MAE 211: Mechatronics

Fall 2002

Introduction

"A blend of mechanics and electronics, mechatronics has come to mean the synergistic use of precision engineering, control theory, computer science, and sensor and actuator technology to design improved products and processes" ME Magazine, May 1997.

Examples

Home Appliances: Controls for microwave, dishwasher, washer/dryer, oven, HVAC, music and video systems, home security systems are all digital electronics now rather than mechanical. These are not just changes in hardware, but additions of functionality, reduction in cost, increase in reliability.

Automobiles: A typical car has 5-10 microprocessors, not counting those in the stereo system. Consider the innovations and improvements in automotive performance, efficiency, safety, and reliability in the last 20 years. What advances will we see in the next 20 years?

Aircraft: Similar advances in aircraft systems are on the way but have been slowed by extremely conservative regulation and approval processes. Navigation and avionics, passenger entertainment are most visible areas.

Industry: Incredible strides in automation, efficiency improvement, pollution prevention, business management. Paperless factories, micro-run production, lights-out manufacturing.

Medicine: MRI, CAT, digital X-rays, laser surgery, robotic surgery, artificial organs, drug delivery systems, G-I pill camera –all of these are examples of mechatronic devises.

Ramifications for Mechanical and Aerospace Engineers:

- We are entering the age of ubiquitous intelligence--smart cards, smart houses, smart highways, smart cars …… All of it is based on embedded microprocessors, miniature sensors, micro-actuators. We have electric motors smaller than the period at the end of this sentence. You will be using, designing and building computers or objects built around computers for the rest of your life. Or not. You could be a plumber.
- All engineers must understand the rudiments of how computers work and have a basic level of skill in programming them. This course is not intended to turn you into EE's, CpE's, or CS students. This course will help make sure you hire them instead of them hiring you.

Chapter 1

Mechanical Concepts

Gears

A gear train is a mechanical assembly of interacting gears that transmits motion or changes the rate or direction of motion. Gear teeth must be matched in shape before they will properly mate or "mesh". The important shape factors are pitch and pressure angle. You will learn about pressure angle in a later course, but gear pitch is a simple concept:

$$\text{Gear Pitch (modulus)} = \frac{\text{number of teeth}}{\text{pitch diameter measured in inches (mm)}} \qquad 1.1$$

In SI units, gears are specified by 'modulus' instead of pitch, but the concept is the same – number of teeth divided by pitch diameter.

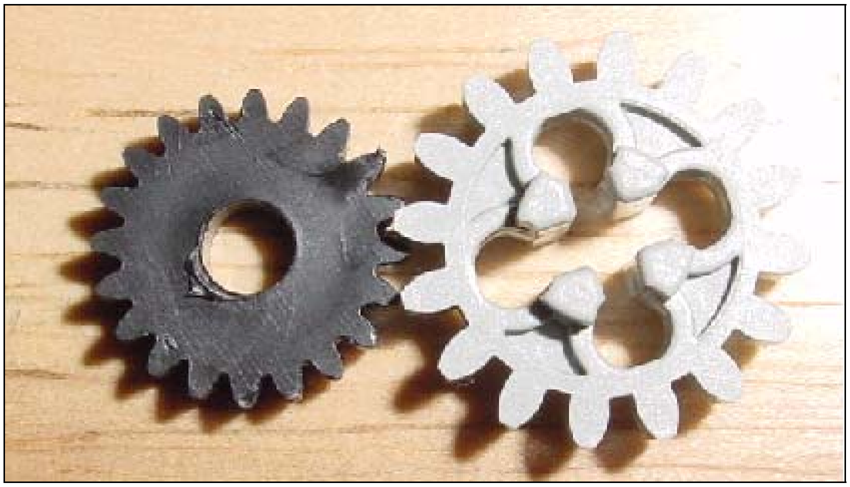


FIGURE 1.1 The two gears pictured here have different gear pitches and will obviously not mate properly.

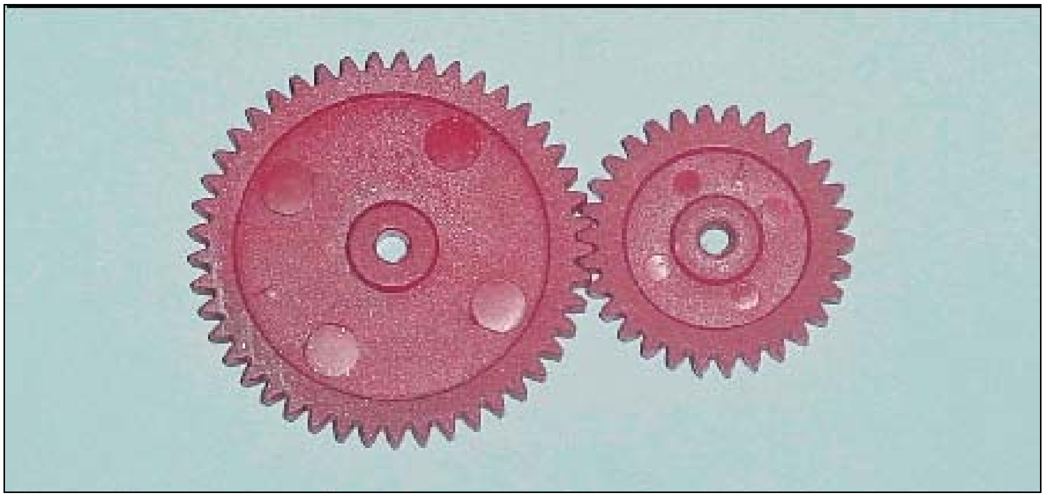


FIGURE 1.2 Properly mated gears have the same gear pitch (modulus)

Pitch Diameter

Look at how gears mate. They don't mate with just the tips of the teeth touching, so outside diameter is not interesting; same for the root diameter (diameter across the bottom of the teeth). What is useful is where the contact between two gears occurs—see figures 1.1 and 1.2. The pitch diameter defines an imaginary circle that cuts through the teeth. The circles thus defined for two properly mated gears are tangent to each other. For two properly mated gears, the shaft centers will be spread half the sum of the pitch diameters apart.

Example 1.1: Suppose I am building a gear train using 24-pitch gears. I want to use a 21-tooth gear to drive a 72-tooth gear. How far apart will the shafts centers be?

Solution: To find the distance apart, add half of the pitch diameter of both gears.

Gear Pitch = # of teeth / Pitch Diameter

24 pitch = 21 teeth / x inches x = .875 inches
24 pitch = 72 teeth / y inches y = 3.00 inches

(x + y) / 2 = 1.94 inches apart

Gear Ratio

In calculating gear ratios, there will always be an input and an output. The input is the side that is supplying the power, and the output is the side that is being driven. The gear ratio is usually given the symbol N, and can be expressed in a number of ways. The calculation is the same, however:

$$\text{Gear Ratio (N)} = \frac{\text{\# teeth on input gear}}{\text{\# teeth on output gear}} \qquad 1.2$$

So, a 12 tooth gear driving a 36-tooth gear would have a gear ratio of N = 12:36 or N = 12/36 or N = 1:3 or N = 1/3 or N = .333. If the numbers of teeth do not turn out to be an integer division, decimal fractions can be used, for example, 1:2.73 was once a common gear ratio for the differential of an automobile.

Example 1.2: Suppose I am building a gear train. I want to use a 21-tooth gear to drive a 72-tooth gear. What is the gear ratio?

Solution: Gear Ratio (N) = 21 teeth/72 teeth = .292

Direction of Rotation

Notice that when two spur gears mate, the output gear turns in the opposite direction from the input gear. In designing a gear train you must keep track of the direction of rotation of each gear.

Multiple Gears

We have talked about a "gear train" but up until now have only considered the use of two gears- which makes a pretty short "train." Obviously, multiple gears can be set up to drive each other. For spur gears, there are two ways in which gears can be connected:

- In-line connection
- Stacked connection

Example 1.3 shows four gears connected in an in-line fashion. We can analyze this gear "train" by taking the gears in pairs. Gear A is the input, and for the first pair we can find the gear ratio very simply: N_1 = (#teeth A)/(# teeth B) = 10/20. Now gear B will turn more slowly than gear A, according to equation (1.2).

$$\boldsymbol{\omega_b/\omega_a = 10/20 \rightarrow \quad \omega_b = ½\, \omega_a}$$

where ω_X is the rotational speed of gear X. Now consider gear C. Gear C is driven by gear B so we can establish the same input / output relationship between gears B and C as we did for A and B.

$$\mathbf{N_{b/c} = 20/10 \qquad \omega_c = \omega_b\,(20/10)}$$

In the same way, we can find a gear ratio for the C / D combination:

$$\mathbf{N_{c/d} = 10/30 \qquad \omega_d = \omega_c\,(10/30)}$$

The gear speeds provide the clue as to how we combine the ratios to get one overall gear ratio for the gear train:

$$\omega_d = \omega_c(N_{c/d}) \quad \text{but} \quad \omega_c = \omega_b(N_{b/c}) \quad \text{and} \quad \omega_b = \omega_a(N_{a/b})$$

Combining these identities we see that:

$$\omega_d = \omega_a(N_{c/d})(N_{b/c})(N_{a/b})$$

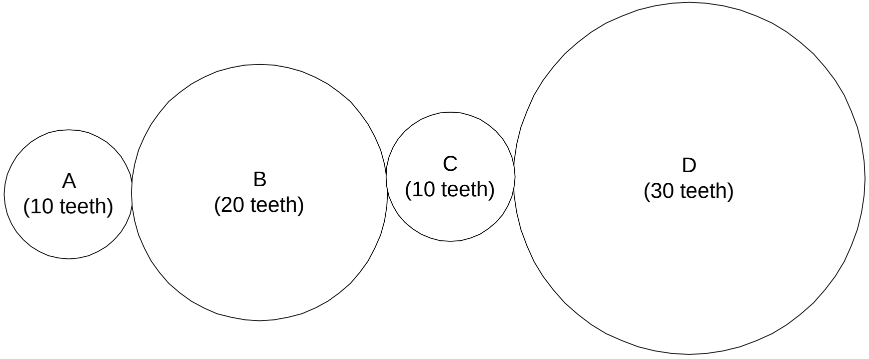
So- the gear ratios in this chain just multiply together. What is interesting, though is to see what happens when gears are combined in this “in-line” fashion. Look at the product of the ratios:

$$(N_{a/b})(N_{b/c})(N_{c/d}) = N_{a/d} \quad = (10/\cancel{20})(\cancel{20}/\cancel{10})(\cancel{10}/30)=(10/30)$$

So- gears B and C have no effect on the gear ratio. Gears in the center of an inline arrangement like this are called **idler gears**. They generally have one of two functions:

- Change the direction of rotation (see figure 1.5)
- Provide physical space to avoid mechanical interference

Example 1.3: Assume that Gear A is the input and Gear D is the output. What is the gear ratio from A:D? What is the direction of rotation of D with respect to that of A?



Solution:
*Gear Ratio (N) = (10 teeth / 20 teeth) * (20 teeth / 10 teeth) * (10 teeth / 30 teeth)*
Gear Ratio (N) = 1/3
Gear D will turn the opposite direction of gear A.

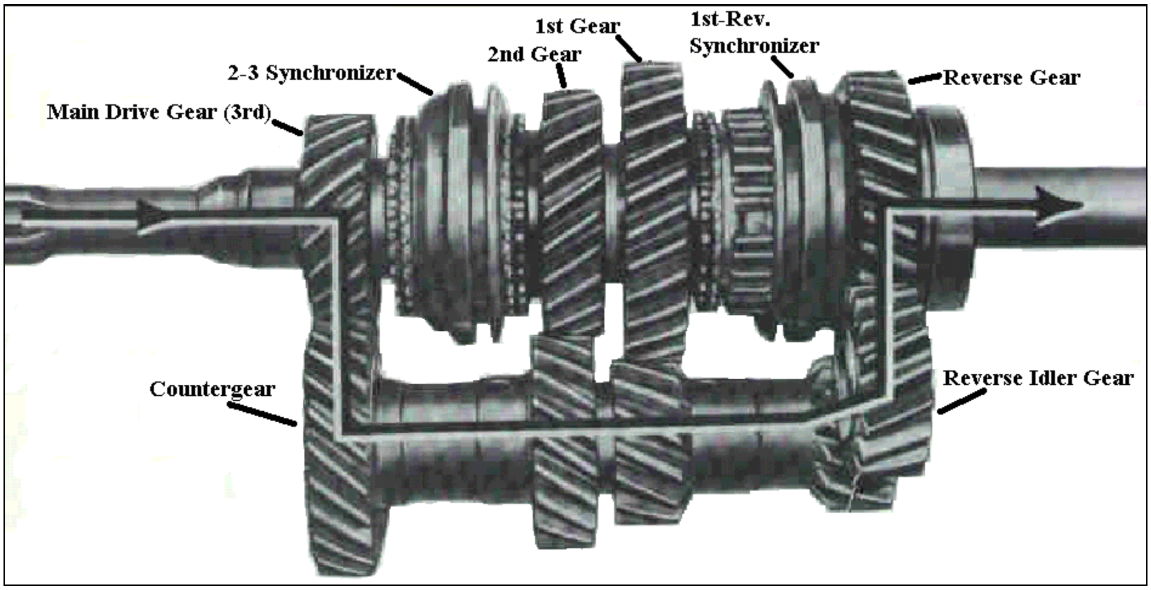


FIGURE 1.5 An idler gear is used in this transmission for the reverse gear.

Stacked Gears

Example 1.4 shows a second way in which gear trains can be constructed, namely with two gears "stacked" on the same shaft. Gear A drives gear B just as before, creating:

$$\mathbf{N_{a/b} = 10/20 \qquad \omega_b = \omega_a (10/20)}$$

But gear B does not mesh with gear C. Since they are on the same shaft, they rotate at the same speed:

$$\mathbf{\omega_c = \omega_b \rightarrow N_{b/c} = 1}$$

Gear C then meshes with gear D, driving it in the normal way:

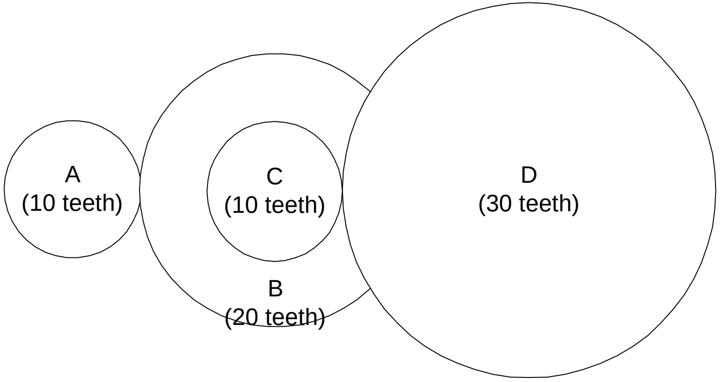
$$\mathbf{N_{c/d} = 10/30 \qquad \omega_d = \omega_c (10/30)}$$

Now, the entire train:

$$\mathbf{N_{a/d} = (10/20)(1)(10/30) = (100/600) \qquad \omega_d = \omega_a (100/600)}$$

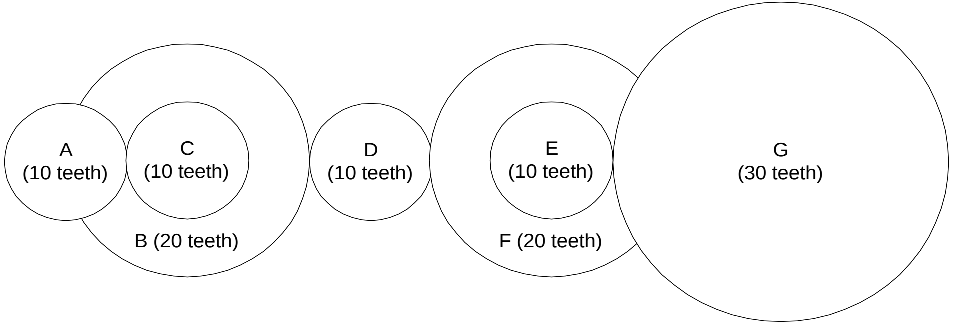
The "stacked" arrangement allows us to multiply two gear ratios together to achieve a greater speed reduction than could have been done with a single gear pair or with an in-line arrangement.

Example 1.4: Now Gear B and Gear C are on the same shaft. Assume that Gear A is the input and Gear D is the output. What is the gear ratio from A:D? What is the direction of rotation of D with respect to that of A?



Solution:
*Gear Ratio (N) = (10 teeth / 20 teeth) *(1) * (10 teeth / 30 teeth)*
Gear Ratio (N) = 1/6
Gear D will turn the same direction of gear A.

Example 1.5: Gear A is turning at 3450 rpm. What is the speed of Gear G? What is the direction of rotation of G with respect to that of A?



Solution:
*Gear Ratio (N) = (10 / 10) * (20 / 10) * (10 / 20) * (10 / 30) = 1/3*

Output Speed / Input Speed = N
Output Speed / 3450 = 1/3
Output Speed =1150 rpm

Gear G will turn the same direction as gear A.

Force

Newton's second law states if the resultant force acting on a particle is not zero, the particle will have an acceleration proportional to the magnitude of the resultant force, which can be stated as:

$F= m * a$ 1.3

Where 'F' is the resultant force acting on the particle (measured by pounds or Newtons), 'm' is the mass of the particle (measured by slugs or kilograms), and 'a' is the acceleration of the particle (measured by feet per seconds squared or meters per second squared). A particular case of importance is that of the attraction of the earth on a particle located on its surface. The acceleration due to earth's gravity is 32.2 ft/s^2 or 9.81 m/s^2. In the following example, pay particular attention to units:

Example 1.6: I have an I-beam that weighs 193 lbs. What is its mass?

*Solution: $F = m*a$ 193 lbs = m * 32.2 ft/s^2*
*m = 5.99 lbs * s^2/ft = 5.99 slugs*

Speed and Torque

The rotational force, or torque, generated at the center of a gear is equal to the product of the gear's radius and the force applied at its circumference. When gears are meshed, their respective radii determine the translation of torque from the driving gear to the driven one.

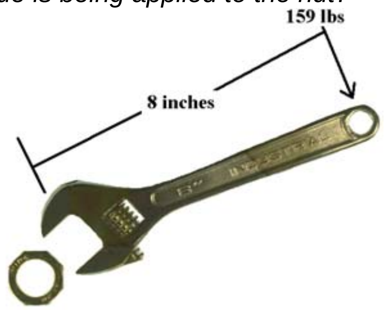
$T=F*R$ 1.4

Where 'T' is torque (measured in foot pounds or Newton meters), 'F' is force (measured in pounds or newtons), and 'R' is the radius of the gear (measured in feet or meters).
The input and output shaft speeds and torques are related to the gear ratios:

$$\frac{\text{output speed}}{\text{input speed}} = \text{gear ratio(N)} \qquad 1.5$$

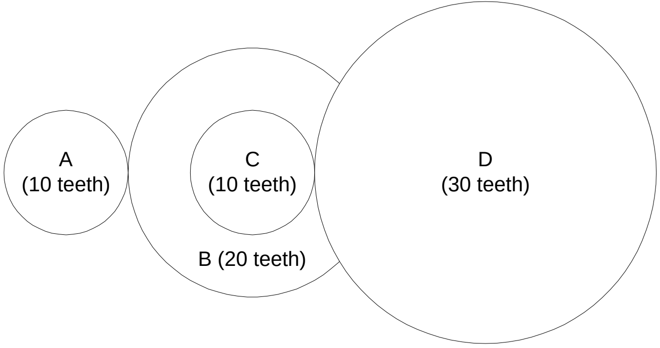
$$\frac{\text{output torque}}{\text{input torque}} = 1/\,\text{gear ratio(N)} \qquad 1.6$$

Example 1.7: What torque is being applied to the nut?



*Solution: T = F*R T = 159 lbs. * 8 inches = 106 ft lbs.*

Example 1.8: Gear A is turning at 1750 rpm and has a torque or 75 ft lbs. What is the speed and torque at gear D?



Solution:
*Gear Ratio (N) = (10 teeth / 20 teeth) * (10 teeth / 30 teeth)*
Gear Ratio (N) = 1/6

Output Speed / Input Speed = N Output Torque / Input Torque = 1 / N
Output Speed / 1750 rpm = 1/6 Output Torque / 75 ft lbs = 6
Output Speed =292 rpm Output Torque = 450 ft lb

Gear Types

Probably the simplest kind of gear train to understand is made of **spur gears**. Spur gears are flat disks with teeth cut into their edges so the teeth are radial to the disk; see figures 1.1 and 1.2.

Bevel gears work the same way as spur gears, but allow the direction of motion to be changed (usually by 90 degrees). The gear teeth are cut into the face of the gear on a 45-degree bevel as shown in figure 1.3. One disadvantage of bevel gears is that one or both of the gears must usually be cantilevered on the shaft and they produce significant side thrust on the shaft and bearings. Bevel gears are generally not to be used for very heavy loads if they can be avoided. For some applications, they are the best solution, however. Automobile differentials are an obvious example- the drive shaft runs from the engine to the rear axle, which must turn at a 90-degree angle to the drive shaft. Automobile differentials contain very robust bevel gears to make this transition.

Worm gears are another means of changing direction and effecting speed reduction; see figure 1.4. They also require careful alignment, but are very rugged and can be used to achieve high gear ratios in a single stage. Precision sets may even machine the worm to wrap around the spur and machine a hollow center in the spur to match the curve of the worm. A unique feature of the worm drives is that they can not be back driven. Worm gears are cut at such a shallow angle that it is generally impossible for the load to drive the motor backward. When meshed with a spur gear, worm gears create an n to 1 reduction where n is the number of teeth on the spur gear.

