, with no interference from differential ground potentials or between the sources themselves. Each input channel of the ADC is electrically independent from the rest. The only limitation to this independence is a certain maximum common-mode voltage between the signal source and the DAQ's own power supply, determined by the design of the instrumentation amplifiers inside the DAQ module.

In summary, we may list some essential rules to follow when connecting analog DAQ inputs to voltage signal sources:

- Ground points in different locations may not actually be common (enough) to each other
- Never create ground loops through wiring (any conductor connected to ground at each end)
- Beware of common-mode (elevated) signal voltages

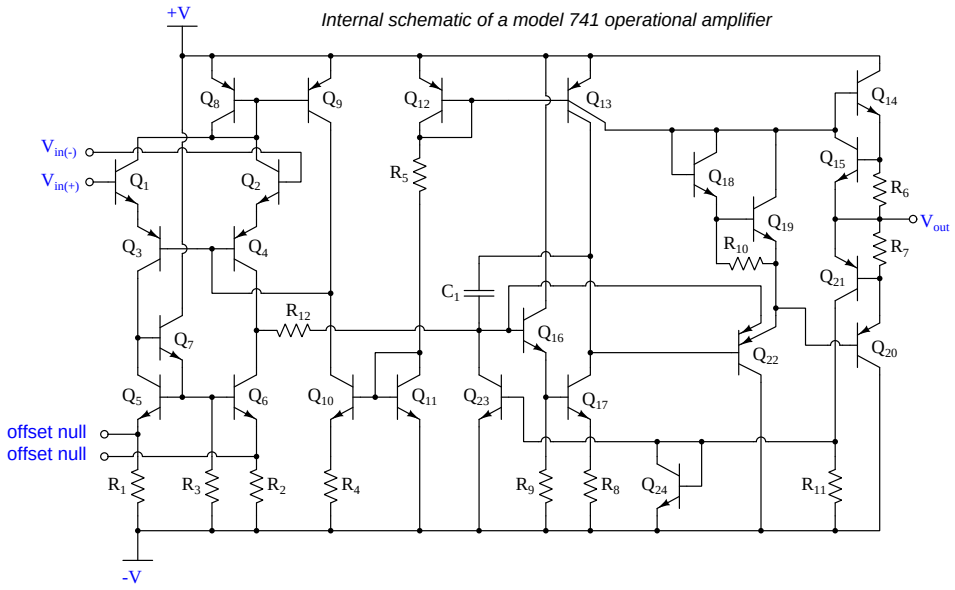
3.4 DAQ input bias currents

One important practical consideration for DAQ differential input channels is that there must be some path for the internal instrumentation amplifiers' bias currents or else the channel will not function properly. This poses a problem where we intend to connect a differential analog input channel to a floating signal source like this:

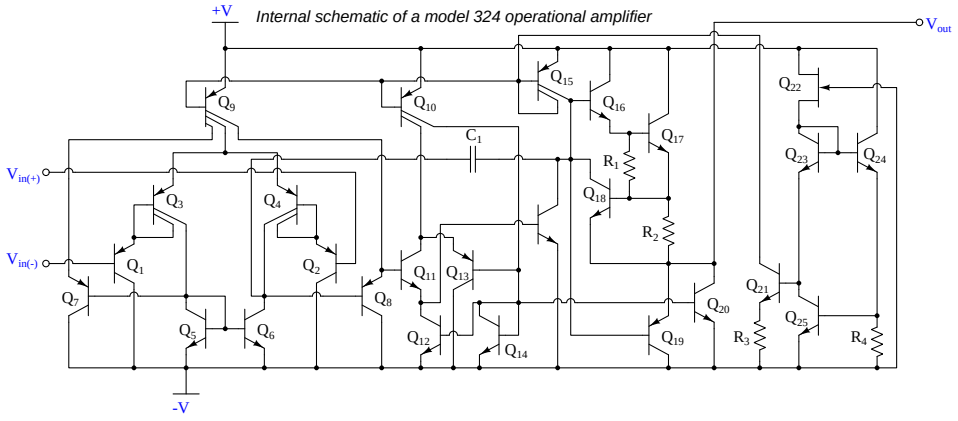


One of the "simplifying assumptions" students learn about operational amplifier circuits is that the input terminals of an opamp draw negligible current. While this may be close enough to the truth when performing calculations on an opamp circuit, it is not true in an absolute sense. All opamps exhibit some amount of *bias current* at their input terminals, small as these currents may be. Without a complete path for these currents, the input transistor stage of the operational amplifier will not be properly biased, and the amplifier will fail to work as designed.

An examination of operational amplifier internal circuitry helps explain why input bias currents exist. Consider the legacy model 741 opamp, with a pair of NPN transistors at its inverting and non-inverting input terminals (Q_1 and Q_2). In order for these two transistors to be in their active modes, there *must* be some small amount of base current entering the terminals of each:

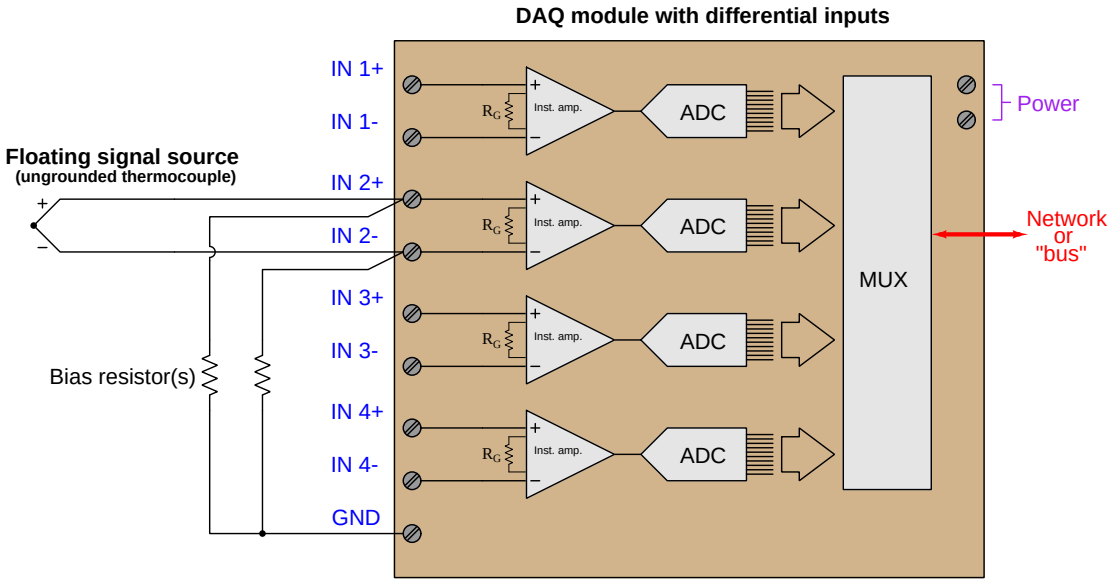


For contrast, consider the internal circuitry of the legacy model 324 opamp, with a pair of PNP transistors at its inverting and non-inverting input terminals (Q_1 and Q_2). Again, in order for any bipolar junction transistor to actively conduct electricity there must be some base current present. In the case of the 324 opamp this necessitates some current *leaving* each of its input terminals:



A DAQ using opamps similar to the 741 therefore require pathways for input bias current from some potential more positive than the opamps' negative power supply rail. A DAQ using opamps similar to the 324 require input bias current pathways to some potential less positive than the opamps' positive power supply rail.

If we happened to be using a DAQ with differential analog inputs, and knew that DAQ's internal operational amplifiers happened to *source* input bias currents (like the 324 opamp), we would need to connect bias resistors to ground when sensing voltage of a floating signal source as shown below:



Note that many DAQs provide bias current pathways internally to avoid the user having to attach extra resistors as shown here. As usual, the manufacturer's literature should give guidance on whether or not this is necessary.

Chapter 4

Derivations and Technical References

This chapter is where you will find mathematical derivations too detailed to include in the tutorial, and/or tables and other technical reference material.

4.1 Ground in electric and electronic circuits

Ground and *grounding* within both electric power and electronic circuits are commonly misunderstood topics, largely because these terms can often have very different meaning depending on context. Here we will explore some of these different meanings and hopefully eliminate some of the confusion surrounding them.

The very first concept we must understand in order to comprehend what "ground" is in any circuit is that *voltage* is fundamentally a *relative* measurement of potential energy different *between two points*. This is really true of all forms of potential energy, because "potential" means energy that has not yet been put to work setting matter into motion, implying a final state yet to be realized. When we say an electrical charge carrier possesses potential energy (i.e. electrical potential), this means it *could* (potentially) release energy to do work if it were to fall into a lower state of energy. The difference in energy held by that electric charge carrier from its original (high-potential) state to some other lower-potential state will be the amount of energy released by that charge carrier to do work, with that amount of work done dependent on both the charge carrier's initial energy level and its final energy level. *Voltage* is the term we use to describe such a difference in electrical potential.

We say that voltage always exists "between two points" or "between two locations" in a circuit because voltage is the amount of potential energy either lost or gained by electric charge carriers from one position in a circuit to another. It is impossible to quantify how much energy will be gained or lost by charge carriers unless we know both the starting point and the ending point for those charge carriers. In order to quantify the absolute amount of potential energy possessed by anything, we would have to know where or how it could lose every bit of its stored energy to be left with zero capacity to do work[1]. This is really unknowable from a practical perspective, and so the best we can do is quantify gains or losses in potential energy between different states.

This fact about voltage is also why we label it with *pairs* of "+" and "−" symbols, the "+" symbol representing the point of *higher potential* and the "−" symbol representing the point of *lower potential*. We cannot place an absolute value on electrical potential at any single location because no one can say where the location of zero energy is in the universe for electric charges. The best we can do is to rate one location as having higher or lower electrical potential than another, quantifying only the difference in potential between those two locations.

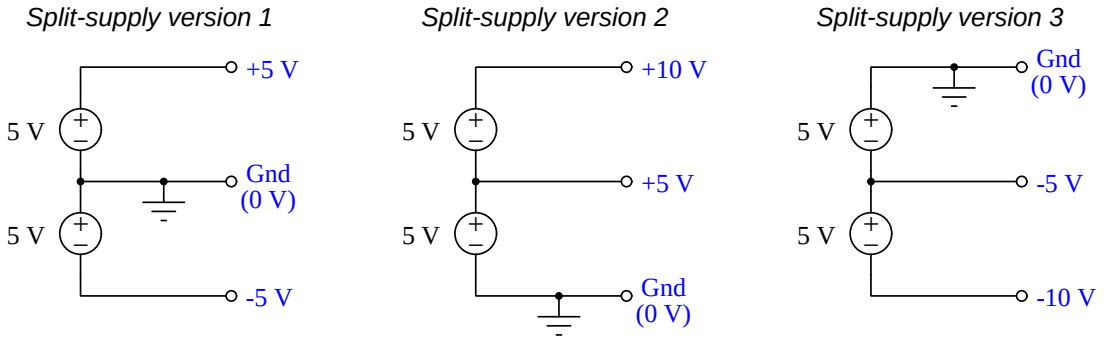
Now that we have reviewed this property of voltage and of potential energy in general, we are ready to explore what is meant by *ground* and *grounding*.

[1] Using gravitational potential energy as an example, a weight hoisted above the Earth's surface certainly possesses potential energy, and we might be tempted to quantify this potential energy by simply measuring its weight and its height above ground level and then multiplying those two quantities (i.e. potential energy equals force times distance). However, there are more places this weight could *potentially* fall than just ground level. That same weight, hoisted above the Earth's surface to the same height, will be assessed to have a far greater amount of potential energy if we consider it potentially falling down a deep hole rather than just falling to ground level. For that matter, one could even argue the weight has a *negative* quantity of potential energy with respect to what it would possess at the summit of the highest mountain! Potential energy, defined as the amount of work that *could* be done, necessarily depends on a system's initial condition *and* any number of different final conditions of which none are known to be truly zero-energy.

How do we rate the height of mountain peaks? Certainly, we may rate one mountain peak as being some number of meters higher or lower than another, but generally what people do instead is measure a mountain's peak height compared to *Mean Sea Level*. Since water is a liquid, and liquid tend to settle at equal heights under the influence of gravity when they are free to move about, the level of oceans throughout the world provide a relatively stable reference point for measurements of altitude. If we imagine a world without oceans, perhaps one with nothing but dry land and no open bodies of water anywhere, people living at different locations in this world would likely choose some flat-land reference points local (to them) for their measurements of nearby mountains, but these localized altitude measurements would be confusing if not meaningless for world-wide comparisons of mountain peaks.

Much the same is true of electrical and electronic circuits, where voltage is an important measure of electrical potential energy as well as a measure of signal strength where electricity is used as an analogue for representing other things[2]. One way to conveniently rate electrical potentials at different points in a circuit is by arbitrarily choosing one of those points to be the "zero-energy" reference point, and calling it *ground*. Ground in an electrical or electronic circuit is analogous to sea level for rating mountain peak heights or a *datum* or *benchmark* used for surveying land: simply a point of reference agreed-upon to be "zero" for the sake of ranking all other potentials to it.

A good example of this is an electrical network popularly known as a *split DC power supply*. Three different versions of this power supply network are shown with the same pair of 5-Volt DC sources, the only difference being the arbitrary location of the "Ground" (Gnd) point:

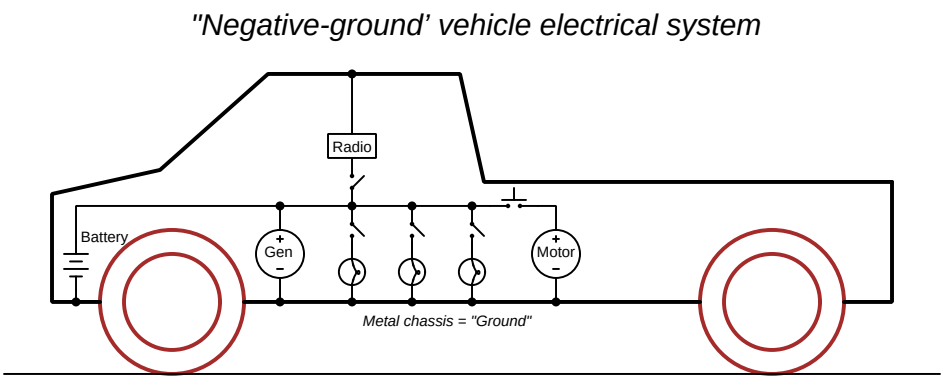


Each terminal's voltage label represents the amount of electrical potential there *relative to the "Ground" terminal*; i.e. "Ground" is simply a "sea-level" reference point against which all other potentials may be measured. The three different versions shown here illustrate just how arbitrary the location of "Ground" really is. Any load connected between the same pairs of terminals on each power supply would experience the same amount of voltage regardless of the "Ground" terminal assignment.

This is one meaning of the word "ground" in electric and electronic circuits: an arbitrary point of reference for the measurement and comparison of electrical power and signal potentials at different locations in the circuit.

[2]Many, many examples exist of voltage being used to represent non-electrical quantities. In digital computer systems, DC voltage levels represent binary "1" and "0" number states. In analog electronic sensor systems, voltage may represent temperature, speed, force, weight, sound pressure, or any number of other physical quantities.

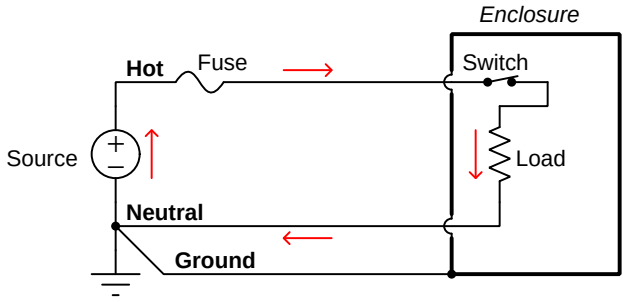
Another meaning of the word "ground" in circuits is a metallic surface used as a common conductor throughout some device or system for reasons of wiring economy. Automobiles with metal frames and chassis typically use their metallic bodies as an electrical conductor for nearly all circuits contained within. Doing so eliminates the need to install separate copper-wire conductors for every source and every load, thereby saving expense and weight:



Modern automotive electrical systems are "negative-ground" which means the vehicle chassis is always the lowest-potential point in any portion of the circuit. Interestingly, some early automobiles used "positive-ground" wiring which simply had the reverse polarity on all sources and loads: the vehicle chassis being the highest-potential point and all other points being either equal to it or at lower (−) potentials. This choice of polarity is really arbitrary, as electric circuits may be designed and built to function just as well either way. The "negative-ground" standardization we see today is really nothing more than automobile manufacturers agreeing to wiring their different vehicles the same way in order to enjoy interoperability of components.

This usage of the word *ground* – describing the metal chassis of some complex system exploited as a common electrical conductor for multiple circuits – is not incompatible with the previous usage of *ground*: a point of reference in a circuit arbitrarily defined to be "zero" potential. An automotive technician troubleshooting circuit faults in a vehicle often connects the black test lead of their voltmeter to the metal chassis, expecting to measure positive or zero voltage values on that meter when probing circuit test points with the red test lead. In a negative-ground electrical system, the chassis' electrical potential is *defined* as zero for the purpose of specifying voltages at other points in the circuit(s).

Yet another electrical meaning of the word "ground" is associated with *safety*, in which one metallic conductor of a power circuit is made electrically common to the Earth through a *grounding rod* buried in the soil. In the following circuit, the ground symbol in the lower-left corner is not merely a reference marker intended for anyone viewing the schematic diagram, but actually represents a metal rod driven deep into the soil and connected to one terminal of the voltage source with intent to achieve equipotentiality with the Earth:



Here the purpose of "grounding" is more important than merely establishing a reference point for voltage measurements. Instead, designating the conductor attached to this metal rod as "Ground" and attaching it to any conductive surface on or around an otherwise energized electrical appliance forces that surface to be equipotential to the same Earth that any humans and animals are standing on. This greatly mitigates the risk of electric shock should there be an internal fault within that appliance by making it impossible to have a difference of potential (i.e. voltage) between the grounded surface and the soil.

The following photograph shows the grounding rod for a commercial building, a large-gauge bare copper conductor clamped to the top of the rod which leads into the building's electrical room:

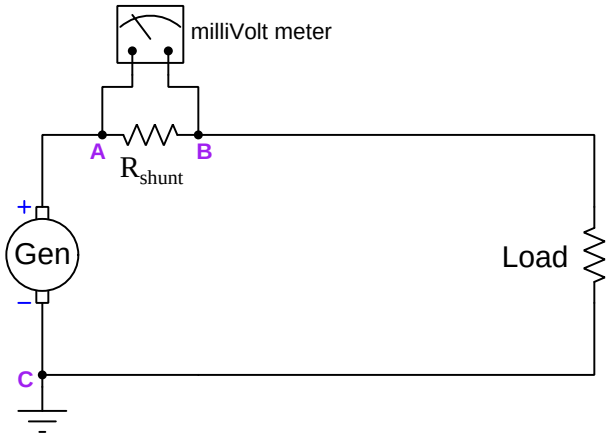
Ground wires in electric power systems should never carry current during normal operation. In fact, the presence of current through a ground conductor indicates a problem in the system! Note the two current-bearing power conductors in the previous schematic, specifically how the current-carrying conductor made common to Earth ground is called the *neutral*, while the other (non-grounded) power conductors is called *hot* by comparison. The Hot/Neutral designations always refer to which is grounded and which is not, rather than polarity. For example, if we were to reverse

the polarity of the DC voltage source shown in that schematic, the upper conductor (now −) would still be considered “hot” and the lower conductor (now +) would still be considered “neutral”.

4.2 Common-mode voltages and currents

A topic of much misunderstanding in electronic circuits, often relevant to precise measurements of voltage and current, is that of *common-mode signals*. The topic is especially confusing in regard to *voltage*, as voltage is fundamentally a differential quantity always existing *between two points* and never at a single location. We will explore this topic in some detail here, through practical examples.

