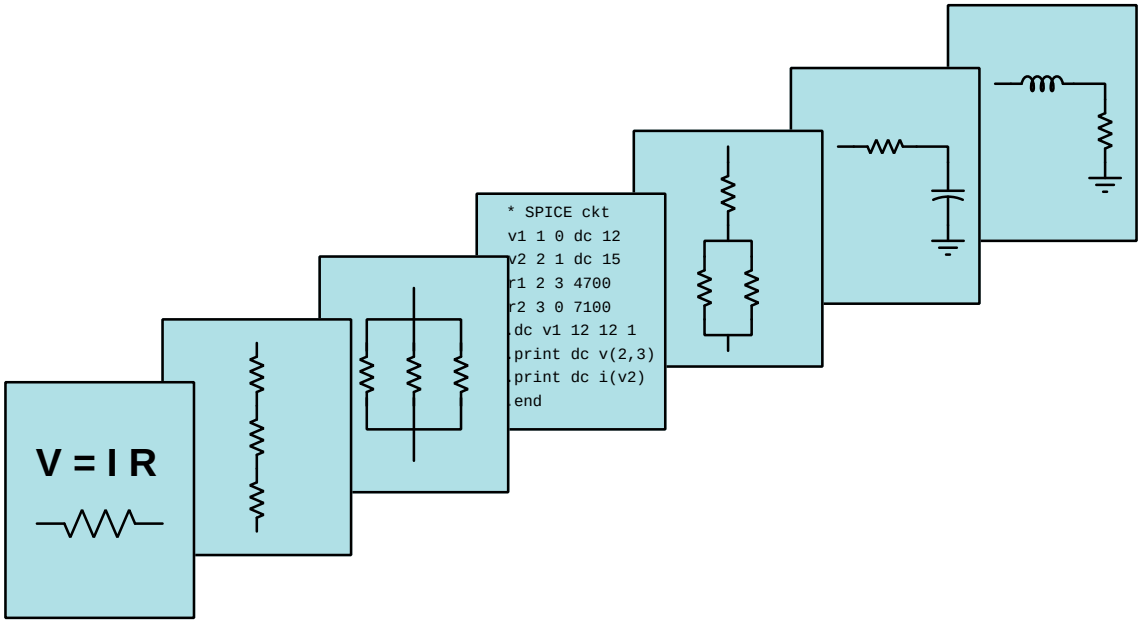
Modular Electronics Learning (ModEL) project



Introduction to Process Control

© 2023-2026 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 7 August 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 process control . . . 5
- 1.3 Recommendations for instructors . . . 6

2 Tutorial **7**
- 2.1 What is Instrumentation? . . . 8
- 2.2 Terms and definitions . . . 10
- 2.3 Control system examples . . . 12
  - 2.3.1 Example: thermostatic room control system . . . 13
  - 2.3.2 Example: boiler water level control system . . . 14
  - 2.3.3 Example: wastewater disinfection . . . 19
  - 2.3.4 Example: chemical reactor temperature control . . . 21
- 2.4 Other types of instruments . . . 24
  - 2.4.1 Indicators . . . 25
  - 2.4.2 Recorders . . . 28
  - 2.4.3 Process switches and alarms . . . 34
  - 2.4.4 Annunciators . . . 40

3 Questions **43**
- 3.1 Conceptual reasoning . . . 47
  - 3.1.1 Reading outline and reflections . . . 48
  - 3.1.2 Foundational concepts . . . 49
  - 3.1.3 Manual water reservoir control . . . 51
  - 3.1.4 Annunciator circuit . . . 53
  - 3.1.5 Boiler control system . . . 54
  - 3.1.6 Choosing proper controller action . . . 56
  - 3.1.7 Negative-feedback PLC control . . . 65
- 3.2 Quantitative reasoning . . . 67
  - 3.2.1 Miscellaneous physical constants . . . 68
  - 3.2.2 Introduction to spreadsheets . . . 69
  - 3.2.3 Diesel engine speed monitor . . . 72

A Problem-Solving Strategies **75**

B Instructional philosophy **77**
B.1 First principles of learning 78
B.2 Proven strategies for instructors 79
B.3 Proven strategies for students 81
B.4 Design of these learning modules 82

C Tools used **85**

D Creative Commons License **89**

E References **97**

F Version history **99**

Index **99**

Chapter 1

Introduction

1.1 Recommendations for students

This module introduces the field industrial instrumentation, which regards the automated monitoring and/or control of certain "process variables" such as temperature, pressure, level, flow rate, etc. Closed-loop control actually pre-dates electronics as a profession, some of the earliest examples being mechanical *governors* used to regulate the speed of steam engines in the 18th and 19th centuries. Practical applications abound for closed-loop control, including vehicle speed regulation (i.e. "cruise control"), engine air/fuel mixture control, powerplant regulation, climate control systems for building interiors, and many, many others.

Important concepts related to industrial instrumentation include **negative feedback**, **sensors**, **transmitters**, **control valves**, **setpoint**, **process variable**, **indicator**, **controller**, **automatic** versus **manual** mode, **process**, **P&ID**, and **loop diagram**.

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

- How might an experiment be designed and conducted to explore the concept of negative feedback? What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?
- What is the definition of a "process" and what are some practical examples of processes?
- What is the purpose of a "setpoint" for an automatic controller?
- Why is it important that feedback be *negative* rather than *positive* for closed-loop control?
- What is a "load" in the context of control systems, and how does this new definition differ from the concept of "load" within electric circuits?
- How do loads necessitate closed-loop control systems?
- What does it mean if a controller is "direct" versus "reverse" acting?
- What is the purpose of a *transducer* in an industrial control system?

- How are instrument locations specified within P&IDs and/or loop diagrams?
- What do "zero" and "span" refer to for an instrument's calibrated range?
- What do "LRV" and "URV" refer to for an instrument's calibrated range?
- What is the purpose of a *heat exchanger*, and where might we find such a device used?
- What variables need to be monitored and controlled for a large steam boiler process?
- What variables need to be monitored and controlled for a water disinfection process?
- What does it mean for a compressor to *surge*, and how may we avoid this potentially destructive condition?
- How does a 4-20 mA electronic signal represent process variable information?
- How does a 3-15 PSI pneumatic signal represent process variable information?
- What does it mean if a control valve is *air-to-open* versus *air-to-close*?
- Why might it be a good idea to intentionally calibrate a process transmitter so that its LRV and URV do *not* align with the physical minimum and maximum limits of the process variable?
- How might you alter one of the closed-loop control systems shown in the text, and then determine the behavior of that altered control system?
- Devise your own question based on the text, suitable for posing to someone encountering this subject for the first time

1.2 Challenging concepts related to process control

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

- **Negative feedback** – this non-intuitive concept is difficult to grasp due its recursive nature: the output of an amplifier is fed back to its input, which affects its output, which affects its input, *ad infinitum*. A particular challenge with negative feedback is comprehending why this does not irresistably lead to oscillations, given the fed-back signal's tendency to counter whatever direction the output signal is changing in. One rationale for the oscillation hypothesis is that we as human beings tend to imagine changes in discrete steps rather than as continuous trends, and so it is understandable why a new student might assume negative feedback causes the amplifier to leap back and forth between the (only) two extremes of signal range they might be envisioning. In reality, as the output of an amplifier with negative feedback drifts in one direction, the fed-back signal has the effect of slowing that drift down until it settles at some value rather than oscillate: as the "error" approaches zero, so does the rate of drift.

- **Faults in a feedback system** – predicting the effect(s) of a given fault in a negative-feedback system can be confusing because of the functional inversion that system provides. For example, if something happens in the feedback signal path to attenuate that signal, the result will be the amplifier generating an ever *larger* output signal to compensate!

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** – Apply the concept of negative feedback to realistic process control applications

  Assessment – Identify the effects of various changes made to a boiler water level control system; e.g. pose problems in the form of the "Boiler control system" Conceptual Reasoning question.

  Assessment – Identify proper controller action (either direct or reverse) to control given processes with given valve and transmitter actions; e.g. pose problems in the form of the "Choosing proper controller action" Conceptual Reasoning question.

- **Outcome** – Independent research

  Assessment – Locate a process controller manual and properly interpret some of the information contained in that document including input signal types, networking capability, operating modes, etc.

Chapter 2

Tutorial

2.1 What is Instrumentation?

Instrumentation is the science of automated measurement and control. Applications of this science abound in modern research, industry, and everyday living. From automobile engine control systems to home thermostats to aircraft autopilots to the manufacture of pharmaceutical drugs, automation surrounds us. This Tutorial explores some of the fundamental principles of industrial instrumentation, and also shows how we are able to implement automated control of physical processes through the use of negative feedback.

The first requirement for any automated system is measurement. If we can't measure something, it is really pointless to try to control it. This "something" usually takes one of the following forms in industry:

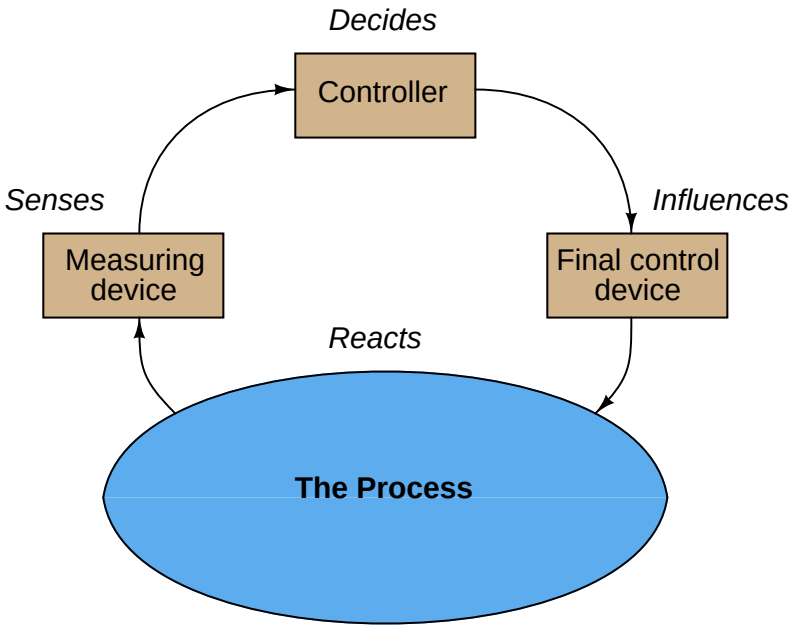
- Fluid pressure
- Fluid flow rate
- The temperature of an object
- Fluid volume stored in a vessel
- Chemical concentration
- Machine position, motion, or acceleration
- Physical dimension(s) of an object
- Count (inventory) of objects
- Electrical voltage, current, or resistance

Once we measure the quantity we are interested in, we usually transmit a signal representing this quantity to an indicating or computing device where either human or automated action then takes place. If the controlling action is automated, the computer sends a signal to a final controlling device which then influences the quantity being measured.

This final control device usually takes one of the following forms:

- Control valve (for throttling the flow rate of a fluid)
- Electric motor
- Electric heater

Both the measurement device and the final control device connect to some physical system which we call the *process*. To show this as a general block diagram:



The common home thermostat is an example of a measurement and control system, with the home's internal air temperature being the "process" under control. In this example, the thermostat usually serves two functions: sensing and control, while the home's heater adds heat to the home to increase temperature, and/or the home's air conditioner extracts heat from the home to decrease temperature. The job of this control system is to maintain air temperature at some comfortable level, with the heater or air conditioner taking action to correct temperature if it strays too far from the desired value (called the *setpoint*).

Most automated control systems employ a general strategy of *negative feedback* to maintain their process variable(s) at setpoint. This term – negative feedback – simply refers to the fact that the control system works to counter-act any undesired changes in the process variable. In the case of a room thermostat system, negative feedback means that any undesired fall in temperature will be counter-acted by the system infusing more heat into the room; similarly, any undesired rise in temperature will be met by a reduction in added heat, or in some systems the activation of an "air conditioner" to remove heat from the room. In any case, the control system reacts *in the opposite direction* to the undesired change.

2.2 Terms and definitions

Industrial measurement and control systems have their own unique terms and standards, which is the primary focus of this lesson. Here are some common instrumentation terms and their definitions:

Process – The physical system we are attempting to control or measure. *Examples: water filtration system, molten metal casting system, steam boiler, oil refinery unit, power generation unit.*

Process Variable, or **PV** – The specific quantity we are measuring in a process. *Examples: pressure, level, temperature, flow, electrical conductivity, pH, position, speed, vibration.*

Setpoint, or **SP** – The value at which we desire the process variable to be maintained at. In other words, the "target" value for the process variable.

Primary Sensing Element, or **PSE** – A device directly sensing the process variable and translating that sensed quantity into an analog representation (electrical voltage, current, resistance; mechanical force, motion, etc.). *Examples: thermocouple, thermistor, bourdon tube, microphone, potentiometer, electrochemical cell, accelerometer.*

Transducer – A device converting one standardized instrumentation signal into another standardized instrumentation signal, and/or performing some sort of processing on that signal. Often referred to as a *converter* and sometimes as a "relay." *Examples: I/P converter (converts 4-20 mA electric signal into 3-15 PSI pneumatic signal), P/I converter (converts 3-15 PSI pneumatic signal into 4-20 mA electric signal), square-root extractor (calculates the square root of the input signal).*

Note: in general science parlance, a "transducer" is any device converting one form of energy into another, such as a microphone or a thermocouple. In industrial instrumentation, however, we generally use "primary sensing element" to describe this concept and reserve the word "transducer" to specifically refer to a conversion device for standardized instrumentation signals.

Transmitter – A device translating the signal produced by a primary sensing element (PSE) into a *standardized* instrumentation signal such as 3-15 PSI air pressure, 4-20 mA DC electric current, Fieldbus digital signal packet, etc., which may then be conveyed to an indicating device, a controlling device, or both.

Lower- and Upper-range values, abbreviated **LRV** and **URV**, respectively – the values of process measurement deemed to be 0% and 100% of a transmitter's calibrated range. For example, if a temperature transmitter is calibrated to measure a range of temperature starting at 300 degrees Celsius and ending at 500 degrees Celsius, its LRV would be 300 ^{o}C and its URV would be 500 ^{o}C.

Zero and **Span** – alternative descriptions to LRV and URV for the 0% and 100% points of an instrument's calibrated range. "Zero" refers to the beginning-point of an instrument's range (equivalent to LRV), while "span" refers to the width of its range (URV − LRV). For example, if a temperature transmitter is calibrated to measure a range of temperature starting at 300 degrees Celsius and ending at 500 degrees Celsius, its zero would be 300 ^{o}C and its span would be 200 ^{o}C.

Controller – A device receiving a process variable (PV) signal from a primary sensing element

(PSE) or transmitter, comparing that signal to the desired value (called the setpoint) for that process variable, and calculating an appropriate output signal value to be sent to a final control element (FCE) such as an electric motor or control valve.

Final Control Element, or **FCE** – A device receiving the signal output by a controller to directly influence the process. *Examples: variable-speed electric motor, control valve, electric heater.*

Manipulated Variable, or **MV** – The quantity in a process we adjust or otherwise manipulate in order to influence the process variable (PV). Also used to describe the output signal generated by a controller; i.e. the signal commanding ("manipulating") the final control element to influence the process.

Load – any uncontrolled factor affecting the process variable's value. *Example: a window opening in a room letting warm air out and cold air in, affecting that room's temperature in such a way that that thermostatic heating system must compensate to maintain the room's temperature at setpoint.*

Automatic mode – When the controller generates an output signal based on the relationship of process variable (PV) to the setpoint (SP).

Manual mode – When the controller's decision-making ability is bypassed to let a human operator directly determine the output signal sent to the final control element.

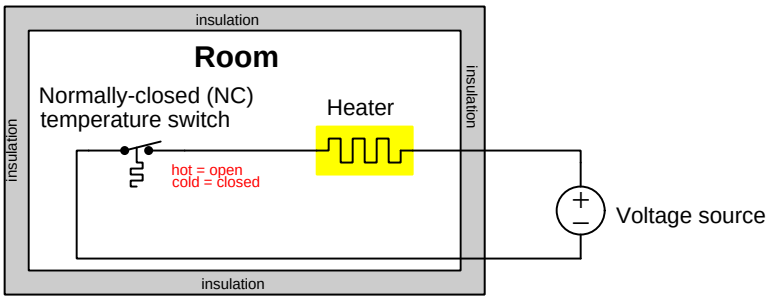
Loop – This widely-used term unfortunately has multiple meanings. In one sense it refers to the complete electrical circuit comprising a 4-20 mA analog measurement or control signal. In another sense it refers to the circular flow of information in any negative feedback regulation system.