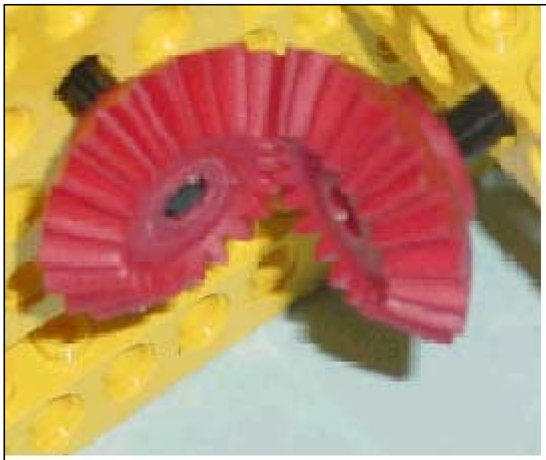


FIGURE 1.3 bevel gears

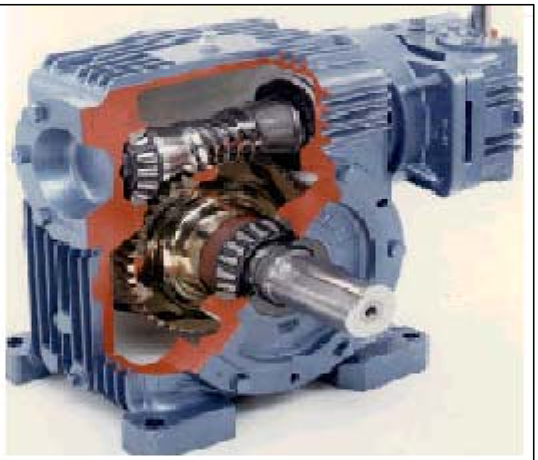
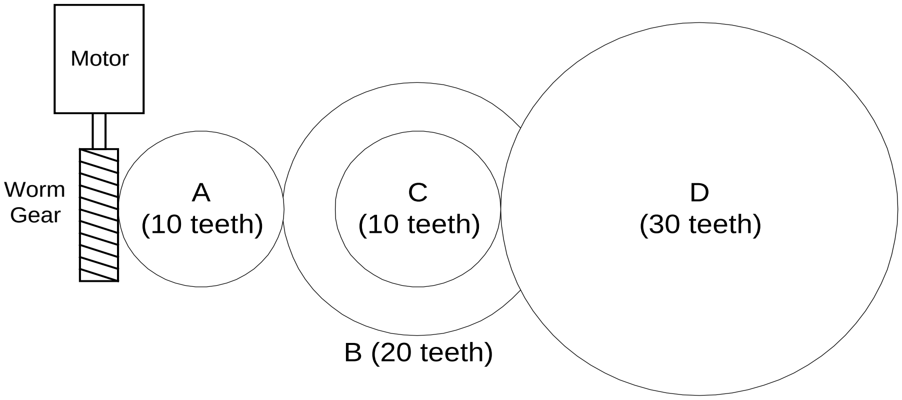


FIGURE 1.4 worm gear mechanism

Example 1.9: What is the gear ratio between the motor and gear D?



Solution:
*Gear Ratio (N) = (1 tooth / 10 teeth) (10 teeth / 20 teeth) * (10 teeth / 30 teeth)*
Gear Ratio (N) = 1/60

Differentials are most commonly used in automobiles, (see figures 1.6 and 1.7). A differential allows two wheels on the same axle to travel at different speeds. If there is equal force on both tires, the tires will travel at the same speed. If there is an unbalanced force between the wheels, then one wheel can slip while the other one is being driven. This is very useful in going around corners, where the outside wheel must travel faster than the inside wheel. One disadvantage is that on a slippery road, the differential allows one tire to spin while the other does nothing.

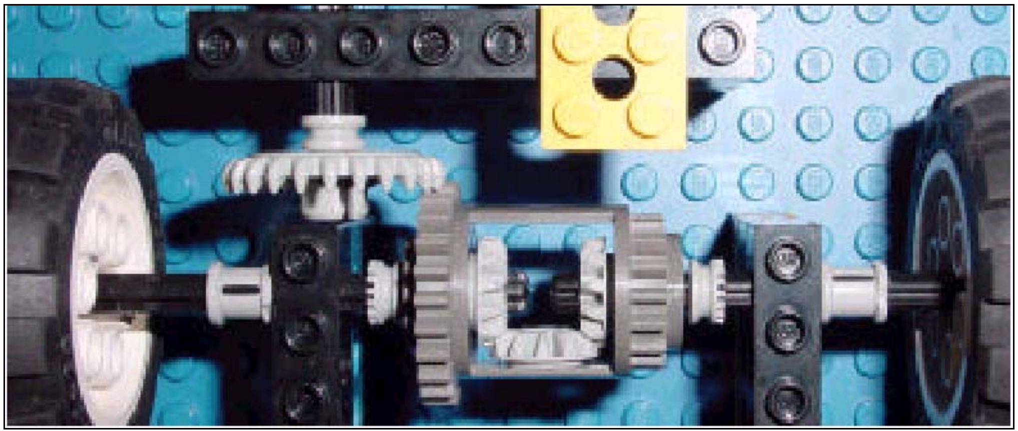


FIGURE 1.6 Lego's version of a differential

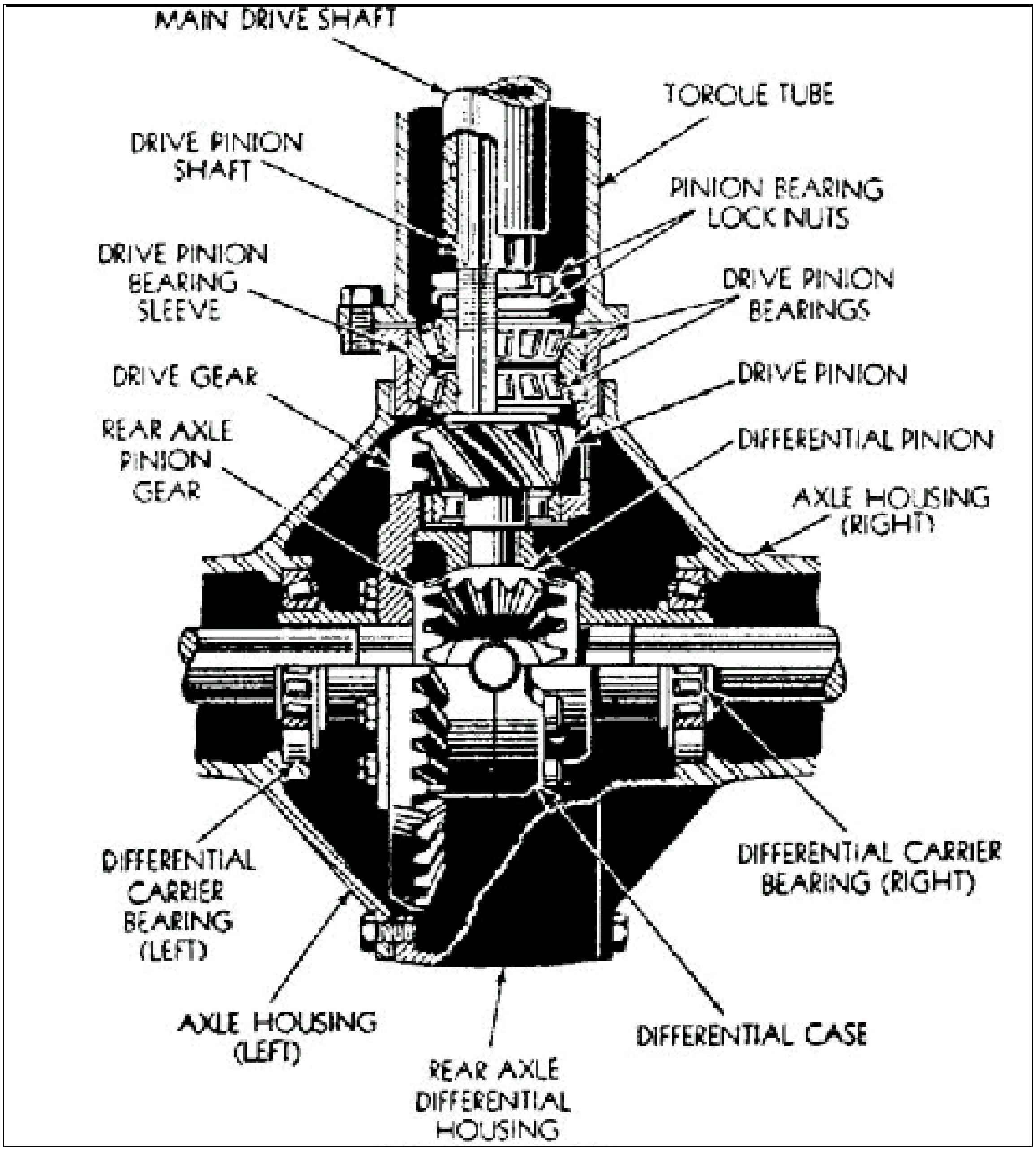


FIGURE 1.7 Cutaway view of automobile differential.

Linear Motion

Conversion of Rotary to Linear Motion

Conversion of rotary motion to linear motion and vice versa is an important and common function in mechanical design. This is because motors of one type or another are the most common forms of work generators, and they inherently provide rotary motion. The purpose of this unit is to examine a number of standard mechanisms for translating rotational motion to linear motion. Some of them can also work in the opposite direction, and some cannot. You will recognize many devices from your everyday experience, which use these mechanisms.

Rack and pinion

The rack and pinion is one of the most common mechanisms for converting rotary to linear motion. The pinion gear is nothing more than a spur gear like you have seen in the previous unit. The rack is a kind of “unrolled” gear with the gear teeth cut into a straight rod. Figure 1.8 is a photo of a rack and pinion. The McMaster Carr catalog (http://www.mcmaster .com) has examples of rack and pinion sets in a variety of sizes and materials.

Like regular gears, the rack and the pinion teeth must mesh correctly, and thus the concept of gear pitch extends to the rack directly. Recall that the units of pitch are teeth per inch. However, this is misleading because the pitch of a round gear is measured in teeth per inch of pitch diameter. Thus, a rack that mates with a 12-pitch pinion gear will not have 12 teeth per inch of rack, rather it will have 12/π teeth per inch of rack—slightly fewer than 4.

Clearly, the pinion is the rotating part of the pair, but there are two ways the system can be set up. If the center of rotation of the pinion is fixed in space, then the rack must be free to translate as the pinion turns. This is the arrangement in a rack and pinion steering system in an automobile. The pinion is attached to the steering wheel shaft, and the rack is driven left or right as the driver turns the wheel, and causes the front wheels of the car to turn left or right, accordingly. Another example of this type of mechanism is the quill of a drill press.

Conversely, the rack can be held stationary, and the motor or mechanism driving the pinion can be mounted on a moving platform. The worktable of many drill presses can be adjusted up and down using this kind of arrangement. Some construction scaffolds and certain elevators also work on this arrangement. Check out the following link to see some examples: http://www.alimakamericas.com/. The most common example is one that is an extension of this concept. If you think of a wheel as a pinion with very small teeth, and a road or railroad track as the rack, then it is easy to see that all of our common forms of land conveyance work on the same principle.

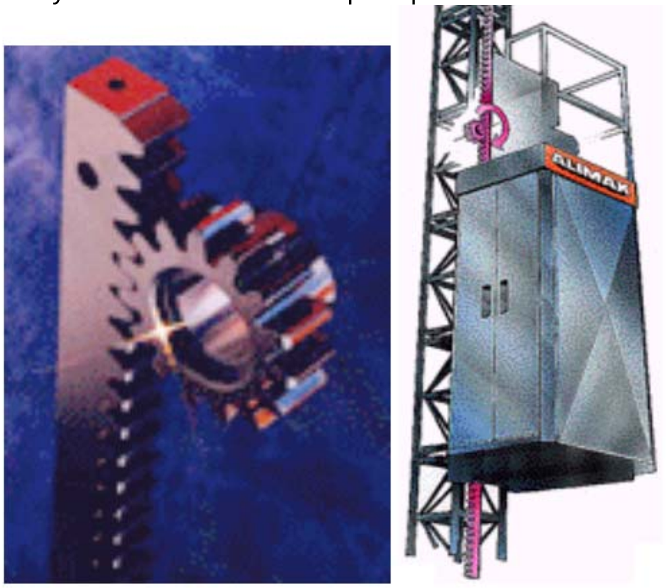


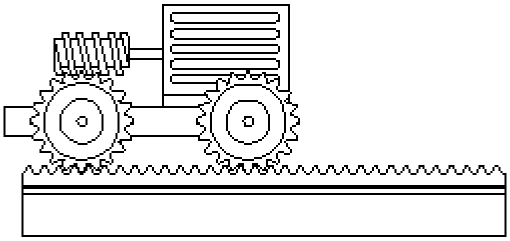
FIGURE 1.8 rack and pinion

It is often necessary to relate the rotational and linear motions mathematically. This is not difficult. As the pinion rotates one revolution, the relative motion between the pinion and the rack must correspond to the circumference of the pinion, since there is no slip between pinion and rack.

$$S = \pi D_p \theta \qquad 1.7$$

Where S is the linear displacement of either the rack or of the pinion along the fixed rack, D_p is the pitch diameter of the pinion and θ is the angular displacement of the pinion measured in number of revolutions (radians/2π).

Example 1.9 The sketch below shows a Lego motor set up to drive the carriage for a model X-Y plotter/recorder. All gears are modulus 1. If the motor turns 10 turns, how far does the carriage move?



Solution:
The two spur gears have 20 teeth each. Since the worm gives a gear ratio of 1:N where N is the number of teeth of the spur, the spur gear turns:

θ = (1/20)(10 revolutions) = ½ revolution*

The distance moved will be $S = \pi D_p \theta$ *millimeters*

*To find D, recall that modulus = (# teeth)/D in millimeters and these gears are modulus 1 so D = 20 mm. Then S = 10 * π = 31.4 mm*

Rack and pinion systems are very rugged, dependable systems. Rack and pinions can be made any length, by just butting rack sections end-to-end. They are relatively simple, but somewhat expensive due to the precision machining that must be done on the parts. Check McMaster Carr for an idea of the costs for different sizes and types of materials: http://www.mcmaster.com Go to the “Power Transmission” section of the catalog, and look at the gears pages—at the top of the page is a pull down menu which will include rack and pinions.

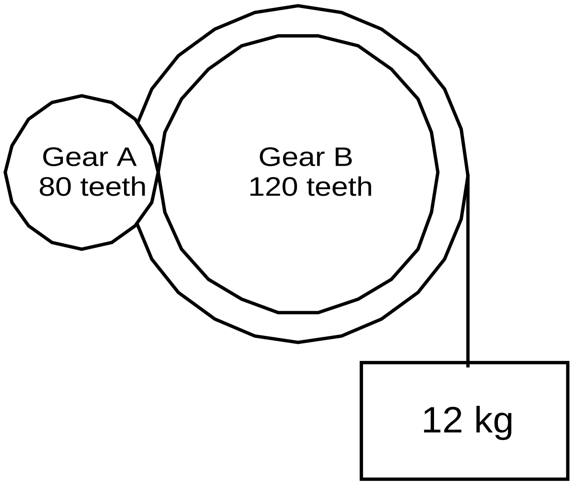
Belts or Cables and Pulleys, Chains and Sprockets

Belt or cable and pulley systems are very common and familiar to all of us. Conveyor belts, escalators, elevators, garage door openers and hundreds of other devices use these mechanisms. Belt systems have the advantage of being relatively cheap, versatile and reliable. They are also much more forgiving of pinion misalignment problems than many other types of drives. Belt drives can operate with the rotary element providing the power or being driven. The relationship between rotational and linear displacement is the same for belts as for rack and pinion systems, namely:

$$S = \pi D_p \theta \qquad 1.8$$

where D_p is now the diameter of the pulley. There are many different types of belts –flat, V, cog, round, etc. The exact pulley measurement used will depend on the type of belt, but the principle is the same. You will study them again in more detail in later ME courses.

Example 1.10: Gear A is turning at 120 rpm counterclockwise and has a modulus of 2. Gear B is on the same shaft as a drum with a diameter of 75mm. A 12 kilogram mass is attached to a cable that wraps around the drum. How many revolutions of gear B will it take to lower the weight 100 meters? How long will it take? What is the torque being applied by the weight to the shaft of gear A?



Solution:

Gear A

Modulus = # teeth/ pitch diameter
2 = 80 teeth / pitch diameter
pitch diameter = 40 mm

Gear B

Modulus = # teeth / pitch diameter
2 = 120 teeth / pitch diameter
pitch diameter = 60 mm

Gear Ratio

N = 80 teeth / 120 teeth
N = 2/3

Speed of Gear B

*Speed of gear A * gear ratio*
*120rpm * 2/3 = 80rpm*

Time Calculation

$S = \pi * D * \theta$
$100\ m = \pi * 75\ mm * \theta$
$\theta = 424.$ *revolutions*
424. revolutions / 80 rpm = 5.3 minutes

Weight of Mass

$F = m * a$
$F = 12\ kg * 9.81 m/s^2$
F = 118. Newtons

Torque on shaft B from drum

*T = F * R*
*T = 118 Newtons * (37.5mm)*
T = 4.43 Newton meters

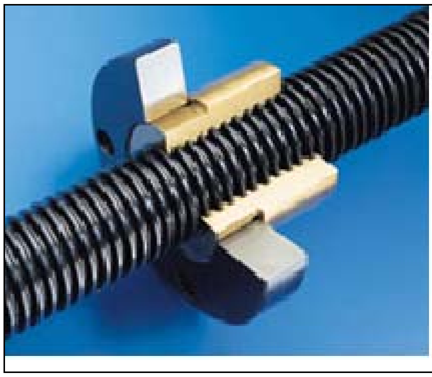
Torque on shaft A

output torque / input torque = 1 / gear ratio
4.43 Newton meters / input torque = 3 / 2
input torque = 2.95 Newton meters

Lead Screws

We generally think of a screw as a clamping or fastening device, but they also find wide use as rotary to linear motion converters. The principle of operation is simple: as the screw turns, the nut is not allowed to rotate, and so it must slide along the screw and push or pull whatever it is attached to along with it. In the lab, we will use plain threaded rod and weld nuts for lead screws, but for industrial or precision purposes, specially designed screws and nuts are available. ACME threads are square looking and can support heavier loads without stripping than conventional screw threads. They require special nuts, obviously, and are available through the McMaster Carr catalogue. Figure 1.9 shows a low-friction plastic used in the lead nut. This type of system is lower cost and lower friction, but lower capacity than an all-steel system.

For precision work or for high-speed applications, ball screws can be purchased like those in Figure 1.10. These have rounded grooves in which ball bearings circulate through a fairly complicated nut. Ball screws have good load handling capability and very low friction, but can be somewhat noisy in operation. Lead screws can offer fairly precise positioning, but the range of motion is limited by the length of the screw, and screw length is in turn limited by a combination of the diameter of the screw and the maximum rotational speed to which it will be subjected. When rotated at high speeds, the screw tends to bow like a jump rope and whip about the rotational axis. Excessive vibration and stress on bearings result. The screw should not be used to bear transverse loads, and so it must also be precisely aligned with the supports for the carriage of the moving element.



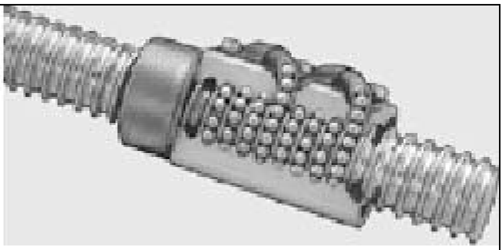


Figure 1.9 (left): Plastic lead nut insert
Figure 1.10 (above): Ball screw and nut

Machine Screw Basics

Before getting into lead screws, it would be useful to understand a little about regular screws and bolts used for fastening things together. Screws are generally classified according to what they are to be used for and also according to the type of head the screw has. For example, in the hardware aisle at Lowe's you will find wood screws, sheet metal screws, machine screws, hex-head cap screws, carriage bolts, and numerous other variants. All of these come in a wide range of sizes, often in several different materials (steel, brass, stainless steel, zinc coated steel, galvanized steel, etc. etc.). Browse through the McMaster Carr catalog (http://www.mcmaster.com), look under "fastening and sealing". A great reference for finding out more than you ever wanted to know about dimensions and standards for threaded fasteners is Machinery's Handbook, Industrial Press, Inc., New York, N.Y. ISBN 0-8311-1155-0. We will be most interested in the general class of screws called machine screws. By convention, fasteners with a diameter smaller than ¼ inch are called screws, and those with a diameter of ¼ inch or larger are generally called bolts, although you will sometimes see all sizes referred to as screws. The following discussion will pertain to all sizes of screws/bolts.

Machine screws are specified in terms of their diameter and the number of threads per inch (also called pitch). For example, a ¼-20 screw has a nominal diameter of ¼ inch and 20 threads per inch. Smaller screws, with a diameter less than ¼ inch are given number sizes ranging from number 12-24 (i.e. diameter "12", 24 threads per inch) down to the very tiny #2-56 screws. There are actually screws even much smaller than #2, but you are not likely to deal with them very often. With only one exception, all screw sizes are even numbers, i.e. 12-24, 10-24, 8-32, 6-32, 4-40, and 2-56. The only odd number size of which I am aware is a #5-40. There is no "formula" that I know of connecting the numbers to a diameter in inches. If you need to know what size hole to drill for clearance or to tap for threads, you need to look it up, or do what most people do and hold the screw up next to a drill bit and see if it looks right.

Lead Screw Pitch and Travel

All of this has been leading up to the topic of lead screw pitches and how to know how far the lead nut will move for each turn of the lead screw. The calculations are pretty easy—threaded rod has single entry threads (i.e. one spiral the length of the rod). Therefore, for our ¼ -20 lead screws, the lead nut will travel one inch for each 20 turns of the threaded rod.

If you purchase a new lead screw or ball screw, you can check the specifications to see what the relationship between rotation and linear travel are. If you are using a screw you have found or salvaged, you need to examine it carefully. Some screws use multiple-entry threads, so you can't just count the number of peaks or valleys per inch and assume the same relationship will hold. With multiple entry threads, there are two or more parallel "nested" sets of threads, which result in a longer pitch than it would appear from casual observation. To see if the screw has multiple entries, examine the end of the screw and see if there is more than one place where the threads "start" at the end. If so, count them. Suppose your screw has 2 entries. Then every other peak or valley belongs to the same thread (see figure 1.11). To translate revolutions of the screw to linear travel, measure the distance from the peak or valley of one screw thread to the next peak or valley of the same thread.