Suppose we install a shunt resistor and a milliVolt meter on a power circuit to infer how much current a DC generator sends to a load. Given the existence of three electrically-distinct points in this circuit (**A**, **B**, and **C**) we have three unique voltages – the voltage registered by the milliVolt meter (V_{AB}), the voltage across the load (V_{BC}), and the voltage output by the generator (V_{AC}):

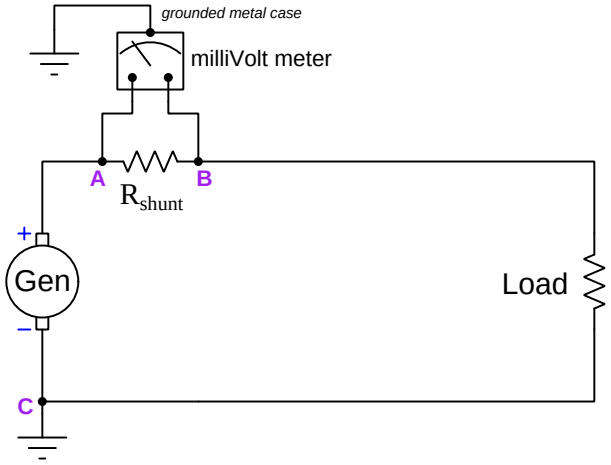


Supposing the generator output 300 Volts DC to a load that happens to be 40 Ohms at some given point in time, with the shunt resistor being a fixed value of 0.1 Ohms, these three voltages would be as follows:

- V_{AB} = 74.81 milliVolts DC
- V_{BC} = 299.25 Volts DC
- V_{AC} = 300 Volts DC

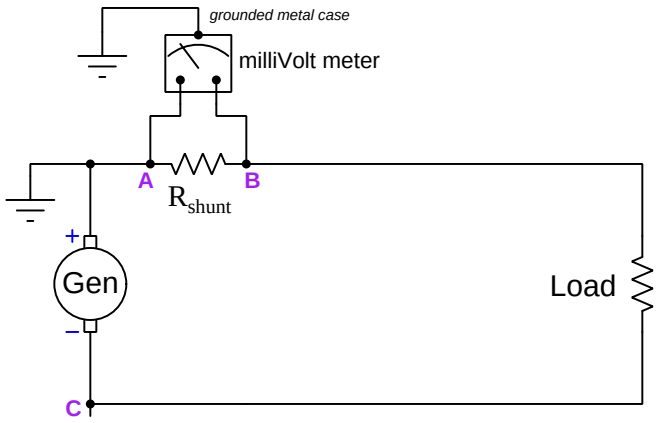
The milliVolt meter's purpose in this circuit, of course, is to infer load current by reporting the amount of potential difference dropped across R_{shunt}. In this case, the registered differential voltage between test points **A** and **B** is 74.81 milliVolts which equates to 7.481 Amperes of load current. The milliVolt meter should be completely immune to any changes in generator voltage, and only respond to current in this circuit.

Now suppose we add one more consideration to this seemingly simple circuit – we consider the milliVolt meter's metal casing attachment to a metal pole planted in the soil. This places the meter's case at ground potential (i.e. the potential of point **C**):



One would expect this to have no effect whatsoever on the meter's proper measurement of differential voltage between its test leads (points **A** and **B**), and in many cases this would be a correct assumption. However, there may be a *safety* concern if the milliVolt meter is not rated to handle such a high amount of voltage between either of its test leads and its metal case, in this instance 300 Volts between **A** and the case, and 299.25 Volts between **B** and the case. If the electrical insulation inside the milliVolt meter were insufficient to handle this much potential difference, it could break down over time and result in a fault where substantial current passes from either test lead to the case, through Earth ground, and back to the generator!

An interesting solution to this dilemma is to relocate the generator's ground connection from point **C** to point **A** as such:

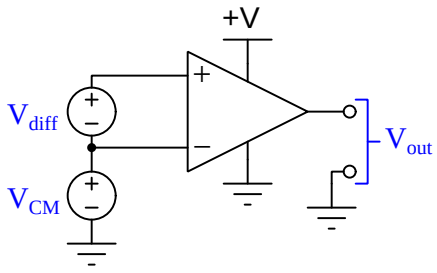


Now the voltage between point **A** and the meter's metal case is 0 Volts, and the potential between point **B** and the meter's metal case is −74.81 milliVolts. Clearly there no longer exists any hazard of electrical insulation breakdown inside the meter! This relocation of Earth-grounding point in the main power circuit has absolutely no effect whatsoever on the generator or on the load or even on the accurate reporting of current as inferred by the shunt resistor, but it certainly has an impact on how much potential difference the meter's internal insulation must withstand, and therefore on the meter's long-term reliability and safety.

By moving the circuit's Earth ground reference point we managed to vastly reduce the amount of electrical insulation stress inside the meter without affecting the generator or the load in the slightest. The meter still registers the shunt resistor voltage drop it's designed to measure, but without the hazard of insulation breakdown we faced before. It would be convenient to have a general term to refer to this change of stress on the meter, which is clearly related to voltage in a general sense but not to the differential voltage across the shunt resistor the meter was set up to read.

Such a term does exist, and it is called *common-mode voltage*. Since in this case we're concerned about the voltage impressed across the insulation protecting wires inside of the meter, the "common" element here would be the meter's metal case, since that is the other pole of any high voltage that any wire inside the meter would experience. We might interpret "common-mode voltage" to mean *any voltage shared in common between the two meter test leads and the metal case*, but this is a somewhat ambiguous definition. How much voltage, exactly is "common" between the 300 Volts (between **A** and case) and 299.29 Volts (between **B** and ground) when this circuit's Earth ground point is located on the generator's negative pole? We might define "common-mode" to be the lower of the two voltages (299.25 Volts), or perhaps the higher of the two (300 Volts), or maybe even the average of the two (299.63 Volts). A compelling argument could be made that the best way to define "common" in this context would be the greater of the two ground-referenced voltages (300 Volts) because what we're ultimately concerned about is over-stressing electrical insulation.

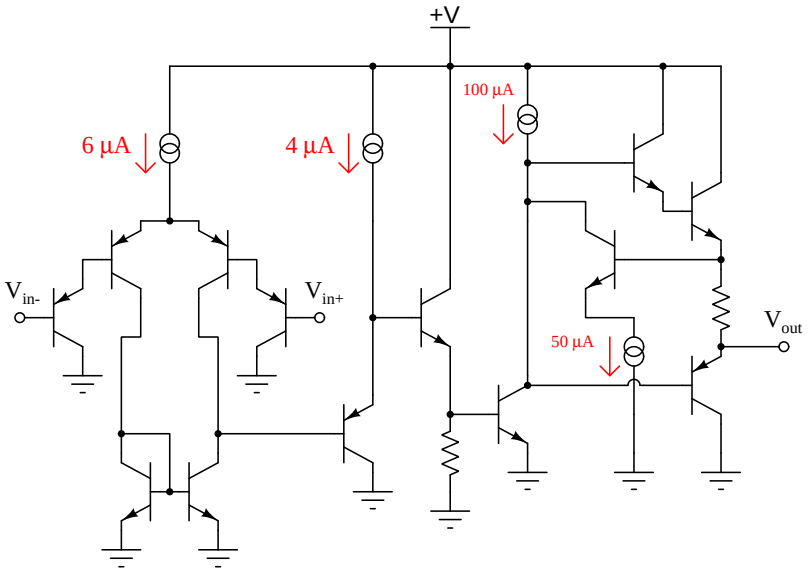
Another practical application of "common-mode voltage" is in differential amplifier circuits, which like the milliVolt meter in the prior example are designed to sense a difference of electrical potential between two input terminals but ignore any voltage with reference to ground or other common points in the circuit. Shown below is a typical illustration of common-mode voltage (V_{CM}) as applied to the input terminals of a differential amplifier (the triangular symbol):



Again, the purpose of any differential voltage amplifier is to respond to V_{diff} but ignore V_{CM}, the ground-referenced output signal V_{out} ideally being a function only of the differential input voltage V_{diff}. This is the ideal situation, but real differential voltage amplifiers do indeed respond (albeit slightly) to common-mode voltage, and if the common-mode voltage becomes too great the amplifier may cease to function entirely. For this reason we must be aware of how common-mode voltage is defined and why it matters to the internal operation of such amplifiers.

If we take this illustration literally, it would mean the lower input terminal of the amplifier experiences V_{CM} with reference to ground, while the upper input terminal experiences V_{diff} more than that, making "common-mode" voltage be the lesser of the two potentials with reference to ground. But, just as in the case of the milliVolt meter in the prior example what might actually *matter* in terms of real consequences might not be voltage measured with reference to ground. In other words, we might express V_{CM} in reference to ground, but there might be some other commonly-shared reference point that actually matters more to the proper operation of the amplifier.

Take for example the internal schematic diagram (simplified) for a model 324 operational amplifier:



Note how the transistors connected to the V_{in+} and V_{in-} input terminals of the amplifier are part of a Darlington pair array, with emitter arrows pointed in such a direction to indicate base currents must flow *out* from the amplifier to whatever device(s) provide the input terminal potentials. In order to ensure base current flows through each of these input transistors, the voltage as measured between either of these terminals and the amplifier's negative DC power terminal (i.e. in this case, ground) must never rise so high as to leave too little voltage across the base-emitter PN junctions of all four input transistors and the 6 μA current regulator for them to operate. This means the electrical potential of either input terminal must always be more than 1.4 Volts *lower* than the +V DC power supply "rail" potential in order to forward-bias two PN junctions (0.7 Volts each) plus provide enough voltage drop across the 6 μA current regulator for it to do its job properly.

One datasheet I consulted on the model LM324 operational amplifier states the common-mode input voltage range as such:

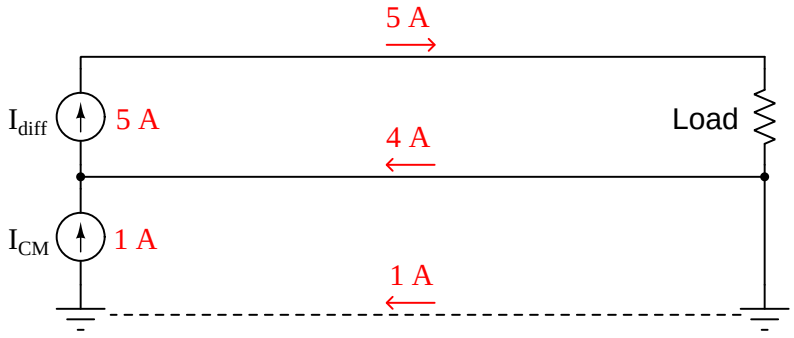
> Maximum V_{CM} common-mode voltage = V_{CC} − 2 Volts *(Where V_{CC} is the positive DC power rail)*

According to this datasheet, this model 324 amplifier circuit requires at least 2 Volts of drop spread between the two base-emitter PN junctions and the 6 μA current regulator in order to maintain a healthy amount of current through them all. So, even though we may define V_{CM} in terms of voltage measured between any amplifier input terminal and ground, what actually matters is maintaining a minimum common-mode voltage drop between any input terminal and the +V DC power supply terminal.

The moral of this story is that common-mode voltage may be defined in more than one way, and also that the way in which we define how it's measured may not necessarily directly relate

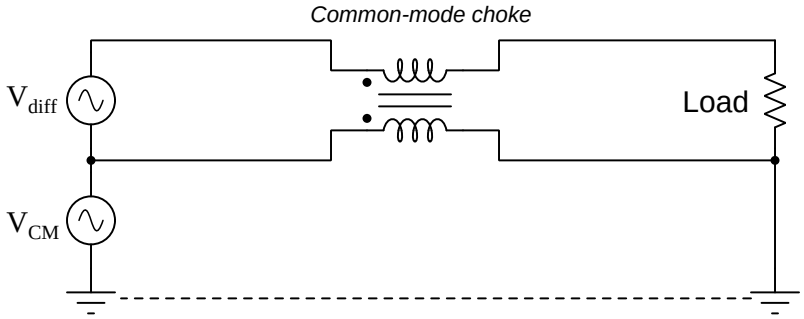
to the threat posed by having too much (or not enough) common-mode voltage! As usual, first principles matter: in the case of the milliVolt meter connected across a high-side shunt resistor we were concerned with possibly breaking down electrical insulation with too much potential difference between a test lead and the meter's metal case; with the model 324 operational amplifier we were concerned with keeping the input transistors in their active operational modes (conducting current).

In case you have been wondering, yes there is such a thing as common-mode *current* as well as common-mode voltage. In the case of common-mode current, it is defined as the net amount of electrical current among a multiple parallel conductors. An illustration is shown below:

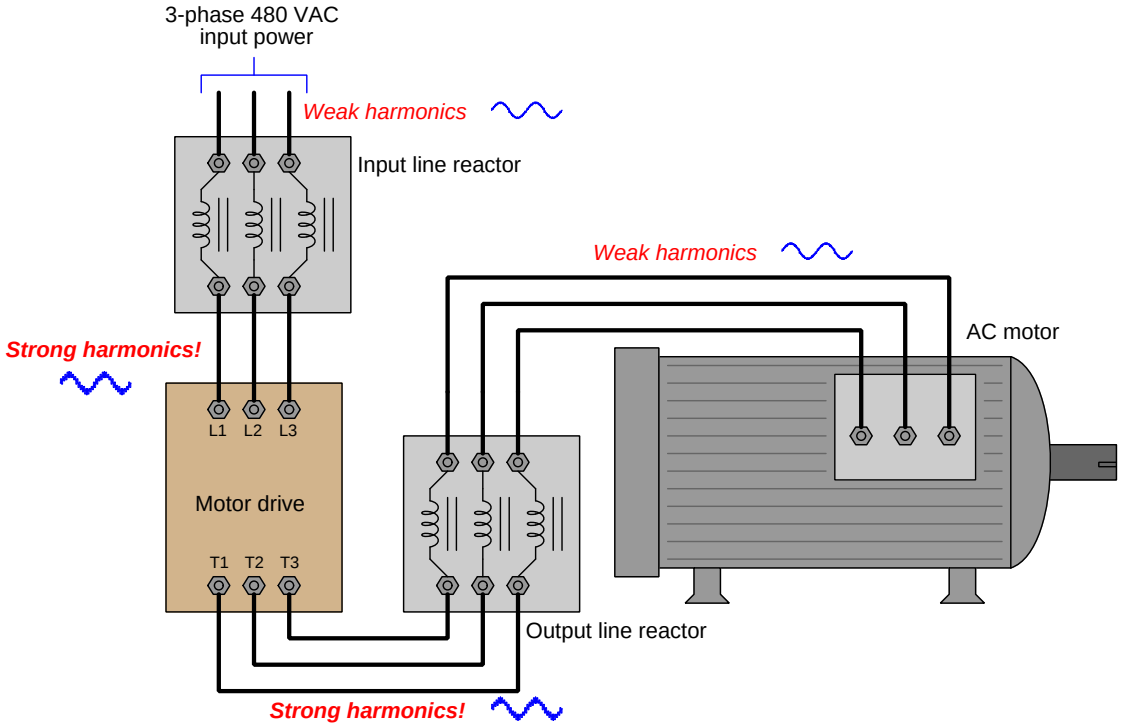


Here the differential current seen by the load is 5 Amperes, but the common-mode current through the two wires connected to that load is 1 Ampere.

In AC circuit applications where common-mode currents are undesirable, a popular way to attenuate them is to use a pair of magnetically-coupled inductors as a *common-mode choke*. The operating principle here is that the inductors' magnetic fields will be equal and opposite – and therefore completely cancel out – for differential current, but will aid each other to create a high impedance standing in the way of any common-mode current:



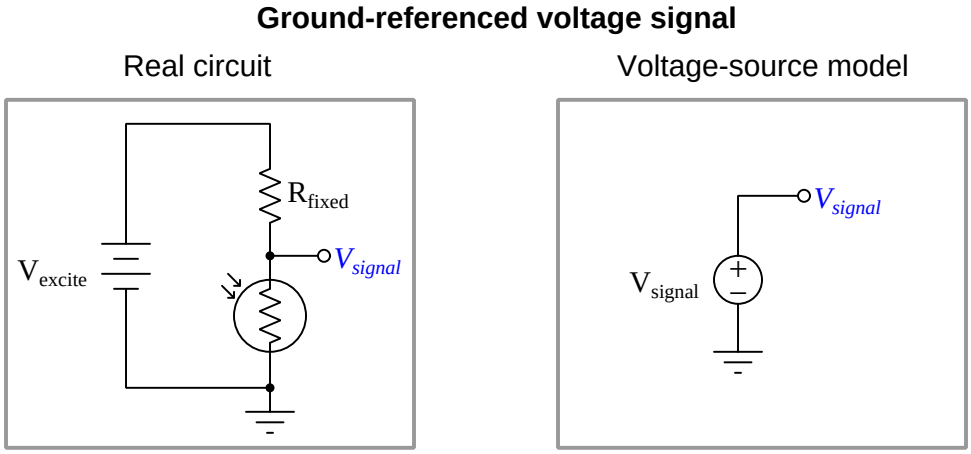
Common-mode chokes are often found in three-phase AC power systems where *triplen harmonics*[3] are found, as these harmonic currents exhibit zero phase shift between each other and so always flow in the same direction at any given time through the three power conductors. An example circuit shown below places two such three-phase common-mode chokes on either side of a variable-frequency motor drive (VFD):



[3] A "triplen" harmonic is any harmonic that is a multiple of three; e.g. 3rd, 6th, 9th, 12th, etc.

4.3 Ground-referenced signal sources

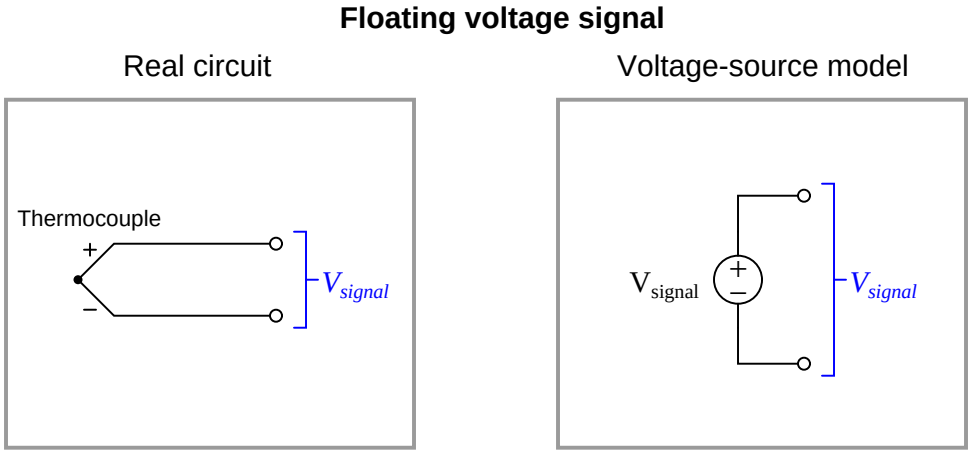
Some analog voltage signals exist with zero common-mode voltage. This is to say, they are *ground-referenced* signals. An example of such a voltage signal is seen in the following diagram, output by a simple voltage-divider circuit containing a fixed-value resistor and a cadmium sulfide photocell (i.e. photoresistor):



If the DAQ shares an electrically-common ground connection with the source, all that is required to interface is a single wire from the source terminal to a single-ended input on the DAQ, assuming all other DAQ parameters (e.g. voltage range) match.

4.4 Floating signal sources

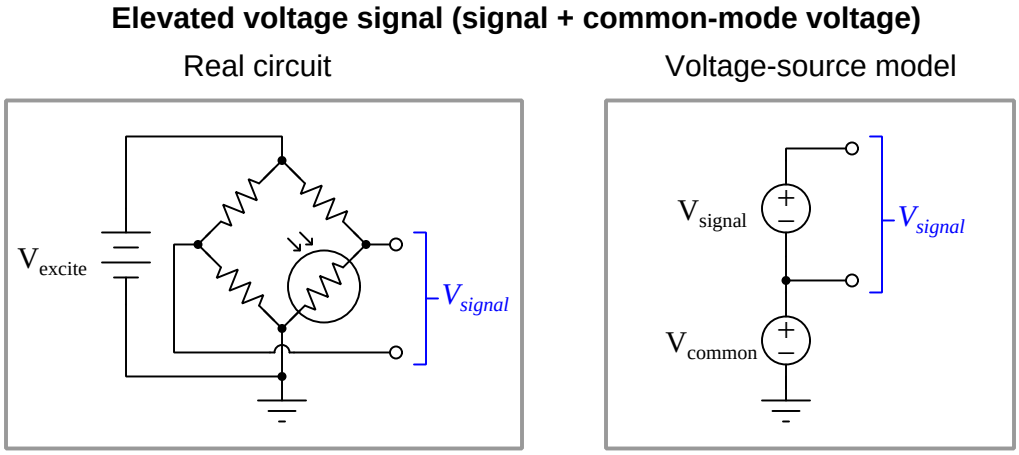
Some analog voltage signals exist in complete electrical isolation from any other conductor or from earth ground. We say that such signals are *floating*. An example of such a device is a *thermocouple* – a pair of dissimilar-metal wires joined together to form a junction. Thermocouple junctions produce small amounts of voltage directly proportional to temperature. As such, they are self-powered devices, needing no "excitation" power sources to operate:



Interfacing a floating source to a DAQ's voltage input requires either grounding one terminal of the source (i.e. make it ground-referenced) and use a single-ended DAQ input, or use a differential DAQ input. Note that bias resistors may be necessary as well to provide paths for differential input bias currents to flow.