2.3 Control system examples

Now we will explore some practical examples of measurement and control systems to give concrete application to these fundamental concepts. Each of the following example systems are quite commonplace in industry, using standard terminology, signal types, and each of them employing a control strategy of *negative feedback* to regulate some process variable at or near a desired *setpoint* value.

2.3.1 Example: thermostatic room control system

Any control system tasked with regulating the temperature of a room at some value comfortable for human and/or animal occupants is generally known as a *thermostatic control system*, "thermo" referring to *temperature* and "static" referring to the control system's task to maintain that room temperature at some steady (static, or unchanging) value:



A simple temperature switch located inside this room senses air temperature, closing the circuit and energizing the electrical heater if the room becomes too cold, and opening the circuit and de-energizing the heater if the room becomes too hot. The electric circuit in this room is another example of a *feedback* system, where its action feeds back on itself: the electric heater doesn't just add thermal energy to the room – it also affects the temperature switch which in turn affects the heater, and so on in a causal loop[1]. Specifically, the thermostatic system shown above is a *negative feedback* system because it counter-acts any thermal "nudge" externally imposed on the room: if cold weather forces the room to cool down, the switch closes to warm it back up. This is analogous to a classic pendulum, whose own weight acts to restore its vertical orientation if ever some external force "nudges" it away from vertical.

An interesting thought experiment is to replace the normally-closed (NC) temperature switch with one that is normally-open (NO). Now, the system exhibits *positive feedback*: if the room cools down, the switch remains open and the heater never energizes; if the room heats up for any reason, the switch closes which energizes the heater and makes the room heat up even more. This is now the thermal equivalent of an inverted pendulum, "falling over" in the direction of the nudge.

Another interesting thought experiment is to return to the normally-closed (NC) temperature switch scenario where we have negative feedback (self-regulation) again, then place some ice near the temperature-sensing switch. The ice, of course, will decrease temperature in the vicinity of the switch, causing the switch to "think" the room is cold. This makes the switch contact close to energize the heater, which in turn adds heat to the room, the end-result being a warmer room. The reverse is true if we place a localized heat source near the temperature switch: the room will cool because the switch is "fooled" into "thinking" the room is warmer than it really is.

[1] If the relationship between a single *cause* and its *effect* may be thought of as an arrow pointing from cause to effect, the relationship between cause and effect in a feedback system may be thought of as a set of curved arrows arranged tip-to-tail, with the last tip pointed toward the first tail to form a loop. In other words, multiple elements in a feedback system play simultaneous roles as causes and effects to each other. Here, the switch causes the heater to operate, and also the heater causes the switch to operate. Causal loops are the essence of all feedback systems, and they are what makes feedback so interesting.

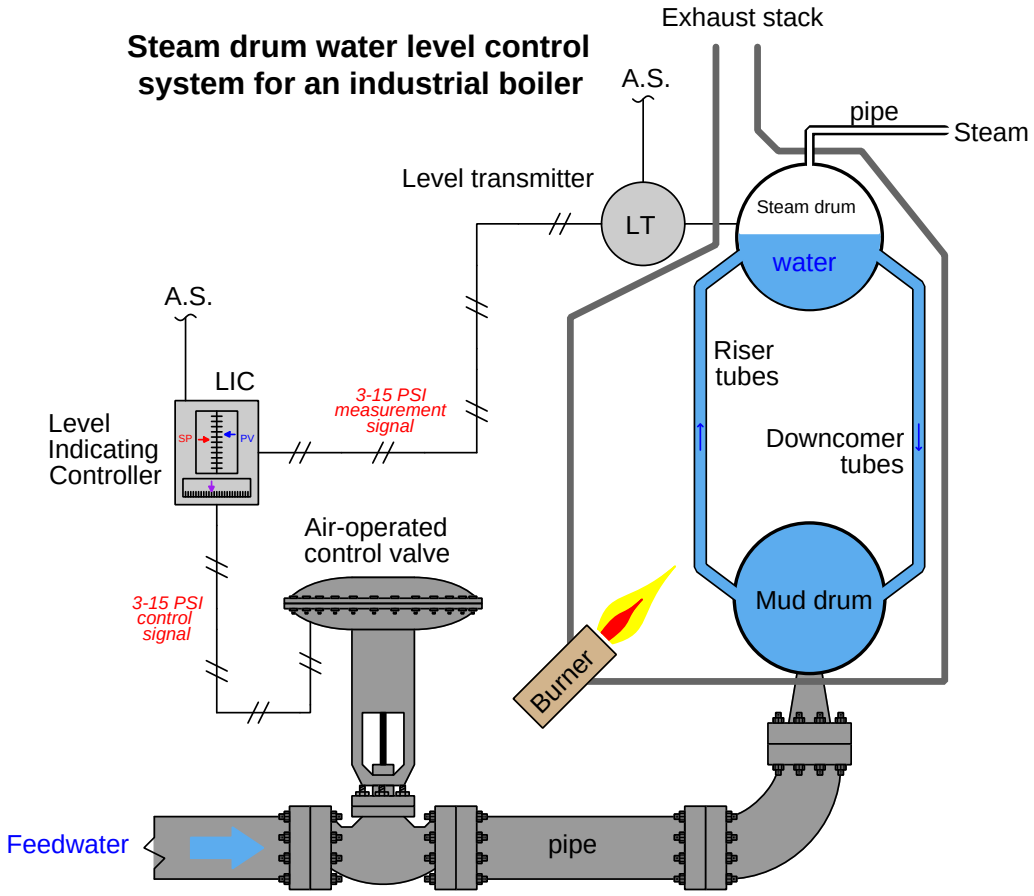
As you can see, feedback adds interesting behavioral characteristics to systems. Another lesson from these examples is the power of *thought experiments* as an analytical tool.

2.3.2 Example: boiler water level control system

Steam boilers are very common in industry, principally because steam power is so useful. Common uses for steam in industry include doing mechanical work (e.g. a steam engine moving some sort of machine), heating, producing vacuums (through the use of "steam ejectors"), and augmenting chemical processes (e.g. reforming of natural gas into hydrogen and carbon dioxide).

The process of converting water into steam is quite simple: heat up the water until it boils. Anyone who has ever boiled a pot of water for cooking knows how this process works. Making steam continuously, however, is a little more complicated. An important variable to measure and control in a continuous boiler is the level of water in the "steam drum" (the upper vessel in a water-tube boiler). In order to safely and efficiently produce a continuous flow of steam, we must ensure the steam drum never runs too low on water, or too high. If there is not enough water in the drum, the water tubes may run dry and burn through from the heat of the fire. If there is too much water in the drum, liquid water may be carried along with the flow of steam, causing problems downstream.

In this next illustration, you can see the essential elements of a water level control system, showing transmitter, controller, and control valve:



The first instrument in this control system is the *level transmitter*, or "LT". The purpose of this device is to sense the water level in the steam drum and report ("transmit") that measurement to the controller in the form of a signal. In this case, the type of signal is *pneumatic*: a variable air pressure sent through metal or plastic tubes. The greater the water level in the drum, the more air pressure output by the level transmitter. Since the transmitter is pneumatic, it must be supplied with a source of clean, compressed air on which to operate. This is the meaning of the "A.S." tube (Air Supply) entering the top of the transmitter.

This pneumatic signal is sent to the next instrument in the control system, the *level indicating controller*, or "LIC". The purpose of this instrument is to compare the level transmitter's signal against a *setpoint* value entered by a human operator representing the desired water level in the steam drum. The controller then generates an *output* signal telling the control valve to either introduce more or less water into the boiler to maintain the steam drum water level at setpoint. As with the transmitter, the controller in this system is pneumatic, operating entirely on compressed

air. This means the output of the controller is also a variable air pressure signal, just like the signal output by the level transmitter. Naturally, the controller requires a constant supply of clean, compressed air on which to run, which explains the "A.S." (Air Supply) tube connecting to it.

The last instrument in this control system is the control valve, operated directly by the air pressure signal output by the controller. Its purpose is to influence the flow rate of water into the boiler, "throttling" the water flow more or less as determined by controller. This particular type of control valve uses a large diaphragm and a large spring to move the valve further open with more signal pressure and further closed with less signal pressure.

When the controller is placed in the "automatic" mode, it will move the control valve to whatever position necessary to maintain a constant steam drum water level. The phrase "whatever position necessary" suggests the relationship between the controller output signal, the process variable signal (PV), and the setpoint (SP) is complex. If the controller senses a water level above setpoint, it will close off the valve as far as necessary to decrease the water level down to setpoint. Conversely, if the controller senses a water level below setpoint, it will open up the valve as far as necessary to raise the water level up to setpoint. This is negative feedback in action.

What this means in a practical sense is that the controller's output signal (equating to valve position) in automatic mode is just as much a function of process load (i.e. how much steam is being used from the boiler) as it is a function of setpoint (i.e. where we wish the water level to be). Consider a situation where the steam demand from the boiler is very low. If there isn't much steam being drawn off the boiler, this means there will be little water boiled into steam and therefore little need for additional feedwater to be pumped into the boiler. Therefore, in this situation, one would expect the control valve to hover near the fully-closed position, allowing just enough water into the boiler to keep the steam drum water level at setpoint. If, however, there is a high demand for steam from this boiler, the rate of evaporation will be much greater. This means the control system must add feedwater to the boiler at a much greater flow rate in order to maintain the steam drum water level at setpoint. In this situation we would expect to see the control valve much closer to being fully-open as the control system "works harder" to maintain a constant water level in the steam drum. Thus, we see how the controller automatically positions the control valve to react to different boiler operating conditions even when the setpoint is fixed.

A human operator supervising this boiler has the option of placing the controller into "manual" mode. In this mode the control valve position is under direct control of the human operator, with the controller essentially ignoring the signal sent from the water level transmitter. Being an indicating controller, the controller faceplate will still show how much water is in the steam drum, but it is now the human operator's sole responsibility to move the control valve to the appropriate position to hold water level at setpoint – *in manual mode the controller takes no corrective action of its own.* Manual mode is useful to human operators during start-up and shut-down conditions. It is also useful to instrument technicians for troubleshooting misbehaving control systems. Placing a controller into manual mode is akin to disengaging the cruise control in an automobile, transferring control of engine power from the car's computer back to the human driver. One can easily imagine an automobile mechanic needing to throttle a car's engine "manually" (i.e. with the cruise control turned off) in order to properly diagnose an engine or drivetrain problem. This is true for industrial processes as well, where instrument technicians may need to place a controller into manual mode in order to properly diagnose transmitter or control valve problems.

As was mentioned before, this is an example of a *pneumatic* (compressed air) control system, where all the instruments operate on compressed air, and use compressed air as the signaling medium. Pneumatic instrumentation is an old technology, dating back to the early twentieth century. While most modern instruments are electronic in nature, pneumatic instruments still find application within industry. The most common industry standard for pneumatic pressure signals is 3 to 15 PSI, with 3 PSI representing low end-of-scale and 15 PSI representing high end-of-scale. Alternative pressure ranges for pneumatic signals sometimes encountered in industry include 3 to 27 PSI, and 6 to 30 PSI. The following table shows the relationship between air signal pressure and steam drum level for this boiler's 3-15 PSI level transmitter:

Transmitter air signal pressure	**Steam drum water level**
3 PSI	0% (Empty)
6 PSI	25%
9 PSI	50%
12 PSI	75%
15 PSI	100% (Full)

It should be noted this table assumes the transmitter measures the *full range* of water level possible in the drum. Usually, this is not the case. Instead, the transmitter will be calibrated so it only senses a narrow range of water level near the middle of the drum. Thus, 3 PSI (0%) will not represent an empty drum, and neither will 15 PSI (100%) represent a completely full drum. Calibrating the transmitter like this helps avoid the possibility of actually running the drum completely empty or completely full in the case of an operator incorrectly setting the setpoint value near either extreme end of the measurement scale.

An example table showing this kind of realistic transmitter calibration appears here:

Transmitter air signal pressure	**Actual steam drum water level**
3 PSI	40%
6 PSI	45%
9 PSI	50%
12 PSI	55%
15 PSI	60%

The boiler's steam drum level controller outputs a pneumatic output signal to the control valve, using the same 3 to 15 PSI standard to command different valve positions:

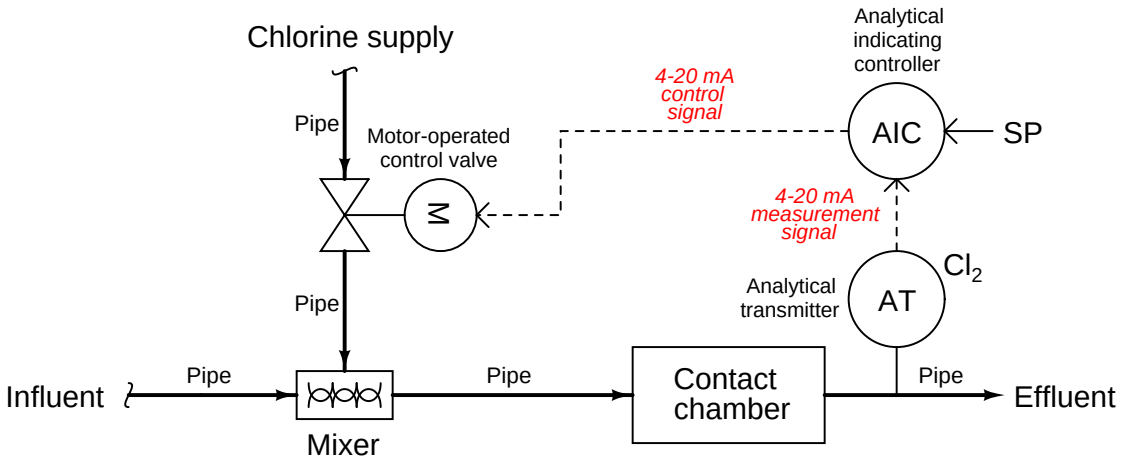
Controller output signal pressure	**Control valve position**
3 PSI	0% open (Fully shut)
6 PSI	25% open
9 PSI	50% open
12 PSI	75% open
15 PSI	100% (Fully open)

Even though the same range of air pressure (3 to 15 PSI) is used to represent water level in the steam drum *and* the position of the control valve, there is no simple correspondence between the two signals. A common misconception for students new to this topic is to assume the transmitter signal (PV) and controller output signal must be identical. *This is not true.* Typically the 3-15 PSI signal representing level will be at some value different from the 3-15 PSI signal driving the valve, because those two signals represent two entirely different variables in the boiler system. As we have seen previously, the output signal from a controller in automatic mode is just as much a function of process conditions as it is a function of the measured variable. This error is akin to thinking the road speed signal in an automobile cruise control system (the "process variable" or PV) must be the same value as the signal sent by the cruise control computer to the engine's accelerator control (the controller's "output" signal). Granted, these two signals are *related* to one another, but since they represent two different parameters in the controlled system we have no reason to expect their values will be *equal* except by chance.

2.3.3 Example: wastewater disinfection

The final step in treating wastewater before releasing it into the natural environment is to kill any harmful microorganisms (e.g. viruses, bacteria) in it. This is called *disinfection*, and chlorine gas is a very effective disinfecting agent. However, just as it is not good to mix too little chlorine in the outgoing water (effluent) because we might not disinfect the water thoroughly enough, there is also danger of injecting too much chlorine in the effluent because then we might begin poisoning animals and beneficial microorganisms in the natural environment.

To ensure the right amount of chlorine injection, we must use a dissolved chlorine analyzer to measure the chlorine concentration in the effluent, and use a controller to automatically adjust the chlorine control valve to inject the right amount of chlorine at all times. The following P&ID (Process and Instrument Diagram) shows how such a control system might look:



Chlorine gas coming through the control valve mixes with the incoming water (influent), then has time to disinfect in the contact chamber before exiting out to the environment.