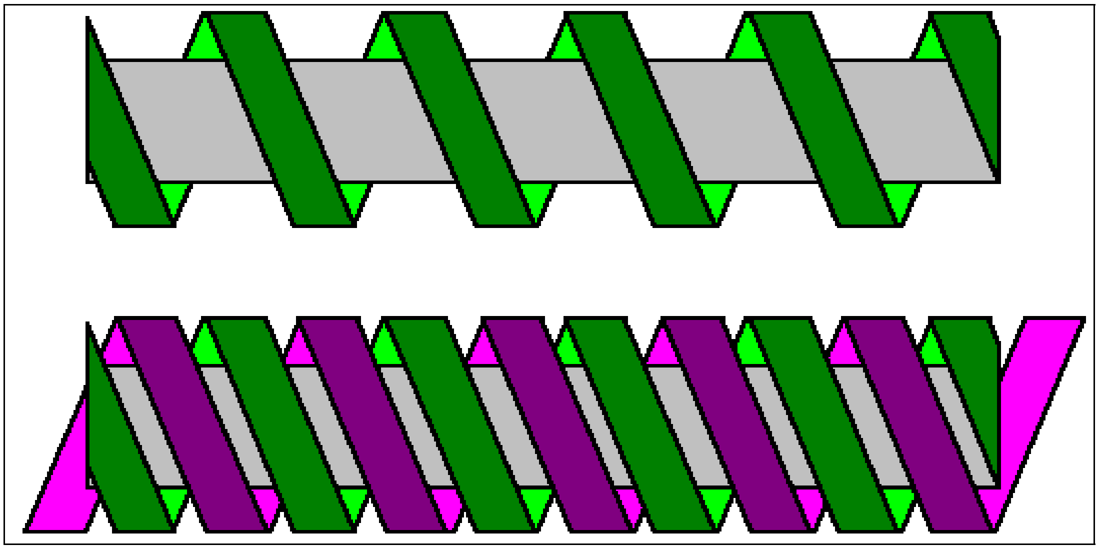


Figure 1.11: (Top) Lead screw with one entry; (bottom) Lead screw with two entries.

Crank and Slider

One of the most common mechanisms for transforming linear to rotary motion or vice versa is the slider-crank mechanism. The piston and crankshaft is the prime example of this mechanism (see figure 1.12) and is found in the vast majority of internal combustion engines, air compressors and refrigeration compressors. Engines and compressors which use piston/crankshaft mechanisms are called reciprocating engines to distinguish them from turbines or jet engines, or from one of the many non-piston types of compressors. The crank and slider is an excellent mechanism for generating repetitive motion, but has a number of drawbacks as a position control machine. The range of linear motion is limited to the throw of the crankshaft (which can be quite large—some stationary or marine diesels have a stroke of several feet). There is a nonlinear relationship between the rotation angle and the linear displacement, given by $S = r \sin\Phi$, with S being the linear displacement and Φ. the crank angle.

An extension of the slider crank is the cam and follower mechanism. A cam is an eccentric element mounted on a rotating shaft. As it turns, it pushes on a rod or roller causing it to move back and forth. Cams are used among other things to open and close the valves in a reciprocating internal combustion engine. Mathematical analysis of cams is non-trivial, and will be covered in one of your machine design courses later in the curriculum.

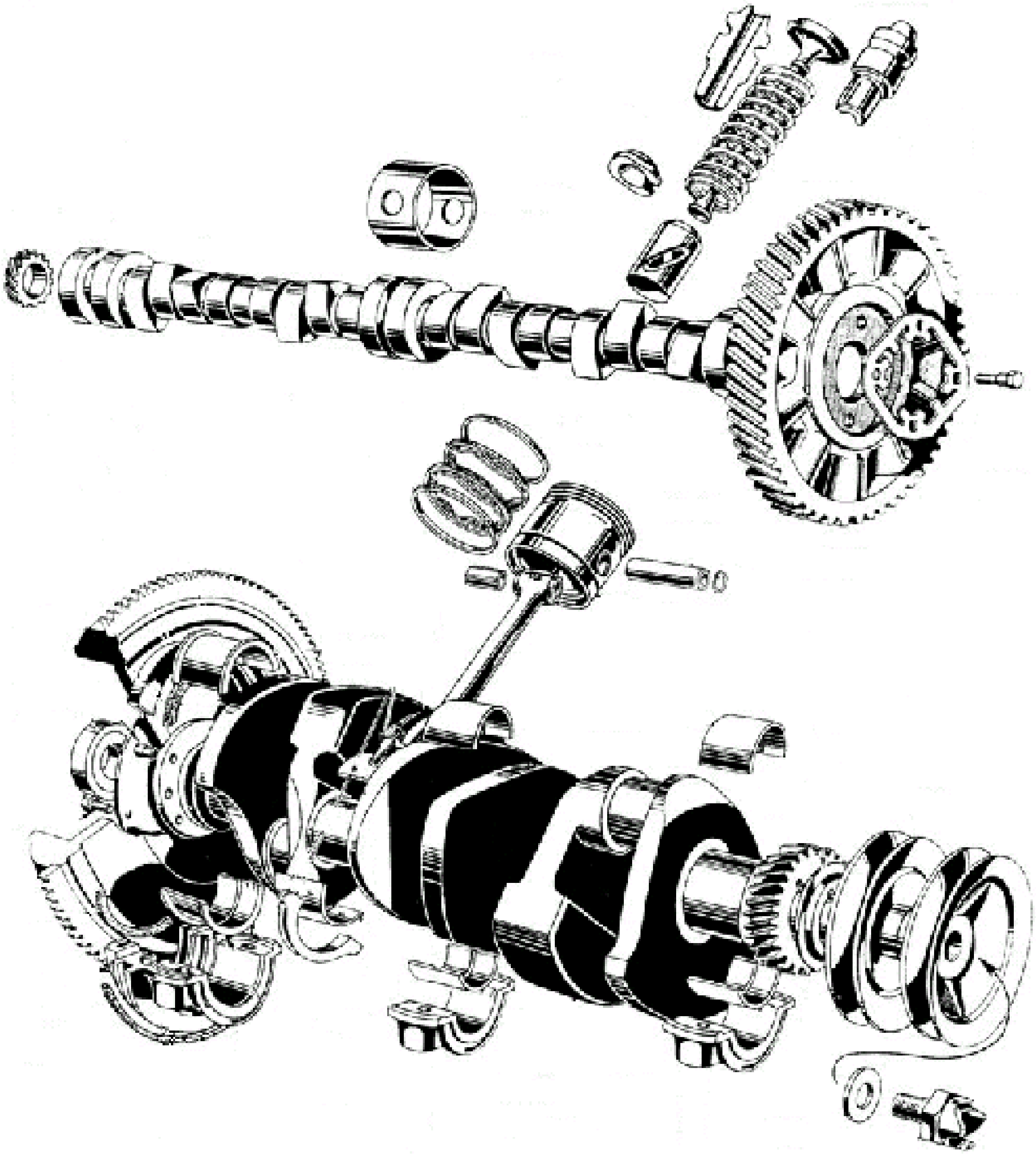


FIGURE 1.12: Automobile crankshaft w/ piston and camshaft w/ valve, exploded view.

Bearings

Bearings are simply places where two parts that are in motion relative to each other come together. Bearings can accommodate both rotating and sliding motion, or even a combination of the two. Some types of bearings are designed to prevent motion in a particular direction while allowing motion in another. Bearings come in a variety of designs, shapes, and sizes; but their primary functions are generally:

- To allow one part to move relative to another while controlling the motion in at least one axis
- To reduce friction and wear as much as possible on moving parts
- To reduce bending stresses on shafts
- To provide a replaceable, sacrificial part in a large or expensive mechanism.

A **plain bearing** is where a shaft simply slides through a hole and is free to either rotate or slide. The shaft is constrained from translation in two axes perpendicular to the axis of the bearing (see figure 1.12). It is free to translate along the axis of the bearing or to rotate about the axis of the bearing. A shaft supported by a single plain bearing is also partially free to rotate about either of the two axes perpendicular to the bearing axis, due to the necessary tolerances between the shaft and the bearing.

Thrust bearings allow rotation of the shaft, but prevent the translation of the shaft along the axis of the shaft/bearing (see figure 1.13). Consider the hinges of a door. Hinges are constructed to act as a thrust bearing, preventing the door from sliding along the hinge pin and resting on the floor. Thrust bearings are found on all motor shafts, swiveling chair seats, "lazy susan" mechanisms and thousands of other places.

Just as for rotary motion, **linear motion** requires bearings to constrain the movement of the components to the proper direction. Numerous options exist. For precision motion, hardened steel rods or rails are often used in conjunction with either plain or linear ball bearings. Plain bearings are generally made of a smooth, low friction plastic such as nylon, Teflon, rulon or others. The McMaster Carr catalog lists many options. Look again in power transmission and find "Linear Bearings". Plain bearings are used when the loading is low and the speeds of travel are low to moderate. For high loads (i.e. heavy stuff) and rapid linear motion, ball bearings are available, in a number of configurations ranging from round shafts to keyed flat or rectangular slides. One of the biggest suppliers of linear bearings and guides is Thomson Industries (http://www.thomsonindustries.com/product_home.htm).

If extreme accuracy is not required, there are many simpler and cheaper solutions to constraining linear motion. From plastic guides on extruded aluminum beams to small casters riding in metal channel, myriad solutions can be found to guide linear motion. Examine everyday objects around you to see how linear motion is controlled. Look at drawer guides, garage door track, sliding van doors, whatever you can find that moves linearly, and see how it works. A company called 80-20 produces a line of products they call "The Industrial Erector Set" (http://www.8020.net).

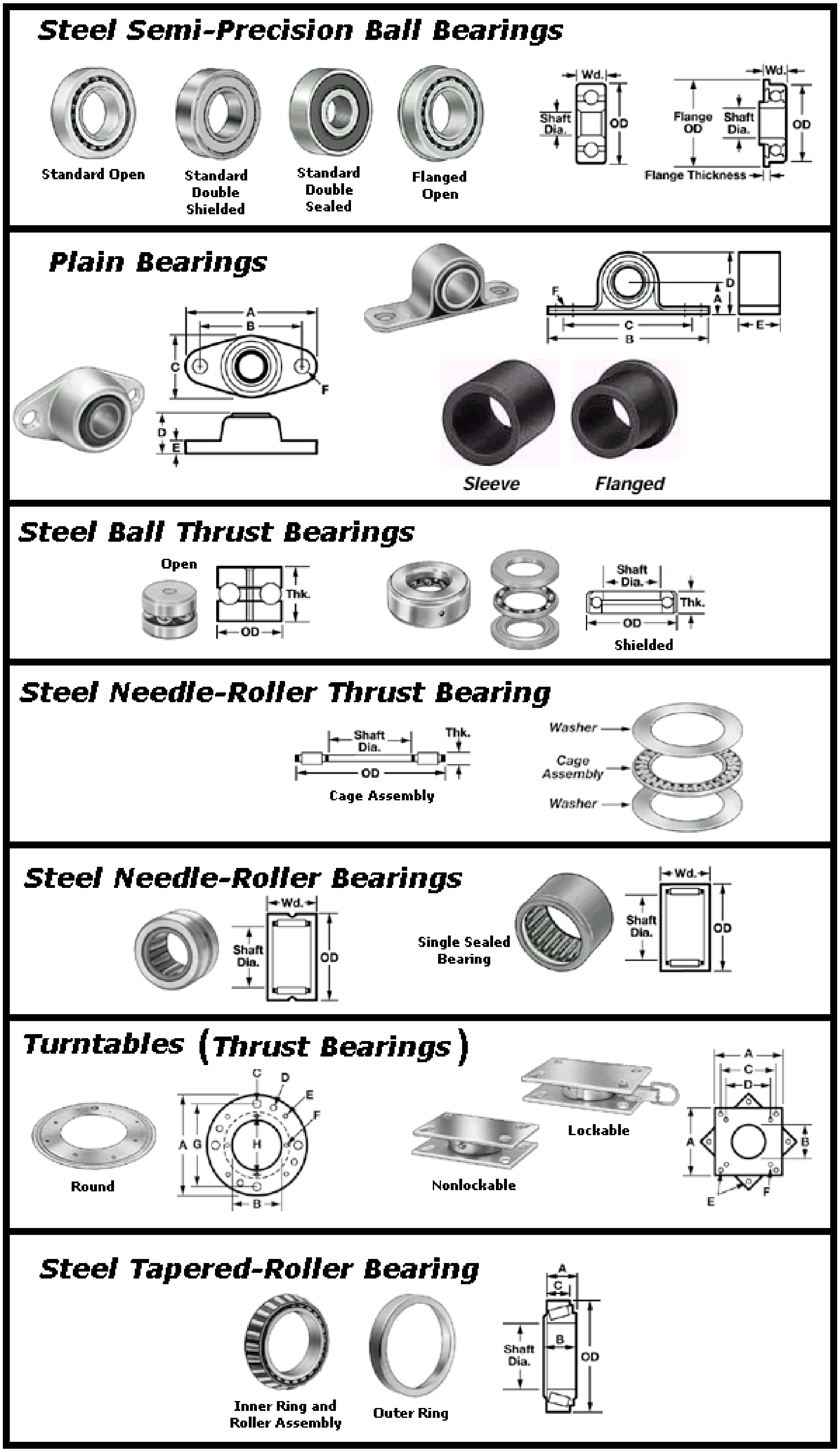


Figure 1.13: Common Bearings (found at: http://www.mcmaster.com/)

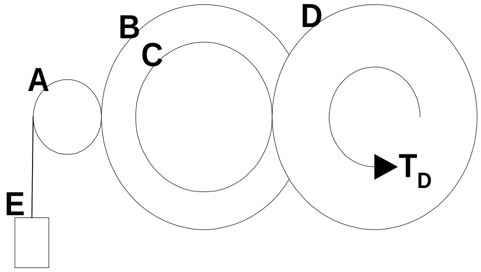
Homework

1. Gear A is a 12 pitch, 28-tooth gear. It needs to be meshed with Gear B, whose shaft is 3.0in from shaft B. How many teeth should Gear B have?

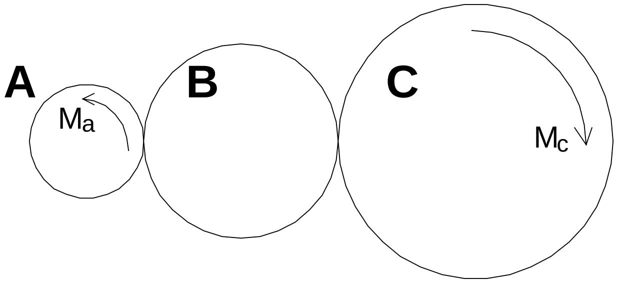
2. A 40-tooth gear and a 16-tooth gear are both 12 pitch and need to be meshed. How far apart should the shafts be?

3. If a 108-tooth gear and a 36-tooth gear are meshed together, with a shaft distance of 2.25in, what is the pitch of the gears?

4. A=8-teeth, B=40-teeth, C=24-teeth, and D=40-teeth. The pitch diameter of gear A is 8mm. Gears B and C are on the same shaft. If block E is 0.5kg, what is the output torque, T_D? If the instantaneous velocity of block E is 0.75m/s, what is the speed of Gear D in rpm's?

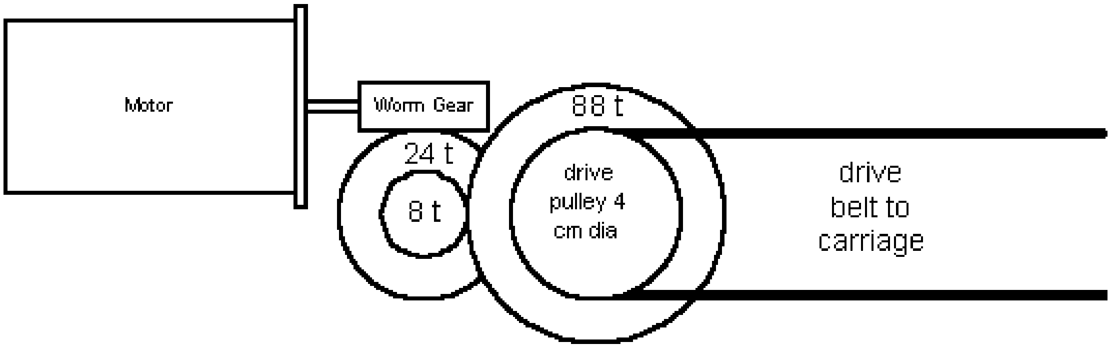


5. If the output torque M_c must be $3M_a$, how many teeth should Gears B and C have? Gear A has 26-teeth.



6. Set up a gear train, whose gear ratio is 2/15. You may choose from the following inventory of gears: one 8-tooth gear, one 16-tooth gear, one 24-tooth gear, one 40-tooth gear, and one 48 tooth gear. You do not have to use all of the gears.

7. If the motor in the diagram below turns at 3000 RPM, find the linear speed of the carriage being driven by the belt.



Chapter 2

Electricity

Basic Electricity Theory

Electricity is the flow of electrons along a conducting material. Electricity is used to produce heat, light, electromechanical motion, and to represent and manipulate information. There are very few mechanical systems that are free of some type of electric or electronic control.

Voltage (Volts)

Voltage is defined as the total work (measured in Joules) per unit charge associated with the motion of charge between two points: ***1 volt = 1 joule/couloumb.*** In more intuitive terms, Volts (V) are a measure of electrical potential or driving force. Using the fluid flow analogy, voltage is like water pressure.

Current (Amperes)

Amps are a measure of electric current. It is literally the number of electrons moving past a fixed point per unit time: ***1 amp = 1 coulomb/second.*** Because an ampere is a great deal of current, electronic circuits are commonly expressed in milliamperes. Using the fluid flow analogy, current is the number of molecules of water flowing past a fixed point per unit time.

Here is a glossary of other common terms that you will encounter:

Direct current, or DC, is the flow of electrons in one direction only. Examples of DC power are automobile batteries, flashlight batteries and the power supplies that you will be using in our laboratory.

Alternating current, or AC, is electron flow that reverses direction at regular intervals. For example, electricity from a standard wall outlet oscillates in a sinusoidal pattern with a frequency of 60 Hertz (cycles per second).

Conductors are materials that permit the relative free flow of electrons. Examples of conductors include copper, silver, gold, aluminum and other metals.

Insulators are materials that do not allow the flow of electrons. Examples of insulators are rubber, glass, ceramic, wood, paper, plastic and air.

Semiconductors are materials or combinations of materials that can act as an electrical conductor under certain conditions, but it is an insulator under other conditions.
Examples of semiconductor materials include: germanium, gallium arsenide, and specially treated silicon. We will explain semiconductors in greater detail in chapter 3.

A **circuit** is a closed path for electron flow consisting of a source of electrons and one or more components. A complete circuit, also called a closed circuit, is required for electricity to flow. A complete circuit is achieved if all of the elements in the circuit are connected by conductors in such a way that an electron can leave its source and flow through a continuous path back to the source. The source is usually a battery or a power supply, at least in the case of DC systems.

A **coulomb** (kōō lōm) is a unit of electrical charge. One electron has 1.60219 $*10^{-19}$ Coulombs. Therefore, 1 Coulomb of charge is approximately equal to the charge of 6.24 * 10^{18} electrons.

A **switch** is a convenient device for allowing a circuit to be either open or closed.

Open and Closed Circuits---Figure 2.1 illustrates the concepts of open and closed circuits, as one would experience them in a flashlight. When the switch is closed, the circuit is complete and electrons will flow from the source, through all of the elements in the circuit and back to the source. When the switch is open, no current can flow, and the bulb does not light. Note that it is the motion of the electrons and not just the electrons themselves that cause the bulb to light.

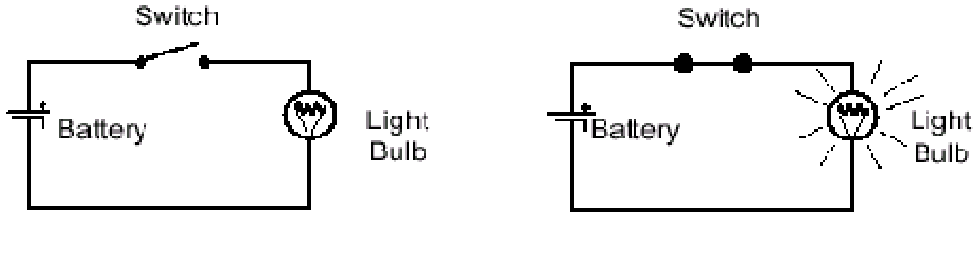


Figure 2.1: Open and closed (incomplete and complete) circuits.

A **short circuit** is a special kind of circuit that should generally be avoided. Suppose we connected the positive and negative terminals of a battery or power supply with a wire or some other good conductor. The current will be very high, heating up the wire, and the batteries. Eventually, either the wire or the batteries will melt, opening the circuit.

A **fuse** is a conducting material that will melt and open the circuit if too much current flows through it. Its purpose is to protect the other components in the circuit from receiving too much current.

A **node** is the junction of two or more electrical components.

Ground is a basic reference point in the circuit to connect to and for taking readings.

A **schematic** is a drawing of an electrical circuit using specific symbols for each type of circuit element. See figure 2.1 for an example of a schematic and figure 2.2 for a list of common symbols.

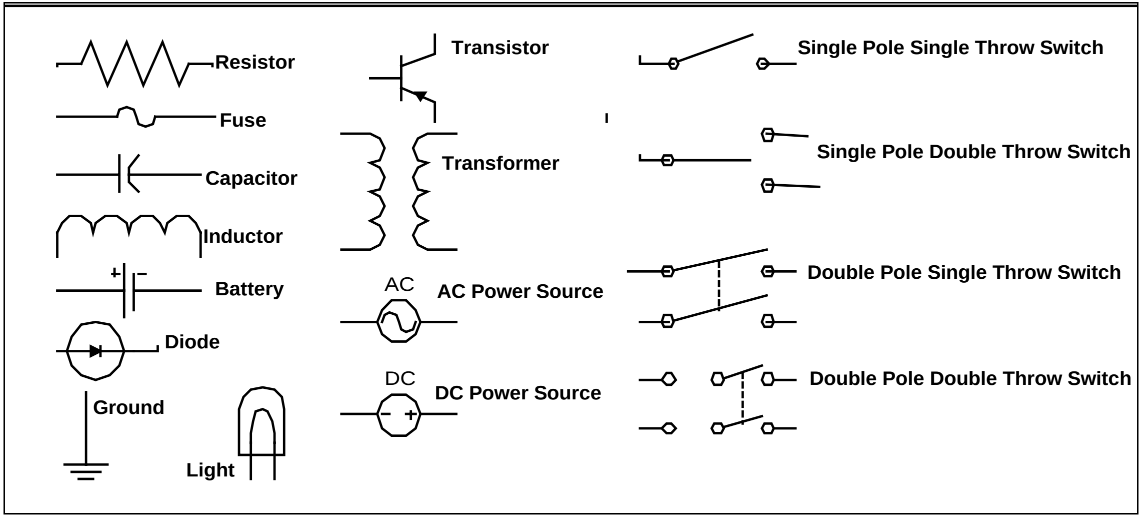

Resistor
Fuse
Capacitor
Inductor
+
−
Battery
Diode
Ground
Light
Transistor
Transformer
AC
AC Power Source
DC
DC Power Source
−
+
Single Pole Single Throw Switch
Single Pole Double Throw Switch
Double Pole Single Throw Switch
Double Pole Double Throw Switch

Figure 2.2 Common circuit element symbols

Series and Parallel---Resistors are said to be in series if they are in a line (sort of head to toe fashion). Resistors are said to be in parallel if all of the heads and all of the tails are connected together. See figure 2.3 for examples of series and parallel resistors.