4.5 Elevated signal sources

Some analog voltage signals are neither isolated nor are they directly referenced to ground, but possess a common-mode "offset" voltage from ground. A great many analog signals are like this, and cannot be measured using a single-ended DAQ input (unless the DAQ's ground potential may be safely elevated to the common-mode voltage of the signal source):



The best solution here is to use a differential voltage input to sense V_{signal}. In many cases, the path formed by the common-mode voltage will allow bias currents from the DAQ's inputs to flow.

4.6 General-purpose opamp recommendations

Operational amplifiers come in many different sizes, models, and capabilities, making selection by hobbyists and students daunting if all they wish to do is build a simple DC or audio-frequency AC proof-of-concept amplifier circuit. The following is a list of popular general-purpose opamps available in DIP packaging[4] that I recommend for students and hobbyists just beginning their exploration of these amazingly capable circuits.

In all cases *please consult manufacturer datasheets for specific ratings* as the information I provide here is generic and may not cover the exact model you happen to have available to you! A good example of this caveat is output current limitations, which are usually dependent on the DC voltage powering the opamp at any given time, and which are typically specified in the form of graphs shown in the datasheet.

4.6.1 TLV27x series

These general-purpose MOSFET-input opamps are available in one, two, or four amplifiers per package. The last digit of the part number specifies that number, for example "TLV272" meaning two opamps in that package. They also offer full rail-to-rail output voltage range.

- *DC supply voltage range* = 2.7 to 16 Volts
- *Rail-to-rail output* = Yes
- *Rail-to-rail input* = No
- *Maximum output current* = 12 mA
- *Slew rate* = 2.4 Volts per microsecond
- *Special characteristics* = MOSFET inputs make these opamps susceptible to damage from static electricity (ESD)

[4] Modern mass-produced electronic circuits typically use surface-mount components for their small footprint and ease of automated assembly. However, SMD opamps are difficult to "breadboard" temporary circuits with, DIP-style packages being far superior in this regard.

4.6.2 TLE214x series

These general-purpose bipolar-transistor opamps have an extremely wide DC power supply voltage range and are available in one, two, or four amplifiers per package. The last digit of the part number specifies that number, for example "TLE2142" meaning two opamps in that package.

- *DC supply voltage range* = 4 to 44 Volts
- *Rail-to-rail output* = No
- *Rail-to-rail input* = No
- *Maximum output current* = 20 mA – short-circuit protected
- *Slew rate* = 27 Volts per microsecond
- *Special characteristics* = input offset adjustment capability for the model TLE2141 (two dedicated IC pins for this called "Offset N1" and "Offset N2")

4.6.3 TL07x / TL08x series

These are good general-purpose opamps with JFET inputs available in one, two, or four amplifiers per package. The last digit of the part number specifies that number, for example "TL082" meaning two opamps in that package.

- *DC supply voltage range* = 10 to 30 Volts
- *Rail-to-rail output* = No
- *Rail-to-rail input* = No
- *Maximum output current* = 26 mA – short-circuit protected
- *Slew rate* = 20 Volts per microsecond
- *Input noise voltage* = 9.2 microVolts peak-peak
- *Special characteristics* = input offset adjustment capability for the single-opamp models TL071 and TL081 (two dedicated IC pins for this called "Offset N1" and "Offset N2")

4.6.4 MCP600x series

These are excellent opamps for circuits operating at modest DC power supply voltage levels, such as TTL and low-voltage CMOS logic circuits where the opamp must operate on the same power supply rails as the digital logic. They have MOSFET inputs, and also offer full rail-to-rail output voltage range and rail-to-rail input voltage range which is important given their rather low power supply voltage range.

- *DC supply voltage range* = 1.8 to 6 Volts
- *Rail-to-rail output* = Yes
- *Rail-to-rail input* = Yes
- *Maximum output current* = 23 mA – short-circuit protected
- *Slew rate* = 0.6 Volts per microsecond
- *Input noise voltage* = 6.1 microVolts peak-peak
- *Special characteristics* = MOSFET inputs make these opamps susceptible to damage from static electricity (ESD)

4.6.5 LM158/258/358

This legacy bipolar-transistor opamp design offers two amplifiers in one package.

- *DC supply voltage range* = 3 to 30 Volts
- *Rail-to-rail output* = No
- *Rail-to-rail input* = No
- *Maximum output current* = 20 mA – short-circuit protected
- *Slew rate* = 0.5 Volts per microsecond
- *Input noise voltage* = 3 microVolts peak-peak

4.6.6 LM324

This legacy bipolar-transistor opamp design offers four amplifiers in one package.

- *DC supply voltage range* = 3 to 30 Volts
- *Rail-to-rail output* = No
- *Rail-to-rail input* = No
- *Maximum output current* = 20 mA – short-circuit protected
- *Slew rate* = 0.5 Volts per microsecond
- *Input noise voltage* = 3 microVolts peak-peak

4.6.7 LM741 and LM1458

The LM741 has just one amplifier in an 8-pin package. The LM1458 is the same opamp design with two amplifiers in one 8-pin package. These are both legacy designs and should be considered a "last-resort" choice for most any circuit requiring an opamp. The high minimum DC supply voltage requirement and relatively slow slew rate specifications are disqualifying for many circuit applications.

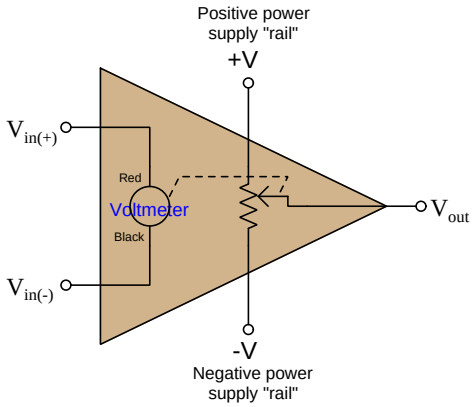
- *DC supply voltage range* = 20 to 44 Volts
- *Rail-to-rail output* = No
- *Rail-to-rail input* = No
- *Maximum output current* = 25 mA – short-circuit protected
- *Slew rate* = 0.5 Volts per microsecond
- *Special characteristics* = input offset adjustment capability for the LM741 (two dedicated IC pins for this called "Offset Null")

4.7 Operational amplifier imperfections

Comparators and operational amplifiers are incredibly useful and practical as electronic circuit building-blocks, but they are not perfect. For a great many applications these imperfections are insignificant, but for others they may be debilitating. Therefore, it is important to understand the ways in which operational amplifiers deviate from ideal behavior.

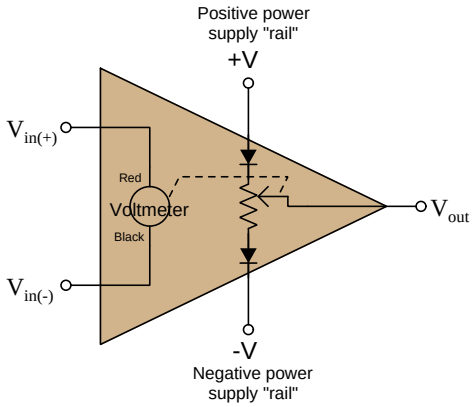
4.7.1 Output voltage limits

The output voltage of any amplifier circuit cannot exceed its DC supply "rail" voltages, because all an amplifier does is *modulate* its supply voltage. A useful model of a differential amplifier is that of a potentiometer with its wiper driven by the pointer of an analog voltmeter driven by the differential voltage applied to the input terminals:



No matter how much voltage is applied between the + and − input terminals, the potentiometer can only "swing" as far as the +V and −V "rails" of the DC power supply. This behavior is often referred to as *rail-to-rail output*, where the V_{out} limits are exactly (or very nearly) equal to the DC rail potentials.

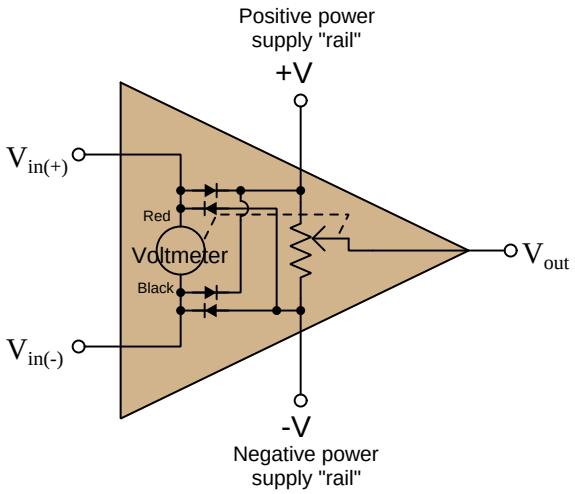
However, many practical amplifier models cannot even swing that far. A more accurate model for many is one where the potentiometer cannot quite reach the +V and −V rails:



With the diodes' voltage drops in effect, V_{out} is only able to "swing" to within approximately 0.7 Volts of each rail. Such internal voltage drops are typical of comparators and operational amplifiers using BJTs in the final output stage. Rail-to-rail capability demands FET output transistors which are capable of operating with mere milliVolts of drain-source voltage drop.

4.7.2 Input voltage limits and latch-up

As a general rule, input voltages for any integrated circuit should remain within the limits of its DC power supply "rail" potentials. Many integrated circuits including comparators and operational amplifiers include "protection" diodes at the input terminals effectively *clamping* input terminal potentials to the power supply rails (to within fractions of a Volt). Expanding upon our previous circuit model, we may show these protection diodes (two per input terminal) connected such that one will begin conducting if the potential at any input exceeds either power supply rail, essentially making that input terminal the new "rail" to power the amplifier:



If the common-mode[5] voltage applied to the inputs of a differential amplifier exceeds the point at which both transistors in the differential pair subcircuit turn off, the amplifier's output may saturate rather than respond properly to the differential input voltage. In some designs this may even lead to a condition where the output remains "latched" in this saturated mode even after the offending input condition ceases – a phenomenon known as *latch-up*[6].

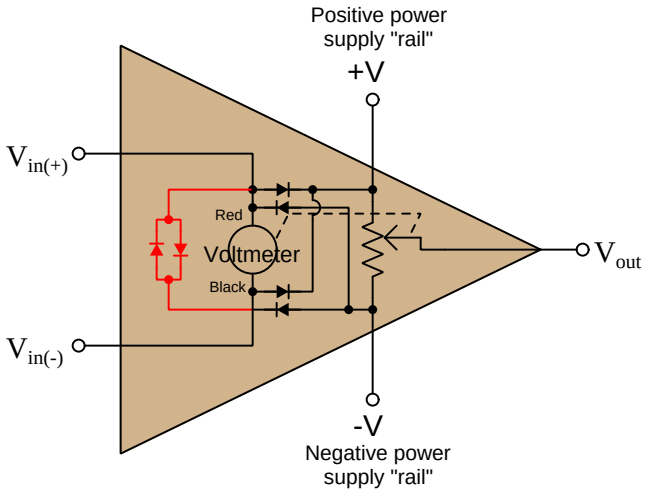
Amplifiers may fail to function properly even with input voltage values falling well within their supplied rail voltages. For example, the popular TL08x series of operational amplifiers specify common-mode voltage limits between +4 Volts above the negative rail voltage and −4 Volts below the positive rail voltage: if we happened to power one of these opamps on ± 12 Volt rails, the permissible input signal voltage range for either of its inputs would be −8 Volts to +8 Volts. In

[5]For a differential amplifier, the *common-mode voltage* is the amount of voltage with respect to ground that is shared among both input terminals, typically calculated as the average of the two inputs' ground-referenced voltages. In other words, if $V_{diff} = V_{in(+)} - V_{in(-)}$ then $V_{CM} = \frac{V_{in(+)} + V_{in(-)}}{2}$. However, for the sake of determining safe limits for amplifier input voltage we may more simply define "common-mode voltage" as being any voltage measured with respect to ground for either of a differential amplifier's two inputs that may be either high or low enough to cause a problem for that amplifier.

[6]Interestingly, the term "latch-up" more properly describes a condition in an integrated circuit where the particularly layering of semiconductor materials forms a PNPN (thyristor) structure accidently triggered by certain abnormal input conditions. Like the "latched" state just described, this triggered-thyristor state can only be reset by cycling power to the circuit.

contrast, other opamp models such as the MCP600x series offer *rail-to-rail inputs* capable of receiving any common-mode voltage up to and including the two DC supply rail voltages.

Some operational amplifiers additionally clamp *differential* input signals using a pair of paralleled protection diodes as shown in red below on the simplified opamp model:

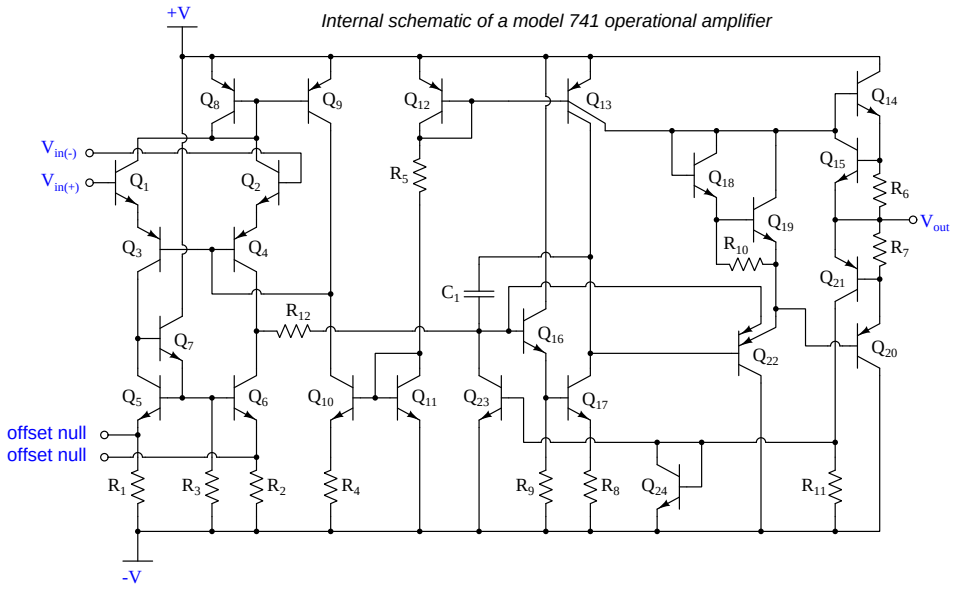


These differential protection diodes attempt to clamp differential voltage to a maximum of 0.7 Volts, which for a typical opamp employing negative feedback is perfectly acceptable because negative feedback should maintain the differential voltage at approximately zero anyway. However, if one attempts to use such a differential-clamped opamp as a *comparator* rather than as an operational amplifier with negative feedback to compare two input potentials differing by more than 0.7 Volts, improper circuit operation will result, with possible damage to the opamp as well!

4.7.3 Input bias current

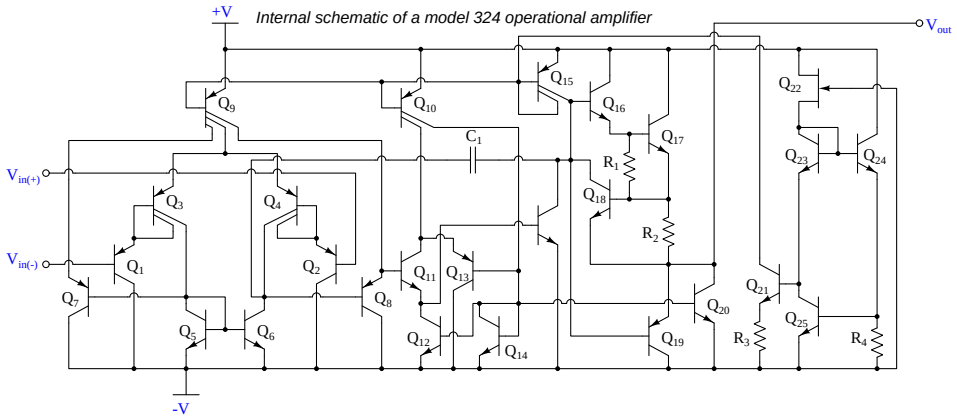
A generally safe assumption for most circuits is that the input terminals of a comparator or operational amplifier carry no current. Like the input of an ideal voltmeter, we assume these inputs have infinite input impedance and therefore pose no load on the sources they sense.

However, this is not entirely true. The input transistor stage of a differential amplifier represents a finite – albeit very large – impedance, and bipolar junction transistors do require *some* amount of base current to function. An inspection of the internal circuitry for the model 741 opamp reveals the source of these currents as well as their directions:



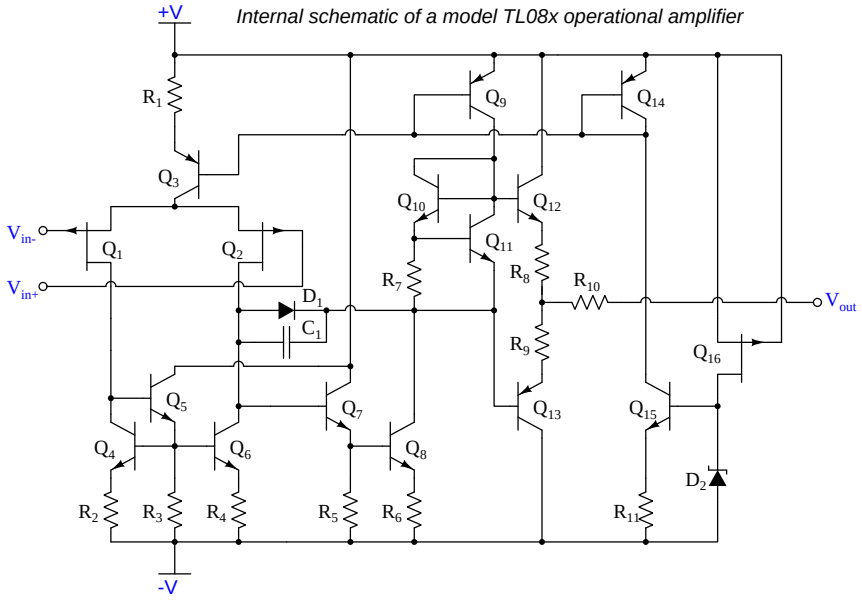
The two input terminals for the 741 connect to the bases of NPN transistors Q_1 and Q_2. In order for these transistors to turn on, current *must* enter the base terminals from outside the opamp. These *bias currents* are quite small – on the order of tens of *nano*Amperes, but they do indeed exist and must find a path through the external circuit.

Another legacy opamp model is the 324. Its internal circuitry is shown below:



Note the two input PNP transistors Q_1 and Q_2. In order for these transistors to turn on, current *must* exit their base terminals and find its way to the negative pole of the power supply through external circuitry.

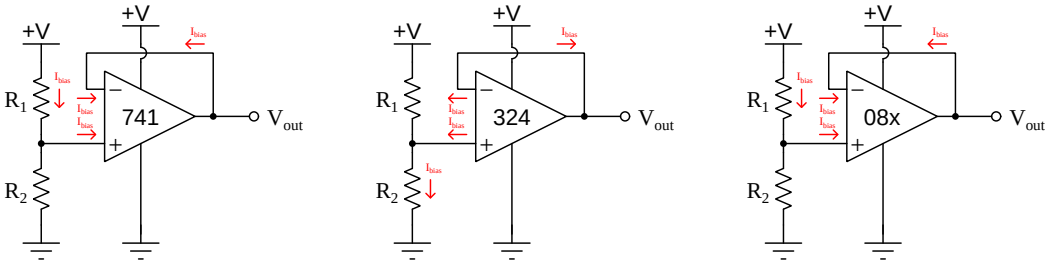
Some comparators and opamps have FET inputs rather than BJT inputs, usually in an attempt to increase input impedance (i.e. decrease bias current). The TL08x opamp series is an example of this design:



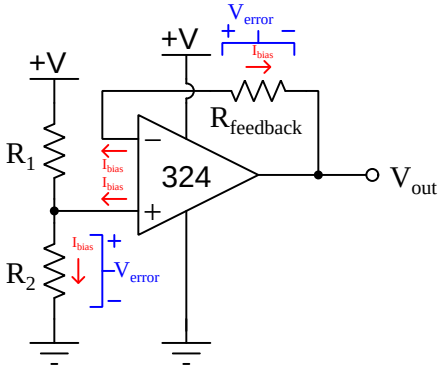
Knowing that JFETs are designed to have their gate-channel PN junctions reverse-biased, we may conclude the two P-channel JFETs Q_1 and Q_2 will exhibit extremely small bias currents *entering* their gate terminals from outside the opamp.