The transmitter is labeled "AT" (Analytical Transmitter) because its function is to *analyze* the concentration of chlorine dissolved in the water and *transmit* this information to the control system. The "Cl_2" (chemical notation for a chlorine molecule) written near the transmitter bubble declares this to be a chlorine analyzer. The dashed line coming out of the transmitter tells us the signal is electric in nature, not pneumatic as was the case in the previous (boiler control system) example. The most common and likely standard for electronic signaling in industry is 4 to 20 milliamps DC, which represents chlorine concentration in much the same way as the 3 to 15 PSI pneumatic signal standard represented steam drum water level in the boiler:

Transmitter signal current	Chlorine concentration
4 mA	0% (no chlorine)
8 mA	25%
12 mA	50%
16 mA	75%
20 mA	100% (Full concentration)

The controller is labeled "AIC" because it is an Analytical Indicating Controller. Controllers are always designated by the process variable they are charged with controlling, in this case the chlorine analysis of the effluent. "Indicating" means there is some form of display that a human operator or technician can read showing the chlorine concentration. "SP" refers to the setpoint value entered by the operator, which the controller tries to maintain by adjusting the position of the chlorine injection valve.

A dashed line going from the controller to the valve indicates another electronic signal: a 4 to 20 mA direct current signal again. Just as with the 3 to 15 PSI pneumatic signal standard in the pneumatic boiler control system, the amount of electric current in this signal path directly relates to a certain valve position:

Controller output signal current	**Control valve position**
4 mA	0% open (Fully shut)
8 mA	25% open
12 mA	50% open
16 mA	75% open
20 mA	100% (Fully open)

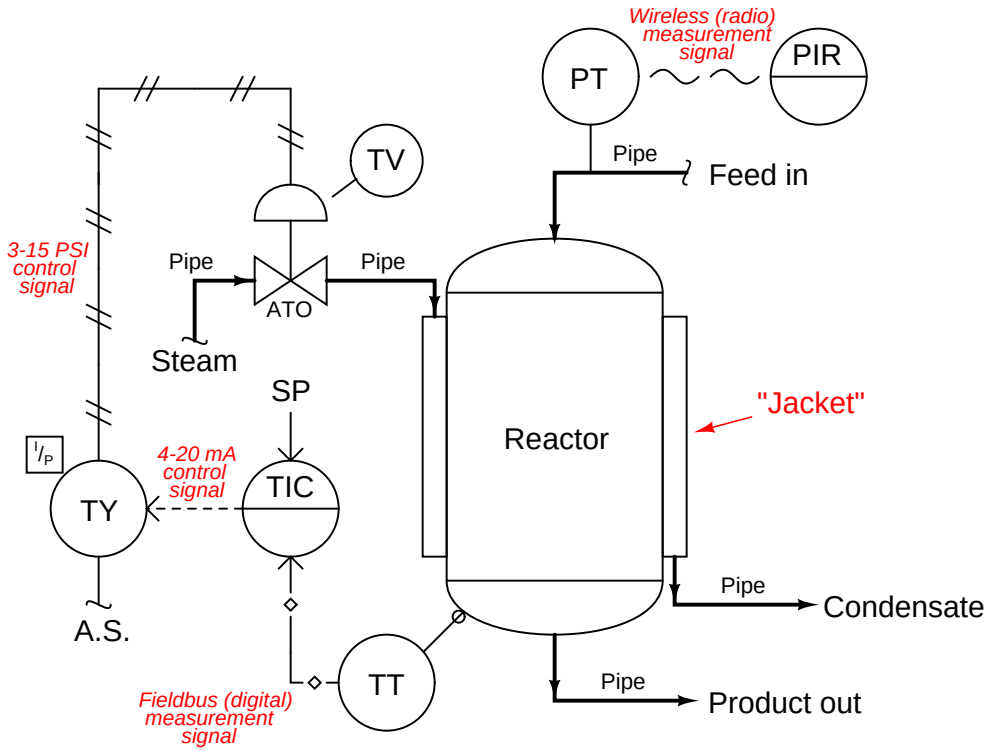
Note: it is possible, and in some cases even preferable, to have either a transmitter or a control valve that responds in reverse fashion to an instrument signal such as 3 to 15 PSI or 4 to 20 milliamps. For example, this valve could have been set up to be wide open at 4 mA and fully shut at 20 mA. The main point to recognize here is that both the process variable sensed by the transmitter and the position of the control valve are proportionately represented by analog signals.

Just as with the 3 to 15 PSI pneumatic signals used to represent water level and control valve position in the boiler seen previously, the two 4 to 20 milliamp current signals in this system represent two different variables in the system and therefore will not be equal to each other except by coincidence. A common misconception for people first learning about analog instrumentation signals is to assume the transmitter's signal ("Process Variable") must be identical in value to the control valve's signal ("Manipulated Variable" or "Output"), but this is not true.

The letter "M" inside the control valve bubble tells us this is a motor-actuated valve. Instead of using compressed air pushing against a spring-loaded diaphragm as was the case in the boiler control system, this valve is actuated by an electric motor turning a gear-reduction mechanism. The gear reduction mechanism allows slow motion of the control valve stem even though the motor spins at a fast rate. A special electronic control circuit inside the valve actuator modulates electric power to the electric motor in order to ensure the valve position accurately matches the signal sent by the controller. In effect, this is another control system in itself, controlling valve position according to a "setpoint" signal sent by another device (in this case, the AIT controller which is telling the valve what position to go to).

2.3.4 Example: chemical reactor temperature control

Sometimes we encounter a diversity of instrument signal standards in one control system. Such is the case with the following chemical reactor temperature control system, where three different signal standards convey information between the instruments. A P&ID (Process and Instrument Diagram) shows the inter-relationships of the process piping, vessels, and instruments:



The purpose of this control system is to ensure the chemical solution inside the reactor vessel is maintained at a constant temperature. A steam-heated "jacket" envelops the reactor vessel, transferring heat from the steam into the chemical solution inside. The control system maintains a constant temperature by measuring the temperature of the reactor vessel, and throttling steam from a boiler to the steam jacket to add more or less heat as needed.

We begin as usual with the temperature transmitter, located near the bottom of the vessel. Note the different line type used to connect the temperature transmitter (TT) with the temperature-indicating controller (TIC): hollow diamonds with lines in between. This signifies a *digital electronic instrument signal* – sometimes referred to as a *fieldbus* – rather than an analog type (such as 4 to 20 mA or 3 to 15 PSI). The transmitter in this system is actually a digital computer, and so is the controller. The transmitter reports the process variable (reactor temperature) to the controller using digital bits of information. Here there is no analog scale of 4 to 20 milliamps, but rather electric voltage/current pulses representing the 0 and 1 states of binary data.

Digital instrument signals are capable of transferring multiple data points rather than single

data points as is the case with analog instrument signals. This means digital instrument signals may convey device status information (such as self-diagnostic test results) as well as the basic measurement value. In other words, the digital signal coming from this transmitter not only tells the controller how hot the reactor is, but it may also communicate to the controller how well the transmitter is functioning.

The dashed line exiting the controller shows it to be analog electronic: most likely 4 to 20 milliamps DC. This electronic signal does not go directly to the control valve, however. It passes through a device labeled "TY", which is a *transducer* to convert the 4 to 20 mA electronic signal into a 3 to 15 PSI pneumatic signal which then actuates the valve. In essence, this signal transducer acts as an electrically-controlled air pressure regulator, taking the supply air pressure (usually 20 to 25 PSI) and regulating it down to a level commanded by the controller's electronic output signal.

At the temperature control valve (TV) the 3 to 15 PSI pneumatic pressure signal applies a force on a diaphragm to move the valve mechanism against the restraining force of a large spring. The construction and operation of this valve is the same as for the feedwater valve in the pneumatic boiler water control system. The letters "ATO" immediately below the valve symbol mean "Air-To-Open," referring to the direction this valve mechanism will move (wider open) as more air signal pressure is applied to its actuator.

A detail not shown on this diagram, yet critically important to the operation of the temperature control system, is the *direction of action* for the controller while in automatic mode. It is possible to configure general-purpose controllers to act either in a *direct* fashion where an increasing process variable signal automatically results in an increasing output signal, or in a *reverse* fashion where an increasing process variable signal automatically results in a decreasing output signal. An effective way to identify the proper direction of action for any process controller is to perform a "thought experiment" whereby we imagine the process variable increasing over time, and then determine which way the controller's output needs to change in order to bring the process variable value back to setpoint based on the final control element's influence within the process.

In this process, let us imagine the reactor temperature increasing for some reason, perhaps an increase in the temperature of the feed entering the reactor. With an increasing temperature, the controller must *reduce* the amount of steam applied to the heating jacket surrounding the reactor in order to correct for this temperature change. With an air-to-open (ATO) steam valve, this requires a *decreased* air pressure signal to the valve in order to close it further and reduce heat input to the reactor. Thus, if an increasing process variable signal requires a decreasing controller output signal, the controller in this case needs to be configured for *reverse* action.

We could easily imagine reasons why the temperature controller in this process might have to be configured for direct action instead of reverse action. If the piping were altered such that the control valve throttled the flow of *coolant* to the reactor rather than steam, an increasing temperature would require a further-open valve, which would only happen if the controller were configured for direct action. Alternatively, if the steam valve were air-to-close (ATC) rather than air-to-open (ATO), an increasing reactor temperature (requiring less steam be sent to the reactor) would necessitate the controller outputting an increased signal to the valve, so that more air signal pressure pushed the valve further closed.

An example of a chemical reaction temperature control system requiring *direct* controller action is shown in the following photograph. Here, we see a jacketed stainless-steel vessel used to ferment beer at cold temperatures. The jacket surrounding this vessel is pumped full of chilled glycol solution (similar to automotive antifreeze), to draw heat away from the fermenting beer and maintain its temperature well below ambient:

If the beer becomes too warm, the controller sends an increased signal to the glycol valve sending more chilled glycol through the vessel's jacket to remove heat from the beer. Since the relationship between the controller's process variable and its output is direct (i.e. rising PV results in rising Output), the controller needs to be configured for direct action.

This is why general-purpose process controllers always provide a user-selectable option for either direct or reverse action: it makes them adaptable to the needs of *any* process, no matter the physics of the process or the behavior of the other loop instruments (e.g. transmitter and final control element).

An additional instrument connected to our hypothetical chemical reactor is a pressure transmitter (PT) on the feed line. While not a part of the temperature control loop, it is shown here to illustrate yet another type of instrumentation signaling: *digital wireless*. Here, the transmitter reports its measurement data to an indicator at the control room via radio signals, using digital codes much like fieldbus to communicate not only the basic process data but also transmitter diagnostic and radio network management data.

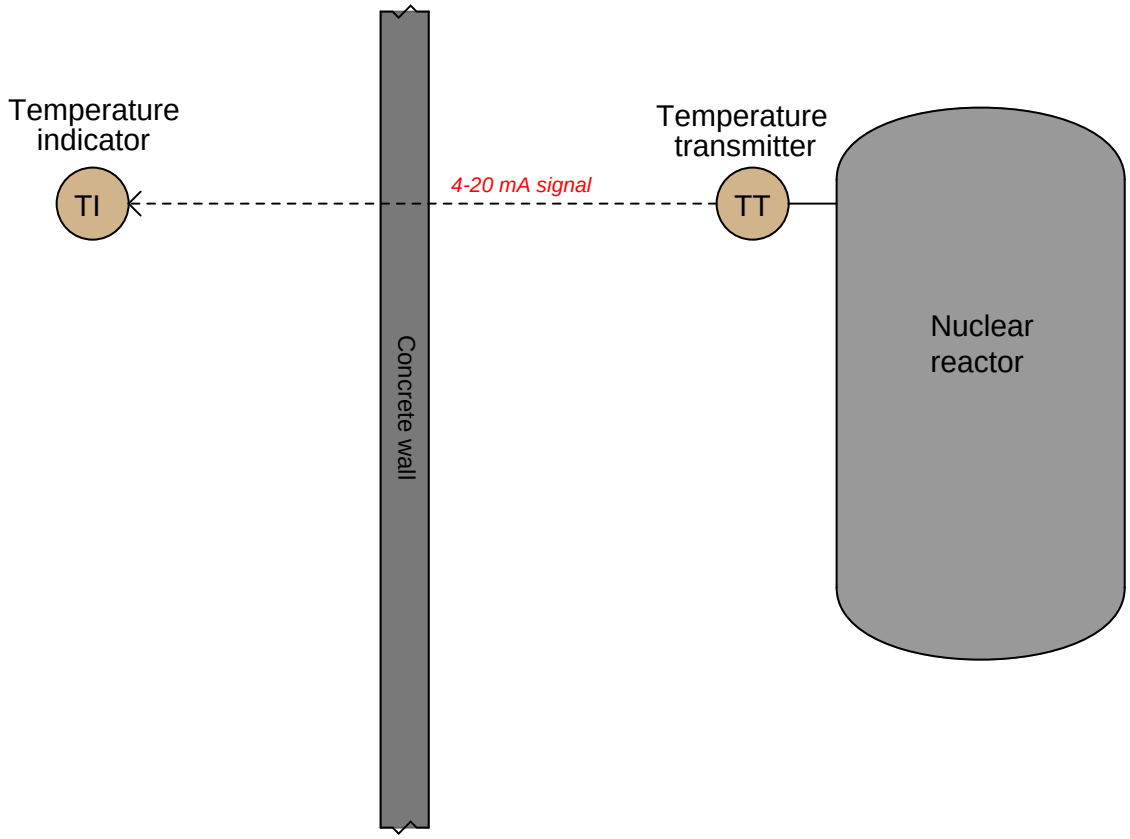
At the time of this writing (2011), wireless instrumentation is not recommended for mission-critical control applications, and finds its greatest use in low-priority monitoring instrumentation. The most obvious advantage of wireless instruments is that they do not require wires of any kind. Since wiring is a major capital cost when installing instruments, this fact makes wireless instrumentation relatively inexpensive to install. Freedom from wires also allows these instruments to be used in applications that would be impossible for wired instruments, such as communicating data from sensors installed in moving vehicles to stationary monitoring or control equipment. However, the elimination of wires means wireless instruments must provide for their own power requirements, usually with long-life batteries. Reliance on battery power alone places restrictions on how frequently these instrument perform their functions: less frequent data transmission results in longer battery life, but correspondingly reduces the instrument's practicality for real-time control. Potential blockage of the radio signals from moving objects such as large vehicles (cranes, lifts, etc.) also poses challenges to signal reliability. Despite these limitations, the total absence of signal or power wiring for a wireless instrument is an essential feature for certain applications. Wireless is just another tool to help us automate processes, and like any other tool it has its advantages and disadvantages.

2.4 Other types of instruments

So far we have just looked at instruments that sense, control, and influence process variables. Transmitters, controllers, and control valves are respective examples of each instrument type. However, other instruments exist to perform useful functions for us.

2.4.1 Indicators

One common "auxiliary" instrument is the *indicator*, the purpose of which is to provide a human-readable indication of an instrument signal. Quite often process transmitters are not equipped with readouts for whatever variable they measure: they just transmit a standard instrument signal (3 to 15 PSI, 4 to 20 mA, etc.) to another device. An indicator gives a human operator a convenient way of seeing what the output of the transmitter is without having to connect test equipment (pressure gauge for 3-15 PSI, ammeter for 4-20 mA) and perform conversion calculations. Moreover, indicators may be located far from their respective transmitters, providing readouts in locations more convenient than the location of the transmitter itself. An example where remote indication would be practical is shown here, in a nuclear reactor temperature measurement system:



It would be unsafe for human beings to approach the nuclear reactor when it is in full-power operation, due to the strong radiation flux it emits. The temperature transmitter is built to withstand the radiation, though, and it transmits a 4 to 20 milliamp electronic signal to an indicating recorder located on the other side of a thick concrete wall blocking the reactor's radiation, where it is safe for human occupancy. There is nothing preventing us from connecting multiple indicators, at multiple locations, to the same 4 to 20 milliamp signal wires coming from the temperature

transmitter. This allows us to display the reactor temperature in as many locations as we desire, since there is no absolute limit on how far we may conduct a DC milliamp signal along copper wires.

A numerical-plus-bargraph indicator appears in this next photograph, mounted in the face of a metal panel inside of a control room:

This particular indicator shows the position of a flow-control gate in a wastewater treatment facility, both by numerical value (98.06%) and by the height of a bargraph (very near full open – 100%). It is directly wired in series with the same 4-20 milliamp current signal sent to the gate actuator.