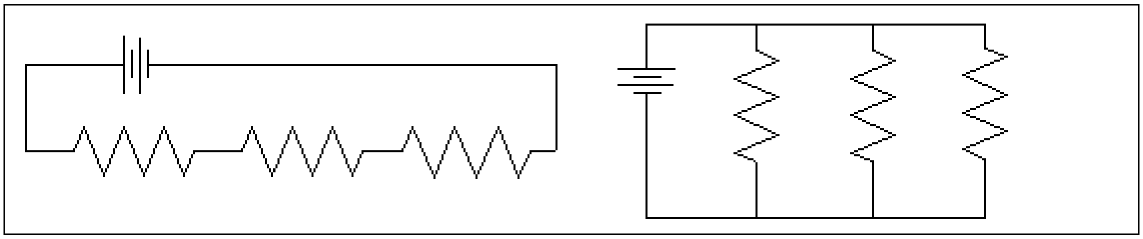


Figure 2.3 (Left) Resistors in series. (Right) Resistors in Parallel.

In a series circuit, the current is the same at every point in the circuit. Look at figure 2.5, Each resistor has the same number of electrons moving through it. There is no other path that they can take.

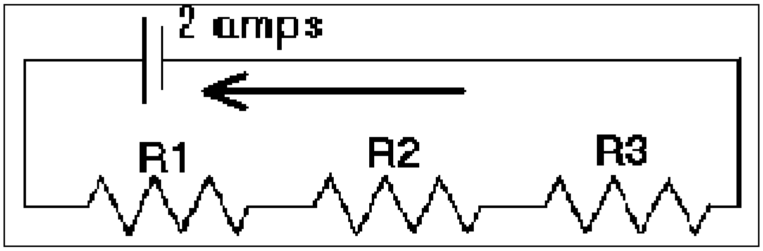


Figure 2.5: Current flow in a series circuit.

Resistance

The resistor is an electronic component that partially obstructs the flow of electrons. A byproduct of this process is that they create heat. Resistance is measured in **Ohms (Ω)**. The resistance of a component or object is a function of its composition and its size. Resistivity is measured in ohm-centimeters. For a given material, the electrical resistance of an object made from that material is given by:

$$\boldsymbol{R = \rho L/A} \qquad 2.1$$

Where p is the material resistivity (ohm-cm), L is the length of the object (cm) and A is the cross-sectional area in cm^2.

We will be using a few different kinds of resistors in our laboratory; besides fixed value resistors, we will also be using potentiometers and photo resistors. A potentiometer is a resistor whose resistance can vary with a knob setting. An example of a potentiometer is the volume knob on older stereos. A photo resistor is a resistor whose value changes with the amount of light it receives. With our photo resistors, an increase in light causes the resistance to decrease. Resistors with fixed values have these values encoded on their bodies as shown in figure 2.4.

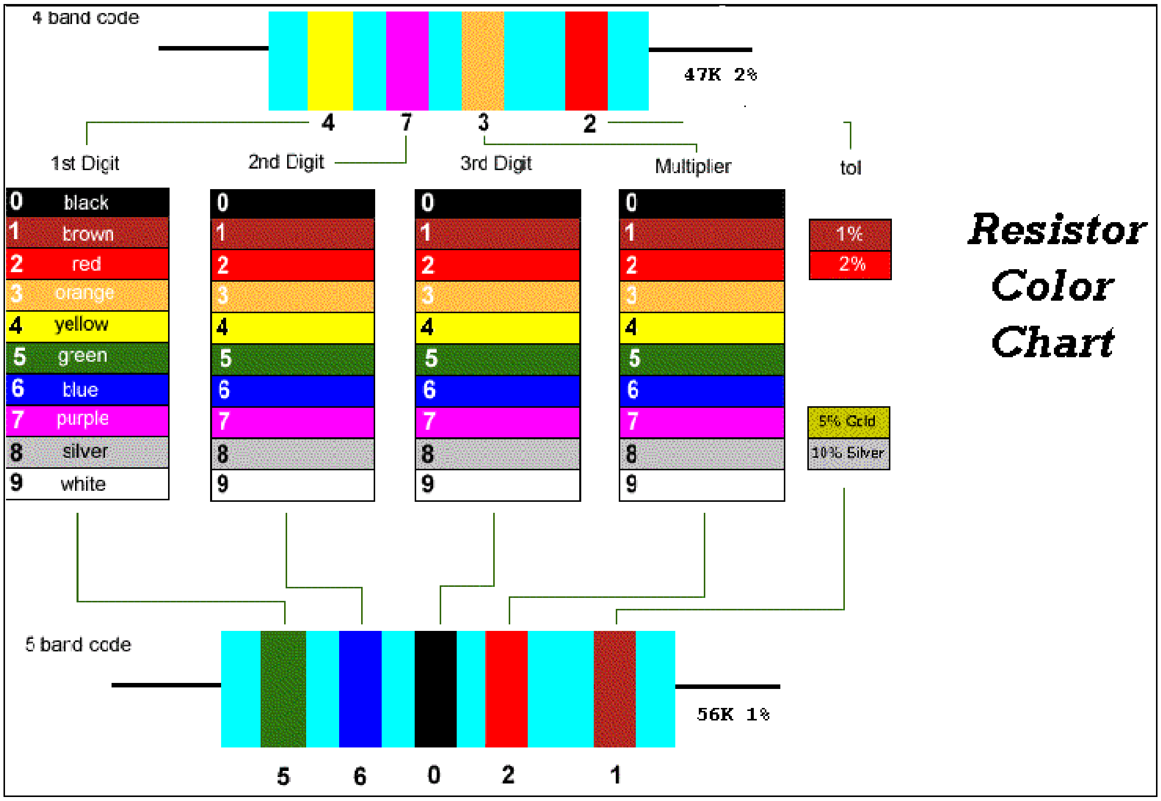
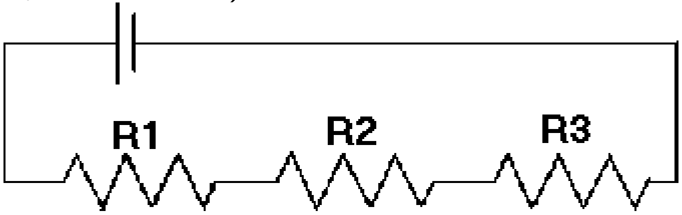


Figure 2.4 Resistor Color Chart

Equivalent Resistance

Any series and/or parallel combination of resistors can be reduced to a single, 'equivalent resistance' (R_{eq}). Replacing the combination by a single resistor with value R_{eq}, would not change the voltage or current distribution in the rest of the circuit. To find the equivalent resistance of a circuit, you need to look at series resistors and parallel resistors separately. Finding the equivalent resistance (R_{eq}) of resistors in series is a simple task; you just add the values of the resistors (see example 2.1).

Example 2.1: Calculate the total resistance of the circuit shown below. (R1 is 100Ω, R2 is 100Ω, and R3 is 200Ω.)



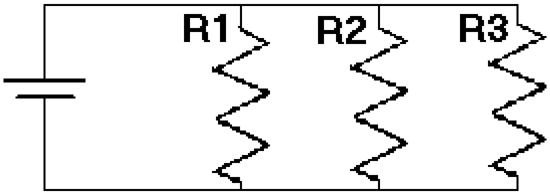
Solution: $R_{eq} = R1 + R2 + R3$

$R_{eq} = 100\Omega + 100\Omega + 200\Omega = 400\Omega$

To find the equivalent resistance of resistors in parallel, use the following formula (demonstrated in example 2.2):

$$\frac{1}{\mathrm{R\ eq}} = \frac{1}{\mathrm{R1}} + \frac{1}{R2} + \frac{1}{\mathrm{R3}} \ldots \qquad 2.2$$

Example 2.2: Calculate the total resistance of the circuit shown below. (R1 is 100Ω, R2 is 100Ω, and R3 is 200Ω.)



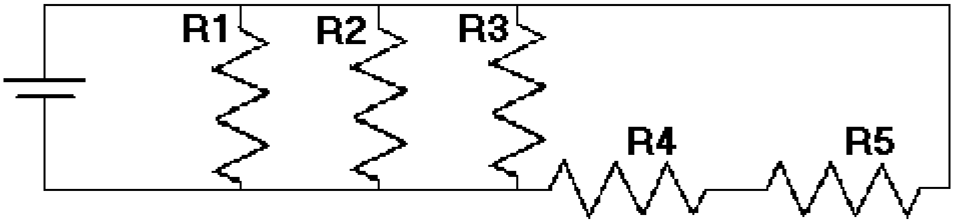
Solution: $1/R_{eq} = 1/R1 + 1/R2 + 1/R3$

$1/R_{eq} = 1/100\Omega + 1/100\Omega + 1/200\Omega = .025$

$R_{eq} = 40\Omega$

When calculating the equivalent resistance of a combination of series and parallel resistors, it is generally easier to start on the outside and work your way in. See example 2.3.

Example 2.3: Calculate the total resistance of the circuit shown below. (R1 is 100Ω, R2 is 100Ω, R3 is 200Ω, R4 is 300Ω, and R5 is 200Ω.)



Solution: First compute the equivalent resistance of the series resistors (R4 and R5)

$R4 + R5 = 300\Omega + 200\Omega = 500\Omega = R45$

Now, Compute the parallel resistance.

$1/R_{eq} = 1/R1 + 1/R2 + 1/R3 + 1/(R45)$

$1/R_{eq} = 1/100\Omega + 1/100\Omega + 1/200\Omega + 1/500\Omega = .027$

$R_{eq} = 37.04\Omega$

Basic Electricity Calculations

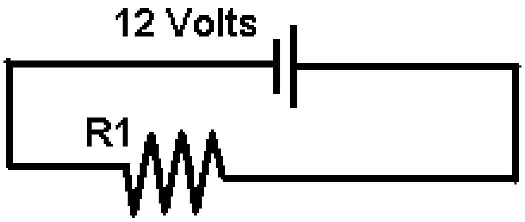
Ohm’s Law

Ohm’s law states that there is a linear relationship between voltage across a resistor and the current that flows in it and current, based on the resistance of the circuit. You must become comfortable with each variation of the following formula:

V = I * R $\mathrm{I} = \frac{\mathrm{V}}{\mathrm{R}}$ $R = \frac{\mathrm{V}}{\mathrm{I}}$ 2.3

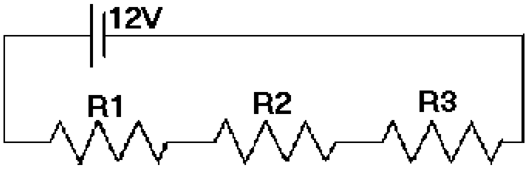
Where v is voltage, I is current, and R is resistance. When solving problems, there is generally several ways to apply ohm’s law to get the same result (see examples 2.4, 2.5, and 2.6).

Example 2.4: Calculate the current at R1. (R1 is 200Ω.)



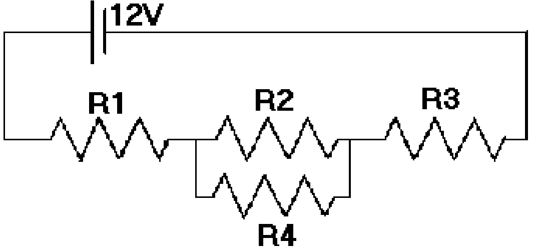
Solution: You can use ohm’s law: ***I = V/R*** *= 12 volts / 200Ω = 60 milliamps.*

Example 2.5: Calculate the current at R1. (R1 is 100Ω, R2 is 100Ω, and R3 is 200Ω.)



Solution: You know that R_{eq} of the circuit is 400Ω. (That’s 100 +100+200)
So, you can use ohm’s law: ***I = V/R*** *= 12 volts / 400Ω = 30 milliamps.*

Example 2.6: Calculate the current at each of the resistors. (R1 is 100Ω, R2 is 100Ω, R3 is 200Ω, and R4 is 300Ω.)



Solution: First, find the equivalent resistance of R2 and R4.

$1/R_{eq}$ = 1/R2 + 1/R4 = 1/100 +1 / 300 R_{eq} = 75Ω

*Voltage across R2 = [75Ω / (100Ω+75Ω+200Ω)] * 12 Volts = 2.4 Volts*

Voltage across R4 = Voltage across R2 = 2.4 Volts

Now apply ohm’s law to R2 and R4:

Ohm’s law at R2: ***I = V/R*** *= 2.4 volts / 100Ω = 24 milliamps*

Ohm’s law at R4: ***I = V/R*** *= 2.4 volts / 300Ω = 8 milliamps*

Now, (I at R1) = (I at R3) = (I at R2 + I at R4) = (24 milliamps + 8 milliamps) = 32 milliamps.

Alternate Solution: To find I at R1 and R3, you could also have found the equivalent resistance of the entire circuit.

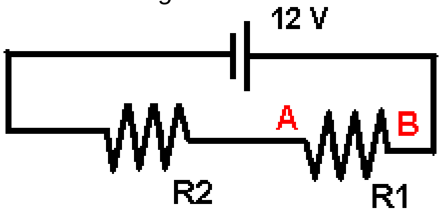
$$R_{eq} = R1 + [1/(1/R2 + 1/R4)] + R3 = 100 + [1/(1/100 + 1/300)] + 200 = 375\Omega$$

Then applied ohm's law to the entire circuit
I= V/R = 12V/375Ω = 32 milliamps *So the current at R1 and R3 is 32 milliamps.*

Kirchoff's Voltage Law (KVL)

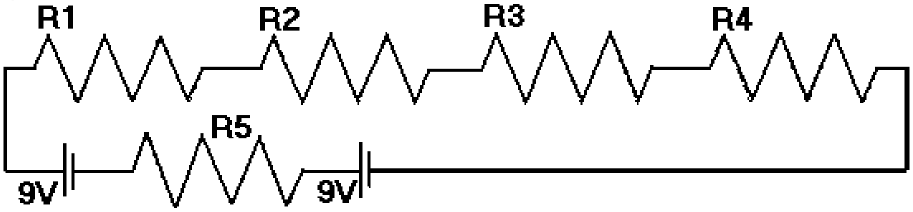
Kirchoff's voltage law states that the sum of all source voltages must be equal to the sum of all load voltages, so that the net voltage is 0. A source voltage means a battery or other power source. The load voltages are voltage changes across each of the resistors (or other components) in the circuit. Look at examples 2.7 and 2.8, basically go around the circuit adding the battery voltages and subtracting the resistor voltages and this must equal 0.

Example 2.7: Look at the schematic, what is the voltage change from point A to point B? You know other voltage losses as follows: R2 (5 volts).



Solution: 12 volts – 5 volts = 0 *R1 = 7 volts*

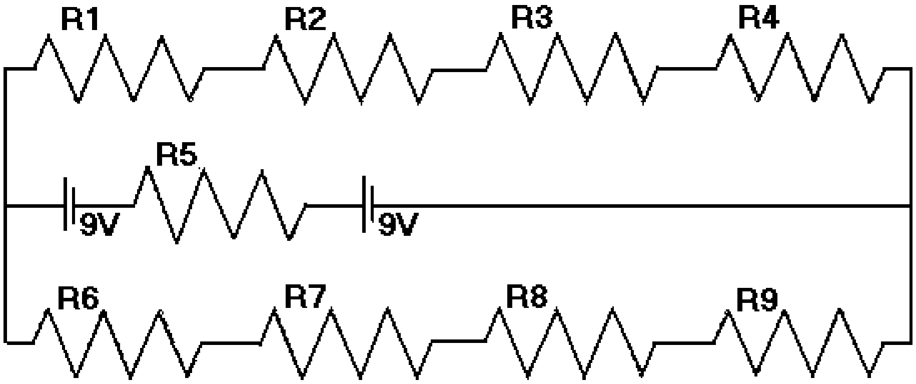
Example 2.8: Look at the schematic, what is the voltage change from point A to point B? You know other voltage losses as follows: R1 (3 volts), R2 (5 volts), R4 (2.75 volts), and R5 (6 volts).



Solution: 9 volts – 3 volts –5 volts –R3 – 2.75 volts + 9 volts –6 volts = 0
R3 = 1.25 volts

Example 2.9 is a little trickier. You must apply KVL twice to solve this problem.

Example 2.9: Look at the schematic, what is the voltage change at R3? You know other voltage losses are as follows: R1 (5 volts), R2 (5 volts), R4 (2.75 volts), R6 (5 volts), R7 (3 volts), R8 (1.25 volts), and R9 (4.5 volts).



Solution:
First, you need to apply KVL to the bottom loop to find the voltage change at R5.

9 – R5 + 9 – 4.5 – 1.25 – 3 – 5 = 0
R5 = 4.25 Volts dropped

Now, you can apply KVL to the top loop to find the voltage change at R3.
9 – 4.25 + 9 –5 – 5 – R3 – 2.75 = 0
R3 = 1 Volts dropped

Kirchoff's Current Law (KCL)

Kirchoff's current law states that the sum of all currents entering a node must equal the sum of all currents leaving the node. Look at figure 2.6, i_3 must equal 20 mA because the current entering node A (50 mA) must equal the current leaving node A (30 mA + i_3).

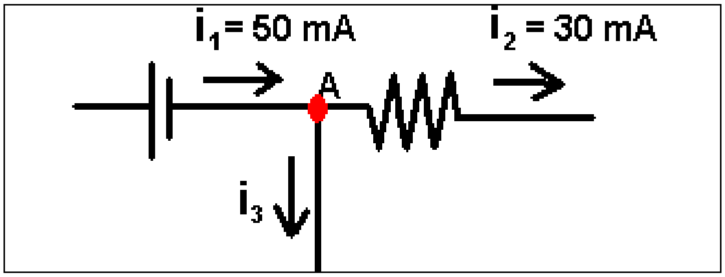


Figure 2.6: Partial schematic showing node A.

Voltage Divider Rule

A voltage divider is just a combination of two or more series resistors across a voltage source. Consider the following circuit (figure 2.7):

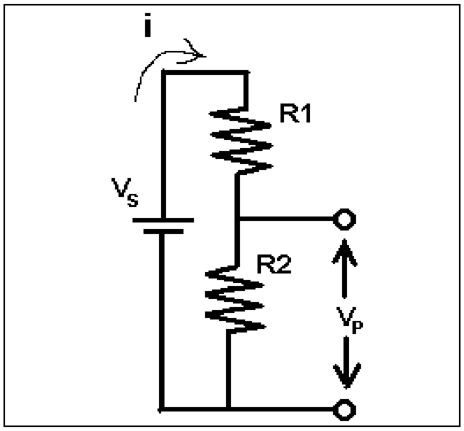


Figure 2.7: Voltage divider circuit

It stands to reason that V_P will be less than V_S but more than zero (provided R2 > 0). V_P should be a function of R1 + R2. From ohm's law we know that:

$$V_P = I * R2 \tag*{2.4}$$

From Kirchoff's law we know that:

$$V_S = I\ (R1 + R2) \tag*{2.5}$$

Then:

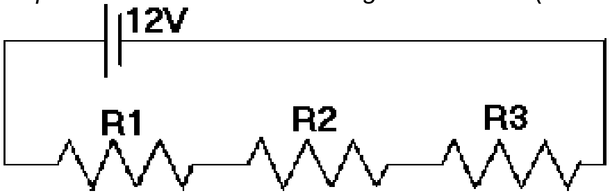
$$I = V_S / (R1 + R2) \tag*{2.6}$$

Which can be substituted into equation 2.6 to get:

$$V_P = [R2 / (R1 + R2)] * V_S \tag*{2.7}$$

See example 2.10 for an example.

Example 2.10: Calculate the voltage across R1. (R1 is 100Ω, R2 is 100Ω, and R3 is 200Ω.)

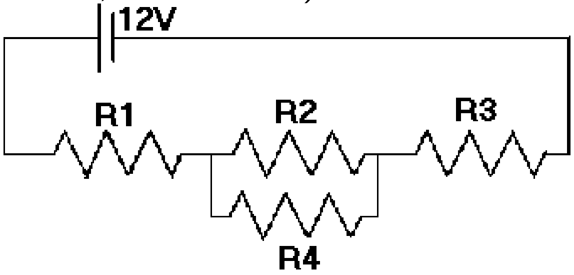


Solution: For this problem, you can use the voltage divider rule:

*Voltage across R1 = [100Ω / (100Ω+100Ω+200Ω)] * 12 Volts*
Voltage across R1 = 3 Volts

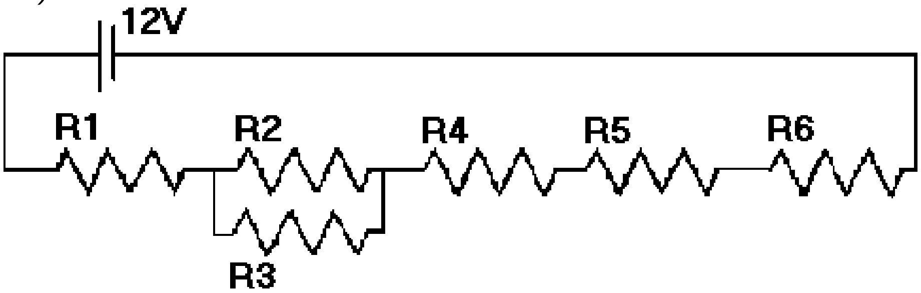
Any circuit can be thought of as a voltage divider, just look at it as a series circuit but find the equivalent resistance of any parallel components first (See examples 2.11 and 2.12).

Example 2.11: Calculate the voltage across each of the resistors. (R1 is 100Ω, R2 is 100Ω, R3 is 200Ω, and R4 is 300Ω.)