Some opamps utilize MOSFETs for their input transistors in order to maximize input impedance and minimize bias currents, the naturally high impedance of a MOSFET stemming from the layer of metal-oxide insulation separating each MOSFET's gate from the conducting channel beneath. In such opamps the main contribution to input terminal bias current actually comes from reverse-leakage current of *protection diodes* added to the circuit (connecting between each of those input terminals and the DC power rails) to protect the sensitive MOSFET gates from electrostatic discharge, rather than from the transistors themselves.

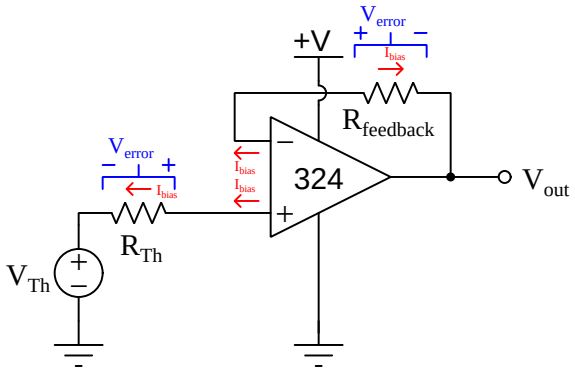
When we connect components to the input terminals of an opamp, we must be sure some pathway exists for these bias currents to flow in the proper directions. Consider the following examples, where a voltage divider provides a voltage signal which each opamp then "buffers" as a voltage follower:



Again, these currents are extremely small in magnitude, and their effects are usually negligible. However, if they must pass through a high impedance, they will cause an undesired voltage drop to develop. A common strategy to mitigate the effects of this voltage drop is to insert an equivalent amount of resistance within the feedback bias current path, assuming the two bias currents will be approximately equal in magnitude, and therefore the *two* unwanted voltage drops should be approximately equal in magnitude but opposite in polarity and so should cancel each other out as common-mode (not differential) voltage at the opamp inputs:



If we regard the voltage divider as a voltage *source* with its own internal Thévenin resistance, we see the bias current must pass through that resistance:



The value of the "source's" Thévenin resistance is equal to the parallel equivalent resistance of R_1 and R_2, and so our feedback resistance needs to be equal to the same:

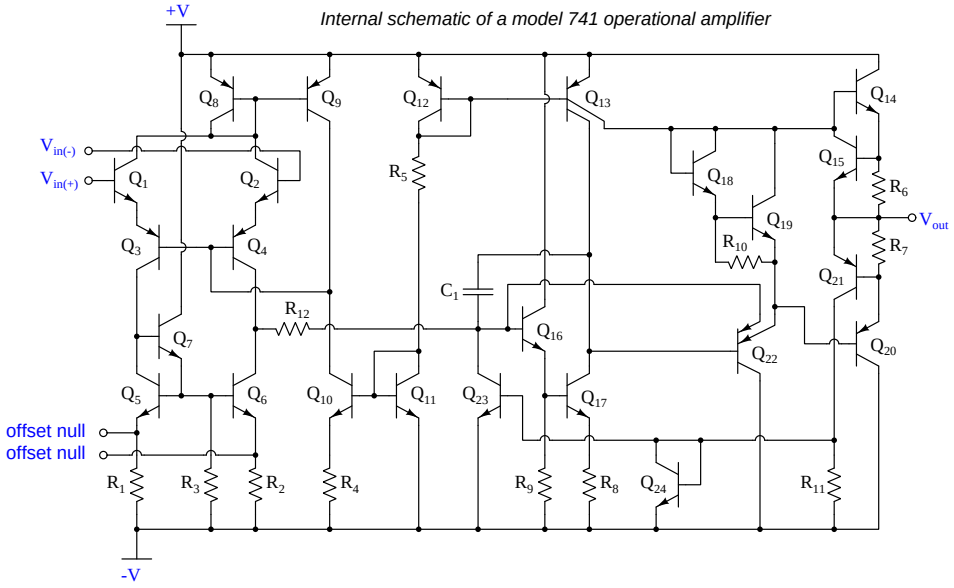
$$R_{feedback} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2}}$$

It is important to note that the assumption of an opamp's two bias currents being equal is exactly that: an *assumption* and not necessarily reality. Small differences in bias current from one input to the other can exist, and this difference of bias current is called *offset current*.

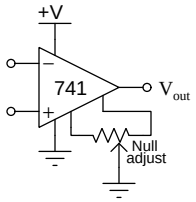
4.7.4 Input offset voltage

If the input terminals of a differential amplifier are shorted together to ensure zero differential input, the amplifier's output should ideally assume a state of zero Volts as well. However, this rarely happens in practice due to asymmetries in the internal circuitry of the amplifier. In practice, a very small amount of differential input voltage is necessary to achieve exactly zero Volts at the output, and this small differential voltage is called the *offset voltage* or *bias voltage* of the amplifier.

Some operational amplifier models provide external terminals for the user to connect a potentiometer, which may be used to "trim" the offset of the amplifier. The legacy 741 opamp is one of those designs, with *offset null* terminals labeled:



Externally, the "nulling" potentiometer connects to these two offset terminals and to the negative power supply terminal:



Essentially, the two portions of the potentiometer become connected in parallel with R_1 and R_2, respectively, inside the opamp. By externally skewing the equivalent resistances of R_1 and R_2, the bias currents through the two halves of the opamp's differential pair input stage may be likewise skewed to achieve the desired negation of offset voltage.

4.7.5 Frequency compensation

A problem common to high-gain amplifier circuits is undesired *oscillation*, and the root of this problem is *positive feedback*: where some of the amplifier's output signal becomes "fed back" to its input with a 360^o phase shift, so that it continually drives itself into oscillation. This principle finds constructive application in sinusoidal oscillator circuits, but for non-oscillator circuits it can be a serious problem.

Most operational amplifier circuits utilize *negative* feedback (i.e. 180^o phase shift) rather than positive feedback, and so it might seem at first oscillations would not be possible in these circuits. However, the 360^o of phase shift necessary for self-sustaining oscillations may occur in a negative-feedback circuit if a total of 180^o *additional* phase shift occurs elsewhere in the opamp circuit. One operating scenario inviting additional phase shift is when the opamp's output must drive a capacitive load. At high enough frequencies, the load's capacitance and stray capacitances in the circuit may conspire to provide the additional 180^o of phase shift necessary to make the total phase shift 360^o.

The criteria necessary for self-oscillation is signal feedback with 360^o of phase shift *and* a total signal gain of at least unity (1). If the 360^o of phase shift cannot be avoided, an effective solution to the problem of unintentional oscillation is to intentionally reduce the opamp's open-loop gain as frequency increases, so that at the frequency necessary to achieve 360^o phase shift the fed-back signal has a voltage gain less than one.

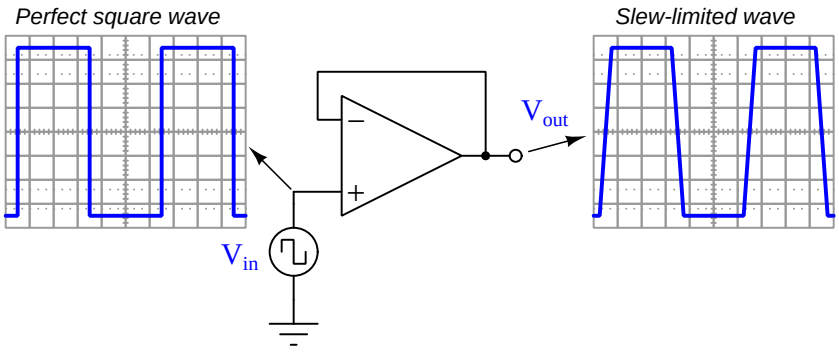
Many operational amplifiers are *internally compensated* for this by the intentional connection of a capacitor between amplification stages. The internal schematic diagrams of three models of opamp shown in previous subsections all contain compensation capacitors. They are easy to spot, being the *only* capacitor in the entire diagram!

4.7.6 Gain-bandwidth product

Operational amplifier open-loop voltage gain is very high, but degrades for multiple reasons as signal frequency increases. The mathematical product of amplifier gain and signal frequency is expressed as the amplifier's *Gain Bandwidth Product*, or *GBWP*.

4.7.7 Slew rate

The maximum rate at which the output voltage of a comparator or opamp is able to rise or fall is called the *slew rate*. If we drive a simple "voltage follower" opamp circuit with a perfect square-wave signal having instantaneous rise and fall times, what we will see on the output is a slew-limited waveform with definite rise and fall *slopes*:



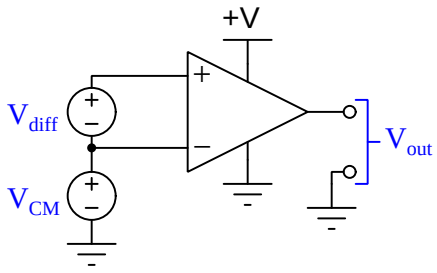
Slope is always measured as a *rise over run*, so for these voltage waveforms we may represent their rising- and falling-edge slopes as a rates-of-change of voltage over time ($\frac{dV}{dt}$), the input waveform having a *infinite* rates of change, and the output waveform having *finite* rates of change. The reason for an opamp's slew rate limitations is based on its internal current limitations and capacitances. Recall the relationship between voltage and current for any capacitance:

$$I_C = C\frac{dV_C}{dt}$$

If a current source drives a constant current through a capacitance, that voltage across that capacitance will either rise or fall (depending on current direction) at a constant *rate*. An opamp's internal transistors are current-limiting devices by nature, and when this limited current energizes or de-energizes capacitance the result is a linearly-sloping waveform rather than an instantaneous jump. A major source of capacitance for any opamp is the *frequency-compensating* capacitance previously discussed, but any capacitance connected to the opamp as part of the load of course also contributes to a limited slew rate. This is why opamp datasheets always specify load capacitance when declaring maximum slew rate (e.g. the model 741 opamp is typically limited to 0.5 Volts per microsecond, or 500,000 Volts per second, while driving a load having no more than 100 picoFarads of capacitance).

4.7.8 Common-mode rejection

Operational amplifiers are *differential* amplifiers, and as such they are only supposed to respond to *differences* in voltage between their two input terminals. Ideally, any common-mode[7] voltage is ignored (i.e. *rejected*) by the amplifier:



In other words, V_{out} for an ideal opamp is strictly a function of V_{diff} and is entirely independent of V_{CM}. However, real opamps are not ideal, and as such their output voltages will be slightly affected by V_{CM}. We may express the relationship between V_{out} and V_{CM} as a *gain*, just as we do for differential voltage:

$$A_{CM} = \frac{\Delta V_{out}}{\Delta V_{CM}} \qquad A_{diff} = \frac{\Delta V_{out}}{\Delta V_{diff}}$$

Differential voltage gain (A_{diff}) is typically referred to as open-loop voltage gain (A_{OL}) and for most operational amplifiers is an extremely large number. Common-mode voltage gain (A_{CM}) by contrast is much smaller (ideally being zero).

The degree to which a differential amplifier rejects a common-mode signal is typically gauged in comparison to its differential voltage gain. The ratio of A_{diff} to A_{CM} is called the *common-mode rejection ratio*, or *CMRR*:

$$\text{CMRR} = \frac{A_{diff}}{A_{CM}}$$

CMRR is more often found expressed as a decibel figure, so:

$$\text{CMRR(dB)} = 20 \log \left(\frac{A_{diff}}{A_{CM}} \right)$$

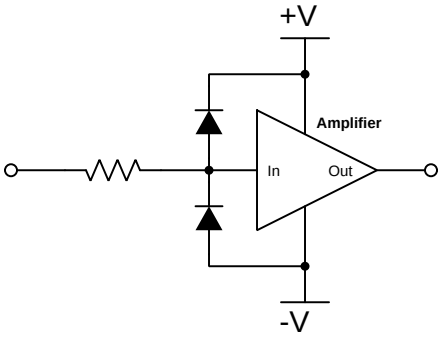
[7] Here, the term "common-mode" refers to a quantity shared in common by both input terminals of the amplifier. This is distinct from "differential" which by definition means something *different* or *distinct* between the input terminals.

4.8 Protecting amplifier inputs from over-voltage

Electronic amplifier circuits are extremely useful devices, comprising portions of many practical integrated-circuit (IC) electronic components such as digital logic gates, comparators, operational amplifiers, signal mixers, etc. Solid-state amplifier circuits use *transistors* to allow one electrical signal to control another, and these transistors tend to be susceptible to damage from excessive applied signal voltage. In the case of MOSFET transistors, an excess of signal voltage may puncture the extremely thin layer of metal-oxide insulation separating the transistor's gate terminal from its current-carrying channel. In the case of bipolar (NPN, PNP) transistors, an excess of signal voltage may break down reverse-biased PN junctions inside the transistor, often causing them to fail in "shorted" states. Excessive applied signal voltage may also cause damage to bipolar transistors if forward-biasing PN junctions to the extent that they conduct high amounts of current which may lead to thermal damage.

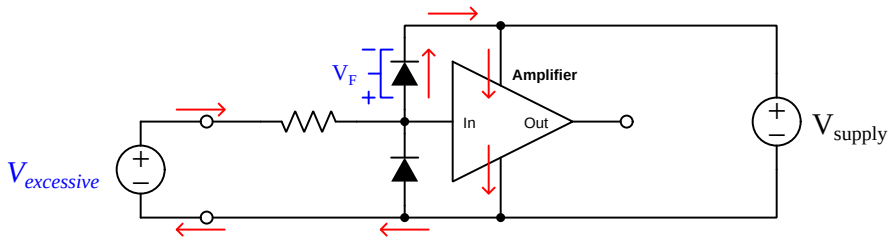
Sources of excessive signal voltage may be broadly categorized as *electro-static discharge* (ESD) or *electrical over-stress* (EOS). ESD happens when some object external to the circuit (including human bodies) accumulates an electro-static charge in its own capacitance, and this capacitively-stored charge is suddenly send to the circuit where it drops a voltage across circuit components high enough to cause damage. EOS is a more broad description of any over-voltage or over-current condition caused by one circuit sourcing energy to another, such as when an electrical test instrument is connected to a signal source that is too great for that instrument to handle.

A simple and effective way to limit both applied voltage and applied current to the input of any electronic amplifier is to connect a current-limiting resistor in series between the input terminal and the amplifier, as well as connect diodes between the input terminal and either DC power supply "rail" terminal:



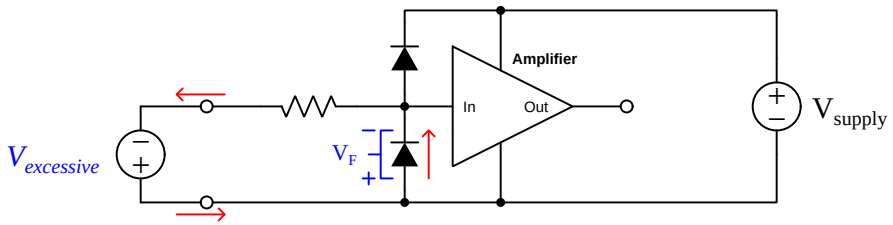
It matters not how the amplifier itself is internally constructed, nor what its larger purpose is – whether it be part of a digital logic gate, a comparator, an operational amplifier, or something even more complex. The purpose of these two diodes and the series resistor is to prevent any component within the amplifier from experiencing either too much voltage and/or too much current. Such resistor-diode networks are generally known as *clipping networks* because they truncate or "clip" an incoming signal's amplitude so as to never exceed some maximum value(s).

To understand how this protection network functions, consider the following circumstance where an excessive positive potential is applied to the input terminal by some external source. Here, "excessive" is defined as any voltage greater in magnitude than the DC power source energizing the amplifier:



The upper diode of the protection network forward-biases to permit current from the offending source to pass into the positive power bus of the amplifier. The resistor's function is to drop however much of the applied voltage exceeds the power supply rail potential plus the protection diode's forward voltage drop ($V_{supply} + V_F$), and to limit the current to a value acceptable to the protection diode and to the circuit's DC power supply so that neither of those is over-stressed. With a normal silicon diode this means a voltage no greater than 0.7 Volts plus the positive DC rail voltage. If Schottky diodes are used in the protection network, it means a voltage no greater than 0.4 Volts beyond the positive rail. Such mild over-voltage conditions are unlikely to cause damage to any of the amplifier's internal transistors.

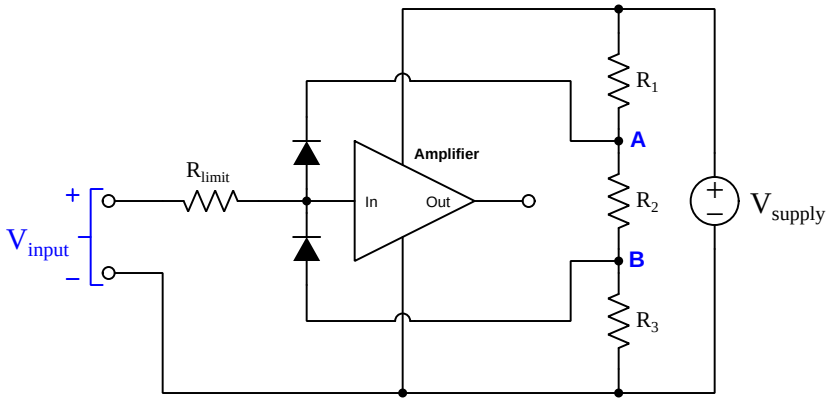
Similarly, if an excessive negative voltage is applied by an external source ("excessive" being any significant potential below that of the negative DC power supply rail):



Here the lower protection diode forward-biases and permits current from the external source to pass through the limiting resistor. While this occurs, the amplifier will never see more than that protection diode's forward voltage drop (V_F) between its input terminal and the negative DC rail bus, which again is unlikely to damage any transistor inside the amplifier. Again, the resistor should be sized to limit current to a value tolerable to the protection diode. However, this resistor should not be made so large that it might impair the normal operation of the amplifier when the protection diodes are not conducting: i.e. it should drop negligible voltage from the amplifier's normal input terminal current.

Such protection networks are often found inside of integrated circuits, sometimes without a series protection resistor in their most basic forms. Protection diodes are standard for CMOS digital logic gates to help protect their constituent MOSFET transistor gates from damage from ESD.

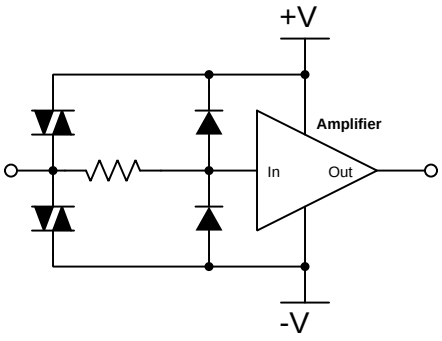
For some devices it is not sufficient for a diode network to clamp the input voltage to some value(s) slightly in excess of the device's power supply rail potential(s). For example, a device operating on a 5 Volt DC supply may tolerate input voltages no lower than zero and no greater than 5 Volts. A relatively simple solution to this problem is to provide the diode network with a set of "protection rails" just shy of the actual power supply rail potentials. Consider the following example as an illustration of this technique:



Here, resistors R_1, R_2, and R_3 form a voltage divider to produce two electrical potentials (labeled **A** and **B**), each one diode forward-voltage drop shy of the nearest rail potentials. For example, if the DC power supply output 5 Volts and the two protection diodes each dropped 0.7 Volts when forward-biased, the voltage divider would need to be designed such that point **A** was +4.3 Volts and point **B** was +0.7 Volts with respect to the DC supply's negative terminal. This would prompt the upper diode to turn on and clamp the amplifier's input potential to +5 Volts if ever the input potential exceeded +5 Volts, the limit resistor dropping the rest. Likewise, the lower diode would turn on if ever the input terminal's potential fell below the power supply's negative rail, clamping the amplifier's input terminal potential to exactly the same as that negative rail with R_{limit} dropping the rest.