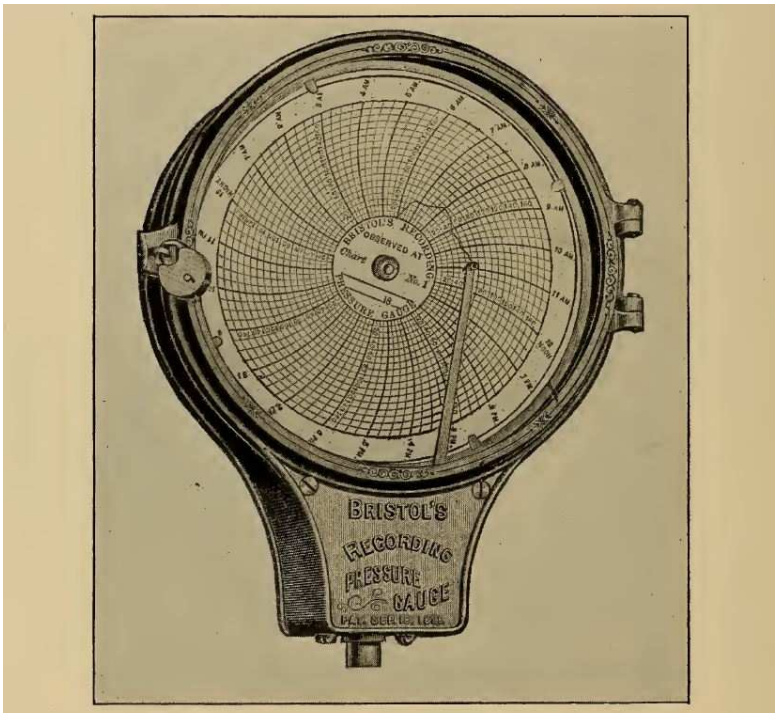
A less sophisticated style of panel-mounted indicator shows only a numeric display, such as this unit shown here:

Indicators may also be used in "field" (process) areas to provide direct indication of measured variables if the transmitter device lacks a human-readable indicator of its own. The following photograph shows a field-mounted indicator, operating directly from the electrical power available in the 4-20 mA loop. The numerical display of this indicator uses LCD technology rather than red-glowing LEDs, in order to use less electrical power:

2.4.2 Recorders

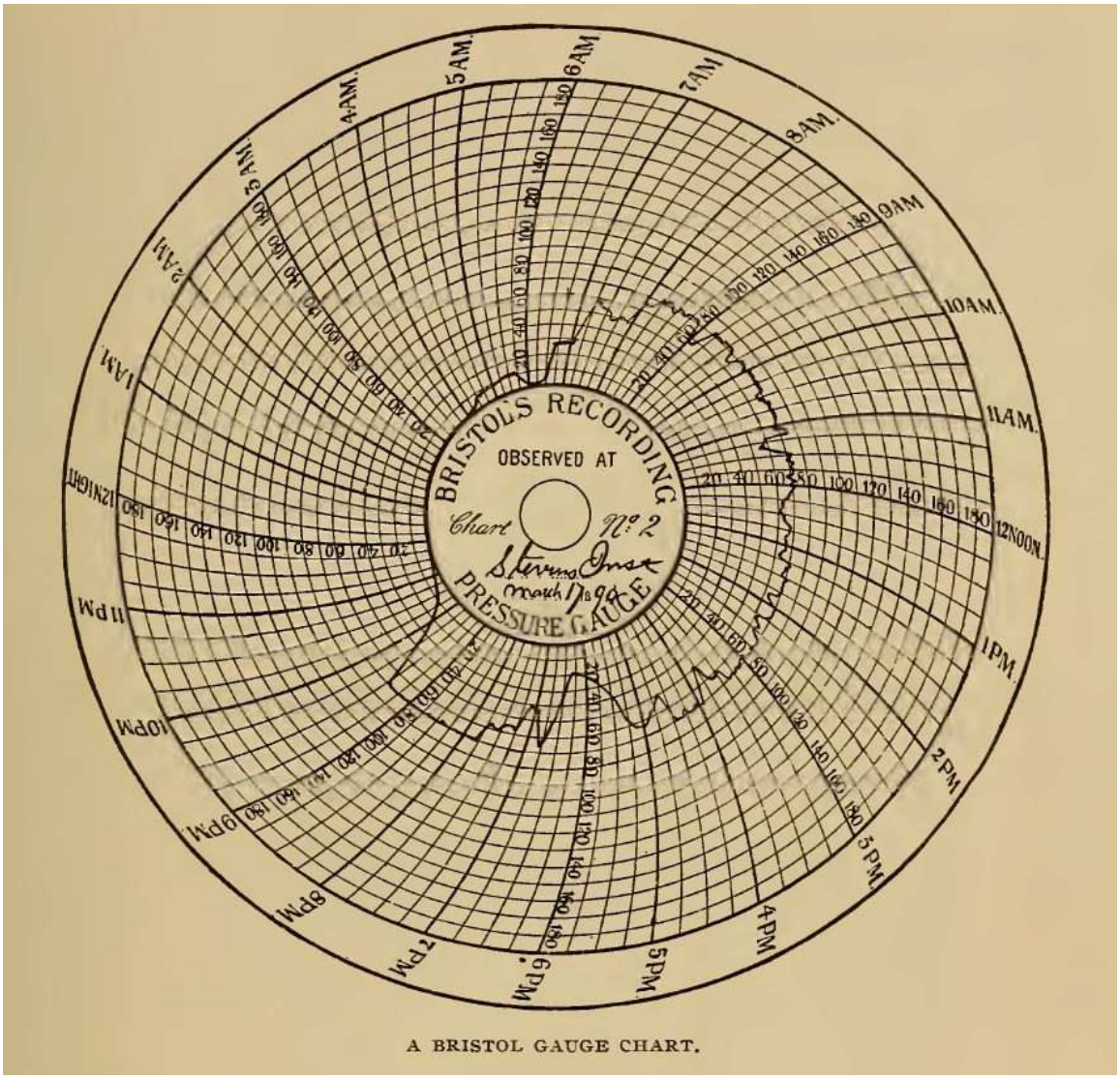
Another common "auxiliary" instrument is the *recorder* (sometimes specifically referred to as a *chart recorder* or a *trend recorder*), the purpose of which is to draw a graph of process variable(s) over time. Recorders usually have indications built into them for showing the instantaneous value of the instrument signal(s) simultaneously with the historical values, and for this reason are usually designated as *indicating* recorders. A temperature indicating recorder for the nuclear reactor system shown previously would be designated as a "TIR" accordingly.

Paper chart recorders are a form of instrumentation with a long history. The following image shows an illustration of a Bristol brand recording pressure gauge found on page 562 of *Cassier's Magazine* volume 8, published in 1895. Note the circular form of the paper chart, allowing the pen to draw a trace as the circular chart slowly spins. A padlock on the front glass cover prevents tampering with the chart recording:



RECORDING GAUGE MADE BY THE BRISTOL COMPANY, WATERBURY, CONN., U. S. A.

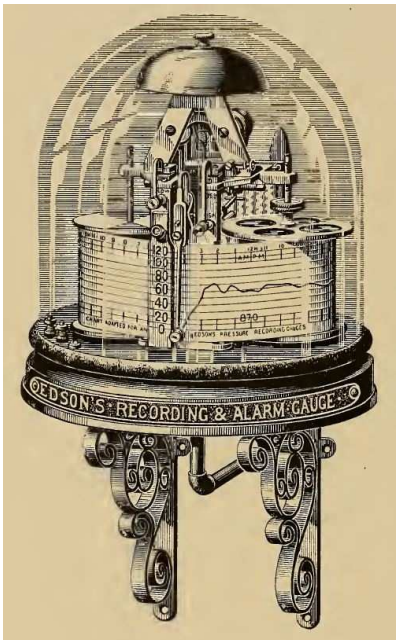
A typical chart from one of these recording devices is shown in this illustration, taken from page 563 of the same engineering periodical:



A BRISTOL GAUGE CHART.

This particular recording is of a steam boiler's pressure over a 24-hour period, showing pressure build-up beginning at 4:00 AM and boiler shut-down at 9:30 PM. This steam boiler's pressure appears to have been operated at approximately 70 PSI. Dips and peaks in the trace reflect changes in steam demand as well as irregularities in the firing of the boiler's coal furnace.

Another design of paper chart recorder is the *strip* style, using a long strip of paper between two spools (one to play out the paper and another to take it up). Like the circular chart recorder design, the strip chart recorder also has a long history. The following illustration from page 560 of the same *Cassier's Magazine* volume:



RECORDING GAUGE MADE BY J. B. EDSON, NEW YORK.

This next image shows a practical example of a strip chart's record for a city water supply company, taken from page 566 of the same periodical:

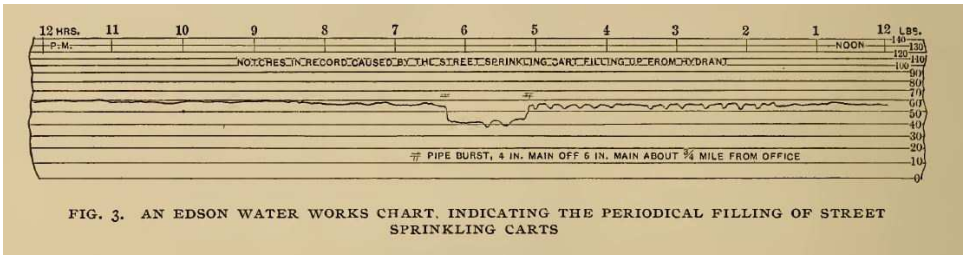


FIG. 3. AN EDSON WATER WORKS CHART, INDICATING THE PERIODICAL FILLING OF STREET SPRINKLING CARTS

Several interesting details[2] may be seen on this particular recording, including pressure fluctuations caused by periodic draws of water from a fire hydrant to fill street carts used to spray

[2] *Many* interesting points may be drawn from these two illustrations. Regarding the strip chart recording instrument itself, it is worth noting the ornate design of the metal frame (quite typical of machinery design from that era), the attractive glass dome used to shield the chart and mechanism from the environment, and the intricate mechanism used to drive the strip chart and move the pen. Unlike a circular chart, the length of a strip chart is limited only by the diameter of the paper roll, and may be made long enough to record many days' worth of pressure measurements. The label seen on the front of this instrument ("Edson's Recording and Alarm Gauge") tells us this instrument has the ability to alert a human operator of abnormal conditions, and a close inspection of the mechanism reveals a bell

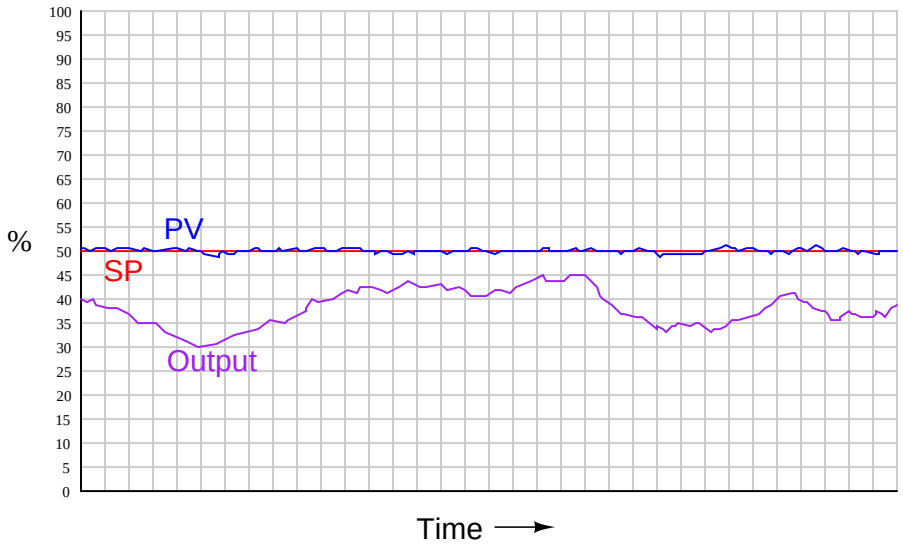
the city's dirt roads with water to minimize dust. Pressure drop caused by a burst 4-inch water pipe is also seen on this recording, between 5:00 PM and 6:15 PM.

Both circular and strip chart recorder designs survive to this day. Two circular chart recorders are shown in the following photograph, used to record temperatures at a powdered milk processing facility:

Two more chart recorders appear in the next photograph, a strip chart recorder on the right and a *paperless* chart recorder on the left. The strip chart recorder uses a scroll of paper drawn slowly past one or more lateral-moving pens, while the paperless recorder does away with paper entirely by plotting graphic trend lines on a computer screen:

on the top which presumably rings under alarm conditions. Regarding the strip chart record, note the "compressed" scale, whereby successive divisions of the vertical scale become closer in spacing, reflecting some inherent nonlinearity of the pressure-sensing mechanism.

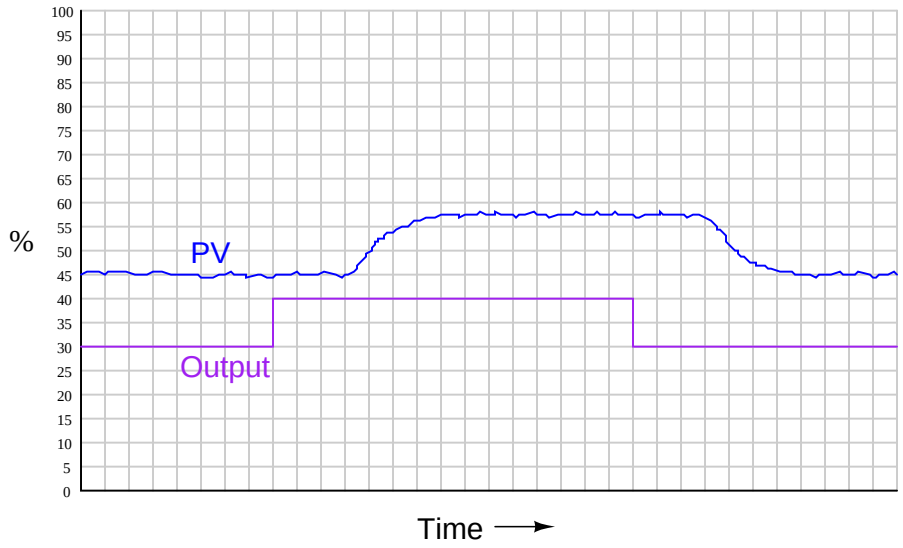
Recorders are extremely helpful for troubleshooting process control problems. This is especially true when the recorder is configured to record not just the process variable, but also the controller's setpoint and output variables as well. Here is an example of a typical "trend" showing the relationship between process variable, setpoint, and controller output in automatic mode, as graphed by a recorder:



Here, the setpoint (SP) appears as a perfectly straight red line, the process variable as a slightly erratic blue line, and the controller output as a moderately erratic purple line. We can see from this trend that the controller is doing exactly what it should: holding the process variable value close to setpoint, manipulating the final control element as far as necessary to do so. The chaotic appearance of the output signal is not really a problem, contrary to most peoples' first impression. The fact that the process variable never deviates significantly from the setpoint tells us the control system is operating quite well. What accounts for the erratic controller output, then? The answer to this question is any factor requiring the controller to change its output in order to maintain the process variable at setpoint. Variations in process load would account for this: as other variables in the process change over time, the controller is forced to compensate for these variations in order to ensure the process variable does not drift from setpoint. Referencing our previous example of a steam boiler water level control system, one of these influencing variables is steam demand. If this trend shows the steam drum water level (PV) and feedwater valve position (Output), variations in the controller's output signal could be indicative of steam demand rising and falling, the controller modulating water flow into the boiler to properly compensate for this load and maintain a fairly constant steam drum water level. A wildly-fluctuating output signal *may* indicate a problem somewhere else in the process (placing undue demands on the control system), but there is certainly no problem with the control system itself: it is doing its job perfectly well.