Solution: First, find the equivalent resistance of R2 and R4.
$1/R_{eq}$ = 1/R2 + 1/R4 = 1/100 +1 / 300
R_{eq} = 75Ω
Now, the voltage divider rule can be used:
*Voltage across R1 = [100Ω / (100Ω+75Ω+200Ω)] * 12 Volts*
Voltage across R1 = 3.2 Volts
*Voltage across R2 = [75Ω / (100Ω+75Ω+200Ω)] * 12 Volts*
Voltage across R2 = 2.4 Volts
*Voltage across R3 = [200Ω / (100Ω+75Ω+200Ω)] * 12 Volts*
Voltage across R3 = 6.4 Volts
Voltage across R4 = Voltage across R2 because voltage is common in parallel
Voltage across R4 = 2.4 Volts

Example 2.12: Calculate the voltage across the combined resistors (R4 & R5) in circuit shown below. (R1 is 100Ω, R2 is 100Ω, R3 is 200Ω, R4 is 300Ω, R5 is 200Ω, and R6 is 150Ω.)



Solution: First compute the equivalent resistance of the parallel resistors (R2 and R3)
$1/R_{eq}$ =1/R2 + 1/R3
R_{eq} = 66.7Ω
Now, use the voltage divider rule

*Voltage across (R4&R5) = [(200Ω+300Ω) / (100Ω+66.7Ω+300Ω+200Ω+150Ω)] * 12 Volts*
Voltage across R(4&5) = 7.35 Volts

Power

Watts (W) is a measure of power. Power is work done per unit time. It can be represented by the following relationship:

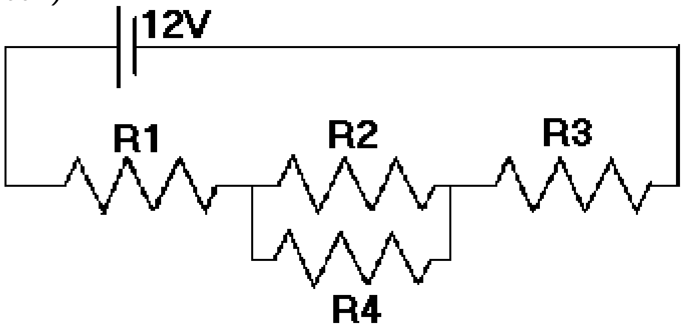
Power=Voltage*Current 2.8

Watts is a measure of the total amount of electricity used. A common use of this measure is that the electric company's bill is based on the number of watts used per month. Look at examples 2.13 and 2.14.

Example 2.13: Suppose I have a 60-Watt light bulb in my desk lamp. A standard wall outlet produces 110 volts. How much current does it use?

*Solution: Power=Voltage*Current 60 Watts = 110 volts * I*
I = .545 amps

Example 2.14: Find the power used by this circuit. (R1 is 100Ω, R2 is 100Ω, R3 is 200Ω, R4 is 300Ω).



Solution: First find the current from the battery.
R_{eq} = 375Ω *I = V/R*
I =12V / 375Ω = 32mA

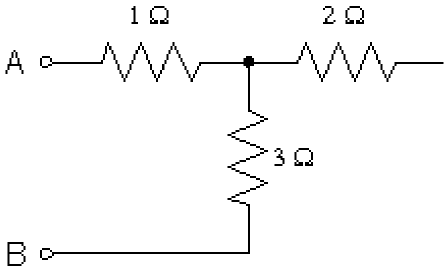
*Now, power = voltage * current*
*Power = 12V * .032A = .384 Watts*

Chapter 2 Homework Set

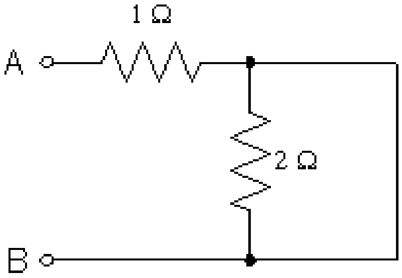
Show work and include proper units with your answers. Round answers to 3 significant digits.

#1. Determine the resistance between A and B for the following figures:

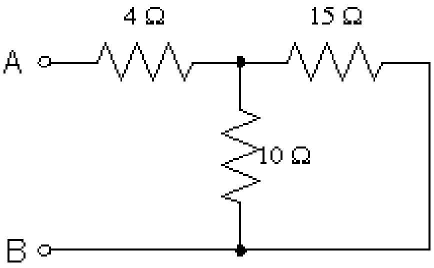
1a) __________



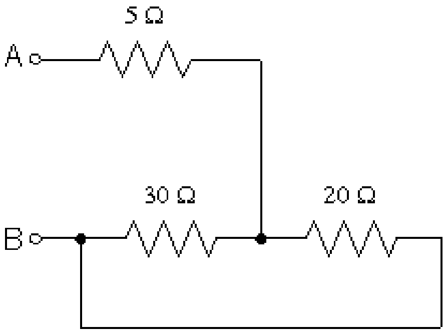
1b) __________



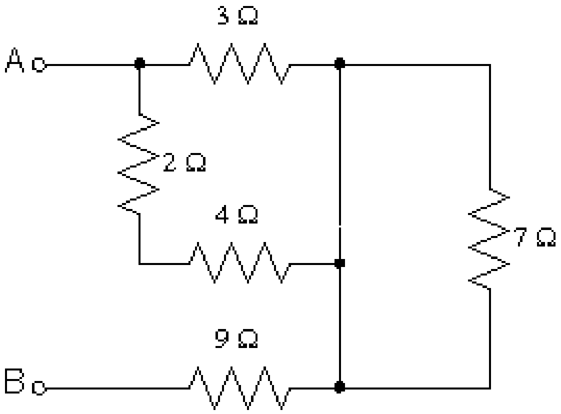
1c) ___________

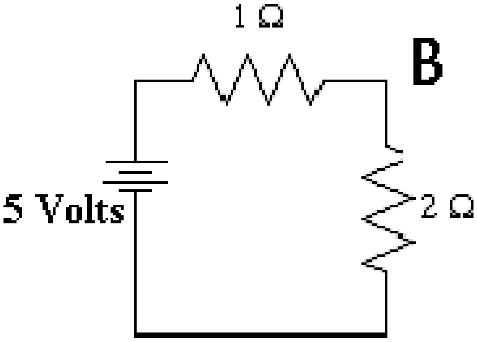


1d) ___________



1e) ___________





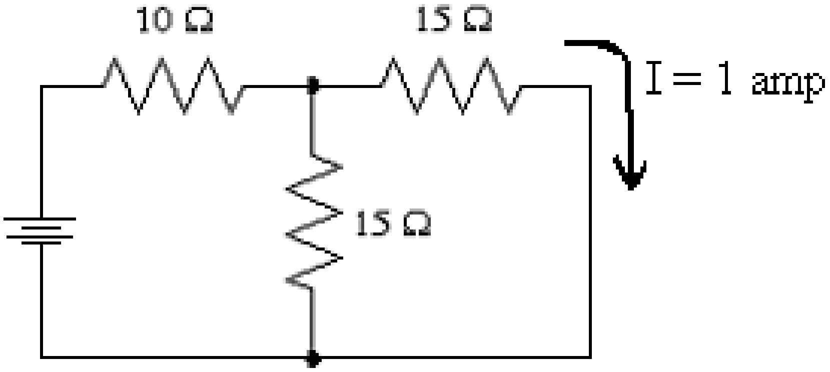
2) Use Ohm's Law to solve these:

2a) Current in circuit? _____________

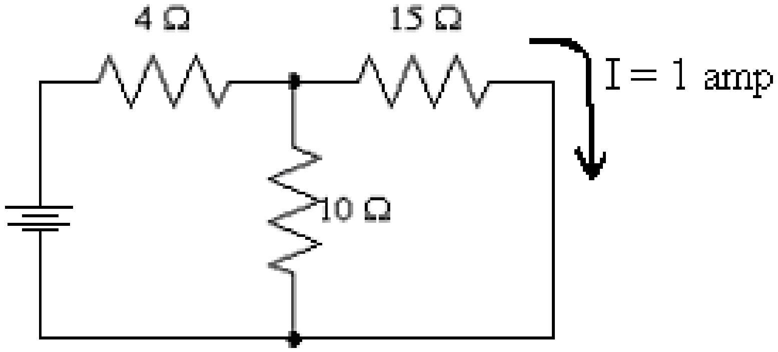
2b) Voltage at point B ______________

3) Determine the voltage of the battery in the following:

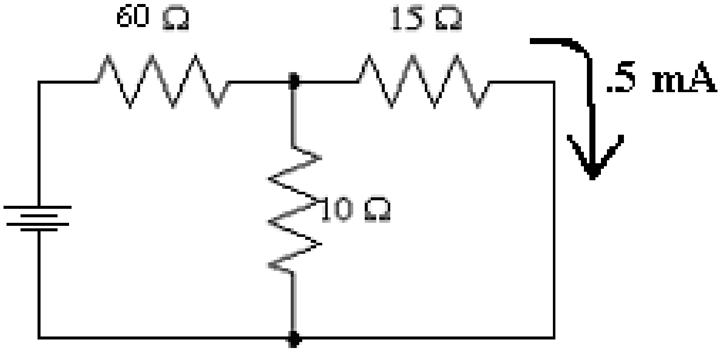
3a) ____________



3b) ____________

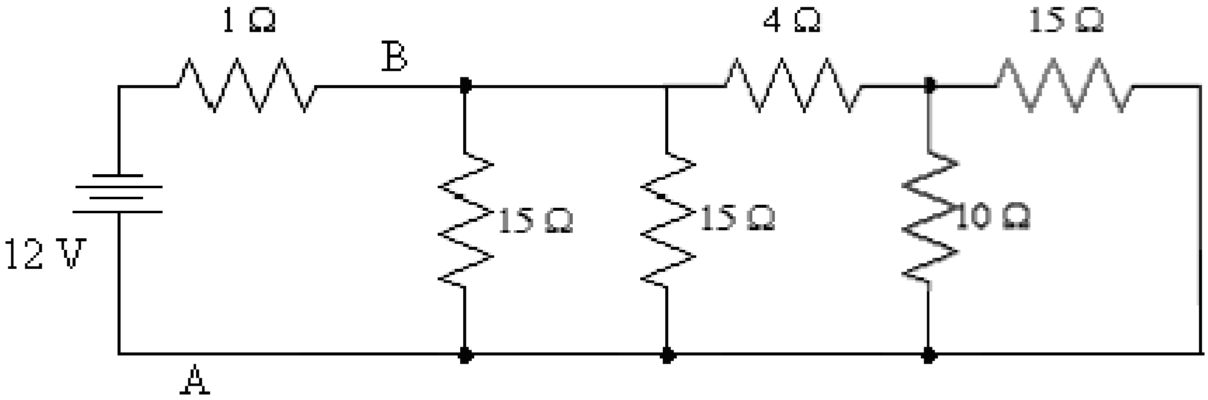


3c) ____________

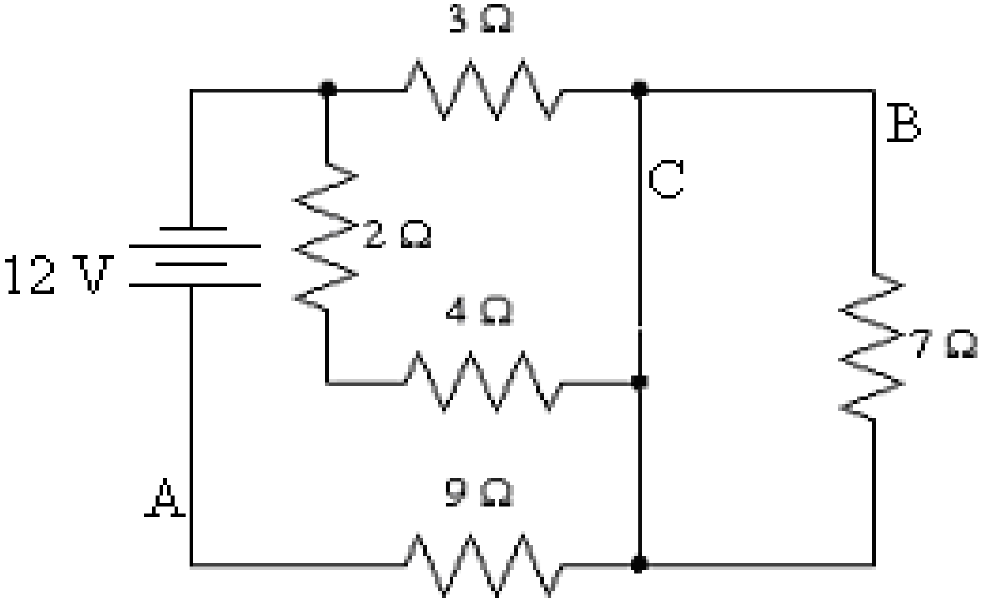


4) Determine the current at the given points for each of the following:

4a) A=_________ B = ___________



4b) A = _________ B = __________ C = __________



Chapter 3

Electrical Components

Electricity and Magnetism

In the 1820s, it was found that an electric current could be produced in a circuit by moving a magnet near the circuit or by changing the current in a nearby circuit. This means that a changing magnetic field creates an electric field. Later, it was found that the reverse was also true, that a changing electric field creates a magnetic field. Have you ever made one of these electromagnets with a battery, piece of wire, and a nail or screwdriver? Try it at home, it works-see figure 3.1. Use small gage wire and many turns around the nail so you don't short circuit the battery.

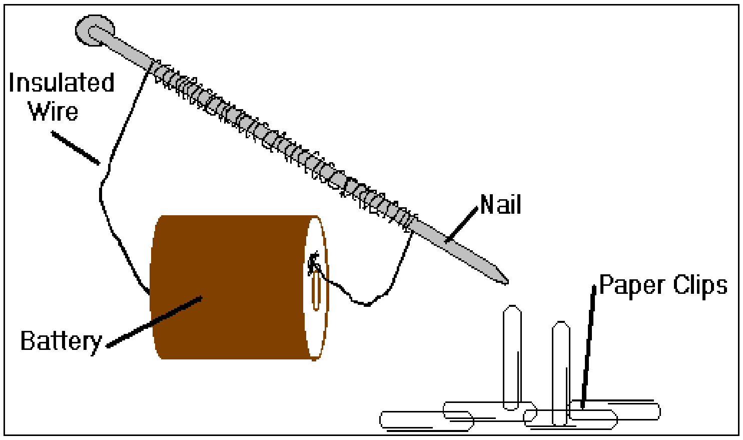


Figure 3.1: Try this at home: a simple electromagnet.

The coil and iron core setup is the basis of many electrical devises that we use today, such as speakers, microphones, inductors, relays, motors, generators and transformers. Here is a glossary listing of several electrical components:

A **speaker** has a large diaphragm attached to the light weight coil. The coil is free to move so that changes in voltage in the coil cause it to vibrate, which causes the diaphragm to vibrate which produces the sound waves.

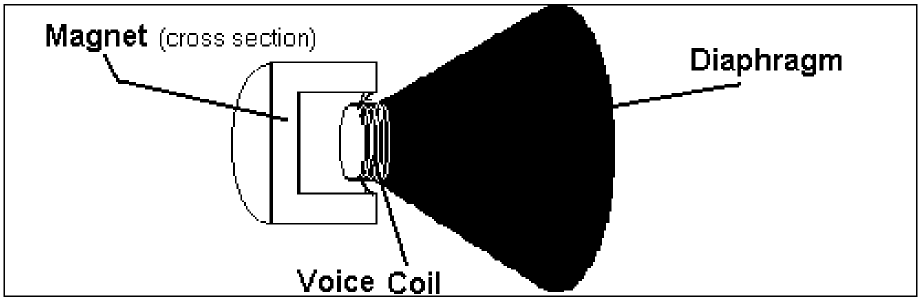


Figure 3.2: Parts of a common speaker.

One type of **microphone** has a small diaphragm attached to the coil. It works exactly opposite a speaker. Sound waves cause the diaphragm to vibrate which causes the iron core to vibrate. These vibrations can be read as voltage changes in the coil. Other microphones work on changing composition or resistance.

A **motor** converts electricity into rotary mechanical motion. For a good description of how the DC motor works go to http://www.howstuffworks.com/inside-motor.htm.

A **generator** converts rotary mechanical motion into electricity. It is very nearly opposite of how a motor works. All automobiles have a kind of generator called an alternator.

An **inductor** is a coil of wire, and may have an air core or a metallic core. The electromagnets, speaker coils and many other devises are inductors. Inductors present a low 'resistance' to direct current but a high resistance to changes in current. The voltage across an inductor is given by:

$$V_L = L * (di / dt) \qquad 3.1$$

where i is the current through the device, t is time, and L is the inductance. Inductance is determined by the core type and the number of turns of wire. For high frequency or rapidly changing signals, (di/dt) is very large and the inductor 'blocks' these signals. For DC, (di/dt) is zero and the voltage drop is due only to the resistance of the wire used in the coil. A good description of inductors can also be found at: http://www.howstuffworks.com/inductor.htm

A **transformer** is a devise that couples two AC circuits magnetically rather than through any direct conductive connection and permits a "transformation" of the voltage and current between one circuit and the other circuit (Note: Transformers do not work for DC!). It consists of an iron core and two coils around it. One coil, the primary, is connected to a power source and the second coil, called the secondary, becomes the power source for the circuit. Step-up transformers step-up (increase) voltage at the expense of decreasing current. Step-down transformers decrease voltage and increase current. These are determined by the ratio of the number of turns(wraps) if the secondary by the number of turns in the primary:

$$\text{Turns Ratio (N)} = \frac{\text{\# turns in secondary (n2)}}{\text{\# turns in primary (n1)}} \qquad 3.2$$

such that:

V_2 (voltage in the secondary) = N * V_1 (voltage in the primary) 3.3

and:

I_1 (current in the primary) = N * I_2 (current in the secondary) 3.4

Example 3.1: Suppose I have a 120 volt power supply and a transformer with 300 turns in the primary and 100 in the secondary. What is the output voltage? What is the current in the secondary if the primary has .5 amps?

Solution:

N = 100 / 300 = .333

*V2 = 120 volts * .333 = 40 volts*

$I_1 = N * I_2$

$.5 \text{ amps} = .333 * I_2$

$I_2 = 1.5 \text{ amps}$

A **relay** is an electromechanical switch. It uses an electromagnet to close a mechanical switch. Figure 3.4 is an example of a double pole double throw relay. The 'throw' is the number of switches the relay has. The 'pole' is the number of contact points for each throw. The relay allows small voltage and current to control the switching of a much larger amount of power. An example of a use for a relay is the ignition of your car. The small wires in your steering column trip a relay that is connected to the huge battery cables.

A **capacitor** is an electrical component that consists of two parallel plates that have a very small gap between them. It can be used as a filter because it blocks DC, but allows AC to pass. It can also be used to temporarily store or build up electrical current.

A **solenoid valves** take electrical signals as inputs and control the flow of fluids as outputs. They are most commonly used to control the flow of high pressure hydraulic oil or compressed air, but can also be used for other liquids or gases, such as water, fuel oil, steam, refrigerant, natural gas, etc.

Stepper Motors

Stepper motors are a special kind of motor designed to move in discrete steps. This can perhaps best be understood by looking at how they are designed. While there are many variations, you will get the general idea by studying a simple type of stepper called the permanent magnet stepper.

Consider the diagram below (figure 3.11). The stepper ***rotor*** is the moving part attached to the shaft. It is a permanent magnet with North (N) and South (S) poles. The stepper ***stator*** surrounds the rotor and is the stationary part of the motor. It consists of several coils of wire wound around iron laminations, which make up electromagnets that can be turned on and off. Recall that like magnetic poles (i.e. N-N and S-S) repel each other while opposite poles (N-S) attract. The magnetic polarity of the stator magnets can be controlled, by controlling the current direction through the coils.

Let us assume that our stepper is set up so that when we turn a coil "on" it generates a magnetic N pole closest to the rotor. Then, if we turn on coil A and leave all of the others off, it is clear that the rotor will try to line itself up with its S pole aligned with coil A, as shown in Figure 3.10 Left. If we leave coil A on, the rotor will come to rest in alignment with coil A and will not move farther. In fact, it will vigorously resist any attempt to manually move it from this position. The amount of external torque the motor can resist is called its ***holding torque.***

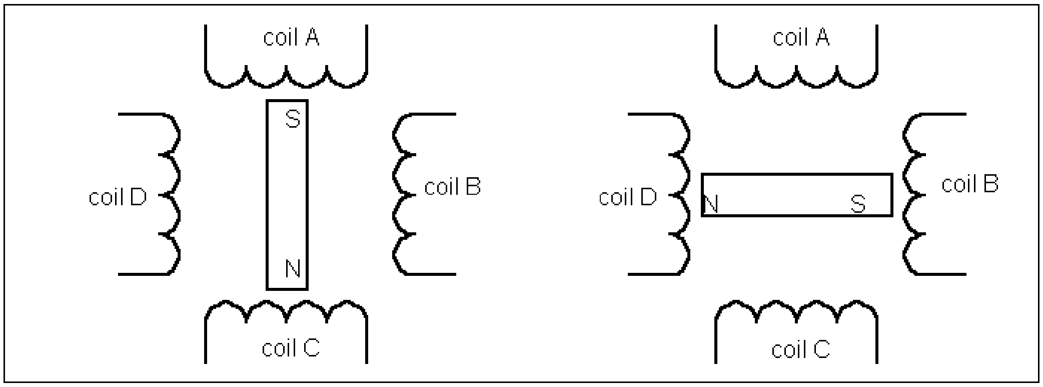


Figure 3.10: Left: Rotor aligned with coil A Right: Rotor aligned with coil B

If we then turn off coil A and turn on coil B, the rotor will turn ¼ rotation to the right and align with coil B, as shown in Figure 3.10 Right. Continuing to rotate the magnetic field around the circle will cause the rotor to align next with coil C then coil D, etc. Each of these locations is a stable position. That is, as long as one of the coils is energized, the rotor will attempt to lock itself in alignment with that coil.

Stepper Resolution

The motor just described has only four stable positions, when operated in the full step mode. It is easy to envision a need to have more options than this, and steppers are available with many more than four steps per revolution. The ones you will be using in your lab have 100 steps per revolution, for example. Many of the variations on the design of the stepper have been created with an eye to improving the resolution and torque/speed characteristics of the motors. The invention of solid state electronic stepper controllers has made it possible to generate literally thousands of steps per revolution from relatively simple stepper motors. One method for doing this is called ***half stepping*** and can be done using your motors. Look again at Figure 1, and imagine that we control the coils in the following way:

1. Energize coil A and wait for the motor to come to equilibrium in alignment with coil A.

2. Without turning off coil A, energize coil B. Assuming that the two coils generate equally powerful magnetic fields, the rotor will come to rest halfway between coils A and B. It has moved exactly ½ step.