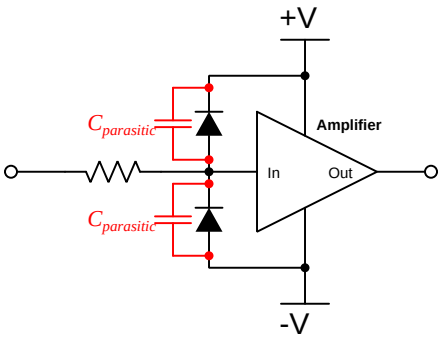
An important caveat to this strategy is that the voltage divider resistors R_1 and R_3 must be relatively small compared to the resistance of R_{limit} in order to ensure that those "protection rail" potentials **A** and **B** do not vary significantly when the protection diodes begin to conduct. A general engineering design principle here is to size R_{limit} at least ten times greater than either R_1 or R_3. If we size the resistors properly, this voltage-divider-based protection strategy may even be made *adjustable* by incorporating potentiometers into the voltage-divider network.

When protection must be provided against extraordinarily strong sources, additional protection may be added in their form of *DIACs* connected between each input terminal and power supply rail:



DIACs are classified as *thyristors*, because once triggered into an electrically conductive state by sufficiently high voltage (and/or sufficiently high *rate of change* of voltage, $\frac{dV}{dt}$) they will remain "on" so long as sufficient "holding" current passes through them even if voltage falls far below the initial triggering value. In summary, a thyristor acts as a very effective snubbing device to tame over-voltage conditions, essentially acting as a near-short to that offending source.

Protection networks, however, are not without their disadvantages. Chief among these is their tendency to adversely affect the signal being sensed by the amplifier being protected. Ideally a protection network should not corrupt the sensed signal at all, and only come into play if and when that signal becomes strong enough to pose threat of damage to the amplifier, but this would only be true if the protection diodes (and DIACs) had no parasitic properties when non-conducting. This is unfortunately untrue, as both diodes and DIACs exhibit *parasitic capacitance* which not only has the effect of storing electrical charge that the amplifier may interpret as a voltage that should not be present, but along with the protection resistor will form a *low-pass filter network* preventing the amplifier from being able to fully sense rapidly-changing input signals:



In other words, even with applied signals weak enough to pose absolutely no threat of harm to the amplifier, the capacitance inherent to the protection diodes will conspire with the series resistor

to "slow down" rates of rise and fall for any voltage arriving at the amplifier's input, thus making the amplifier "think" the signal isn't changing as rapidly as it really is.

In order to minimize the effects of parasitic capacitance within the protection diodes, we must choose diodes with as little of that capacitance as possible and also select a protection resistor with as low a value as possible that still limits maximum current to a value safe for the amplifier.

Chapter 5

Questions

This learning module, along with all others in the ModEL collection, is designed to be used in an inverted instructional environment where students independently read[1] the tutorials and attempt to answer questions on their own *prior* to the instructor's interaction with them. In place of lecture[2], the instructor engages with students in Socratic-style dialogue, probing and challenging their understanding of the subject matter through inquiry.

Answers are not provided for questions within this chapter, and this is by design. Solved problems may be found in the Tutorial and Derivation chapters, instead. The goal here is *independence*, and this requires students to be challenged in ways where others cannot think for them. Remember that you always have the tools of *experimentation* and *computer simulation* (e.g. SPICE) to explore concepts!

The following lists contain ideas for Socratic-style questions and challenges. Upon inspection, one will notice a strong theme of *metacognition* within these statements: they are designed to foster a regular habit of examining one's own thoughts as a means toward clearer thinking. As such these sample questions are useful both for instructor-led discussions as well as for self-study.

[1] Technical reading is an essential academic skill for any technical practitioner to possess for the simple reason that the most comprehensive, accurate, and useful information to be found for developing technical competence is in textual form. Technical careers in general are characterized by the need for continuous learning to remain current with standards and technology, and therefore any technical practitioner who cannot read well is handicapped in their professional development. An excellent resource for educators on improving students' reading prowess through intentional effort and strategy is the book textitReading For Understanding – How Reading Apprenticeship Improves Disciplinary Learning in Secondary and College Classrooms by Ruth Schoenbach, Cynthia Greenleaf, and Lynn Murphy.

[2] Lecture is popular as a teaching method because it is easy to implement: any reasonably articulate subject matter expert can talk to students, even with little preparation. However, it is also quite problematic. A good lecture always makes complicated concepts seem easier than they are, which is bad for students because it instills a false sense of confidence in their own understanding; reading and re-articulation requires more cognitive effort and serves to verify comprehension. A culture of teaching-by-lecture fosters a debilitating dependence upon direct personal instruction, whereas the challenges of modern life demand independent and critical thought made possible only by gathering information and perspectives from afar. Information presented in a lecture is ephemeral, easily lost to failures of memory and dictation; text is forever, and may be referenced at any time.

General challenges following tutorial reading

- Summarize as much of the text as you can in one paragraph of your own words. A helpful strategy is to explain ideas as you would for an intelligent child: as simple as you can without compromising too much accuracy.
- Simplify a particular section of the text, for example a paragraph or even a single sentence, so as to capture the same fundamental idea in fewer words.
- Where did the text make the most sense to you? What was it about the text's presentation that made it clear?
- Identify where it might be easy for someone to misunderstand the text, and explain why you think it could be confusing.
- Identify any new concept(s) presented in the text, and explain in your own words.
- Identify any familiar concept(s) such as physical laws or principles applied or referenced in the text.
- Devise a proof of concept experiment demonstrating an important principle, physical law, or technical innovation represented in the text.
- Devise an experiment to disprove a plausible misconception.
- Did the text reveal any misconceptions you might have harbored? If so, describe the misconception(s) and the reason(s) why you now know them to be incorrect.
- Describe any useful problem-solving strategies applied in the text.
- Devise a question of your own to challenge a reader's comprehension of the text.

General follow-up challenges for assigned problems

- Identify where any fundamental laws or principles apply to the solution of this problem, especially before applying any mathematical techniques.
- Devise a thought experiment to explore the characteristics of the problem scenario, applying known laws and principles to mentally model its behavior.
- Describe in detail your own strategy for solving this problem. How did you identify and organized the given information? Did you sketch any diagrams to help frame the problem?
- Is there more than one way to solve this problem? Which method seems best to you?
- Show the work you did in solving this problem, even if the solution is incomplete or incorrect.
- What would you say was the most challenging part of this problem, and why was it so?
- Was any important information missing from the problem which you had to research or recall?
- Was there any extraneous information presented within this problem? If so, what was it and why did it not matter?
- Examine someone else's solution to identify where they applied fundamental laws or principles.
- Simplify the problem from its given form and show how to solve this simpler version of it. Examples include eliminating certain variables or conditions, altering values to simpler (usually whole) numbers, applying a limiting case (i.e. altering a variable to some extreme or ultimate value).
- For quantitative problems, identify the real-world meaning of all intermediate calculations: their units of measurement, where they fit into the scenario at hand. Annotate any diagrams or illustrations with these calculated values.
- For quantitative problems, try approaching it qualitatively instead, thinking in terms of "increase" and "decrease" rather than definite values.
- For qualitative problems, try approaching it quantitatively instead, proposing simple numerical values for the variables.
- Were there any assumptions you made while solving this problem? Would your solution change if one of those assumptions were altered?
- Identify where it would be easy for someone to go astray in attempting to solve this problem.
- Formulate your own problem based on what you learned solving this one.

General follow-up challenges for experiments or projects

- In what way(s) was this experiment or project easy to complete?
- Identify some of the challenges you faced in completing this experiment or project.

- Show how thorough documentation assisted in the completion of this experiment or project.
- Which fundamental laws or principles are key to this system's function?
- Identify any way(s) in which one might obtain false or otherwise misleading measurements from test equipment in this system.
- What will happen if (component X) fails (open/shorted/etc.)?
- What would have to occur to make this system unsafe?

5.1 Conceptual reasoning

These questions are designed to stimulate your analytic and synthetic thinking[3]. In a Socratic discussion with your instructor, the goal is for these questions to prompt an extended dialogue where assumptions are revealed, conclusions are tested, and understanding is sharpened. Your instructor may also pose additional questions based on those assigned, in order to further probe and refine your conceptual understanding.

Questions that follow are presented to challenge and probe your understanding of various concepts presented in the tutorial. These questions are intended to serve as a guide for the Socratic dialogue between yourself and the instructor. Your instructor's task is to ensure you have a sound grasp of these concepts, and the questions contained in this document are merely a means to this end. Your instructor may, at his or her discretion, alter or substitute questions for the benefit of tailoring the discussion to each student's needs. The only absolute requirement is that each student is challenged and assessed at a level equal to or greater than that represented by the documented questions.

It is far more important that you convey your *reasoning* than it is to simply convey a correct answer. For this reason, you should refrain from researching other information sources to answer questions. What matters here is that *you* are doing the thinking. If the answer is incorrect, your instructor will work with you to correct it through proper reasoning. A correct answer without an adequate explanation of how you derived that answer is unacceptable, as it does not aid the learning or assessment process.

You will note a conspicuous lack of answers given for these conceptual questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your conceptual answers, where applicable, is to use circuit simulation software to explore the effects of changes made to circuits. For example, if one of these conceptual questions challenges you to predict the effects of altering some component parameter in a circuit, you may check the validity of your work by simulating that same parameter change within software and seeing if the results agree.

[3] *Analytical* thinking involves the "disassembly" of an idea into its constituent parts, analogous to dissection. *Synthetic* thinking involves the "assembly" of a new idea comprised of multiple concepts, analogous to construction. Both activities are high-level cognitive skills, extremely important for effective problem-solving, necessitating frequent challenge and regular practice to fully develop.

5.1.1 Reading outline and reflections

"Reading maketh a full man; conference a ready man; and writing an exact man" – Francis Bacon

Francis Bacon's advice is a blueprint for effective education: reading provides the learner with knowledge, writing focuses the learner's thoughts, and critical dialogue equips the learner to confidently communicate and apply their learning. Independent acquisition and application of knowledge is a powerful skill, well worth the effort to cultivate. To this end, students should read these educational resources closely, journal their own reflections on the reading, and discuss in detail their findings with classmates and instructor(s). You should be able to do all of the following after reading any instructional text:

☑ Briefly SUMMARIZE THE TEXT in the form of a journal entry documenting your learning as you progress through the course of study. Share this summary in dialogue with your classmates and instructor. Journaling is an excellent self-test of thorough reading because you cannot clearly express what you have not read or did not comprehend.

☑ Demonstrate ACTIVE READING STRATEGIES, including verbalizing your impressions as you read, simplifying long passages to convey the same ideas using fewer words, annotating text and illustrations with your own interpretations, working through mathematical examples shown in the text, cross-referencing passages with relevant illustrations and/or other passages, identifying problem-solving strategies applied by the author, etc. Technical reading is a special case of problem-solving, and so these strategies work precisely because they help solve any problem: paying attention to your own thoughts (metacognition), eliminating unnecessary complexities, identifying what makes sense, paying close attention to details, drawing connections between separated facts, and noting the successful strategies of others.

☑ Identify IMPORTANT THEMES, especially GENERAL LAWS and PRINCIPLES, expounded in the text and express them in the simplest of terms as though you were teaching an intelligent child. This emphasizes connections between related topics and develops your ability to communicate complex ideas to anyone.

☑ Form YOUR OWN QUESTIONS based on the reading, and then pose them to your instructor and classmates for their consideration. Anticipate both correct and incorrect answers, the incorrect answer(s) assuming one or more plausible misconceptions. This helps you view the subject from different perspectives to grasp it more fully.

☑ Devise EXPERIMENTS to test claims presented in the reading, or to disprove misconceptions. Predict possible outcomes of these experiments, and evaluate their meanings: what result(s) would confirm, and what would constitute disproof? Running mental simulations and evaluating results is essential to scientific and diagnostic reasoning.

☑ Specifically identify any points you found CONFUSING. The reason for doing this is to help diagnose misconceptions and overcome barriers to learning.

5.1.2 Foundational concepts

Correct analysis and diagnosis of electric circuits begins with a proper understanding of some basic concepts. The following is a list of some important concepts referenced in this module's full tutorial. Define each of them in your own words, and be prepared to illustrate each of these concepts with a description of a practical example and/or a live demonstration.

Digital signal

Analog signal

Annotating diagrams as a problem-solving strategy

Resolution

Binary number

Hexadecimal

Count

Electrically common points

Electrically distinct points

Electrically isolated points

Single-ended voltage signal

Differential voltage signal

Common-mode voltage signal

Ground-referenced voltage signal

Floating voltage signal

Elevated voltage signal

Noise

Ohm's Law

Ground loop

Thermocouple

RTD

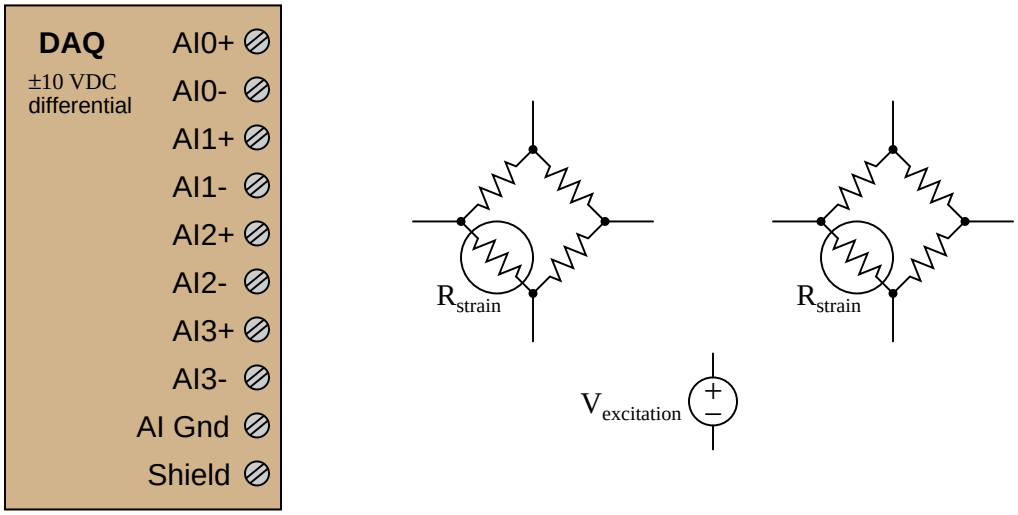
Thermistor

Bridge network

Bias current

5.1.3 Connecting strain gauges to a DAQ

Sketch connecting wires to allow this data acquisition unit (DAQ) to sense strain using quarter-bridge strain gauge circuits on input channels #1 and #3, such that increasing tension on the strain gauge (increasing gauge resistance) generates a more *positive* signal voltage on each channel:

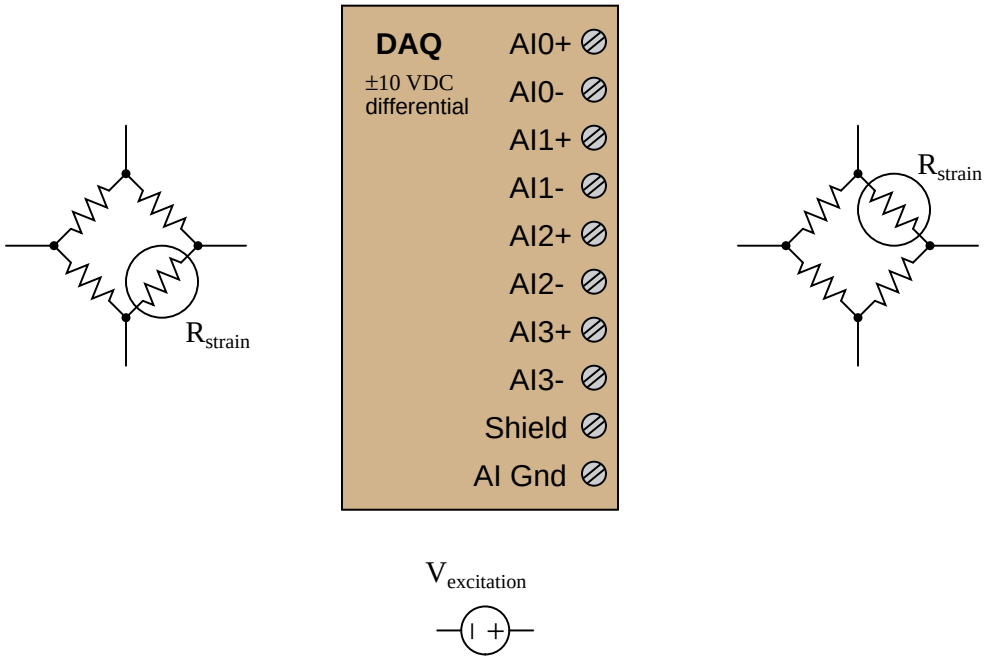


Challenges

- Explain why in order for differential DAQ inputs to work, the power supply must share a common connection with the DAQ power supply.
- Explain what the "AI Gnd" and "Shield" input terminals are generally used for.
- Identify whether or not *bias resistors* are necessary to connect to the DAQ's input terminals. If so, where should those resistors connect, and what should their approximate sizes be?
- After you have sketched your circuit, evaluate the effects of various components failing either open or shorted, one at a time.

5.1.4 Connecting strain gauges to another DAQ

Sketch connecting wires to allow this data acquisition unit (DAQ) to sense strain using quarter-bridge strain gauge circuits on input channels #2 and #3, such that increasing tension on the strain gauge (increasing gauge resistance) generates a more *positive* signal voltage on each channel:

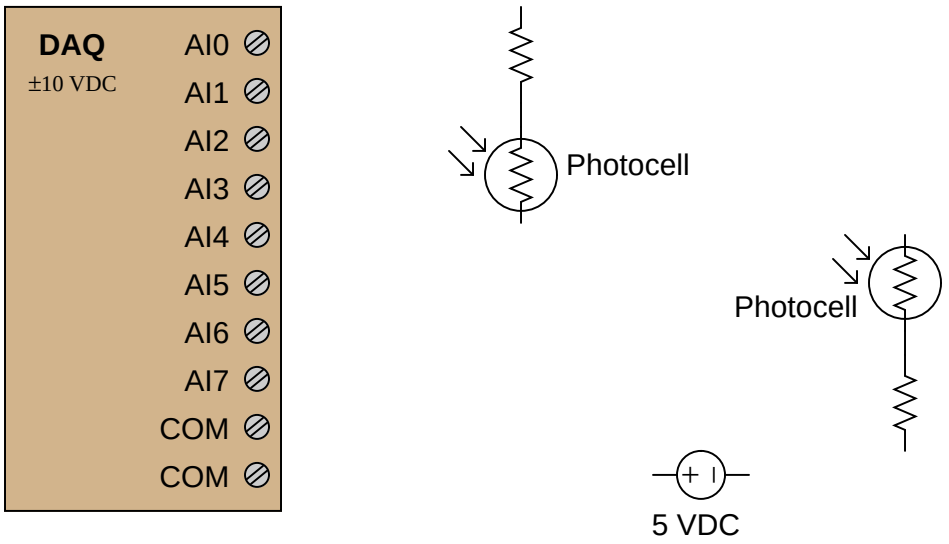


Challenges

- Re-draw the circuit using input channels #0 and #1.