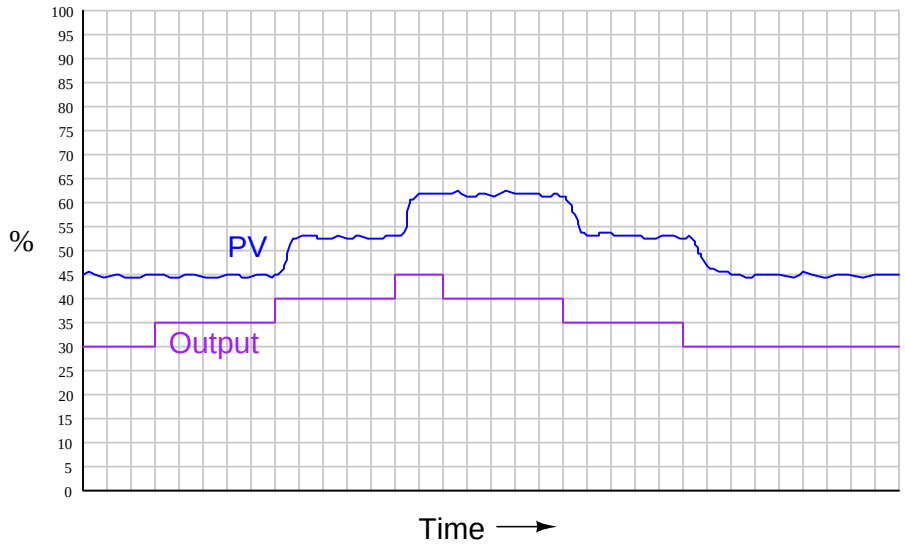
Recorders become powerful diagnostic tools when coupled with the controller's manual control mode. By placing a controller in "manual" mode and allowing direct human control over the final control element (valve, motor, heater), we can tell a lot about a process. Here is an example of a

trend recording for a process in manual mode, where the process variable response is seen graphed in relation to the controller output as that output is increased and decreased in steps:



Notice the time delay between when the output signal is "stepped" to a new value and when the process variable responds to the change. This delay is called *dead time*, and it is generally detrimental to control system performance. Imagine trying to steer an automobile whose front wheels respond to your input at the steering wheel only after a 5-second delay! This would be a very challenging car to drive. The same problem plagues any industrial control system with a time lag between the final control element and the transmitter. Typical causes of this problem include *transport delay* (where there is a physical delay resulting from transit time of a process medium from the point of control to the point of measurement) and mechanical problems in the final control element.

This next example shows another type of problem revealed by a trend recording during manual-mode testing:



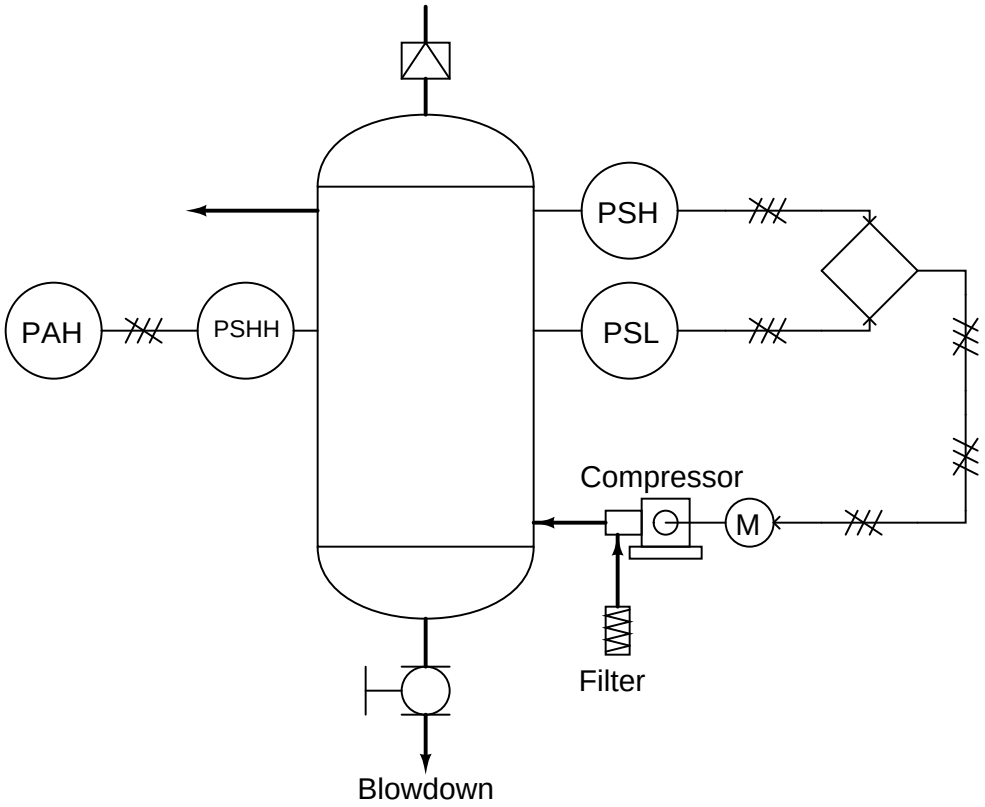
Here, we see the process quickly responding to all step-changes in controller output except for those involving a change in direction. This problem is usually caused by mechanical friction in the final control element (e.g. "sticky" valve stem packing in a pneumatically-actuated control valve), and is analogous to "loose" steering in an automobile, where the driver must turn the steering wheel a little bit extra after reversing steering direction. Anyone who has ever driven an old farm tractor knows what this phenomenon is like, and how it detrimentally affects one's ability to steer the tractor in a straight line.

Sometimes it becomes useful to temporarily place a recorder into an instrumentation system for diagnostic purposes. On the simplest level, this might consist of a digital multimeter (DMM) connected to measure signal voltage or current, with its "minimum/maximum" capture mode engaged. On a more complex level, this might be a personal computer running data graphing software, connected to the instrumentation circuit through a data acquisition (DAQ) module converting the circuit's analog voltage or current signals into digital values readable by the computer.

2.4.3 Process switches and alarms

Another type of instrument commonly seen in measurement and control systems is the *process switch*. The purpose of a switch is to turn on and off with varying process conditions. Usually, switches are used to activate alarms to alert human operators to take special action. In other situations, switches are directly used as control devices.

The following P&ID of a compressed air control system shows both uses of process switches:

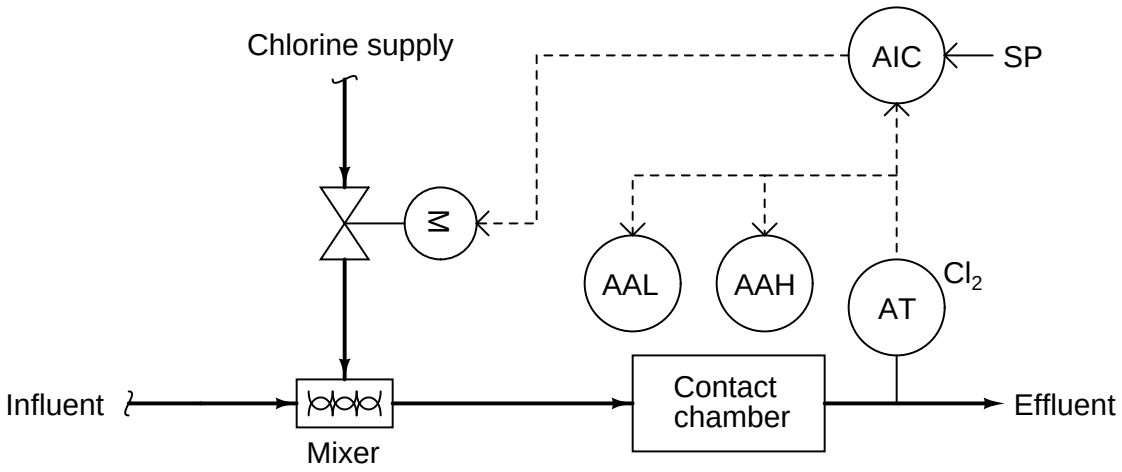


The "PSH" (*pressure switch, high*) activates when the air pressure inside the vessel reaches its high control point. The "PSL" (*pressure switch, low*) activates when the air pressure inside the vessel drops down to its low control point. Both switches feed discrete (on/off) electrical signals to a logic control device (symbolized by the diamond) which then controls the starting and stopping of the electric motor-driven air compressor.

Another switch in this system labeled "PSHH" (*pressure switch, high-high*) activates only if the air pressure inside the vessel exceeds a level beyond the high shut-off point of the high pressure control switch (PSH). If this switch activates, something has gone wrong with the compressor control system, and the high pressure alarm (PAH, or *pressure alarm, high*) activates to notify a human operator.

All three switches in this air compressor control system are directly actuated by the air pressure in the vessel: in other words, these are direct *process-sensing* switches. It is possible, however, to build switch devices that interpret standardized instrumentation signals such as 3 to 15 PSI (pneumatic) or 4 to 20 milliamps (analog electronic), allowing us to build on/off control systems and alarms for *any* type of process variable measurable with a transmitter.

For example, the chlorine wastewater disinfection system shown earlier may be equipped with a couple of electronic alarm switches to alert an operator if the chlorine concentration ever exceeds pre-determined high or low limits:



The labels "AAL" and "AAH" refer to *analytical alarm low* and *analytical alarm high*, respectively. Note how the diagram shows these two alarm units connected to the electronic (4-20 mA) signal output by the chlorine analyzer (AT). This tells us the AAL and AAH alarm units are really just electronic circuits, alarming if the analytical transmitter's 4-20 mA analog signal falls below (AAL) or exceeds (AAH) certain pre-set limits. As such, the AAL and AAH alarms do not *directly* sense the chlorine concentration in the water, but rather *indirectly* sense it by monitoring the chlorine analyzer's 4-20 milliamp output signal.

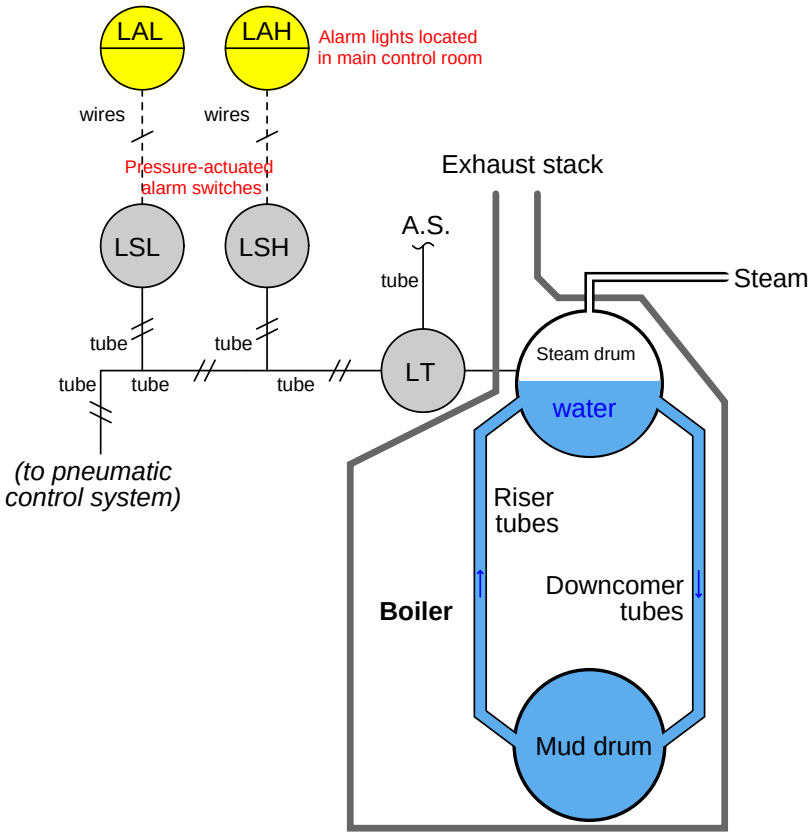
Since both alarms work off the 4 to 20 milliamp electronic signal output by the chlorine analytical transmitter (AT) rather than directly sensing the process, their construction is greatly simplified. If these were process-sensing switches, each one would have to be equipped with the analytical capability of directly sensing chlorine concentration in water. In other words, each switch would have to be its own self-contained chlorine concentration analyzer, with all the attendant complexity.

An example of an electronic alarm module (triggered by a 4-20 mA current signal coming from a transmitter) is the Moore Industries model SPA ("Site Programmable Alarm"), shown here:

In addition to providing alarm capability, this SPA module also provides a digital display (a small LCD screen) to show the analog signal value for operational or diagnostic purposes.

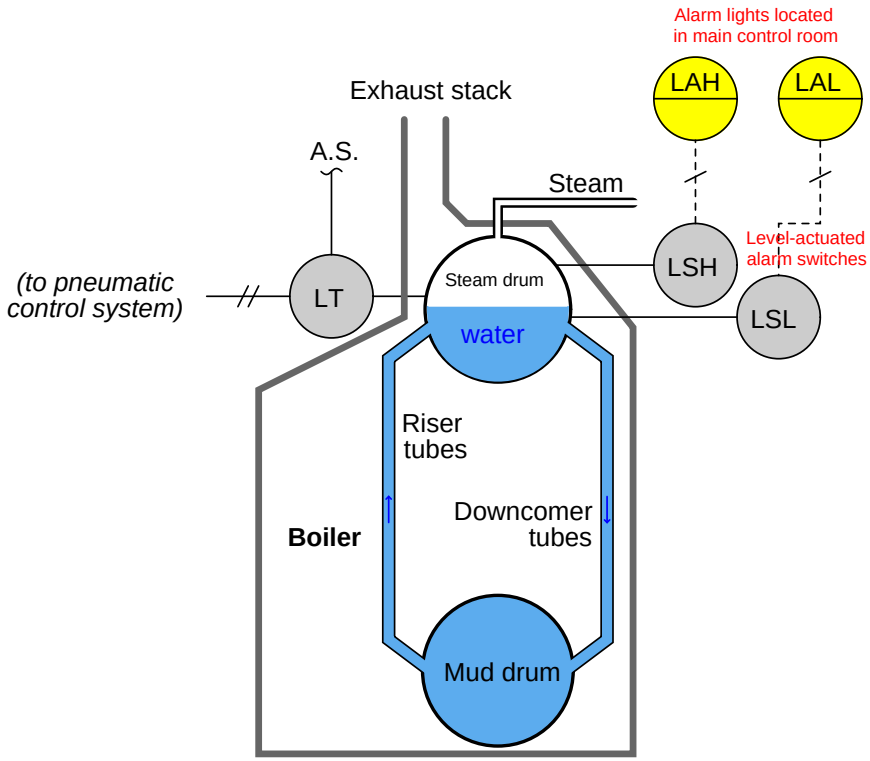
Like all current-operated alarm modules, the Moore Industries SPA may be configured to "trip" electrical contacts when the current signal reaches a variety of different programmed thresholds. Some of the alarm types provided by this unit include high process, low process, out-of-range, and high rate-of-change.

In a similar manner, we may add pressure-actuated process alarm switches to pneumatic (3-15 PSI) signal lines coming from pneumatic transmitters to add alarm capability to a system designed for continuous measurement. For example, if high- and low-alarm capability were desired for the steam drum water level process described earlier in this chapter, one could add a pair of pressure-actuated switches to the pneumatic level transmitter's 3-15 PSI output signal line:



These two pressure-actuated switches serve as *water level* alarms, because the air pressure signal actuating them comes from the pneumatic *level* transmitter, which outputs an air pressure signal in direct proportion to water level in the boiler's steam drum. Even though the physical stimulus actuating each switch is an air *pressure*, the switches still serve the purpose of liquid *level* alarm signaling because that air pressure is an analogue (representation) of water level in the steam drum. In other words, these two alarm switches (LSL and LSH) *indirectly* sense water level by monitoring the pneumatic signal pressure output by the level transmitter (LT).

The alternative to pressure-actuated water level alarm switches would be independent level-sensing switches attached directly to the steam drum, each switch equipped with its own means[3] of directly sensing water level:



It should be mentioned that the choice between using process alarm switches directly actuated by the process versus alarm switches actuated by a transmitter's analog signal is not arbitrary. In the system where the two alarm switches actuate from the transmitter's 3-15 PSI output signal, the integrity of the water level control *and* that of the high- and low-level alarms all depend on the proper function of one transmitter. If that one transmitter were to fail, all three system functions would be compromised. This elevates the importance of a single instrument, which is generally not desirable from the perspective of reliability and process safety. In the system where each level alarm switch independently senses steam drum water level, one device may fail without compromising either of the other two functions. This independence is desirable because it greatly reduces the probability of "common-cause" failures, where a single fault disables multiple system functions. The final determination should be based on a rigorous analysis of device versus system reliability, which is typically the task of a process engineer.