3. Now turn off coil A, leaving only coil B energized, and the rotor will move on to align itself with coil B.

Microstepping

The same idea can be extended by the use of electronic current controls. Suppose instead of having equal currents flowing in coil A and coil B, that the current in coil A were twice as large as the current in coil B. Now the rotor would come to rest a little more than ¼ of the way between A and B ($\tan^{-1}(0.5) = 26.5$ degrees). By controlling the relative strengths of the currents between two adjacent coils, the rotor can be made to move in even smaller increments than the half steps described above.

Unipolar/Bipolar Windings

The motors you will use are unipolar. That means that you only have access to one end of each winding individually. Schematically, the motor windings look like Figure (3.11 Right), with one wire being the common lead for all of the windings. It is possible using split power supplies and special controllers to control the direction of current flow through these motors, but nobody would bother with that. In a unipolar motor, the positive supply voltage is usually fed to the common wire, and each of the windings is then connected to a driver transistor, which acts as an on/off switch.

Bipolar motors are available which provide access to both ends of each winding individually. These motors allow relatively simple H-bridge controllers to switch the direction of the current through the windings, allowing even more flexibility in the control of the motor, and offering the possibility of achieving higher torque from a given frame size. Returning to Figure 3.10, suppose that while we were energizing coil A to be a North pole, we reversed the current in coil C and caused it to be a South pole. Depending on the geometry of the motor and the state of magnetic saturation of the rotor iron, additional torque might be derived from the motor in this way.

Winding Geometries

Unipolar motors such as yours accomplish high resolutions by incorporating multiple A,B,C and D poles for each winding. Figure (3.11 Left) shows symbolically how this works. In reality, the stator laminations have complicated shapes, which generate the multiple magnetic "poles". The point is, when the A coils are energized, there are many stable states to which the rotor can move. It merely "selects" the nearest one. The different windings are often stacked along the axis of the motor as shown schematically in Figure (3.11 Middle).

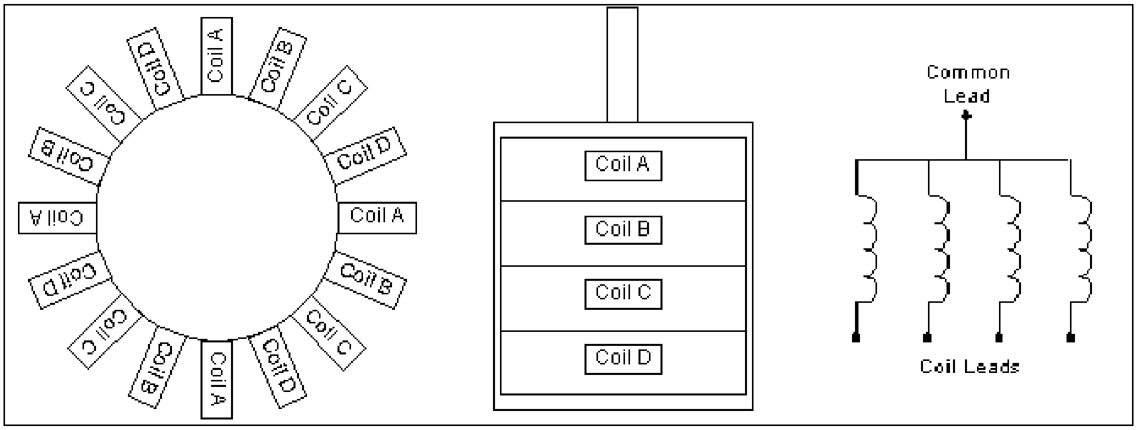


Figure 3.11: Stepper motor wiring.
Left: Conceptual diagram of multiple poles for each winding.
Middle: Stator coils sometimes are stacked along the motor axis.
Right: Motor wiring diagram.

Semiconductors

Semiconductors are materials or combinations of materials that can act as an electrical conductor under certain conditions and an insulator under other conditions. Examples of semiconductor materials include: germanium, gallium arsenide, and specially treated silicon.

Some semiconductors contain combinations of 'n-type' and' p-type' materials. These materials are produced by chemically treating or 'doping' them. Doping consists of adding a precise amount of impurities to the crystalline structure of the semiconductor. When used in combination, these materials make diodes and transistors. In your EE courses you will learn why they work. For now, it is important to understand their uses:

A **diode** is a semiconductor device that allows current to flow in one direction only, from the anode to the cathode (see figure 3.3). It has one section of n-type material and one section of p-type material. The primary application of the diode is to convert alternating current into direct current.

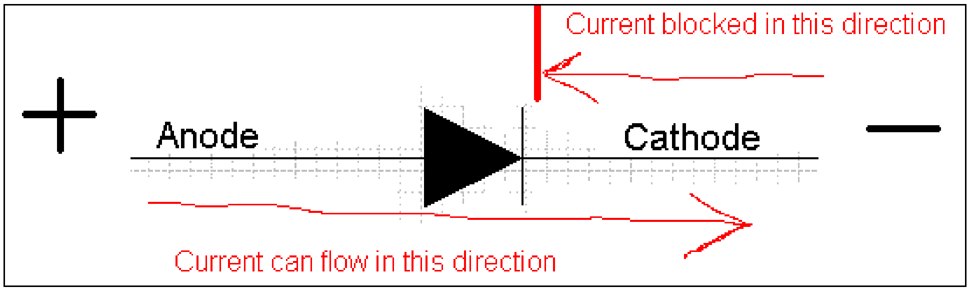

Current blocked in this direction
+
Anode
Cathode
—
Current can flow in this direction

Figure 3.3: Current flow at diode.

Unlike the resistor, the relationship between current and voltage for the diode is nonlinear (Ohm's Law does not apply to diodes!). Figure 3.4 shows the IV plot for a typical diode. As 'forward' voltage is applied, no current flows until the 'conduction voltage V_c is reached. For silicon diodes, V_c is typically about 0.7V but can be in excess of 1.0V for specialty diodes and LEDs. Germanium diodes have a lower V_c, typically 0.4V or less.

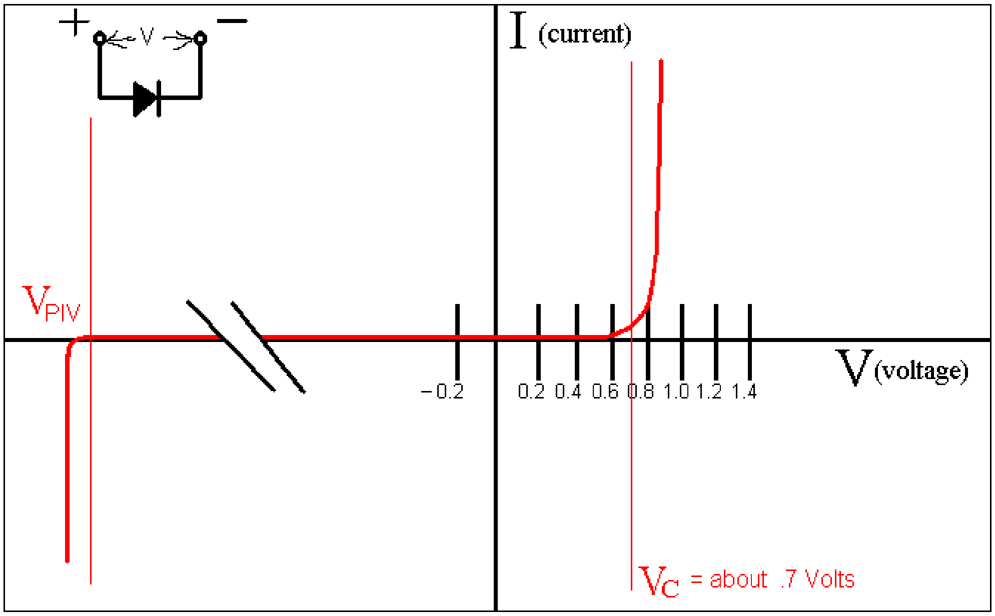
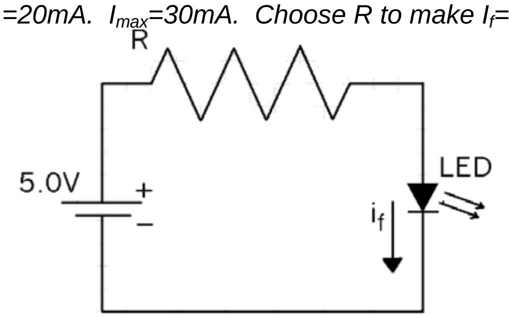


Figure 3.4: IV plot for typical diode.

Once V_c is reached, the diode becomes an almost perfect conductor, and the forward current I_f rises dramatically until the current capacity of the device (I_{max}) is reached. If I_{max} is exceeded, the diode will be destroyed. For this reason, it is critical to have a current-limiting element (like a resistor) in series with a diode. The value of the resistor is determined by selecting the desired I_f and using the Kirchoff and Ohm's Law equations (see example 3.2).

Example 3.2: According to the manufacturer's specs, the LED has a V_f of 1.0V when I_f =20mA. I_{max}=30mA. Choose R to make I_f=20mA.



Solution: From Kirchoff,

$$0 = V_B + V_R + V_{LED}$$

$$0 = 5V + (--I_fR) - 1.0V$$

$$R = 4V/20mA = 200\Omega$$

Exercise: Find the minimum value of R to keep the LED from being destroyed.

Return to figure 3.4 and look what happens if we reverse the voltage on the diode. The current is blocked until V_{PIV} is reached, and then the diode will be destroyed. PIV stands for 'Peak Inverse Voltage'. Diodes can have a PIV ranging from tens of volts to thousands of volts.

Two main specifications are critical when purchasing a diode: PIV and I_{max}. For rectifiers, the circuit must be analyzed to determine the maximum current the diode must carry and the maximum reverse voltage to which the device will be subjected (if any). Specify the diode to exceed these maxima. For LEDs, PIV is generally not a consideration and the rest of the circuit is designed to meet the LED's requirements for forward current.

A **light emitting diode (LED)** is a diode that emits light when a current passes through it.

A **transistor** is an analog three terminal semiconductor component that has two basic functions: amplification and switching. The NPN transistor is probably the most common transistor and gets its name from the types of material that it is made from: it has two n-regions (collector and emitter) separated by a p-region (base). A PNP transistor is opposite, with two P regions sandwiching the N region.

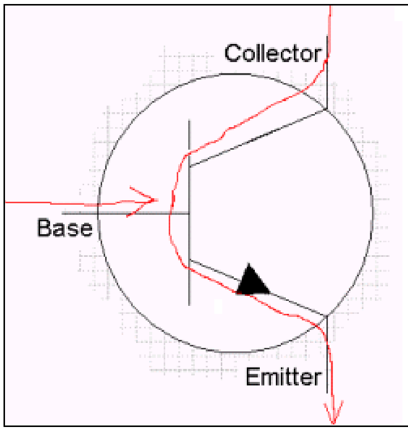


Figure 3.5: NPN Transistor

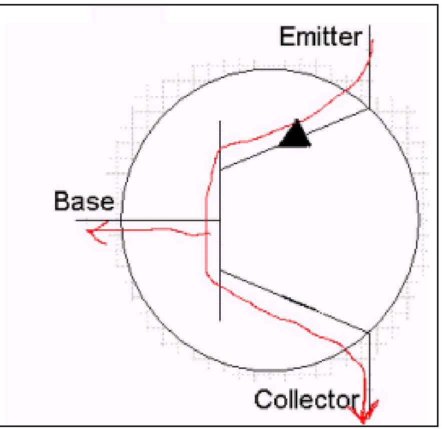


Figure 3.6: PNP Transistor

Both types of transistors provide amplification of the current applied to the base. The amount of amplification is called gain and is often given the symbol β.

$$\beta = I_C / I_B \qquad 3.5$$

Where I_C = collector current and I_B = base current. Typical β ranges from 50 to 1000 or more. In your EE classes you will learn to design simple transistor amplifier circuits. In this class we are most interested in using the transistor as a switch.

Many digital devices can provide only a few milliamps (mA) of drive current—not enough to run a motor or a solenoid valve. A simple transistor can provide the current amplification needed to effect the control.

Consider the situation in figure 3.7. We need to control a solenoid valve using the output of a digital logic gate, but there are two problems. The valve needs 150 mA at 12V to work, but the gate can only supply 10 mA @4.5V. A small NPN transistor can solve our problem. Almost any NPN transistor would work for this job so long as it could carry at least 150 mA of collector current (I_c). A 2N2222 is a common "workhorse" transistor costing about $.07 in quantities of 100. It has a β of more than 100 and comes in a variety of packages.

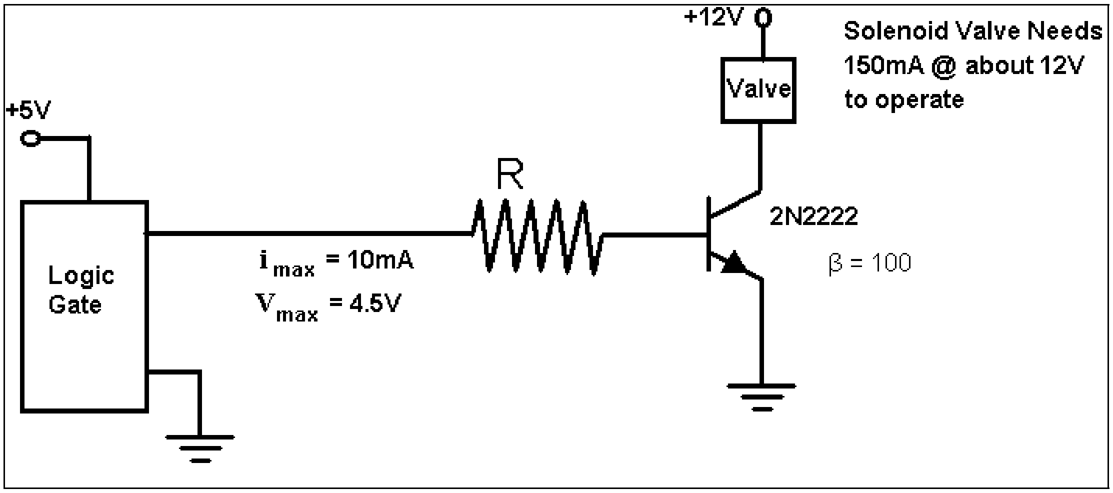


Figure 3.7: Logic gate schematic.

We can connect the transistor as shown in figure 3.7. To find R remember that:

$$\beta = 100 = I_C / I_B \rightarrow I_B = .150A / 100 = .0015A \qquad 3.6$$

The voltage V_{BE} is just a diode drop, or about 0.7V (actually a little lower but use 0.7). So to turn the transistor on, V_B must be about 0 + 0.7V where 0V is the voltage at the emitter. The voltage drop across R must then be:

$$V_R = 4.5V – 0.7V = 3.8V \qquad 3.7$$

R must be chosen to give V_R = 3.8V when I_B + .0015 A, so:

$$R = V / I = 3.8V / .0015A = 2530\ \Omega \qquad 3.8$$

Since 2400 Ω is a standard value, choose R = 2400Ω. For switching applications, the value of R is not critical. It must be large enough to limit I_B so the transistor is not destroyed but small enough to allow adequate drive current for the load.

An **optical sensor** (optical encoder) is a switch that is activated by the presence of any object in its target area. The optical sensor is basically a combination of an LED and a phototransistor. The LED sends out a beam of light and the phototransistor will allow current to flow through it if it 'senses' this light. The light takes the place of the base current in a 'normal' transistor. They are often used for position control of mechanical devises. The major advantage of optical sensors over mechanical switches is that they have no moving parts so there is practically no wear so they can last for very extended periods of time. There are two basic types of optical sensors: through pass and reflective.

The through beam sensors (figure 3.7) are U-shaped and will switch when an object goes between the arms. The reflective sensors will switch when its signal is reflected off an object back at itself. Care must be taken when mounting the reflective sensor because an object that is too close, too far away, or oblique to the sensor will not be detected (figure 3.8).

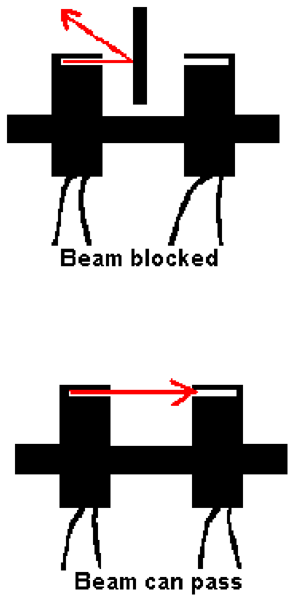


Figure 3.7: through beam

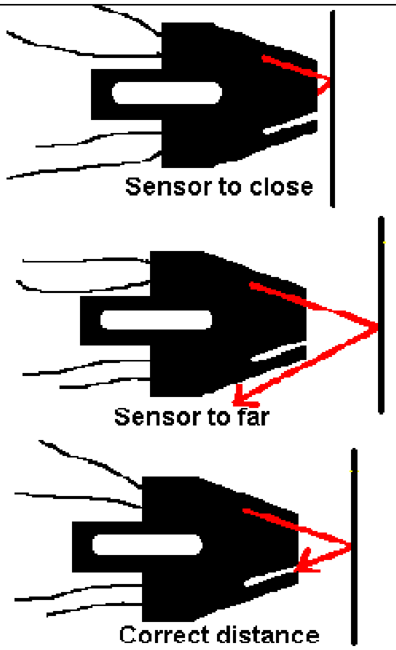


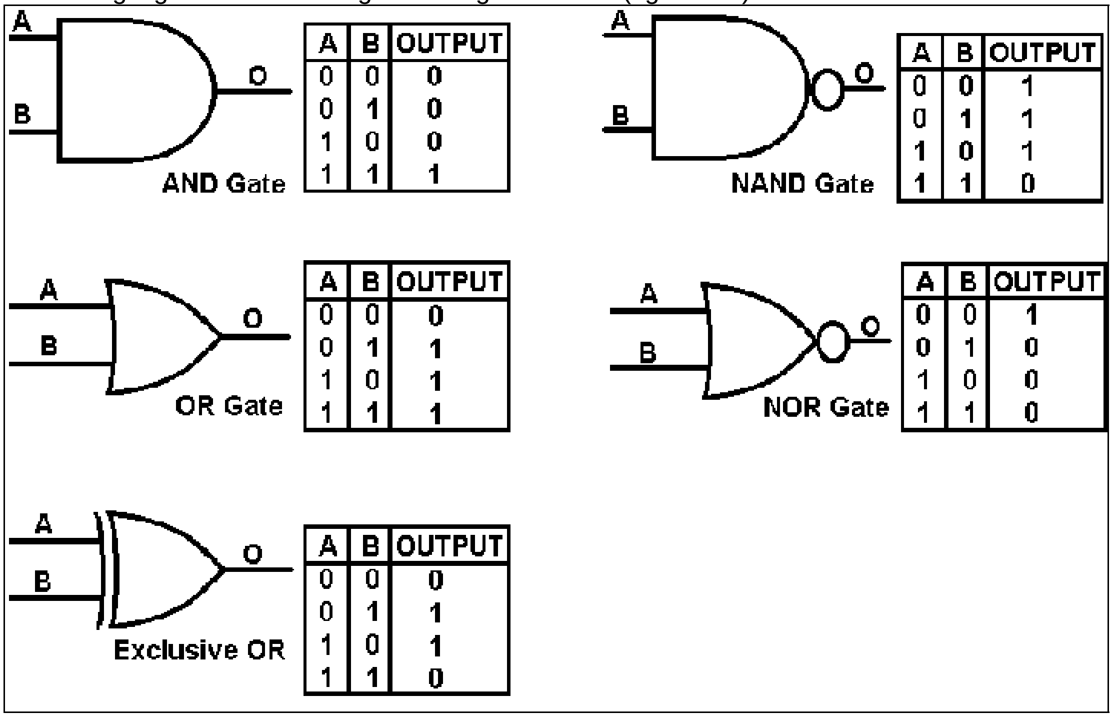
Figure 3.8: reflective

There are many different types of sensors that we have not talked about here, but you may encounter them in another class. All sensors measure some sort of environmental ‘thing’ and report the results as an electrical current for a computer to read. Examples include: the LM34 temperature sensor that we will use in the laboratory, pressure sensors, oxygen sensors, and many others.

Digital Electronics

Digital electronics involves two things: signal processing and miniaturization. Miniaturization is taking many of the components that we previously discussed and fitting them onto an integrated circuit ‘chip’. Most of the work currently being done to improve these chips involves reducing the current necessary for them to function. This will allow further miniaturization as well as reduce the heat produced while they are operating.

Digital electronics use gates, which are logical operators. The most common are: AND, OR, NOT, NOR, NAND, and NOR. Each type of logic gate produces a specific output for each possible combination of inputs. For example, the AND gate will give a ‘true’ or logic ‘1’ output only if both its inputs are ‘true’. The NAND gate is it’s logical negative of the AND (NAND = Not AND), thus it gives an output of 1 for every combination of A and B except both true. The operation of logic gates can be compactly described using a ‘truth table’ or logic table. Logic gates for several gates are given below (figure 3.9).



A	B	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

A	B	OUTPUT
0	0	1
0	1	1
1	0	1
1	1	0

A	B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	1

A	B	OUTPUT
0	0	1
0	1	0
1	0	0
1	1	0

A	B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

Figure 3.9: Logic gates with truth tables.