5.1.5 Connecting photocells to a DAQ

Sketch connecting wires to allow this data acquisition unit (DAQ) to sense light on input channels #3 and #7, such that increasing light at each photocell generates an *increasing positive* signal voltage on each channel. Note that all the analog input channels on this DAQ are single-ended, bipolar:

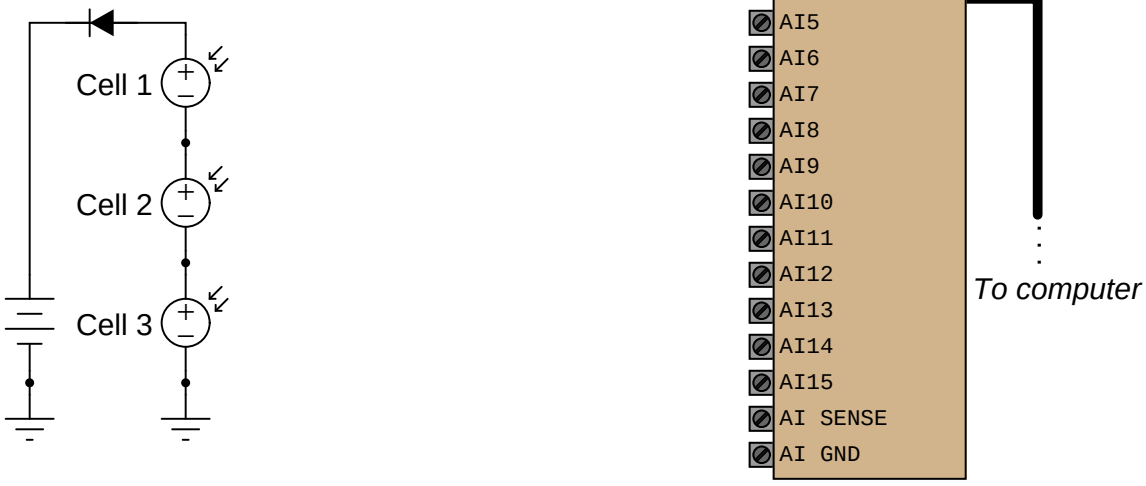


Challenges

- Identify factors which would help you select appropriate resistors for the fixed-value resistors in this circuit.
- A good problem-solving technique to apply in cases where we need to determine the direction of a change for a component's resistance in order to design a functioning circuit is to consider *limiting cases* for that component's resistance. For example, instead of asking ourselves what would happen if the intensity of light *slightly*, we ask ourselves what would happen if light intensity changed *dramatically*. Explain how this problem-solving technique applies to this particular system.
- After you have sketched your circuit, evaluate the effects of various components failing either open or shorted, one at a time.

5.1.6 Connecting solar cells to a DAQ

Identify suitable input terminals, proper modes, and necessary connecting wires to allow this National Instruments E-series data acquisition unit (DAQ) to independently sense the voltages of the three series-connected solar cells shown:



Note: the available modes for the input channels are `RSE`, `NRSE`, and `DIFF`:

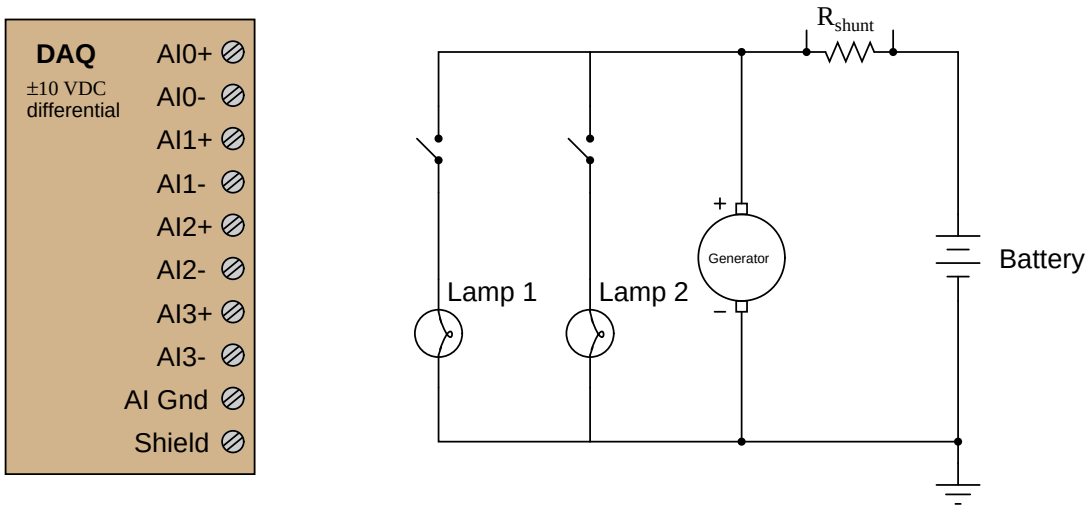
- Channel 0 = (mode, terminals) =
- Channel 1 = (mode, terminals) =
- Channel 2 = (mode, terminals) =

Challenges

- Describe the differences between `RSE`, `NRSE`, and `DIFF` modes for this model of DAQ.

5.1.7 DC circuit monitor

Sketch connecting wires to allow this data acquisition unit (DAQ) to sense current (using the "shunt" resistor as a primary sensing element) on input channel #2 and also generator output voltage on input channel #3:



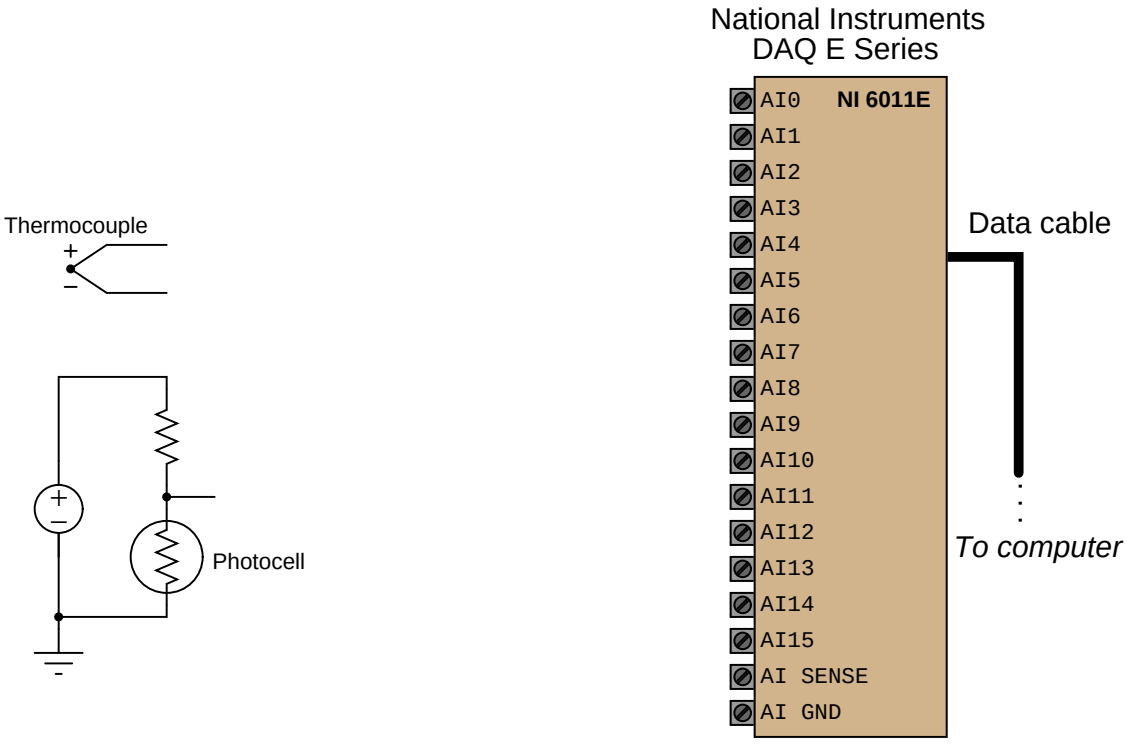
Your circuit should be wired in such a way that a *discharging* battery produces a *negative* signal at channel #2, and a *charging* battery produces a *positive* signal at channel #2. Draw an arrow in the circuit diagram showing the direction of current through the battery (conventional flow notation) while it is charging.

Challenges

- Identify a practical application for this circuit, where two lamps, a generator, and a battery connect to each other.
- Some differential-input DAQs have programmable voltage gain values for each input. Identify how this might be a useful feature for this measurement application.

5.1.8 Measuring light and temperature with a DAQ

Identify suitable input terminals, proper modes, and necessary connecting wires to allow this National Instruments E-series data acquisition unit (DAQ) to sense the two voltage sources shown:



Note: the available modes for the input channels are `RSE`, `NRSE`, and `DIFF`:

- Channel 0 = (mode, terminals) =
- Channel 1 = (mode, terminals) =

Challenges

- Will the light signal increase or decrease as more light strikes the photocell?

5.2 Quantitative reasoning

These questions are designed to stimulate your computational thinking. In a Socratic discussion with your instructor, the goal is for these questions to reveal your mathematical approach(es) to problem-solving so that good technique and sound reasoning may be reinforced. Your instructor may also pose additional questions based on those assigned, in order to observe your problem-solving firsthand.

Mental arithmetic and estimations are strongly encouraged for all calculations, because without these abilities you will be unable to readily detect errors caused by calculator misuse (e.g. keystroke errors).

You will note a conspicuous lack of answers given for these quantitative questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. My advice is to use circuit simulation software such as SPICE to check the correctness of quantitative answers. Refer to those learning modules within this collection focusing on SPICE to see worked examples which you may use directly as practice problems for your own study, and/or as templates you may modify to run your own analyses and generate your own practice problems.

Completely worked example problems found in the Tutorial may also serve as "test cases[4]" for gaining proficiency in the use of circuit simulation software, and then once that proficiency is gained you will never need to rely[5] on an answer key!

[4] In other words, set up the circuit simulation software to analyze the same circuit examples found in the Tutorial. If the simulated results match the answers shown in the Tutorial, it confirms the simulation has properly run. If the simulated results disagree with the Tutorial's answers, something has been set up incorrectly in the simulation software. Using every Tutorial as practice in this way will quickly develop proficiency in the use of circuit simulation software.

[5] This approach is perfectly in keeping with the instructional philosophy of these learning modules: *teaching students to be self-sufficient thinkers*. Answer keys can be useful, but it is even more useful to your long-term success to have a set of tools on hand for checking your own work, because once you have left school and are on your own, there will no longer be "answer keys" available for the problems you will have to solve.

5.2.1 Miscellaneous physical constants

Note: constants shown in **bold** type are *exact*, not approximations. Values inside of parentheses show one standard deviation (σ) of uncertainty in the final digits: for example, the magnetic permeability of free space value given as $1.25663706212(19) \times 10^{-6}$ H/m represents a center value (i.e. the location parameter) of $1.25663706212 \times 10^{-6}$ Henrys per meter with one standard deviation of uncertainty equal to $0.0000000000019 \times 10^{-6}$ Henrys per meter.

Avogadro's number (N_A) = **6.02214076** $\times 10^{23}$ **per mole** (mol^{-1})

Boltzmann's constant (k) = **1.380649** $\times 10^{-23}$ **Joules per Kelvin** (J/K)

Electronic charge (e) = **1.602176634** $\times 10^{-19}$ **Coulomb** (C)

Faraday constant (F) = **96,485.33212...** $\times 10^{4}$ **Coulombs per mole** (C/mol)

Magnetic permeability of free space (μ_0) = $1.25663706212(19) \times 10^{-6}$ Henrys per meter (H/m)

Electric permittivity of free space (ϵ_0) = $8.8541878128(13) \times 10^{-12}$ Farads per meter (F/m)

Characteristic impedance of free space (Z_0) = 376.730313668(57) Ohms (Ω)

Gravitational constant (G) = $6.67430(15) \times 10^{-11}$ cubic meters per kilogram-seconds squared ($m^3/kg\text{-}s^2$)

Molar gas constant (R) = **8.314462618... Joules per mole-Kelvin** (J/mol-K) = 0.08205746(14) liters-atmospheres per mole-Kelvin

Planck constant (h) = **6.62607015** $\times 10^{-34}$ **joule-seconds** (J-s)

Stefan-Boltzmann constant (σ) = **5.670374419...** $\times 10^{-8}$ **Watts per square meter-Kelvin**^{4} ($W/m^2 \cdot K^4$)

Speed of light in a vacuum (c) = **299,792,458 meters per second** (m/s) = 186282.4 miles per second (mi/s)

Note: All constants taken from NIST data "Fundamental Physical Constants – Complete Listing", from `http://physics.nist.gov/constants`, National Institute of Standards and Technology (NIST), 2018 CODATA Adjustment.

5.2.2 Introduction to spreadsheets

A powerful computational tool you are encouraged to use in your work is a *spreadsheet*. Available on most personal computers (e.g. Microsoft Excel), *spreadsheet* software performs numerical calculations based on number values and formulae entered into cells of a grid. This grid is typically arranged as lettered columns and numbered rows, with each cell of the grid identified by its column/row coordinates (e.g. cell `B3`, cell `A8`). Each cell may contain a string of text, a number value, or a mathematical formula. The spreadsheet automatically updates the results of all mathematical formulae whenever the entered number values are changed. This means it is possible to set up a spreadsheet to perform a series of calculations on entered data, and those calculations will be re-done by the computer any time the data points are edited in any way.

For example, the following spreadsheet calculates average speed based on entered values of distance traveled and time elapsed:

	A	B	C	D
1	Distance traveled	46.9	Kilometers	
2	Time elapsed	1.18	Hours	
3	Average speed	= B1 / B2	km/h	
4				
5				

Text labels contained in cells `A1` through `A3` and cells `C1` through `C3` exist solely for readability and are not involved in any calculations. Cell `B1` contains a sample distance value while cell `B2` contains a sample time value. The formula for computing speed is contained in cell `B3`. Note how this formula begins with an "equals" symbol (`=`), references the values for distance and speed by lettered column and numbered row coordinates (`B1` and `B2`), and uses a forward slash symbol for division (`/`). The coordinates `B1` and `B2` function as *variables*[6] would in an algebraic formula.

When this spreadsheet is executed, the numerical value `39.74576` will appear in cell `B3` rather than the formula `= B1 / B2`, because 39.74576 is the computed speed value given 46.9 kilometers traveled over a period of 1.18 hours. If a different numerical value for distance is entered into cell `B1` or a different value for time is entered into cell `B2`, cell `B3`'s value will automatically update. All you need to do is set up the given values and any formulae into the spreadsheet, and the computer will do all the calculations for you.

Cell `B3` may be referenced by other formulae in the spreadsheet if desired, since it is a variable just like the given values contained in `B1` and `B2`. This means it is possible to set up an entire chain of calculations, one dependent on the result of another, in order to arrive at a final value. The arrangement of the given data and formulae need not follow any pattern on the grid, which means you may place them anywhere.

[6] Spreadsheets may also provide means to attach text labels to cells for use as variable names (Microsoft Excel simply calls these labels "names"), but for simple spreadsheets such as those shown here it's usually easier just to use the standard coordinate naming for each cell.

Common[7] arithmetic operations available for your use in a spreadsheet include the following:

- Addition (`+`)
- Subtraction (`-`)
- Multiplication (`*`)
- Division (`/`)
- Powers (`^`)
- Square roots (`sqrt()`)
- Logarithms (`ln()` , `log10()`)

Parentheses may be used to ensure[8] proper order of operations within a complex formula. Consider this example of a spreadsheet implementing the *quadratic formula*, used to solve for roots of a polynomial expression in the form of $ax^2 + bx + c$:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

	A	B
1	x_1	= (-B4 + sqrt((B4^2) - (4*B3*B5))) / (2*B3)
2	x_2	= (-B4 - sqrt((B4^2) - (4*B3*B5))) / (2*B3)
3	a =	9
4	b =	5
5	c =	-2

This example is configured to compute roots[9] of the polynomial $9x^2 + 5x - 2$ because the values of 9, 5, and -2 have been inserted into cells `B3`, `B4`, and `B5`, respectively. Once this spreadsheet has been built, though, it may be used to calculate the roots of *any* second-degree polynomial expression simply by entering the new a, b, and c coefficients into cells `B3` through `B5`. The numerical values appearing in cells `B1` and `B2` will be automatically updated by the computer immediately following any changes made to the coefficients.

[7] Modern spreadsheet software offers a bewildering array of mathematical functions you may use in your computations. I recommend you consult the documentation for your particular spreadsheet for information on operations other than those listed here.

[8] Spreadsheet programs, like text-based programming languages, are designed to follow standard order of operations by default. However, my personal preference is to use parentheses even where strictly unnecessary just to make it clear to any other person viewing the formula what the intended order of operations is.

[9] Reviewing some algebra here, a *root* is a value for x that yields an overall value of zero for the polynomial. For this polynomial ($9x^2 + 5x - 2$) the two roots happen to be $x = 0.269381$ and $x = -0.82494$, with these values displayed in cells `B1` and `B2`, respectively upon execution of the spreadsheet.

Alternatively, one could break up the long quadratic formula into smaller pieces like this:

$$y = \sqrt{b^2 - 4ac} \qquad z = 2a$$

$$x = \frac{-b \pm y}{z}$$

	A	B	C
1	x_1	= (-B4 + C1) / C2	= sqrt((B4^2) - (4*B3*B5))
2	x_2	= (-B4 - C1) / C2	= 2*B3
3	a =	9	
4	b =	5	
5	c =	-2	

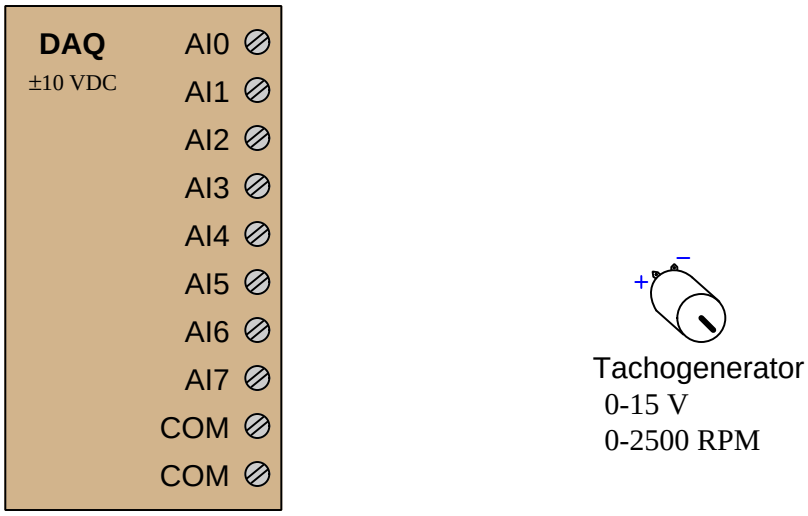
Note how the square-root term (y) is calculated in cell C1, and the denominator term (z) in cell C2. This makes the two final formulae (in cells B1 and B2) simpler to interpret. The positioning of all these cells on the grid is completely arbitrary[10] – all that matters is that they properly reference each other in the formulae.

Spreadsheets are particularly useful for situations where the same set of calculations representing a circuit or other system must be repeated for different initial conditions. The power of a spreadsheet is that it automates what would otherwise be a tedious set of calculations. One specific application of this is to simulate the effects of various components within a circuit failing with abnormal values (e.g. a shorted resistor simulated by making its value nearly zero; an open resistor simulated by making its value extremely large). Another application is analyzing the behavior of a circuit design given new components that are out of specification, and/or aging components experiencing drift over time.

[10] My personal preference is to locate all the "given" data in the upper-left cells of the spreadsheet grid (each data point flanked by a sensible name in the cell to the left and units of measurement in the cell to the right as illustrated in the first distance/time spreadsheet example), sometimes coloring them in order to clearly distinguish which cells contain entered data versus which cells contain computed results from formulae. I like to place all formulae in cells below the given data, and try to arrange them in logical order so that anyone examining my spreadsheet will be able to figure out *how* I constructed a solution. This is a general principle I believe all computer programmers should follow: *document and arrange your code to make it easy for other people to learn from it.*

5.2.3 Measuring speed with a DAQ

Show how to connect this tachogenerator to channel #2 on the DAQ so that the DAQ will output its maximum count value at a shaft speed of 2000 RPM:

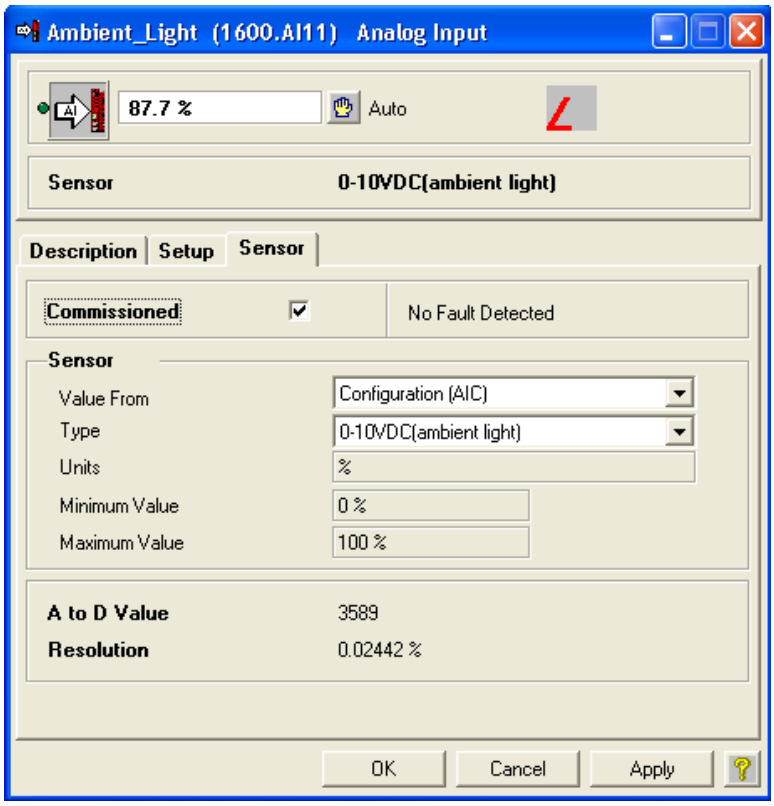


Challenges

- Will this system be able to measure shaft speed in both directions, or just in one direction?
- Tachogenerator voltage signals are often "noisy" due to the sliding contact of the generator's brushes against the commutator bars as the shaft rotates. How would you mitigate this electrical noise so that the DAQ does not get fooled into "thinking" the speed is actually spiking up and down as the noise might imply?