[3]These might be float-driven switches, where each switch is mechanically actuated by the buoyancy of a hollow metal float resting on the surface of the water. Another technology uses metal electrodes inserted into the water from above, sensing water level by electrical conductivity: when the water level reaches the probe's tip, an electrical circuit is closed. For more information on liquid level switches, refer to section **??** beginning on page **??**.

2.4.4 Annunciators

Process alarm switches may be used to trigger a special type of indicator device known as an *annunciator*. An annunciator is an array of indicator lights and associated circuitry designed to secure a human operator's attention[4] by blinking and sounding an audible buzzer when a process switch actuates into an abnormal state. The alarm state may be then "acknowledged" by an operator pushing a button, causing the alarm light to remain on (solid) rather than blink, and silencing the buzzer. The indicator light does not turn off until the actual alarm condition (the process switch) has returned to its regular state.

This photograph shows an annunciator located on a control panel for a large engine-driven pump. Each white plastic square with writing on it is a translucent pane covering a small light bulb. When an alarm condition occurs, the respective light bulb flashes, causing the translucent white plastic to glow, highlighting to the operator which alarm is active:

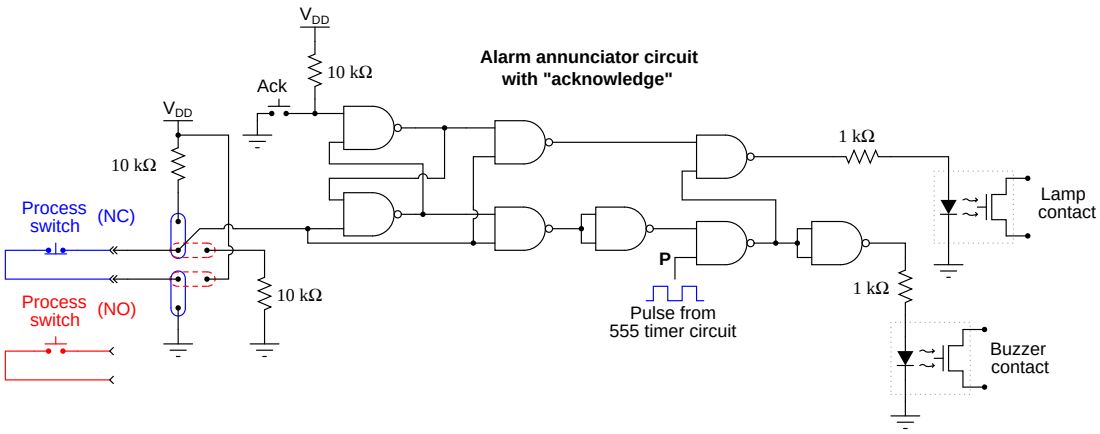
Note the two pushbutton switches below labeled "Test" and "Acknowledge." Pressing the "Acknowledge" button will silence the audible buzzer and also turn any blinking alarm light into a steady (solid) alarm light until the alarm condition clears, at which time the light turns off completely. Pressing the "Test" button turns all alarm lights on, to ensure all light bulbs are still functional.

[4] D.A. Strobhar, writing in *The Instrument Engineers' Handbook* on the subject of alarm management, keenly observes that alarms are the only form of instrument "whose sole purpose is to alter the operator's behavior." Other instrument devices work to control the process, but only alarms work to *control the human operator*.

Opening the front panel of this annunciator reveals modular relay units controlling the blinking and acknowledgment latch functions, one for each alarm light:

This modular design allows each alarm channel to be serviced without necessarily interrupting the function of the other channels in the annunciator panel.

A simple logic gate circuit illustrates the acknowledgment latching feature (here implemented by an S-R latch circuit) common to all process alarm annunciators:



Panel-mounted annunciators are becoming a thing of the past, as computer-based alarm displays replace them with advanced capabilities such as time logging, first-event[5] recording, and multiple layers of acknowledgment/access. Time logging is of particular importance in the process industries, as the sequence of events is often extremely important in investigations following an abnormal operating condition. Knowing *what happened, and exactly when it happened* is much more informative than simply knowing which alarms have tripped.

[5] When a complex machine or process with many shutdown sensors automatically shuts down, it may be difficult to discern after the fact *which* shutdown device was responsible. For instance, imagine an engine-powered generator automatically shutting down because one of the generator's "trip" sensors detected an under-voltage condition. Once the engine shuts down, though, *multiple* trip sensors will show abnormal conditions simply because the engine is not running anymore. The oil pressure sensor is one example of this: once the engine shuts down, there will no longer be any oil pressure, thus causing that alarm to activate. The under-voltage alarm falls into this category as well: once the engine shuts down, the generator will no longer be turning and therefore its output voltage *must* be zero. The problem for any human operator encountering the shut-down engine is that he or she cannot tell *which* of these alarms was the initiating cause of the shutdown versus *which* of these alarms simply activated after the fact once the engine shut off. An annunciator panel showing both an under-voltage and a low oil pressure light does not tell us which event happened first to shut down the generator. A "first-event" (sometimes called a "first-out") annunciator, however, shows which trip sensor was the *first* to activate, thus revealing the initiating cause of the event.

Chapter 3

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?

3.1 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.

3.1.1 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.

3.1.2 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.

Energy

Conservation of Energy

Simplification as a problem-solving strategy

Thought experiments as a problem-solving strategy

Limiting cases as a problem-solving strategy

Annotating diagrams as a problem-solving strategy

Interpreting intermediate results as a problem-solving strategy

Graphing as a problem-solving strategy

Converting a qualitative problem into a quantitative problem

Converting a quantitative problem into a qualitative problem

Working "backwards" to validate calculated results

Reductio ad absurdum

Re-drawing schematics as a problem-solving strategy

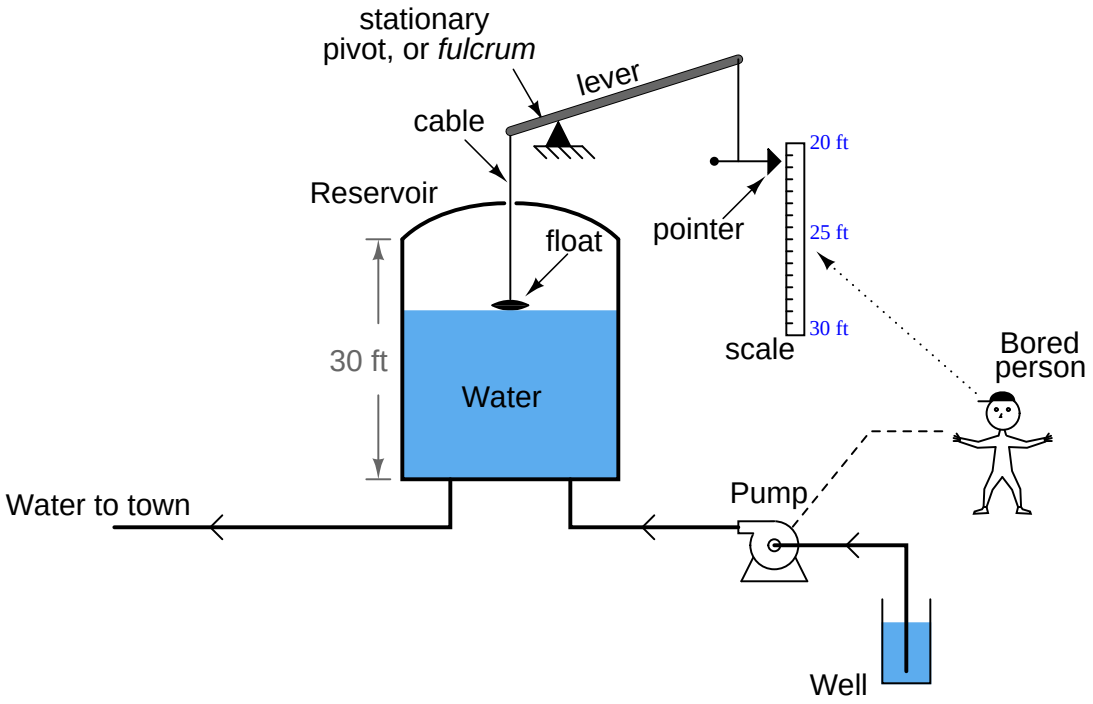
Cut-and-try problem-solving strategy

Algebraic substitution

???

3.1.3 Manual water reservoir control

A water reservoir located high on a hill stores fresh water for a town's drinking needs. A float connected to a lever provides visual indication of the water level inside the reservoir. Nearby this reservoir, a person has the most boring job in the world: to turn the pump on when the water level gets too low, and to turn the pump off when the water level gets too high. Note that the float mechanism showing water level in the reservoir cannot show the entire capacity of the reservoir, but only the top ten feet (from 20 feet to 30 feet of level):



As crude as it is, this system contains *instrumentation*, and we may apply standard instrumentation terms to its components. Apply the following terms to this water-supply system, as best as you can:

- Process
- Primary sensing element
- Final control element
- Measurement range
- Lower-Range Value (LRV)
- Upper-Range Value (URV)

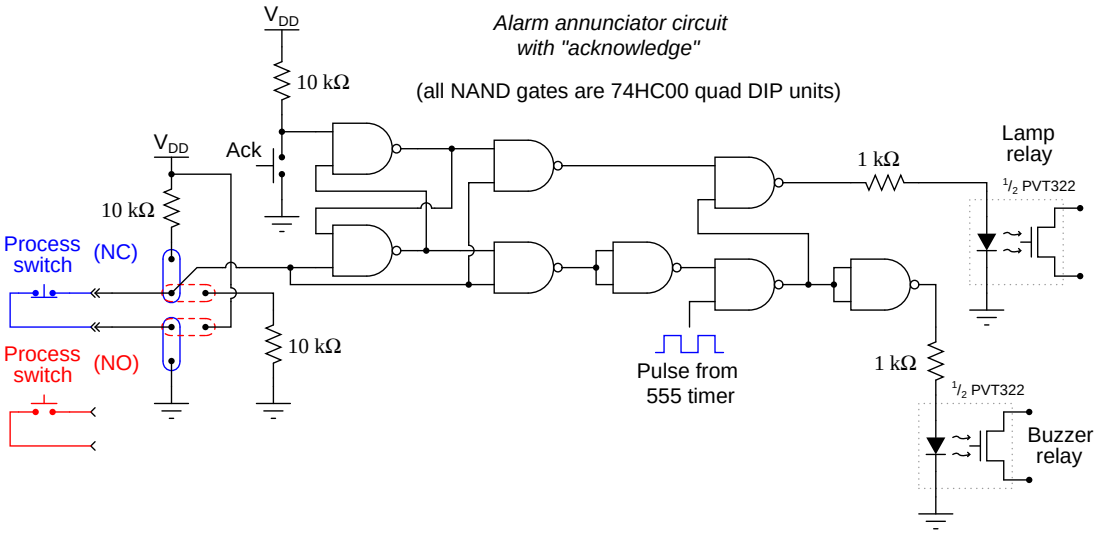
- Measurement span
- Indicator
- Transmitter
- Controller
- Measured Variable (or Process Variable)
- Controlled Variable (or Manipulated Variable)

Challenges

- Identify ways to alleviate the person's boredom by automating this process and giving that person a more interesting job to do.

3.1.4 Annunciator circuit

Explain how the following annunciator circuit works:



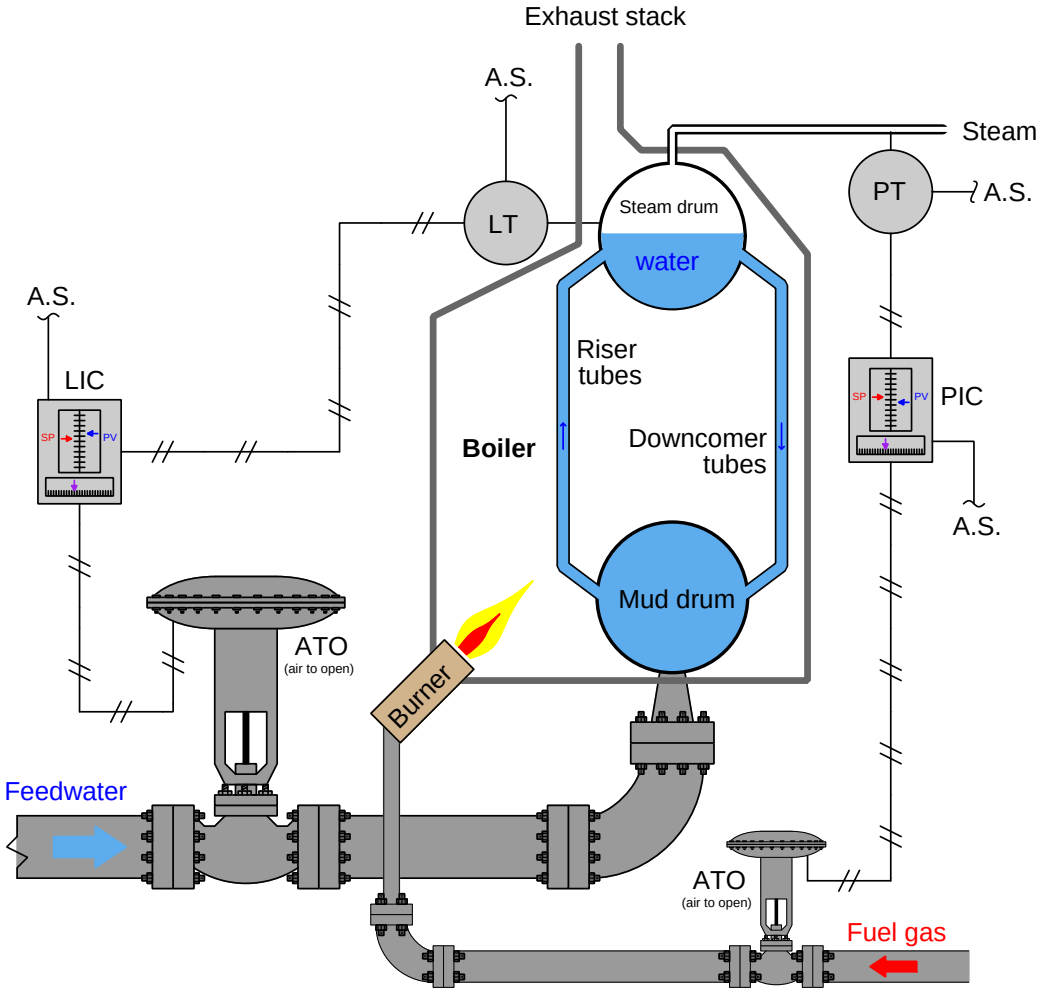
Note the jumper options shown in the diagram: one set of jumper positions configures the alarm for a process switch that alarms when its contacts open, and the other positions configures the alarm for a process switch that alarms when its contacts close. In either case, the circuit is designed to indicate an alarm status when the line going in to the lower-left NAND gate goes *high*.

Challenges

- What is the purpose of such an annunciator circuit, when we could simply wire an indicating lamp to the process switch to obtain visual indication of its status?

3.1.5 Boiler control system

Examine these two pneumatic control "loops" (transmitter-controller-valve systems) for an industrial boiler, controlling both water level and steam pressure:



If the PIC setpoint is 225 PSI and the measured pressure begins to fall below that value, how should the PIC respond, and how will this response bring the steam pressure back up to setpoint?

If the pump supplying feedwater to the boiler begins to wear down over time, becoming less and less effective at providing water pressure to the level control valve, how do you suspect the LIC will respond over time as it works to maintain steam drum water level at setpoint?

Describe a situation where *manual mode* might be useful to either the boiler operator, or to an instrument technician tasked with maintaining either of these control loops.

Suppose the level transmitter's calibration was 12 to 22 inches of water level while the level indicating controller's calibration was 10 to 20 inches of water level. How many inches of water level would the LIC indicate when the actual steam drum water level was 17 inches?