Chapter 4

Computer Operation

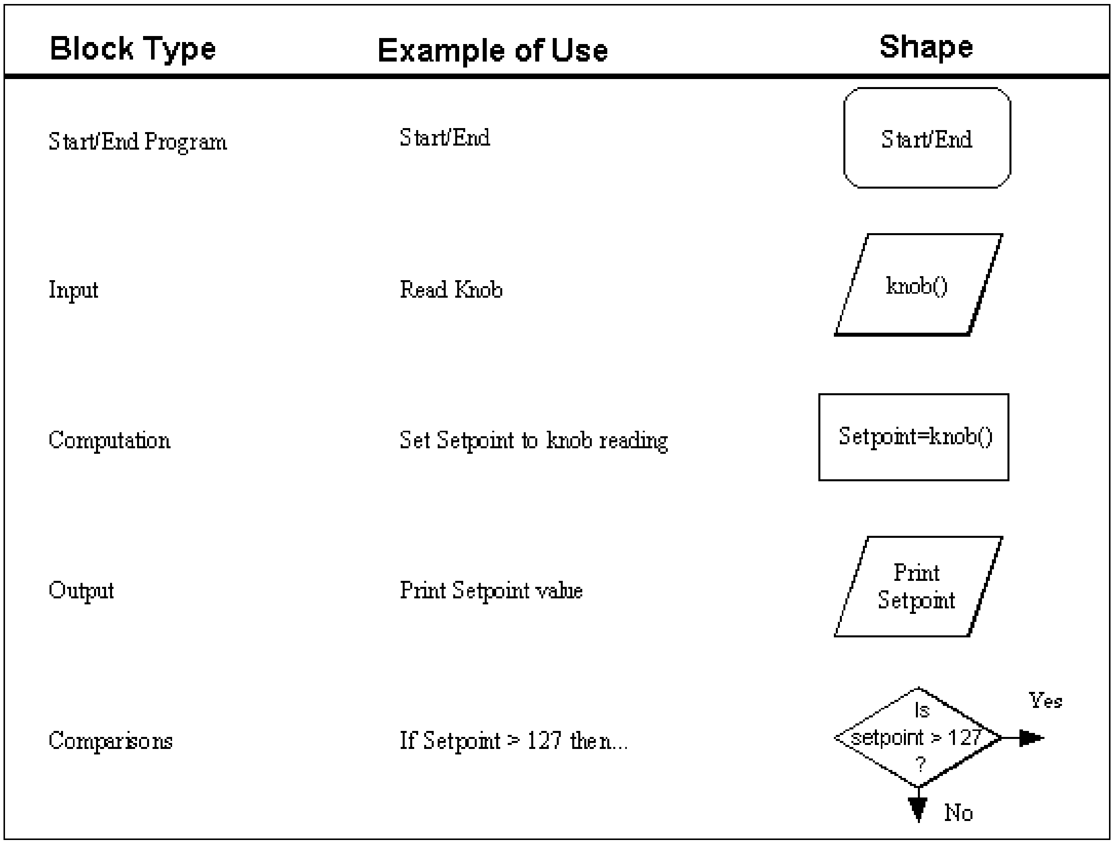
Handy Board Operation

The Handy Board is a hand-held, battery-powered microcontroller board ideal for personal and educational robotic projects. Based on the Motorola 68HC11 microprocessor, the Handy Board includes 32K of battery-backed static RAM, outputs for four DC motors, inputs for a variety of sensors, and a 16x2 character LCD screen. The Handy Board runs Interactive C, a cross-platform, multi-tasking version of the C programming language.

Detailed operation of the Handyboard will be discussed during the laboratory experiments. Basically, you write a C program on the desktop computer and then download it to the handy board. We will frequently be using flowcharts to describe computer programs. For a detailed description of making flowcharts and writing C programs consult the prerequisite ‘Engineering 102’ material; however, the following sections contain a brief review. The few differences between the ‘C++’ from Engineering 102 and the Interactive C (IC) that we use will also be discussed.

Flowchart Review

A flowchart is a diagram used to describe the steps of a program. It is used to simplify a difficult problem, making it easier to write a program. Figure 4.1 shows the common symbols used and has descriptions based on handyboard programming. Figures 4.2, 4.3, and 4.4 show basic flowchart examples.



Block Type	Example of Use	Shape
Start/End Program	Start/End	Start/End
Input	Read Knob	knob()
Computation	Set Setpoint to knob reading	Setpoint=knob()
Output	Print Setpoint value	Print Setpoint
Comparisons	If Setpoint > 127 then...	Is setpoint > 127 ? Yes No

Figure 4.1: Flowchart symbols

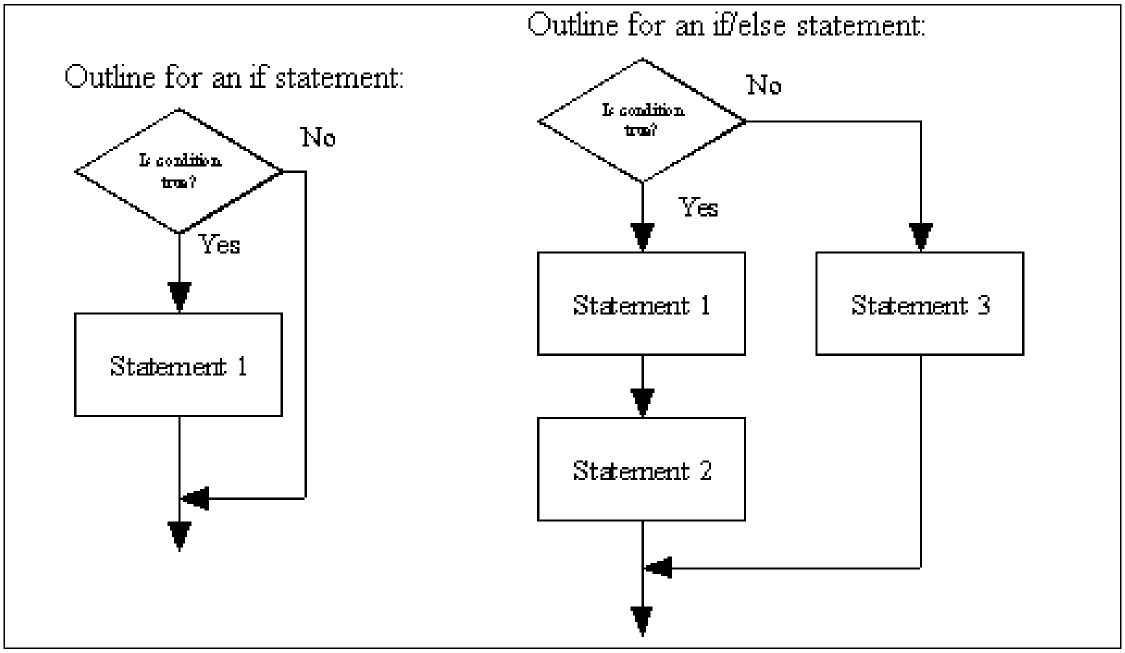


Figure 4.2: Generic flowchart examples

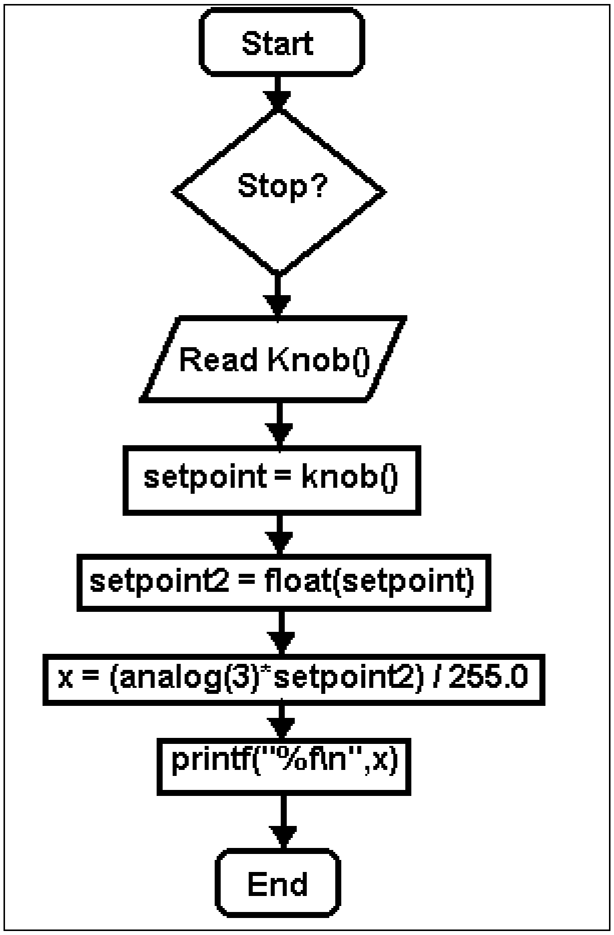

Start
Stop?
Read Knob()
setpoint = knob()
setpoint2 = float(setpoint)
x = (analog(3)*setpoint2) / 255.0
printf("%f\n",x)
End

Figure 4.3: Outline for a for loop.

IC Programming

The difference between C++ and IC are the compilers that are used. When using IC, a nice list of compilers are automatically included. For detailed descriptions of ‘C++’ programming consult the Help menu of ‘IC’, the Handy Board manual, or your ‘Engineering 102’ material. Here are a couple basic examples (4.1 and 4.2).

Example 4.1: Count to 5 on the LCD screen with one second between each number using a ‘for loop’.

Solution:

```
void main()
{    int x;
     for(x=1;x<=5;x++)
     {         printf(“%d\n”,x);
               sleep(1.0);
     }
 }
```

Example 4.2: Count to 5 on the LCD screen with one second between each number using a ‘while loop’.

Solution:

```
void main()
{    int x=1;
     while(x<=5)
     {         printf(“%d\n”,x);
               sleep(1.0);
               x++;
     }
 }
```

Microprocessor Architecture

Computers have come through a major evolution in performance and price over the past decade, but the principles behind how they work have not changed significantly since they were first invented. All computers use the same basic types of components—a Central Processing Unit (CPU), random access memory(RAM), one or more types of disk memory, (e.g. floppy, CD ROM, Zip drive, etc.) and various input/output (I/O) devises such as a monitor, mouse, keyboard, speakers, printer, etc. All of these devices are connected to two sets of wires called buses (see figure 4.?). The data bus is used to pass information back and forth from the CPU to and from other devices. The address bus is used to tell the various devices which one is being written to or read from.

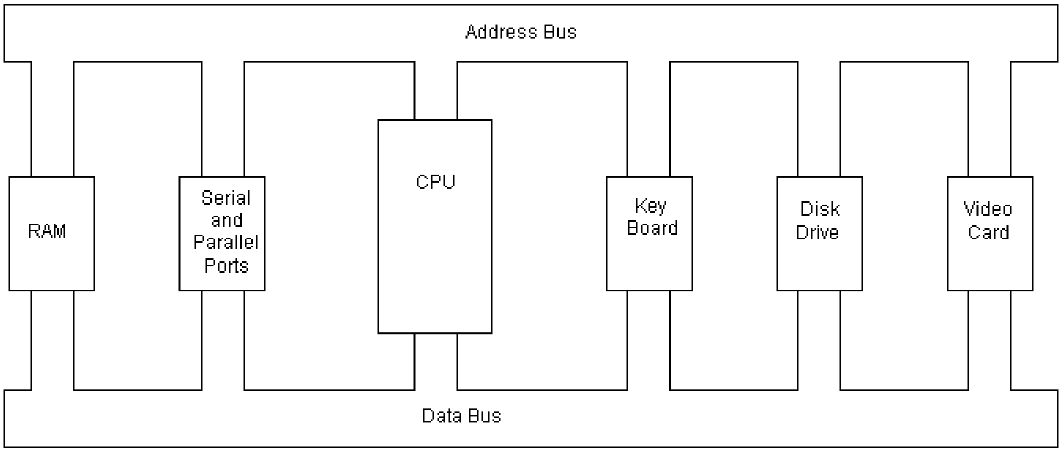


Figure 4.4: General personal computer architecture

The 'bus' is nothing more than a set of wires connected to each of the peripheral devices and to the CPU (see figure 4.?). Each wire corresponds to one bit of information and logic '1' or '0' bits are represented by the voltage level on the wire. For the handyboard, a voltage of less than about 1 Volt represents a logic '1' or 'true' and a voltage greater than 3.5 Volts represents a logic '0' or 'false'. Voltage levels and logic assignments vary from system to system. Newer computers, especially portables are being created to run on lower voltages as a means of conserving power and prolonging battery life. Many low power processors now run with power supply voltages in the 3 Volt range, and newer models are emerging which will run on 2 Volts.

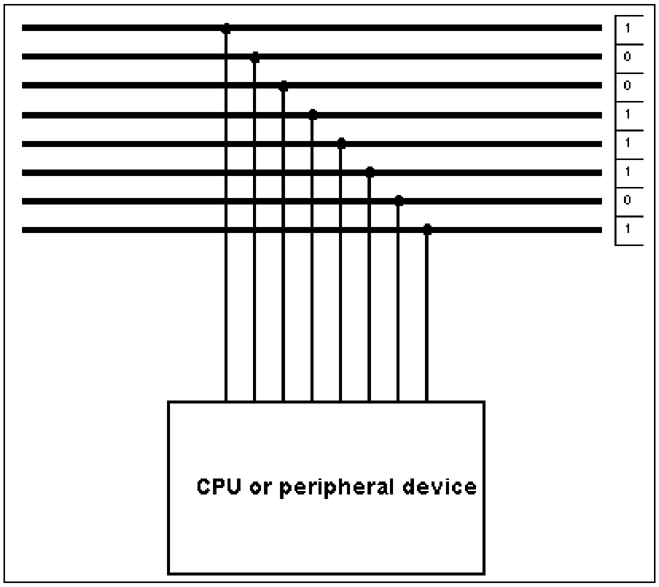


Figure 4.5: Bus detail

The data bus, as well as the parallel ports, use parallel data flow. Parallel data flow is sending several ‘bits’ of information (usually 8) to and from the CPU or other device at the same time. These 8 bits are called a byte and 4 bits are sometimes called a nibble. It is important to be able to represent a byte with one number instead of a series of eight numbers so you need to learn hexadecimal notation. One ‘hex’ number can stand for 4 binary bits, creating a convenient and compact notation.

Serial data transfer occurs through the computer’s serial ports. Serial ports have only one wire and send only one bit of information at a time. The advantage of serial communication is that it only requires on data wire, but the disadvantage is that it takes considerably longer to send and receive information. The data rate of a serial transmission line is measured in bits per second, since the data is transmitted on bit at a time. The unit of 1 bit/second is called a **baud**. Most PC users should recognize this because modem speeds are measures this way (i.e. 56k modem).

There is a number of ways to detect and correct errors in data transfer. Basically, it involves sending extra information along with the required information. The more extra information you send, the better the error detection and correction will be. However, it is a tradeoff because the more extra information that is sent, the longer the data transfer will take.

The most common error detection involves sending a ‘parity’ bit. Seven bits of the byte are information and the last bit is for error detection. It works like this: add the ones of the first 7 bits; if this number is odd the parity bit becomes a one. Other wise, the parity bit becomes a zero.

Binary/Decimal/Hexadecimal

Converting decimal and binary: Method 1

To understand converting from binary to decimal and back, you first need to understand the decimal system. It is 2472 miles from Morgantown, WV to San Diego, CA. The 2472 miles is the same as saying: $(2*10^3) + (4*10^2) + (7*10^1) + (2*10^0)$ miles. Any time you use a number system other than base ten, you need to put a subscript after it showing the number system that you used. For example, 10110101 binary would look like this: 10110101_2.

The binary system is based on powers of two. Therefore, the binary number 10110101_2 is equal to $(1*2^7) + (0*2^6) +(1*2^5) + (1*2^4) + (0*2^3) +(1*2^2) + (0*2^1) + (1*2^0)$ or 181 base ten. To convert from decimal to binary you do the following:

1. Divide the number by 2 and write the remainder off to the side.
2. Divide the new number by 2 and write the remainder off to the side.
3. Repeat step 2 until the number is 0.
4. Your answer is the remainders, read from bottom to top.

This may sound confusing, but example 4.3 Should clear up any confusion.

Example 4.3: Convert 69 to binary.

Solution:	*69 / 2 = 34*	*with a remainder of 1*
	34 / 2 = 17	*with a remainder of 0*
	17 / 2 = 8	*with a remainder of 1*
	8 / 2 = 4	*with a remainder of 0*
	4 / 2 = 2	*with a remainder of 0*
	2 / 2 = 1	*with a remainder of 0*
	1 / 2 = 0	*with a remainder of 1*

Now read the remainders from bottom to top: 1000101_2

Converting decimal and binary: Method 2

There is a second method that also works; you can choose either one. You make a small chart, start at the left, and see which numbers your number is divisible by. See example 4.4—69 is not divisible by 128 so write a 0 below 128, 69 is divisible by 64 (and you have a remainder of 5) so write a 1 below 64, 5 is not divisible by 32, 16, or 8 so write a 0 below them, 5 is divisible by 4 (with a remainder of 1) so write a 1 below 4, 1 is not divisible by 2 so write a 0 below 2, and 1 is divisible by 1 so write a 1 below the one.

Example 4.4: Convert 69 to binary.

Solution:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	*64*	*32*	*16*	*8*	*4*	*2*	*1*
	0	*1*	*0*	*0*	*0*	*1*	*0*	*1*

1000101_2

Converting Hexadecimal and decimal

Hexadecimal is base 16. When you run out of numbers, start using letters. You can count in base 16 as follows: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23… So A=10, B=11, C=12, D=13, E=14, F=15. Converting from base 16 to base 10 is just like converting from base 2 to base 10 (see example 4.5). Note that $10_{16} = (1*16^1) + (0 * 16^0) = 16$.

Example 4.5: Convert the Hex number E42A$_{16}$ to decimal.

Solution: $(14*16^3) + (4*16^2) + (2*16^1) + (10*16^0) = 58410$

Converting from base 10 to base 16 is very similar to converting from base 10 to base 2, as shown in example 4.6.

Example 4.6: Convert 1628 to Hex.

Solution: Method 1:

	1628 / 16 = 101	*with a remainder of 12*
	101 / 16 = 6	*with a remainder of 5*
	6 / 16 = 0	*with a remainder of 6*

Now read the remainders from bottom to top: 65C$_{16}$

Method 2:	16^3	16^2	16^1	16^0	
	4096	*256*	*16*	*1*	
	0	*6*	*5*	*12*	*(12 = C)*

So, again 65C$_{16}$

Converting Hexadecimal and Binary

To convert from binary to Hex it is easiest to convert to decimal first (look at example 4.7).

Example 4.7: Convert 10111010$_2$ to Hex.

Solution: First, convert binary to decimal--

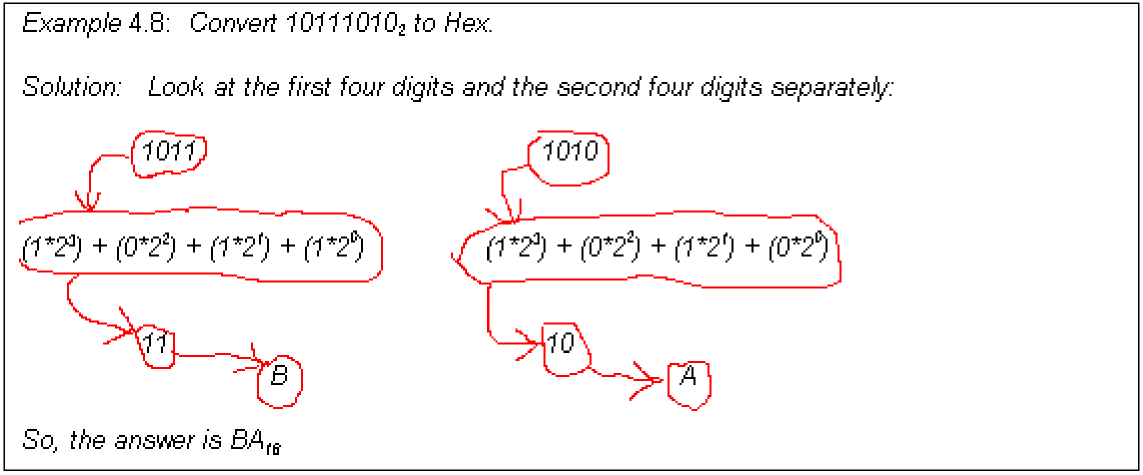
$(1*2^7) + (0*2^6) +(1*2^5) + (1*2^4) + (1*2^3) +(0*2^2) + (1*2^1) + (0*2^0) = 186$

Then, convert decimal to Hex—

186 / 16 = 11	*with a remainder of 10*
11 / 16 = 0	*with a remainder of 11*

So, the answer is BA$_{16}$

There is a trick for converting from binary to Hex. The last four digits of the binary number become the last digit of the Hex number, and the next four of the binary becomes the next one in Hex. It sounds complicated, but example 4.8 Will clear it up.



Example 4.8: Convert 10111010$_2$ to Hex.

Solution: Look at the first four digits and the second four digits separately:

So, the answer is BA$_{16}$

To convert from Hex to Binary, it is simplest to convert from Hex to decimal and then from decimal to binary.

Binary, Decimal, and Hex Homework

1.) Convert the following binary numbers (base 2) to decimal (base 10):
a.) 01101011
b.) 101110
c.) 11111

2.) Convert the following binary number (base 2) to hex’ notation (base 16):
a.) 11010101
b.) 10011010
c.) 11111110

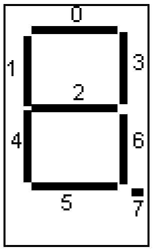
3.) Convert the following decimal numbers (base 10) to binary (base 2):
a.) 64
b.) 226
c.) 24

4.) Convert the following decimal number (base 10) to hex’ notation (base 16):
a.) 1804
b.) 13
c.) 17

5.) Convert the following hexadecimal notations (base 16) to decimal numbers(base 10):
a.) A7
b.) B2
c.) 65

6.) Convert the following hexadecimal notations (base 16) to binary numbers(base 2):
a.) 1B
b.) 45
c.) AC

The "7-segment" display is a cheap and popular way to display numbers for clocks, timers, ovens, etc. The display consists of 7 LED or LCD bars and sometimes a decimal point, as shown in the diagram below. Assume that we have a crude setup in which each display segment is connected to one bit of an 8-bit digital output port, according to the numbering scheme as shown. What hexadecimal number would have to be written to the computer port to make the display show the number 3?



7 6 5 4 3 2 1 0

8-bit parallel digital output port. Bit numbers correspond to display segment numbers shown. For example, bit zero of the ouput port is connected to segment zero of the display. Bit 7 of the port is connected to the decimal point, etc.

7.) Write a flowchart for a computer program that counts to ten with a 2 second delay between each number.

8.) Write a flowchart describing the inner workings of the common household commode.

9.) Write a flowchart describing the inner workings of a pop machine (including returning the correct change).