5.2.4 DDC light measurement

The following screenshot shows the configuration window for an analog input on a Delta model DSC-1280 DDC ("Direct Digital Control") building automation controller[11]:



Based on the parameters you see in this configuration window, how many bits does the analog-digital converter inside the DDC controller use?

Challenges

- Calculate the count value associated with a measured signal value of 48.3%.
- Calculate the percentage value associated with a count value of 1327 (decimal).

[11] This type of control system is generally applied to HVAC (heating, ventilation, and air-conditioning) scenarios.

5.3 Diagnostic reasoning

These questions are designed to stimulate your deductive and inductive thinking, where you must apply general principles to specific scenarios (deductive) and also derive conclusions about the failed circuit from specific details (inductive). In a Socratic discussion with your instructor, the goal is for these questions to reinforce your recall and use of general circuit principles and also challenge your ability to integrate multiple symptoms into a sensible explanation of what's wrong in a circuit. Your instructor may also pose additional questions based on those assigned, in order to further challenge and sharpen your diagnostic abilities.

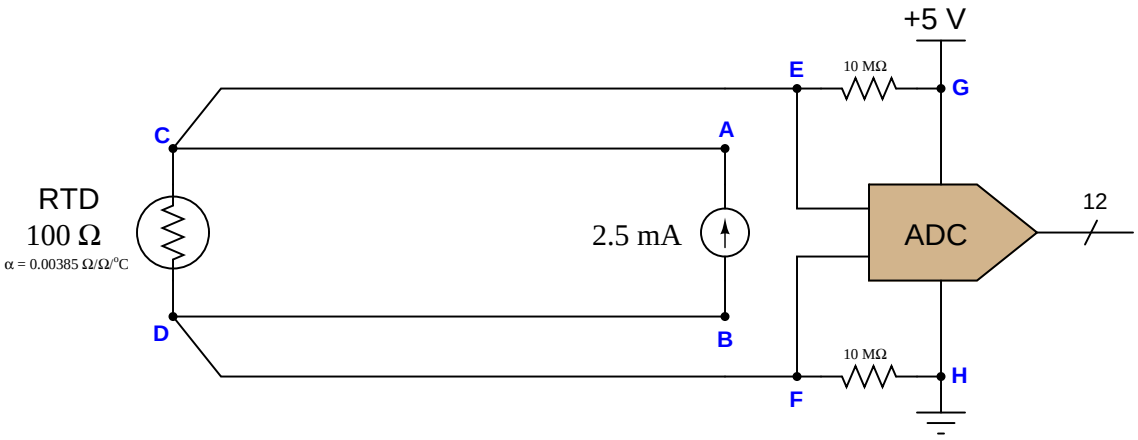
As always, your goal is to fully *explain* your analysis of each problem. Simply obtaining a correct answer is not good enough – you must also demonstrate sound reasoning in order to successfully complete the assignment. Your instructor's responsibility is to probe and challenge your understanding of the relevant principles and analytical processes in order to ensure you have a strong foundation upon which to build further understanding.

You will note a conspicuous lack of answers given for these diagnostic questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your diagnostic answers, where applicable, is to use circuit simulation software to explore the effects of faults placed in circuits. For example, if one of these diagnostic questions requires that you predict the effect of an open or a short in a circuit, you may check the validity of your work by simulating that same fault (substituting a very high resistance in place of that component for an open, and substituting a very low resistance for a short) within software and seeing if the results agree.

5.3.1 Faulty temperature measurement system

Suppose a digital temperature measurement system is not functioning as it should. The ADC is outputting a count value of `0x002` when the sensor is known to be at a temperature of 20 degrees Celsius (room temperature). The sensor is an RTD: a variable-resistance device which is supposed to exhibit 100 Ohms at 0 ^{o}C and 138.5 Ohms at 100 ^{o}C, linearly proportional to temperature. The analog input range of the ADC is 0 to 0.5 Volts DC. Your first diagnostic test is to check the voltage between test points **G** and **H**, and there you measure 5.08 Volts DC:



Identify the likelihood of each specified fault for this circuit. Consider each fault one at a time (i.e. no coincidental faults), determining whether or not each fault could independently account for *all* measurements and symptoms in this circuit.

- RTD failed open
- RTD failed shorted
- Broken wire between **C** and **E**
- Broken wire between **C** and **A**
- Broken wire between **D** and **B**
- Broken wire between **D** and **F**
- Dead current source

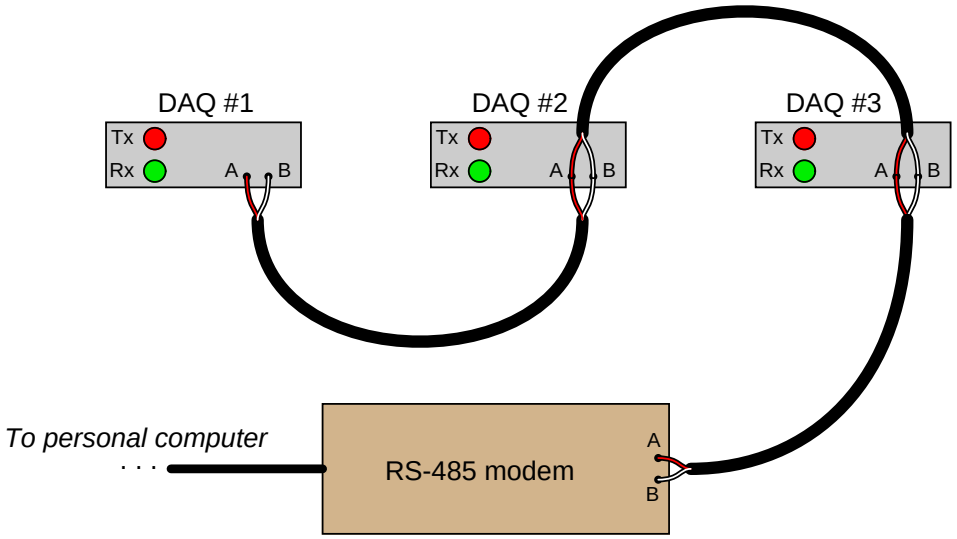
Finally, identify the *next* diagnostic test or measurement you would make on this system. Explain how the result(s) of this next test or measurement help further identify the location and/or nature of the fault.

Challenges

- What should the ADC's count value be at a temperature of 100 degrees Celsius?
- Why are four wires being used to connect the RTD sensor to the rest of the circuit rather than using just two?

5.3.2 Networked DAQ modules

A data acquisition system consisting of three DAQ[12] modules connected to a personal computer using RS-485 (multipoint) through a modem has a problem. The DAQ software running in the personal computer does not seem to "recognize" DAQ #1, although it does recognize the other two DAQ modules and is receiving data from them:



Looking at the LED indicators, you notice that the green "Rx" LEDs on all three DAQ units blink. You also notice that the red "Tx" LEDs on DAQ modules #2 and #3 are blinking as well, but that the "Tx" LED on DAQ #1 is constantly off.

Identify the likelihood of each specified fault for this system. Consider each fault one at a time (i.e. no coincidental faults), determining whether or not each fault is compatible with *all* measurements and symptoms in this network.

- Open cable between DAQ #1 and DAQ #2 =
- Open cable between DAQ #2 and DAQ #3 =
- Open cable between DAQ #3 and modem =
- Incorrect serial settings (e.g. parity, bit rate) in DAQ #1 =
- Incorrect serial settings (e.g. parity, bit rate) in DAQ #2 =
- Incorrect serial settings (e.g. parity, bit rate) in DAQ #3 =
- Incorrect serial settings (e.g. parity, bit rate) in computer =

[12] A "DAQ" module is a *Data AcQuisition* circuit that usually takes in analog or digital data from sensors and converts it into digital data readable to a computer. This functionality is not important to the question of networking.

- Missing termination resistor =

Challenges

- Is this network capable of simplex, half-duplex, or full-duplex operation?
- Identify another possible fault not present in this list.
- Identify how a digital multimeter (DMM) could be used to perform useful diagnostic tests in an RS-485 network such as this. What sorts of things can we measure using a DMM connected to a digital network? What sorts of things *can't* we measure using a DMM connected to a digital network?

Appendix A

Problem-Solving Strategies

The ability to solve complex problems is arguably one of the most valuable skills one can possess, and this skill is particularly important in any science-based discipline.

- Study principles, not procedures. Don't be satisfied with merely knowing how to compute solutions – learn *why* those solutions work.
- Identify what it is you need to solve, identify all relevant data, identify all units of measurement, identify any general principles or formulae linking the given information to the solution, and then identify any "missing pieces" to a solution. Annotate all diagrams with this data.
- Sketch a diagram to help visualize the problem. When building a real system, always devise a plan for that system and analyze its function *before* constructing it.
- Follow the units of measurement and meaning of every calculation. If you are ever performing mathematical calculations as part of a problem-solving procedure, and you find yourself unable to apply each and every intermediate result to some aspect of the problem, it means you don't understand what you are doing. Properly done, every mathematical result should have practical meaning for the problem, and not just be an abstract number. You should be able to identify the proper units of measurement for each and every calculated result, and show where that result fits into the problem.
- Perform "thought experiments" to explore the effects of different conditions for theoretical problems. When troubleshooting real systems, perform *diagnostic tests* rather than visually inspecting for faults, the best diagnostic test being the one giving you the most information about the nature and/or location of the fault with the fewest steps.
- Simplify the problem until the solution becomes obvious, and then use that obvious case as a model to follow in solving the more complex version of the problem.
- Check for exceptions to see if your solution is incorrect or incomplete. A good solution will work for *all* known conditions and criteria. A good example of this is the process of testing scientific hypotheses: the task of a scientist is not to find support for a new idea, but rather to *challenge* that new idea to see if it holds up under a battery of tests. The philosophical

principle of *reductio ad absurdum* (i.e. disproving a general idea by finding a specific case where it fails) is useful here.

- Work "backward" from a hypothetical solution to a new set of given conditions.
- Add quantities to problems that are qualitative in nature, because sometimes a little math helps illuminate the scenario.
- Sketch graphs illustrating how variables relate to each other. These may be quantitative (i.e. with realistic number values) or qualitative (i.e. simply showing increases and decreases).
- Treat quantitative problems as qualitative in order to discern the relative magnitudes and/or directions of change of the relevant variables. For example, try determining what happens if a certain variable were to increase or decrease before attempting to precisely calculate quantities: how will each of the dependent variables respond, by increasing, decreasing, or remaining the same as before?
- Consider limiting cases. This works especially well for qualitative problems where you need to determine which direction a variable will change. Take the given condition and magnify that condition to an extreme degree as a way of simplifying the direction of the system's response.
- Check your work. This means regularly testing your conclusions to see if they make sense. This does *not* mean repeating the same steps originally used to obtain the conclusion(s), but rather to use some other means to check validity. Simply repeating procedures often leads to *repeating the same errors* if any were made, which is why alternative paths are better.

Appendix B

Instructional philosophy

B.1 First principles of learning

- **Anyone can learn anything** given appropriate time, effort, resources, challenges, encouragement, and expectations. Dedicating time and investing effort are the student's responsibility; providing resources, challenges, and encouragement are the teacher's responsibility; high expectations are a responsibility shared by both student and teacher.

- **Transfer is not automatic.** The human mind has a natural tendency to compartmentalize information, which means the process of taking knowledge learned in one context and applying it to another usually does not come easy and therefore should never be taken for granted.

- **Learning is iterative.** The human mind rarely learns anything perfectly on the first attempt. Anticipate mistakes and plan for multiple tries to achieve full understanding, using the lessons of those mistakes as feedback to guide future attempts.

- **Information is absorbed, but understanding is created.** Facts and procedures may be memorized easily enough by repeated exposure, but the ability to reliably apply principles to novel scenarios only comes through intense personal effort. This effort is fundamentally creative in nature: casting new concepts in one's own words, running mental experiments to perceive patterns and trends, etc. These acts of making knowledge "one's own" need not be perfect in order to be effective, as the value lies in the activity and not necessarily the finished product. Depth of understanding and permanence of memory are both directly related to mental effort during the learning process – i.e. *easy learning is shallow and fleeting.*

- **Education trumps training.** There is no such thing as an entirely isolated subject, as all fields of knowledge are connected. Training is narrowly-focused and task-oriented. Education is broad-based and principle-oriented. When preparing for a life-long technical career, education beats training every time.

- **Character matters.** Poor habits are more destructive than deficits of knowledge or skill. This is especially true in collective endeavors, where a team's ability to function depends on trust between its members. Simply put, no one wants an untrustworthy person on their team. An essential component of education then, is character development.

- **People learn to be responsible by bearing responsibility.** An irresponsible person is someone who has never *had* to be responsible for anything that mattered enough to them. Just as anyone can learn anything, anyone can become responsible if the personal cost of irresponsibility becomes high enough.

- **What gets measured, gets done.** Accurate and relevant assessment of learning is key to ensuring all students learn. Therefore, it is imperative to measure what matters.

- **Failure is nothing to fear.** Every human being fails, and fails in multiple ways at multiple times. Eventual success only happens when we don't stop trying.

B.2 Proven strategies for instructors

- Assume every student is capable of learning anything they desire given the proper conditions. Treat them as capable adults by granting real responsibility and avoiding artificial incentives such as merit or demerit points.
- Create a consistent culture of high expectations across the entire program of study. Demonstrate and encourage patience, persistence, and a healthy sense of self-skepticism. Anticipate and de-stigmatize error. Teach respect for the capabilities of others as well as respect for one's own fallibility.
- Replace lecture with "inverted" instruction, where students first encounter new concepts through reading and then spend class time in Socratic dialogue with the instructor exploring those concepts and solving problems individually. There is a world of difference between observing someone solve a problem versus actually solving a problem yourself, and so the point of this form of instruction is to place students in a position where they *cannot* passively observe.
- Require students to read extensively, write about what they learn, and dialogue with you and their peers to sharpen their understanding. Apply Francis Bacon's advice that "reading maketh a full man; conference a ready man; and writing an exact man". These are complementary activities helping students expand their confidence and abilities.
- Use artificial intelligence (AI) to challenge student understanding rather than merely provide information. Find productive ways for AI to critique students' clarity of thought and of expression, for example by employing AI as a Socratic-style interlocutor or as a reviewer of students' journals. Properly applied, AI has the ability to expand student access to critical review well outside the bounds of their instructor's reach.
- Build frequent and rapid feedback into the learning process so that students know at all times how well they are learning, to identify problems early and fix them before they grow. Model the intellectual habit of self-assessing and self-correcting your own understanding (i.e. a cognitive *feedback loop*), encouraging students to do the same.
- Use "mastery" as the standard for every assessment, which means the exam or experiment or project must be done with 100% competence in order to pass. Provide students with multiple opportunity for re-tries (different versions of the assessment every time).
- Require students to devise their own hypotheses and procedures on all experiments, so that the process is truly a scientific one. Have students assess their proposed experimental procedures for risk and devise mitigations for those risks. Let nothing be pre-designed about students' experiments other than a stated task (i.e. what principle the experiment shall test) at the start and a set of demonstrable knowledge and skill objectives at the end.
- Have students build as much of their lab equipment as possible: building power sources, building test assemblies[1], and building complete working systems (no kits!). In order to provide

[1] In the program I teach, every student builds their own "Development Board" consisting of a metal chassis with DIN rail, terminal blocks, and an AC-DC power supply of their own making which functions as a portable lab environment they can use at school as well as take home.

this same "ground-up" experience for every new student, this means either previous students take their creations with them, or the systems get disassembled in preparation for the new students, or the systems grow and evolve with each new student group.

- Incorporate external accountability for you and for your students, continuously improving the curriculum and your instructional methods based on proven results. Have students regularly network with active professionals through participation in advisory committee meetings, service projects, tours, jobshadows, internships, etc. Practical suggestions include requiring students to design and build projects for external clients (e.g. community groups, businesses, different departments within the institution), and also requiring students attend all technical advisory committee meetings and to dialogue with the industry representatives at those meetings.
- Repeatedly explore difficult-to-learn concepts across multiple courses, so that students have multiple opportunities to build their understanding.
- Relate all new concepts, whenever possible, to previous concepts and to relevant physical laws. Challenge each and every student, every day, to *reason* from concept to concept and to explain the logical connections between. Challenge students to verify their conclusions by multiple approaches (e.g. double-checking their work using different methods). Ask *"Why?"* often.
- Maintain detailed records on each student's performance and share these records privately with them. These records should include academic performance as well as professionally relevant behavioral tendencies.
- Hold mandatory "check-in" meetings between all program faculty and each new student during their first term. Offer these to all other students as an option, except for any students continuing to manifest unprofessional behaviors, poor academic performance, or who have some other need for a face-to-face meeting with faculty.
- Address problems while they are small, before they grow larger. This is equally true for tutoring technical concepts as it is for helping students build professional habits.
- Build rigorous quality control into the curriculum to ensure every student masters every important concept, and that the mastery is retained over time. This includes (1) review questions added to every exam to re-assess knowledge taught in previous terms, (2) cumulative exams at the end of every term to re-assess all important concepts back to the very beginning of the program, and (3) review assessments in practical (hands-on) coursework to ensure critically-important skills were indeed taught and are still retained. What you will find by doing this is that it actually boosts retention of students by ensuring that important knowledge gets taught and is retained over long spans of time. In the absence of such quality control, student learning and retention tends to be spotty and this contributes to drop-out and failure rates later in their education.
- Finally, *never rush learning.* Education is not a race. Give your students ample time to digest complex ideas, as you continually remind yourself of just how long it took you to achieve mastery! Long-term retention and the consistently correct application of concepts are always the result of *focused effort over long periods of time* which means there are no shortcuts to learning.

B.3 Proven strategies for students

The single most important piece of advice I have for any student of any subject is to take responsibility for your own development in all areas of life including mental development. Expecting others in your life to entirely guide your own development is a recipe for disappointment. This is just as true for students enrolled in formal learning institutions as it is for auto-didacts pursuing learning entirely on their own. Learning to think in new ways is key to being able to gainfully use information, to make informed decisions about your life, and to best serve those you care about. With this in mind, I offer the following advice to students:

- **Approach all learning as valuable**. No matter what course you take, no matter who you learn from, no matter the subject, there is something useful in every learning experience. If you don't see the value of every new experience, you are not looking closely enough!
- **Continually challenge yourself**. Let other people take shortcuts and find easy answers to easy problems. The purpose of education is to stretch your mind, in order to shape it into a more powerful tool. This doesn't come by taking the path of least resistance. An excellent analogy for an empowering education is productive physical exercise: becoming stronger, more flexible, and more persistent only comes through intense personal effort.
- **Master the use of language**. This includes reading extensively, writing every day, listening closely, and speaking articulately. To a great extent language channels and empowers thought, so the better you are at wielding language the better you will be at grasping abstract concepts and articulating them not only for your benefit but for others as well.
- **Do not limit yourself to the resources given to you**. Read books that are not on the reading list. Run experiments that aren't assigned to you. Form study groups outside of class. Take an entrepreneurial approach to your own education, as though it were a business you were building for your future benefit.
- **Express and share what you learn**. Take every opportunity to teach what you have learned to others, as this will not only help them but will also strengthen your own understanding[2].
- Realize that **no one can give you understanding**, just as no one can give you physical fitness. These both must be *built*.
- **Above all, recognize that learning is hard work, and that a certain level of frustration is unavoidable**. There are times when you will struggle to grasp some of these concepts, and that struggle is a natural thing. Take heart that it will yield with persistent and varied[3] effort, and never give up! That concepts don't immediately come to you is not a sign of something wrong, but rather of something right: that you have found a worthy challenge!