Challenges

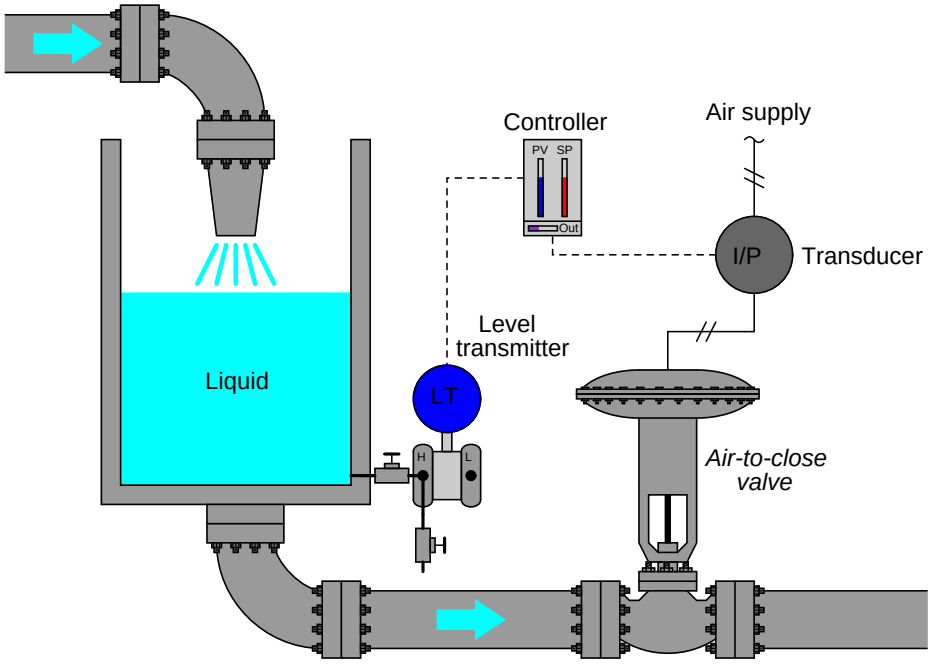
- How would the two control valves react to a loss in pneumatic supply air pressure?

3.1.6 Choosing proper controller action

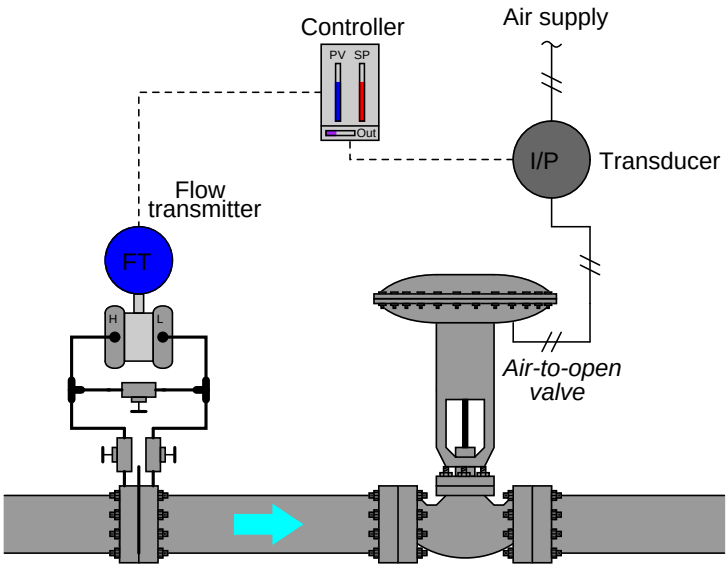
In each of these process control examples, the transmitter produces an increasing signal for an increase in process measurement (level, pressure, temperature, etc.), and the I/P transducer produces an increasing air pressure signal out for an increasing current signal in.

Your task is to determine the proper action for the process controller, either *direct-acting* or *reverse-acting*. Remember, a direct-acting controller produces an increasing output signal with an increasing process variable input. A reverse-acting controller produces a decreasing output signal for an increasing process variable input. It is essential for stability that the controller have the correct direction of action!

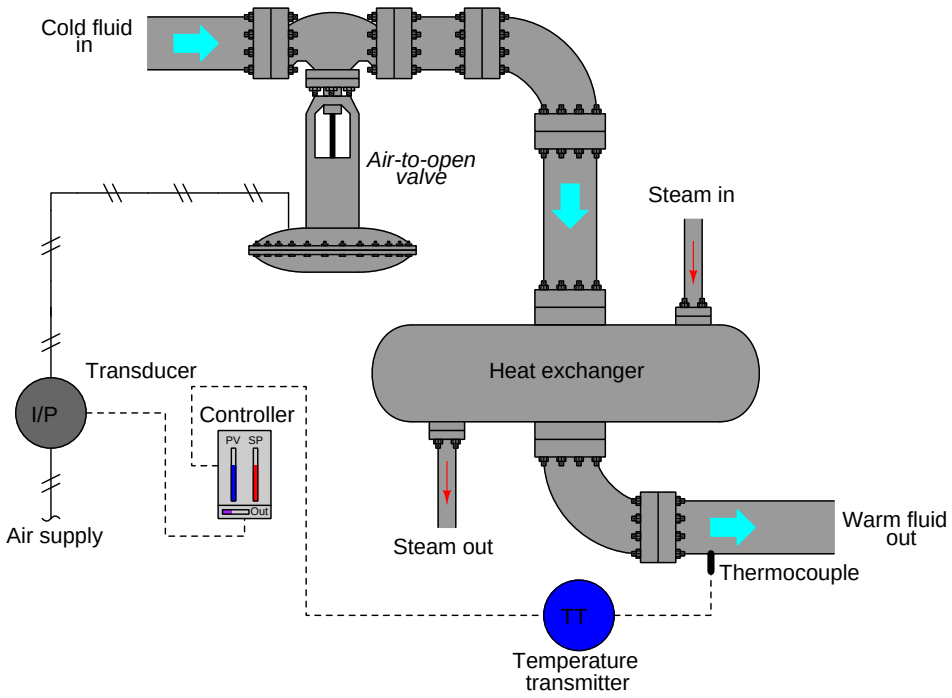
Example 1:



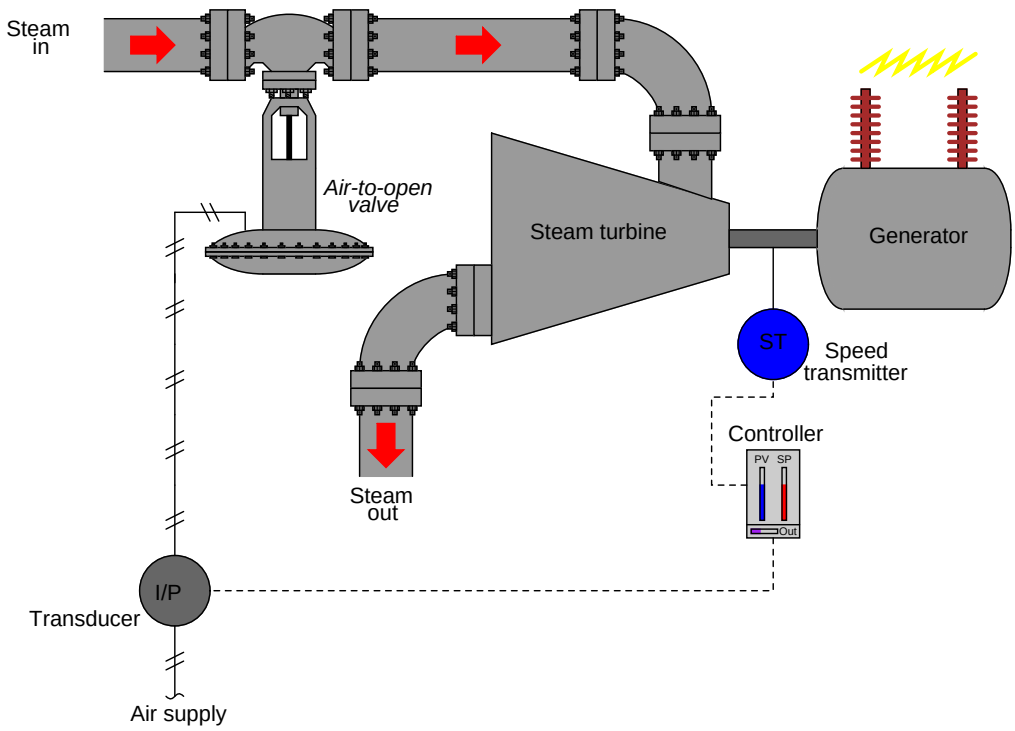
Example 2:


Controller
PV SP
Out
Air supply
I/P
Transducer
Flow
transmitter
FT
H
L
Air-to-open
valve

Example 3:

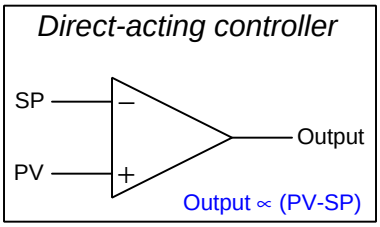

Cold fluid in
Air-to-open valve
Steam in
Heat exchanger
Transducer
I/P
Controller
PV SP
Out
Air supply
Steam out
Warm fluid out
Thermocouple
TT
Temperature transmitter

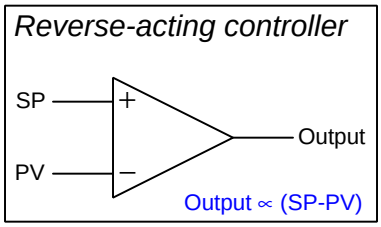
Example 4:


Steam
in
Air-to-open
valve
Steam turbine
Generator
ST
Speed
transmitter
Controller
PV SP
Out
Steam
out
I/P
Transducer
Air supply

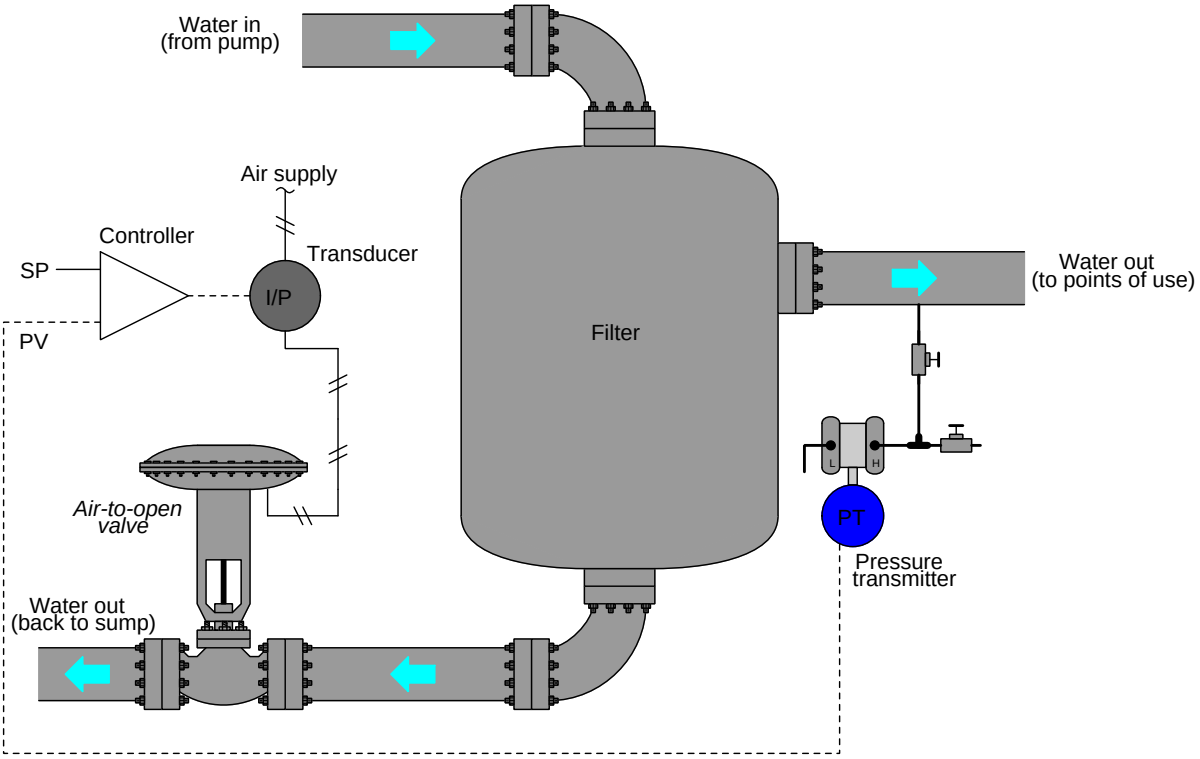
A concept familiar to students of electronics is the *differential amplifier*, a device built to compare two input signals and generate an output signal proportional to that comparison. The most common form of differential amplifier is the so-called *operational amplifier* or "opamp", drawn as a triangle with two inputs labeled "+" and "−" to show the relative influence of each input signal on the output. A process controller may be thought of as a kind of differential amplifier, sensing the difference between two input signals (the process variable and the setpoint) and generating an output signal proportional to the difference between PV and SP to drive a final control element.

The following process control examples replace the controller symbol with an amplifier symbol. Your task is to figure out appropriate labels for the amplifier's input terminals (e.g. "+" and "−"). Remember that a controller is defined as being "direct-acting" if an increase in PV causes an increase in output and "reverse-acting" if an increase in PV causes a decrease in output. Following opamp labeling, this means the PV input of a direct-acting controller should bear a "+" mark while the PV input of a reverse-acting controller should bear a "−" mark.

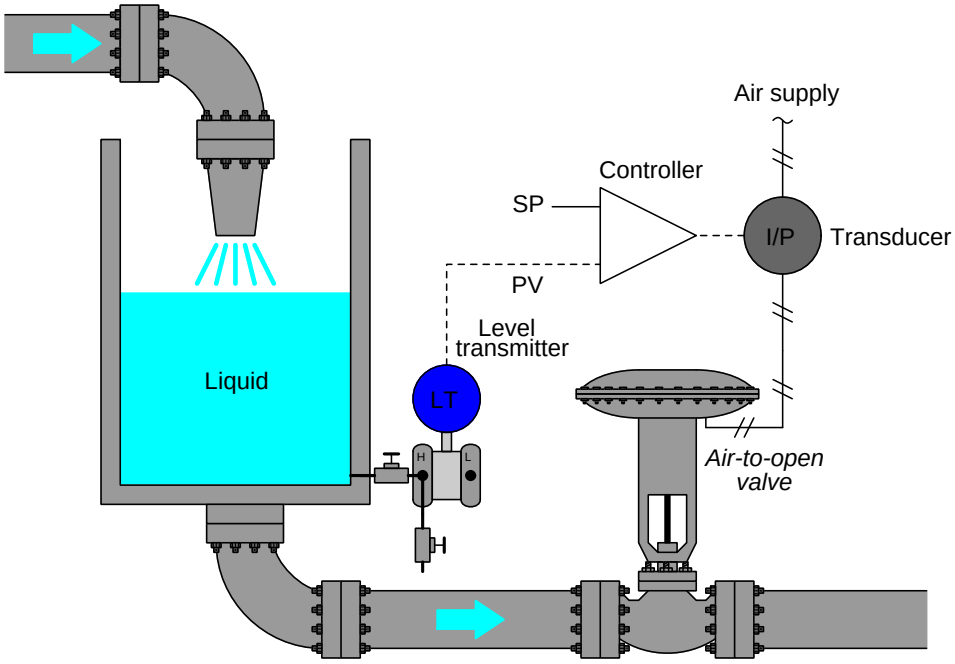




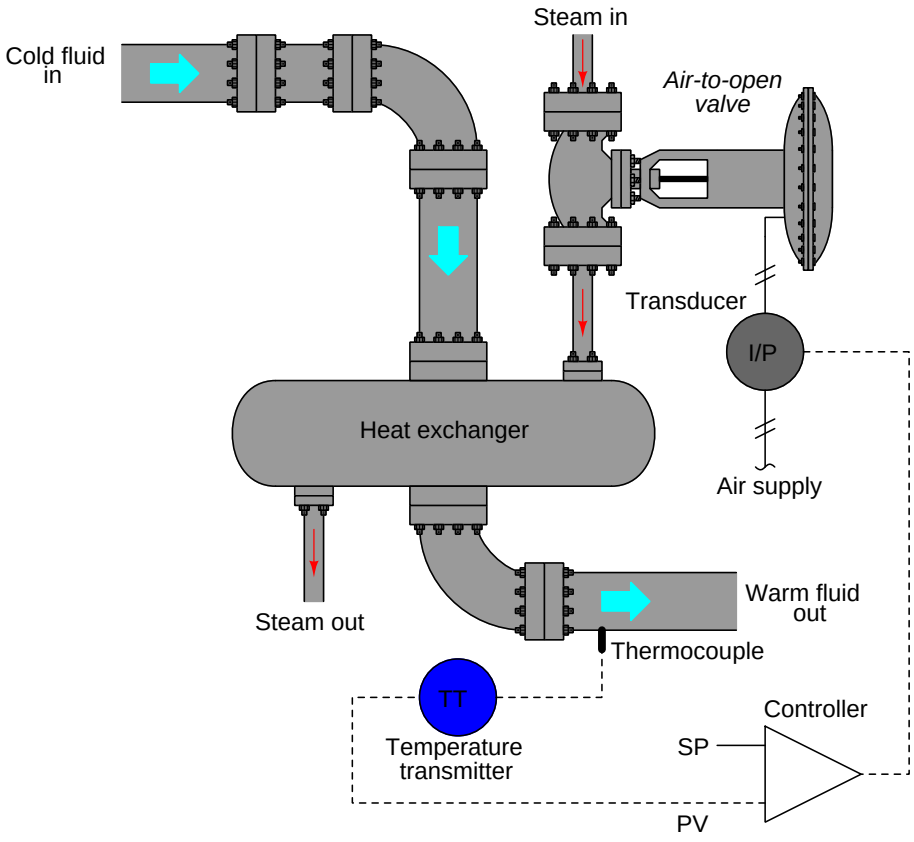
Example 5: *Label the PV & SP amplifier inputs for the correct controller action*