[2] On a personal note, I was surprised to learn just how much my own understanding of electronics and related subjects was strengthened by becoming a teacher. When you are tasked every day with helping other people grasp complex topics, it catalyzes your own learning by giving you powerful incentives to study, to articulate your thoughts, and to reflect deeply on the process of learning.

[3] As the old saying goes, "Insanity is trying the same thing over and over again, expecting different results." If you find yourself stumped by something in the text, you should attempt a different approach. Alter the thought experiment, change the mathematical parameters, do whatever you can to see the problem in a slightly different light, and then the solution will often present itself more readily.

B.4 Design of these learning modules

"The unexamined circuit is not worth energizing" – Socrates (if he had taught electricity)

These learning modules, although useful for self-study, were designed to be used in a formal learning environment where a subject-matter expert challenges students to digest the content and exercise their critical thinking abilities in the answering of questions and in the construction and testing of working circuits. Every effort has been made to embed the following instructional and assessment philosophies within:

- The first goal of education is to enhance clear and independent thought, in order that every student reach their fullest potential in a highly complex and inter-dependent world. Robust reasoning is *always* more important than particulars of any subject matter, because its application is universal.
- Literacy is fundamental to independent learning and thought because text continues to be the most efficient way to communicate complex ideas over space and time. Those who cannot read with ease are limited in their ability to acquire knowledge and perspective.
- Articulate communication is fundamental to work that is complex and interdisciplinary.
- Faulty assumptions and poor reasoning are best corrected through challenge, not presentation. The rhetorical technique of *reductio ad absurdum* (disproving an assertion by exposing an absurdity) works well to discipline student's minds, not only to correct the problem at hand but also to learn how to detect and correct future errors.
- Important principles should be repeatedly explored and widely applied throughout a course of study, not only to reinforce their importance and help ensure their mastery, but also to showcase the interconnectedness and utility of knowledge.

These learning modules were expressly designed to be used in an "inverted" teaching environment[4] where students first read the introductory and tutorial chapters on their own, then individually attempt to answer the questions and construct working circuits according to the experiment and project guidelines. The instructor never lectures, but instead meets regularly with each individual student to review their progress, answer questions, identify misconceptions, and challenge the student to new depths of understanding through further questioning. Regular meetings between instructor and student should resemble a Socratic[5] dialogue, where questions serve as scalpels to dissect topics and expose assumptions. The student passes each module only after consistently demonstrating their ability to logically analyze and correctly apply all major concepts in each question or project/experiment. The instructor must be vigilant in probing each student's understanding to ensure they are truly *reasoning* and not just *memorizing*. This is why "Challenge" points appear throughout, as prompts for students to think deeper about topics and as starting points for instructor queries. Sometimes these challenge points require additional knowledge that hasn't been covered in the series to answer in full. This is okay, as the major purpose of the Challenges is to stimulate analysis and synthesis on the part of each student.

The instructor must possess enough mastery of the subject matter and awareness of students' reasoning to generate their own follow-up questions to practically any student response. Even completely correct answers given by the student should be challenged by the instructor for the purpose of having students practice articulating their thoughts and defending their reasoning. Conceptual errors committed by the student should be exposed and corrected not by direct instruction, but rather by reducing the errors to an absurdity[6] through well-chosen questions and thought experiments posed by the instructor. Becoming proficient at this style of instruction requires time and dedication, but the positive effects on critical thinking for both student and instructor are spectacular.

An inspection of these learning modules reveals certain unique characteristics. One of these is a bias toward thorough explanations in the tutorial chapters. Without a live instructor to explain concepts and applications to students, the text itself must fulfill this role. This philosophy results in lengthier explanations than what you might typically find in a textbook, each step of the reasoning process fully explained, including footnotes addressing common questions and concerns students raise while learning these concepts. Each tutorial seeks to not only explain each major concept in sufficient detail, but also to explain the logic of each concept and how each may be developed

[4] In a traditional teaching environment, students first encounter new information via *lecture* from an expert, and then independently apply that information via *homework*. In an "inverted" course of study, students first encounter new information via *homework*, and then independently apply that information under the scrutiny of an expert. The expert's role in lecture is to simply *explain*, but the expert's role in an inverted session is to *challenge*, *critique*, and if necessary *explain* where gaps in understanding still exist.

[5] Socrates is a figure in ancient Greek philosophy famous for his unflinching style of questioning. Although he authored no texts, he appears as a character in Plato's many writings. The essence of Socratic philosophy is to leave no question unexamined and no point of view unchallenged. While purists may argue a topic such as electric circuits is too narrow for a true Socratic-style dialogue, I would argue that the essential thought processes involved with scientific reasoning on *any* topic are not far removed from the Socratic ideal, and that students of electricity and electronics would do very well to challenge assumptions, pose thought experiments, identify fallacies, and otherwise employ the arsenal of critical thinking skills modeled by Socrates.

[6] This rhetorical technique is known by the Latin phrase *reductio ad absurdum*. The concept is to expose errors by counter-example, since only one solid counter-example is necessary to disprove a universal claim. As an example of this, consider the common misconception among beginning students of electricity that voltage cannot exist without current. One way to apply *reductio ad absurdum* to this statement is to ask how much current passes through a fully-charged battery connected to nothing (i.e. a clear example of voltage existing without current).

from "first principles". Again, this reflects the goal of developing clear and independent thought in students' minds, by showing how clear and logical thought was used to forge each concept. Students benefit from witnessing a model of clear thinking in action, and these tutorials strive to be just that.

Another feature of these learning modules is that they do not treat topics in isolation. Rather, important concepts are introduced early in the series, and appear repeatedly as stepping-stones toward other concepts in subsequent modules. This helps to avoid the "compartmentalization" of knowledge, demonstrating the inter-connectedness of concepts and simultaneously reinforcing them. Each module is fairly complete in itself, reserving the beginning of its tutorial to a review of foundational concepts.

To high standards of education,

Tony R. Kuphaldt

Appendix C

Tools used

I am indebted to the developers of many open-source software applications in the creation of these learning modules. The following is a list of these applications with some commentary on each.

You will notice a theme common to many of these applications: a bias toward *code*. Although I am by no means an expert programmer in any computer language, I understand and appreciate the flexibility offered by code-based applications where the user (you) enters commands into a plain ASCII text file, which the software then reads and processes to create the final output. Code-based computer applications are by their very nature *extensible*, while WYSIWYG (What You See Is What You Get) applications are generally limited to whatever user interface the developer makes for you.

The `GNU/Linux` computer operating system

> There is so much to be said about Linus Torvalds' `Linux` and Richard Stallman's `GNU` project. First, to credit just these two individuals is to fail to do justice to the *mob* of passionate volunteers who contributed to make this amazing software a reality. I first learned of `Linux` back in 1996, and have been using this operating system on my personal computers almost exclusively since then. It is *free*, it is completely *configurable*, and it permits the continued use of highly efficient `Unix` applications and scripting languages (e.g. shell scripts, Makefiles, `sed`, `awk`) developed over many decades. `Linux` not only provided me with a powerful computing platform, but its open design served to inspire my life's work of creating open-source educational resources.

Bram Moolenaar's `Vim` text editor

> Writing code for any code-based computer application requires a *text editor*, which may be thought of as a word processor strictly limited to outputting plain-ASCII text files. Many good text editors exist, and one's choice of text editor seems to be a deeply personal matter within the programming world. I prefer `Vim` because it operates very similarly to `vi` which is ubiquitous on `Unix`/`Linux` operating systems, and because it may be entirely operated via keyboard (i.e. no mouse required) which makes it fast to use.

Donald Knuth's TEX typesetting system

Developed in the late 1970's and early 1980's by computer scientist extraordinaire Donald Knuth to typeset his multi-volume magnum opus *The Art of Computer Programming*, this software allows the production of formatted text for screen-viewing or paper printing, all by writing plain-text code to describe how the formatted text is supposed to appear. TEX is not just a markup language for documents, but it is also a Turing-complete programming language in and of itself, allowing useful algorithms to be created to control the production of documents. Simply put, TEX *is a programmer's approach to word processing*. Since TEX is controlled by code written in a plain-text file, this means anyone may read that plain-text file to see exactly how the document was created. This openness afforded by the code-based nature of TEX makes it relatively easy to learn how other people have created their own TEX documents. By contrast, examining a beautiful document created in a conventional WYSIWYG word processor such as Microsoft `Word` suggests nothing to the reader about *how* that document was created, or what the user might do to create something similar. As Mr. Knuth himself once quipped, conventional word processing applications should be called WYSIAYG (What You See Is *All* You Get).

Leslie Lamport's LATEX extensions to TEX

Like all true programming languages, TEX is inherently extensible. So, years after the release of TEX to the public, Leslie Lamport decided to create a massive extension allowing easier compilation of book-length documents. The result was LATEX, which is the markup language used to create all ModEL module documents. You could say that TEX is to LATEX as `C` is to `C++`. This means it is permissible to use any and all TEX commands within LATEX source code, and it all still works. Some of the features offered by LATEX that would be challenging to implement in TEX include automatic index and table-of-content creation.

Tim Edwards' `Xcircuit` drafting program

This wonderful program is what I use to create all the schematic diagrams and illustrations (but not photographic images or mathematical plots) throughout the ModEL project. It natively outputs PostScript format which is a true vector graphic format (this is why the images do not pixellate when you zoom in for a closer view), and it is so simple to use that I have never had to read the manual! Object libraries are easy to create for `Xcircuit`, being plain-text files using PostScript programming conventions. Over the years I have collected a large set of object libraries useful for drawing electrical and electronic schematics, pictorial diagrams, and other technical illustrations.

`Gimp` graphic image manipulation program

Essentially an open-source clone of Adobe's `PhotoShop`, I use `Gimp` to resize, crop, and convert file formats for all of the photographic images appearing in the ModEL modules. Although `Gimp` does offer its own scripting language (called `Script-Fu`), I have never had occasion to use it. Thus, my utilization of `Gimp` to merely crop, resize, and convert graphic images is akin to using a sword to slice bread.

`SPICE` circuit simulation program

`SPICE` is to circuit analysis as TEX is to document creation: it is a form of markup language designed to describe a certain object to be processed in plain-ASCII text. When the plain-text "source file" is compiled by the software, it outputs the final result. More modern circuit analysis tools certainly exist, but I prefer `SPICE` for the following reasons: it is *free*, it is *fast*, it is *reliable*, and it is a fantastic tool for *teaching* students of electricity and electronics how to write simple code. I happen to use rather old versions of `SPICE`, version 2g6 being my "go to" application when I only require text-based output. `NGSPICE` (version 26), which is based on Berkeley `SPICE` version 3f5, is used when I require graphical output for such things as time-domain waveforms and Bode plots. In all `SPICE` example netlists I strive to use coding conventions compatible with all `SPICE` versions.

Andrew D. Hwang's `ePiX` mathematical visualization programming library

This amazing project is a `C++` library you may link to any `C/C++` code for the purpose of generating PostScript graphic images of mathematical functions. As a completely free and open-source project, it does all the plotting I would otherwise use a Computer Algebra System (CAS) such as `Mathematica` or `Maple` to do. It should be said that `ePiX` is *not* a Computer Algebra System like `Mathematica` or `Maple`, but merely a mathematical *visualization* tool. In other words, it won't determine integrals for you (you'll have to implement that in your own `C/C++` code!), but it can graph the results, and it does so beautifully. What I really admire about `ePiX` is that it is a `C++` programming library, which means it builds on the existing power and toolset available with that programming language. Mr. Hwang could have probably developed his own stand-alone application for mathematical plotting, but by creating a `C++` library to do the same thing he accomplished something much greater.

`gnuplot` mathematical visualization software

Another open-source tool for mathematical visualization is `gnuplot`. Interestingly, this tool is *not* part of Richard Stallman's GNU project, its name being a coincidence. For this reason the authors prefer "gnu" *not* be capitalized at all to avoid confusion. This is a much "lighter-weight" alternative to a spreadsheet for plotting tabular data, and the fact that it easily outputs directly to an X11 console or a file in a number of different graphical formats (including PostScript) is very helpful. I typically set my `gnuplot` output format to default (X11 on my Linux PC) for quick viewing while I'm developing a visualization, then switch to PostScript file export once the visual is ready to include in the document(s) I'm writing. As with my use of `Gimp` to do rudimentary image editing, my use of `gnuplot` only scratches the surface of its capabilities, but the important points are that it's *free* and that it *works well*.

`Python` programming language

Both Python and C++ find extensive use in these modules as instructional aids and exercises, but I'm listing Python here as a *tool* for myself because I use it almost daily as a *calculator*. If you open a Python interpreter console and type `from math import *` you can type mathematical expressions and have it return results just as you would on a hand calculator. Complex-number (i.e. *phasor*) arithmetic is similarly supported if you include the complex-math library (`from cmath import *`). Examples of this are shown in the Programming References chapter (if included) in each module. Of course, being a fully-featured programming language, Python also supports conditionals, loops, and other structures useful for calculation of quantities. Also, running in a console environment where all entries and returned values show as text in a chronologically-ordered list makes it easy to copy-and-paste those calculations to document exactly how they were performed.

Appendix D

Creative Commons License

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or

limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

A. reproduce and Share the Licensed Material, in whole or in part; and

B. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures.

For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors,

whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully

be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at `creativecommons.org/policies`, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at `creativecommons.org`.

Appendix E

References

"DAQ E Series – E Series User Manual", document 370503K-01, National Instruments Corporation, Austin, TX, 2007.

Appendix F

Version history

This is a list showing all significant additions, corrections, and other edits made to this learning module. Each entry is referenced by calendar date in reverse chronological order (newest version first), which appears on the front cover of every learning module for easy reference. Any contributors to this open-source document are listed here as well.

13 May 2026 – added more to the Tutorial regarding the technique of envisioning a DAQ as a multi-channel multimeter.

13 April 2026 – minor edit to labeling within image_4151.

3 November 2025 – edited the Tutorial where it discusses DAQ input bias currents, pointing out how some opamp inputs source current while others sink current. This was set up as a new section in the Tutorial. Also added some instructor comments.

24 April 2025 – added another Technical References section, this one on common-mode signals.

8 April 2025 – added a new Challenge question to the "DC circuit monitor" Conceptual Reasoning question.

13 November 2024 – added a new Technical Reference section on general opamp recommendations.

6 November 2024 – corrected an error in one of the Challenge questions in the "Connecting strain gauges to another DAQ" Conceptual Reasoning question.

29 October 2024 – divided the Introduction chapter into sections, one with recommendations for students, one with a listing of challenging concepts, and one with recommendations for instructors. Also added a new Technical Reference section on ground in electric and electronic circuits, and made minor edits to the Tutorial.

9 April 2024 – added a new Tutorial section discussing the purpose of DAQs, and also added two Diagnostic Reasoning questions.

26 March 2024 – added a Case Tutorial chapter with a section showing the use of a LabJack model U3 DAQ controlled by Python scripts.

20 January 2024 – added a new Technical Reference section on protecting amplifier inputs against overvoltage from electro-static discharge (ESD) as well as electrical over-stress (EOS) from improperly connected signal sources.

12 November 2023 – included a Technical Reference section on operational amplifier imperfections.

29 November 2022 – placed questions at the top of the itemized list in the Introduction chapter prompting students to devise experiments related to the tutorial content.

10 May 2021 – commented out or deleted empty chapters.

18 March 2021 – minor additions to the Introduction, as well as added Ground terminals to all the differential DAQ illustrations.

5 October 2020 – significantly edited the Introduction chapter to make it more suitable as a pre-study guide and to provide cues useful to instructors leading "inverted" teaching sessions. Also, defined "unipolar" and "bipolar" ranges in the Tutorial, and added a footnote on bridge networks.

15 July 2020 – added more Foundational Concepts to the list.

14 July 2020 – document first created.

Index

ADC, 17
Adding quantities to a qualitative problem, 96
Aliasing, 18
Analog input, single-ended, 20
Annotating diagrams, 95

Benchmark, 33
Bias current, 55
Bias voltage, opamp, 59
Bipolar range, 18
Bipolar transistor, 63
Bridge network, 24

Capacitance, parasitic, 66
Channel, 18
Checking for exceptions, 96
Checking your work, 96
Choke, common-mode, 42
Clipping network, 63
CMOS, 64
CMRR, 62
Code, computer, 105
Common-mode choke, 42
Common-mode current, 42
Common-mode rejection ratio, 62
Common-mode signal, 37
Common-mode voltage, 19, 21, 24, 26, 37, 54, 62
Current, bias, 55
Current, common-mode, 42

DAC, 17
DAQ, 8, 17, 18
DAQ input, single-ended, 20
Data acquisition unit, 17, 18
Data acquisition unit (DAQ), 8
Datum, 33
DIAC, 66
Dimensional analysis, 95
Diode, protection, 53, 57, 63
Diode, Schottky, 64

Edwards, Tim, 106
EIA/TIA-232, 18
Electric potential, 32
Electrical over-stress, 63
Electro-static discharge, 63
EOS, 63
ESD, 63
Ethernet, 18

Field-effect transistor, 63
Filter, low-pass, 66
Floating voltage signal, 45

Gain-bandwidth product, 60
GBWP, 60
Graph values to solve a problem, 96
Greenleaf, Cynthia, 69
Ground, 33
Ground loop, 22
Ground, different definitions of, 32

Hot, 36
How to teach with these modules, 103
Hwang, Andrew D., 107

Identify given data, 95
Identify relevant principles, 95
Input protection diode, 53, 57, 63
Input, rail-to-rail, 54
Intermediate results, 95
Inverted instruction, 103

Knuth, Donald, 106

LabJack, 8
Lamport, Leslie, 106
Latch-up, 54
Limiting cases, 96
Low-pass filter, 66

Mean Sea Level, 33
Metacognition, 74
Moolenaar, Bram, 105
MOSFET, 63
Murphy, Lynn, 69

Neutral, 36
NPN, 63

Offset currrent, opamp, 58
Offset voltage, opamp, 59
Ohm's Law, 22
Open-source, 105
Output, rail-to-rail, 52

Parasitic capacitance, 66
PCI bus, 18
Photocell, 44
Photoresistor, 44
PNP, 63
Potential, electric, 32
Problem-solving: annotate diagrams, 95
Problem-solving: check for exceptions, 96
Problem-solving: checking work, 96
Problem-solving: dimensional analysis, 95
Problem-solving: graph values, 96
Problem-solving: identify given data, 95
Problem-solving: identify relevant principles, 95
Problem-solving: interpret intermediate results, 95
Problem-solving: limiting cases, 96
Problem-solving: qualitative to quantitative, 96
Problem-solving: quantitative to qualitative, 96
Problem-solving: reductio ad absurdum, 96
Problem-solving: simplify the system, 95
Problem-solving: thought experiment, 95
Problem-solving: track units of measurement, 95
Problem-solving: visually represent the system, 95
Problem-solving: work in reverse, 96
Protection diode, 53, 57, 63

Qualitatively approaching a quantitative problem, 96

Rail, DC power supply, 63
Rail-to-rail input, 54
Rail-to-rail output, 52
Reading Apprenticeship, 69
Reductio ad absurdum, 96, 102, 103
RTD, 20

Schoenbach, Ruth, 69
Schottky diode, 64
Scientific method, 74
Sea Level, Mean, 33
Signal, common-mode, 37
Signal, floating, 45
Simplifying a system, 95
Single-ended analog input, 20
Socrates, 102
Socratic dialogue, 103
SPICE, 69
Stallman, Richard, 105

Thévenin resistance, 58
Thermistor, 20
Thermocouple, 45
Thought experiment, 95
Thyristor, 66
Torvalds, Linus, 105
Transistor, 63

Unipolar range, 18
Units of measurement, 95
USB, 18

Visualizing a system, 95
Voltage, 32
Voltage follower, 57, 61
Voltage signal, floating, 45
Voltage, common-mode, 37, 54, 62

Work in reverse to solve a problem, 96
WYSIWYG, 105, 106