Example 6: *Label the PV & SP amplifier inputs for the correct controller action*



Example 7: *Label the PV & SP amplifier inputs for the correct controller action*



Challenges

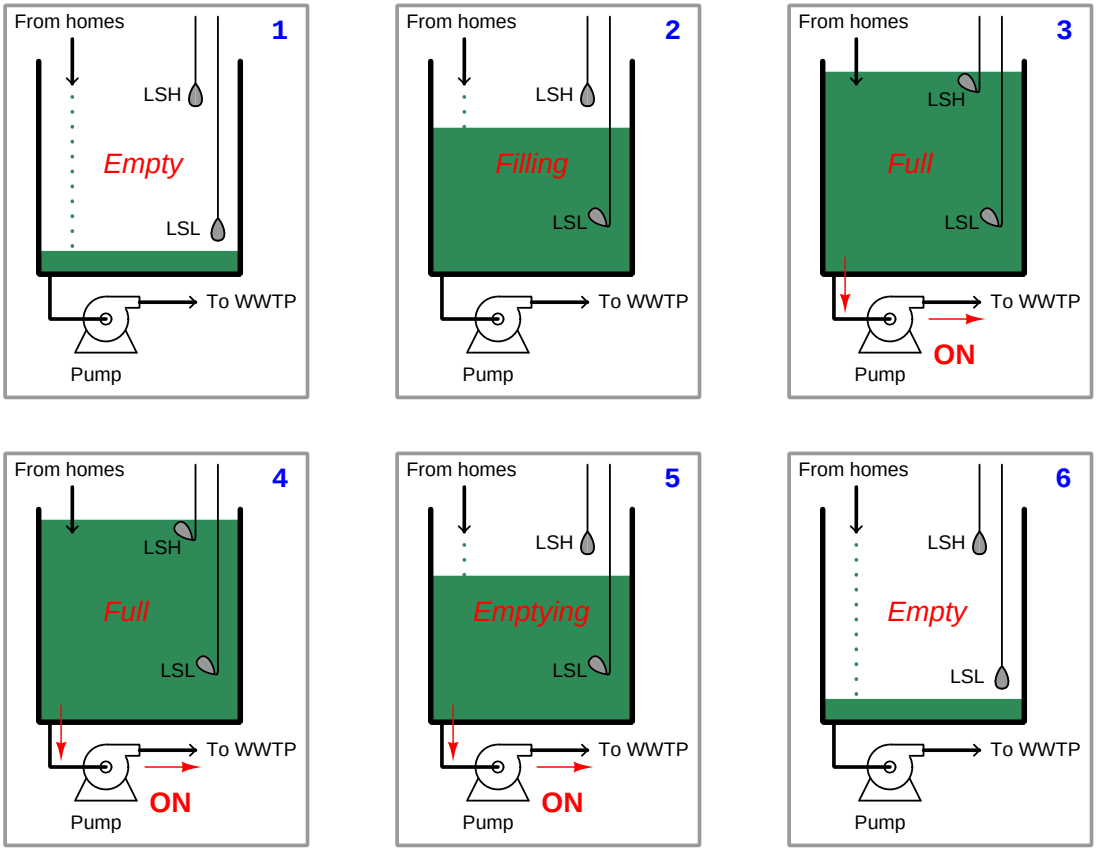
- As always, what is more important than arriving at the correct answer(s) is to develop a clear and logical *reason* for your correct answers. Explain the problem-solving technique(s) you used to determine correct controller action in each of these process control examples.
- A powerful problem-solving technique is performing a *thought experiment* where you mentally simulate the response of a system to some imagined set of conditions. Describe a useful "thought experiment" for any of these process control loops, and how the results of that thought experiment are helpful to answering the question.
- Explain how to reliably identify the process variable (PV) in any controlled process presented to you.
- Explain how to reliably identify the manipulated variable (MV) in any controlled process presented to you.

- Identify and explain the deleterious effect(s) caused by a process controller configured with the wrong action.
- Identify an instrument mis-calibration or mis-configuration that could cause the process variable to settle at a greater value than it should be, assuming all other components in the system are functioning properly.
- Once you have identified the proper controller action for any given process example, identify something that could be altered about the process to require the *other* control action.

3.1.7 Negative-feedback PLC control

Programmable Logic Controllers (PLCs) are industrial control computers well-suited for on/off control, inputting and outputting discrete logic signals. A good application for a PLC is in the control of a *lift station*: an underground reservoir with an automatically-controlled electric pump that collects and transports sewage from neighborhoods to a centralized wastewater treatment plant (usually located miles away).

The following sequence of illustrations shows the lift station reservoir going from empty to full, the pump turning on when the water level reaches the Level Switch High (LSH):

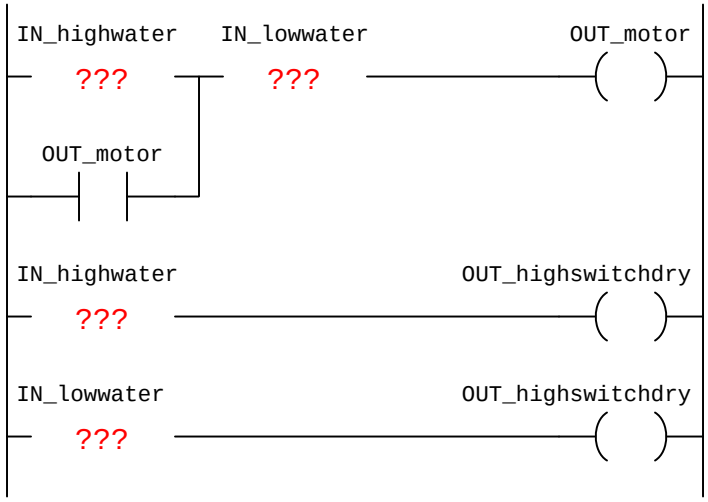


This pump, once started, continues to run until the reservoir's water drops below the Level Switch Low (LSL). This means the PLC must implement a "latching" function to maintain the pump in its energized state until the reservoir returns to an empty condition.

First, explain how such a control system applies the principle of *negative feedback*.

Next, assuming both water level switches are normally-closed (i.e. electrically closed when dry, opening up only when water reaches or exceedstheir level), complete the following PLC program

with the appropriate types of contact instructions (normally-open or normally-closed) to make the pump turn on and off as it should as well as control a pair of indicator lights visible to human operators showing when each switch is *dry*:



Challenges

- A very common misconception among students new to PLC contact/coil programming is that the contact type of the PLC program instruction needs to be the same as the contact type of the real-world switch contact. Explain why this is incorrect as a general rule, but why contact types may have to match for certain circumstances.

3.2 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.

3.2.1 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.

3.2.2 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.

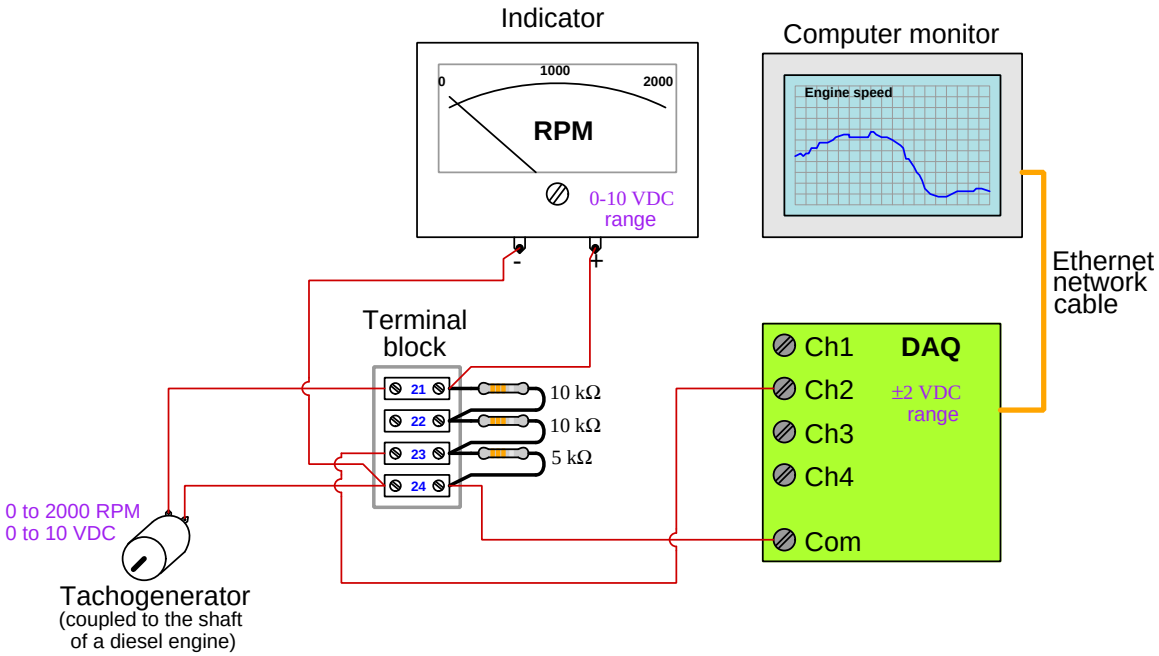
Spreadsheets are particularly useful for situations where the same set of calculations representing a circuit or other system must be repeated for different initial conditions. The power of a spreadsheet is that it automates what would otherwise be a tedious set of calculations. One specific application 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.*

3.2.3 Diesel engine speed monitor

A *tachogenerator* is a small DC generator designed to output a voltage directly proportional to the speed of a rotating shaft. These instruments are used to generate an analog electrical signal representing the rotary speed of a mechanism. An *indicator* is an instrument used to display a measured variable to a human. A *recorder* is a similar instrument used to display a measured variable as a "trend" graph over time. A *Data Acquisition Unit* (abbreviated *DAQ*) inputs one or more analog electrical signal and outputs a digital number representing those signals, essentially a set of analog-to-digital converters combined with digital networking circuitry. DAQ units are often used in *telemetry* systems where various measurements must be taken and reported over long distances via a digital network such as Ethernet or radio.

With these definitions in mind, examine the following pictorial diagram and explain the purpose of each component within the system:



Suppose the diesel engine happens to be running at full speed (2000 revolutions per minute, or 2000 *RPM*). Identify the amount of voltage we would expect to measure between the following pairs of points in the circuit at this engine speed:

- V_{21-24} =
- V_{22-COM} =
- V_{24-COM} =

Challenges

- More important than obtaining the correct solution to a problem is to *devise an effective problem-solving strategy*. Describe at least one strategy useful in this problem.

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

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.

7 August 2026 – added a new Conceptual Reasoning question asking how to program a PLC to properly control wastewater level in a lift station.

5 May 2026 – added some instructor notes.

3 December 2024 – commented out some placeholder text and images for instructor notes in problems that have not yet been created for this module.

18 November 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 commented out the Diagnostic Reasoning section of the Questions chapter which had no content yet.

7 April 2024 – added more questions.

30 March 2024 – added a section to the Tutorial showing an example of a thermostatic room temperature control system.

1 May 2023 – edited Tutorial and added questions.

29 April 2023 – document first created from content in the Lessons In Industrial Instrumentation textbook.

Index

Action, controller, 22
Adding quantities to a qualitative problem, 76
Alarm, annunciator, 40
Annotating diagrams, 75
Annunciator, 40
Automatic mode, 11

Cassier's Magazine, 30
Causal loop, 13
Chart recorder, 28
Checking for exceptions, 76
Checking your work, 76
Circular chart recorder, 28, 31
Code, computer, 85
Controller, 11
Controller action, direct vs. reverse, 22
Converter, 10

DAQ, 34
Data acquisition module, 34
Dead time, 33
Digital multimeter, 34
Dimensional analysis, 75
Direct-acting controller, 22
DMM, 34

Edwards, Tim, 86

Feedback, negative, 8, 9, 12, 16
Fieldbus, 21
Final Control Element, 11

Graph values to solve a problem, 76
Greenleaf, Cynthia, 43

How to teach with these modules, 83
Hwang, Andrew D., 87

Identify given data, 75
Identify relevant principles, 75
Indicator, 25
Intermediate results, 75
Inverted instruction, 83

Jacket, reactor vessel, 21

Knuth, Donald, 86

Lamport, Leslie, 86
Limiting cases, 76
Load, 11
Loop, 11
Loop, causal, 13
Lower range value, 10
LRV, 10

Manipulated variable, 11
Manual mode, 11
Metacognition, 48
Moolenaar, Bram, 85
Moore Industries model SPA alarm module, 37
Murphy, Lynn, 43
MV, 11

Negative feedback, 8, 9, 12, 16

Open-source, 85

Paperless chart recorder, 31
Pneumatic control system, 17
Primary sensing element, 10
Problem-solving technique: thought experiment, 22
Problem-solving: annotate diagrams, 75
Problem-solving: check for exceptions, 76
Problem-solving: checking work, 76

Problem-solving: dimensional analysis, 75
Problem-solving: graph values, 76
Problem-solving: identify given data, 75
Problem-solving: identify relevant principles, 75
Problem-solving: interpret intermediate results, 75
Problem-solving: limiting cases, 76
Problem-solving: qualitative to quantitative, 76
Problem-solving: quantitative to qualitative, 76
Problem-solving: reductio ad absurdum, 76
Problem-solving: simplify the system, 75
Problem-solving: thought experiment, 13, 75
Problem-solving: track units of measurement, 75
Problem-solving: visually represent the system, 75
Problem-solving: work in reverse, 76
Process, 10
Process switch, 35
Process variable, 10

Qualitatively approaching a quantitative problem, 76

Reading Apprenticeship, 43
Recorder, 28
Red Lion Controls panel-mounted indicator, 27
Reductio ad absurdum, 76, 82, 83
Relay, 10
Reverse-acting controller, 22
Rosemount field-mounted indicator, 27

Schoenbach, Ruth, 43
Scientific method, 48
Setpoint, 10
Simplifying a system, 75
Socrates, 82
Socratic dialogue, 83
Span, 10
SPICE, 43
Stallman, Richard, 85
Steam jacket, 21
Strip chart recorder, 30
Switch, process, 35

Thought experiment, 13, 22, 75
Torvalds, Linus, 85
Transducer, 10
Transmitter, 10
Transport delay, 33
Trend recorder, 28

Units of measurement, 75
Upper range value, 10
URV, 10

Visualizing a system, 75

Wastewater disinfection, 19
Weschler panel-mounted bargraph indicator, 26
Work in reverse to solve a problem, 76
WYSIWYG, 85, 86

Zero, 10