Chapter 5: I/O and Control Programming

Digital Input/Output (I/O)

Digital or discrete control systems are all around us. Lights, fire alarms, elevator doors, elevator motors, heating system fans and dozens of other devices operate on a basic on/off mode. Lots of industrial systems are also primarily binary—either the part is present or not, the E-stop button is pushed or not, the conveyor is on or not. We use a variety of sensors to determine whether or not the part is there or the worker’s hands are out of the way, etc. In the final analysis though, all of them are nothing more than a switch. What causes the switch to actuate may be mechanical force, or heat, or the presence of a ferrous material, or reflected light, or a variety of other things. But in the end, what happens is an electrical switch either closes or opens.

Digital output consists of a series of binary switches that can be used singularly (as a simple on or off) or combined to output more complex data like numbers or letters. All digital “conversations” between sensors, processors, and the controlled device consist of the transfer of on/off or high/low signals. Digital input is a high/low or on/off signal received by a device. For example, your car is equipped with an alarm that sounds if you leave your keys in the ignition while the car is not running and the door is open. The digital conversation goes something like this: When the car is turned off, the processor asks the sensor in the ignition if the keys are still there. If the answer (digital input) is yes, then the processor asks the sensor in the door if the door is open. If the input from the door sensor is “true” (meaning a binary 1 or “yes the door is open”) then the processor sends a signal (digital output) to the alarm, which will turn on the bell. The bell will continue to ding until you remove the keys from the ignition or shut the door (hopefully not locking yourself out of the car). Figures 5.1 and 5.2 illustrate this process.

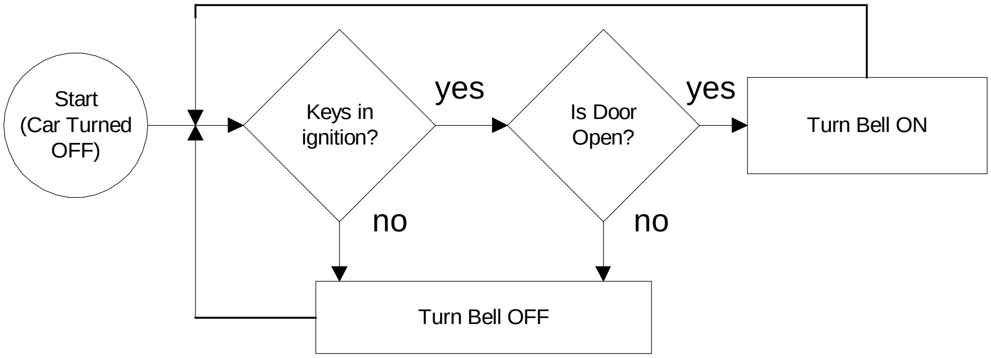


Figure 5.1: Flowchart of feedback system for warning bell in a car

Door	Ignition	To Bell
1	1	**1**
1	0	**0**
0	1	**0**
0	0	**0**

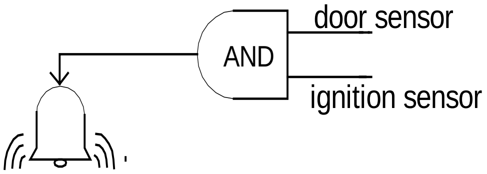


Figure 5.2: Logic Diagram and truth table for warning bell system

In order for the processor to turn off the bell, it has to continually recheck the input from each sensor. Therefore the program that runs on the control computer will take the form of a loop. In C, this would be a “while” loop because it will execute as long as the computer is in the “automatic” mode.

Control Program Logic Flow

The fundamental purpose of the microcontroller is to control something (duh!). Since everything the microcontroller “does” is the result of a program, it is clear that the actual control happens via the algorithms that make up the program. At their core, all control programs share some elements of structure, which we will examine

Loops and Loops: the *while* loop

The word “loop” will be used in two different ways when we talk about control programs. One way is in the programming sense, talking about a repetitive cycle of a computer program, usually set up for controls as a while loop:

```
while (test condition)
{
statement1;
statement2;
statement3:
}
statement4;
```

When the program comes to this section, it first goes to see if the test condition is true. If not, it will execute statement1, statement2 and statement3. The closed curly bracket tells the program that the loop is over, and it goes back to the top and checks the test condition again. If the condition is still true, it goes back through the loop. If the condition is false, the program skips over the loop and goes directly to statement4, and continues on down the line from there. Generally, the test condition is something that allows the system to jump out of automatic control and shut down or enter a manual control mode. In the case of the Handy Board, the stop button provides a useful input to allow this action. Any other input would work, however. The use of the stop button is illustrated towards the end of this section.

Loops and Loops: *open loop* vs *closed loop* control

In controls jargon, we also talk about open loop and closed loop control. Figure 5.3 is a schematic for an open loop control system. In open loop control, the controller reads the setpoint and calculates a control signal that “should” give the correct output. No measurement of the actual output is made. The water temperature control in a shower is a good example. You can set the hot/cold valves to give a comfortable water temperature, and everything works fairly well. But there is no feedback - no sensor measuring the actual temperature of the water. So if someone turns on the dishwasher while you are in the shower, and the hot water pressure decreases, there is no automatic adjustment to drive the system back toward the setpoint—you must manually readjust the valves.

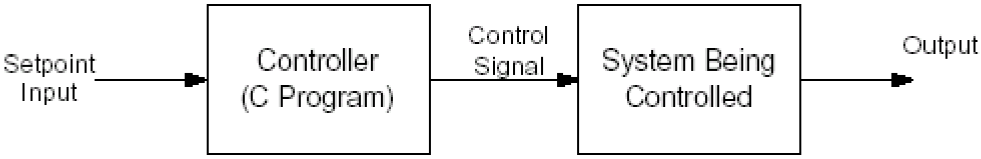


Figure 5.3: Open loop control system. There is a setpoint input and a control signal, but no sensor to measure the actual output of the system.

In closed loop control, sometimes called feedback control, a sensor is used to measure the system output and compare the actual output with the setpoint. Figure 5.4 is a schematic of a closed loop system. Generally, the comparison takes the form of a subtraction operation to create what is called the error signal—

error = setpoint – actual state

The error signal forms the basis for the control signal, with the objective being to drive the error signal to zero. It is easy to see from the schematic where the terms closed loop and feedback come from. The closed loop system can adjust for changes in system or environmental parameters—for example climbing or descending a hill with the cruise control actuated in your car.

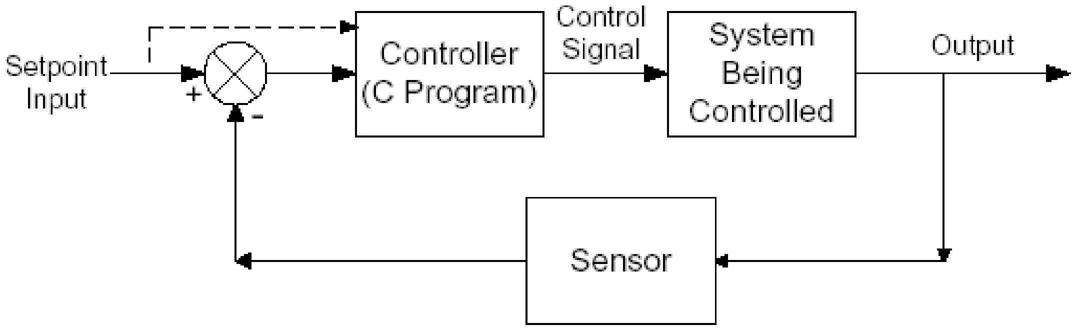


Figure 5.4: Closed loop control system. A sensor provides *feedback* to help adjust for system changes. The input to the controller is an error signal, and sometimes the setpoint also, depending on the control scheme.

Loops Either Way: Control Program Flow

Whether the control system you are devising is open loop or closed loop, the program that actually performs the decision-making will be set up with a program loop as its heart. Examples of both open and closed loop control structures are given below. Note the stop_button() function is used as the condition for the while loop.

Open Loop Control Example

```
while(!stop_button()) /* Recall that ! means logical NOT */
{
      read setpoint;
      control = fn(setpoint);
      output control;
      delay;
}
shut down;
```

Closed Loop Control Example

```
while(!stop_button())
{
      read setpoint;
      read feedback;
      error = setpoint – feedback;
      control = fn(setpoint, error);
      output control;
      delay;
}
shut down;
```

Note that in both of the examples above the control signal is given as a function of either the setpoint or the error signal or both. The form of that function can be quite simple or extremely complex.. The simplest form of control is proportional control, and the function is simply a multiplication:

output = K*setpoint; /* for open loop case */

or

output = K*error or K1*setpoint + K2*error; /*for closed loop case */

K is some constant, called the gain.

Also note that in both the examples, the loop included a delay step. This is because the computer can execute this little loop so fast that most of the other components can't keep up. If you are using the Handy Board to control the temperature of the room, there is no point in trying to adjust the temperature every 0.00001 seconds. In fact, there's not a lot you can do to anything mechanical in that short a time. So chill. In an advanced class, we would learn how to send the program off to check the outside temperature, make sure the windows were closed, check to see if you were out of the shower, if so turn on the dishwasher, if not, dial the office and download your email, etc, etc, then come back and do the loop again. In here, we'll just work on the loop. But do stick in a little delay—say 0.1 sec or so. The easiest way to do this is with the sleep( ) function provided by IC. Check out sleep ( ) and msleep( ). [HBM p 23-4].

Setpoints: Tweaking the Dials

An important concept within a control structure is the setpoint. A setpoint can be a fixed or adjustable point, within the range of the machine that is being controlled, which serves as the "goal" for the control system. Some examples of setpoints are the temperature at which you set your thermostat in your house, the speed at which you set the cruise control on your car and the number of rings that it takes for your answering machine to kick on. The controllers in each one of those machines "watches" for that setpoint and once the setpoint is reached, there is a change in the operation of the machine – either the furnace turns off, the fuel supply is decreased or the answering machine answers the incoming call. In the case of the answering machine, once the greeting begins to play, the control loop ends and does not reset until after the incoming message is recorded. Because the loop ends without checking the outside variable (whether or not someone actually recorded a message is irrelevant to the processor) the answering machine is an open-loop system. In the case of the thermostat and the cruise control, the systems are considered closed-loop because the controller will continue to check the status of the setpoint by comparing it to feedback from a sensor until the device (the furnace or the cruise control system) is turned off.

So, where does the setpoint come from? In some control systems, it is fairly obvious: you set the dial on the thermostat for a certain temperature; you set the volume knob on your stereo for a certain level; you push the elevator button for the floor you want. In other systems, the setpoint is sometimes the hardest function to spot. Take the cover off your toilet tank and find the knob that sets the water level in the tank...The key to remember is that the setpoint is connected with the desired state of the system. Sometimes there is a knob, sometimes not. Sometimes, as in a car cruise control, we push a button when the speed is correct. The computer then reads the car speed sensor and stores that value as the setpoint. In computer-controlled equipment, the setpoint will be just a number—like everything else in the computer.

You will learn more about setpoints in a controls class. For now, we will use the little user knob on your Handy Board as a way to put in the setpoint. The knob is just a potentiometer connected to one of the analog input channels (that's why you only have 7). It returns a value from 0 to 255, like any other analog input. You can read the knob setting by using the knob() function (see p 22 of the HB Manual). Try it: with the HB running, go to the command line and type:

```
printf("In = %d \n", knob( ))
```

Recall that knob( ) is a function which returns an integer value. So to use it, you must assign the value of the function to a variable you have declared as an integer.

Example:

```
void main()
{
      /* declare setpoint as an integer */
      int setpoint;
      setpoint = knob();
      printf("Knob set at %d \n", setpoint);
}
```

Feedback: Where it's at

So, we have the setpoint. To "close the loop" we have to compare the actual state of the system to the setpoint. Usually there is some scaling involved, too. (Refer to the section in this chapter on scaling) Collecting feedback is fairly obvious (although not always simple). We just need to stick some kind of sensor in/on the system and measure the state. No big deal, provided you can find a sensor to measure what it is you are looking for. You will learn about sensors in MAE 411. In the meantime, we'll measure simple stuff like temperature, pressure, speed, time, etc. The big deal here is that you have to scale the feedback and the setpoint signals so they are compatible with each other. For example, when you set the temperature control on your oven, it's probably a potentiometer, like the user knob on the HB. You are setting a voltage that corresponds to the temperature you want in the oven. Similarly, the feedback device may be a gas bulb thermometer (pressure output), a thermocouple (millivolt output), an RTD (voltage change output), etc. Each of these devices provides an output that represents the temperature in the oven. It is necessary to scale the output of the feedback device so that it matches the input from the setpoint transducer before you can generate a meaningful error signal. For linear systems, this is usually not hard - it just takes a little algebra.

DATA CONVERSION

Although computers communicate in a digital language, we do not live in a digital world. Most of the information that we want to manipulate with computers exists in analog form. Things like temperature, pressure and speed, have infinitely many values in a continuum and the computer must be able to deal with these kinds of signals as well, if it is to control many types of machinery.

Computers cannot deal directly with analog signals. Like a foreign language, the analog signals must be translated into a format that the computer can understand. Analog to Digital Converters (ADC) perform this operation by measuring the input voltage of the analog signal and comparing it to a reference voltage. A number is then generated that indicates the relative level of the input to the reference. In the case of the HB, the reference voltage is 5 V, and the numbers representing the input level range from 0 to 255 (i.e. 8 bits). That is, when the input voltage is 0, the number assigned to represent it will be 0. When the input voltage is 5 V, the number representing it will be 255. Ideally, all values in between 0 and 5V would be linearly proportional as in Figure 5.5.

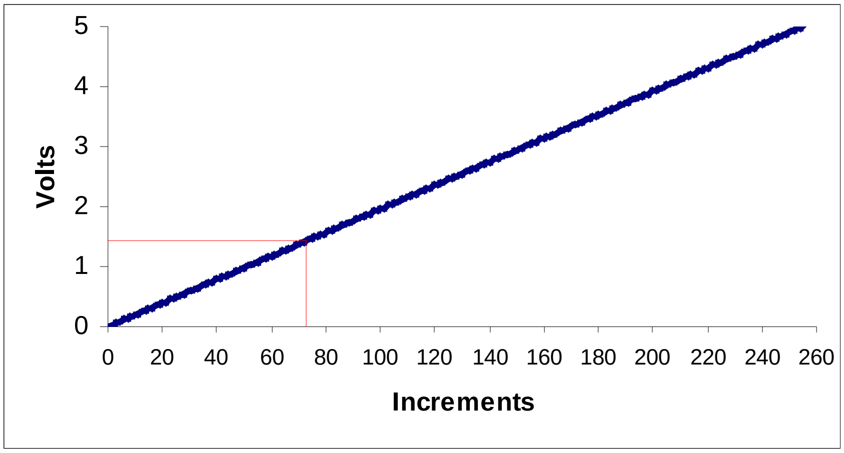


Figure 5.5: Linear relationship between voltage and bit-size

For example, the number 73 would represent: (73/255)*5 = 1.43 Volts. Unfortunately, the world is not ideal. Once an analog signal is converted to digital values, each of digital output represents a range of analog values dependent upon the resolution of the converter.

Resolution is the number of binary bits output by the converter. Because ADC circuits take in an analog signal, which is continuously variable, and resolve it into one of many discrete steps, it is important to know how many of these steps there are in total. For example, an ADC with a 10-bit output can represent up to 1024 (210) unique conditions of signal measurement. Over the range of measurement from 0% to 100%, there will be exactly 1024 unique binary numbers output by the converter (from 0000000000 through 1111111111). An 11-bit ADC can send twice as many states to its output - 2048, or 211. Resolution is very important in data acquisition systems because it determines the accuracy of the data. The amount of resolution needed is dependent upon what is being measured or monitored and how that information is to be used. A controller monitoring the amount of a medicine that passes through an IV line must be more accurate than the controller that monitors the amount of water going to a sprinkler system on a golf course. The controller on the IV may have to detect increments as small as a fraction of a milliliter whereas the sprinkler system may only need to monitor gallons.

Converter resolution is closely related to quantization error. **Quantization** is the division of a continuous event (such as an analog signal) into a series of discrete steps. The problem arises when the analog value being sampled falls between two of the digital "steps." When this happens, the analog value must be represented by the nearest digital value, resulting in a very slight error. In other words, the difference between the continuous analog waveform, and the stair-stepped digital representation is quantization error. Quantization error (q error) is simply an electronic form of rounding error. When solving mathematical equations, rounding the intermediate answers results in a discrepancy at the end of the calculation. And like rounding error, the more times the number is quantized (rounded) the more discrepancy there will be at the end of the conversion. Figure 5.6 illustrates how quantization occurs.

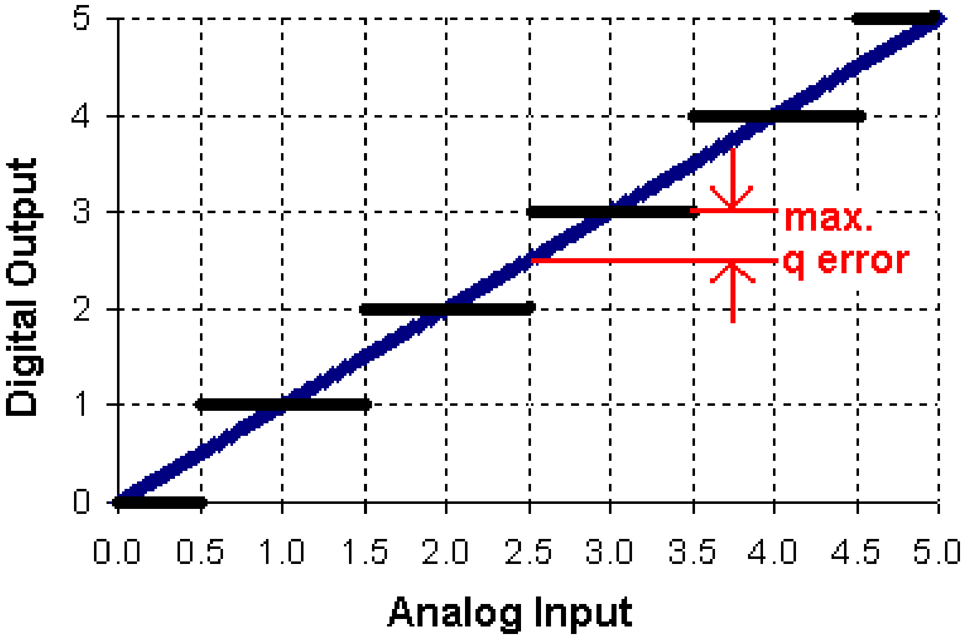


Figure 5.6: Illustration of quantization error. The digital output can take on only integer values, so each output represents a range of input values.

The diagonal line represents the analog signal being sampled and the stair-stepped horizontal lines represent the available digital output. The analog input and digital output differ at all times where the lines do not intersect. The digital output will be the value of the horizontal line that is closest to the diagonal (analog) input line. The largest error that can occur is one half as big as one of the digital steps.

The general equation for finding q is:

q = (0.5)* (Vmax-Vmin)/ (2N – 1)

where:

Vmax, Vmin = maximum and minimum input for the ADC

N = number of “bits” of ADC resolution

For example, in an 8-bit converter with a reference voltage of 0-5Volts has a maximum resolution of 0.02 Volts/step ((5-0)/(256-1) = 0.0196). The amount of quantization error would be equal to:

(0.5steps) * ((5-0)Volts/(256-1steps) = 0.01Volts, approximately +/-0.2%.

Notice that the quantization error of the ADC is a function of two things. We have covered the number of bits of conversion. But the q-error is also a function of the input voltage range. The HB input range is 0-5V, but converters are available with many possible ranges. Common ones are: -10 to +10V, -5 to +5V, -1 to +10V, 0 to 1V, 0 to 5V, and 0 to 10V.

Suppose we have an 8 bit converter with an input range of 0-10V. What is the quantization error (in volts) of this device?

q = (0.5)* (10-0)/(28-1) = 5/255 = 0.02Volts – twice the error of the 0-5V converter. For this reason, it is important to match the input range of the sensors and the input range of the ADC as closely as possible.

Audio recordings are another common use for an ADC. All sounds are analog signals in the form of waves. When a microphone collects acoustic waves, they are sent to the ADC in the form of varying voltages. The converter must then change those varying voltages into a binary representation of the original sound wave in order to store the information for later use. Figure 5.7 shows an analog sine wave and its digital representation.

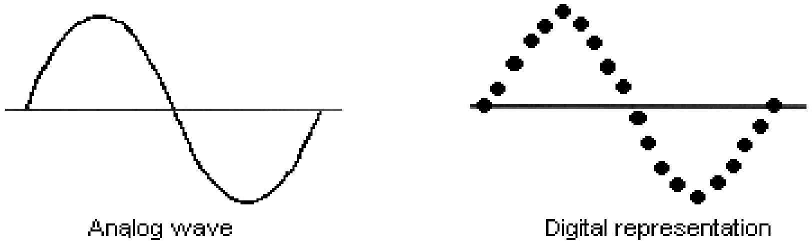


Figure 5.7: Analog sine wave and its digital representation

The ultimate goal in recording sound (or any other data) is to be able to reproduce the output signal so that it is identical to the original input. However, perfect playback is impossible because as with any language conversion, information is lost in the translation. As you have already learned, one factor affecting the accuracy of the reproduction is resolution. Another is quantization because as in mathematical equations, once a number is rounded, it is impossible to reverse the process and get an accurate answer. A third factor affecting the accuracy ADC circuitry is sample frequency, or conversion rate. This is simply the speed at which the converter outputs a new binary number. Like resolution, this consideration is linked to the specific application of the ADC. If the converter is being used to measure slowly changing signals such as level in a water storage tank, it could probably have a very slow sample frequency and still perform adequately. But if it is being used to digitize an audio frequency signal cycling at several thousand times per second, the converter needs to be considerably faster. When the sample period is too long (too slow), substantial details of the analog signal will be missed. It is very important that an ADC's sample time is fast enough to catch essential changes in the analog waveform. The highest-frequency waveform that an ADC can theoretically capture is the Nyquist frequency, which is equal to one-half of the ADC's sample frequency. Therefore, if an ADC circuit has a sample frequency of 50000 Hz, the highest-frequency waveform it can successfully resolve will be the Nyquist frequency of 25000 Hz. Naturally, the faster the ADC and/or the higher the number of bits of resolution, the more expensive it is.

If an ADC is subjected to an analog input signal whose frequency exceeds the maximum frequency that ADC can handle, the converter will output a digitized signal of falsely low frequency. This phenomenon is known as aliasing. Figure 5.8 is a graphical representation of how aliasing occurs.

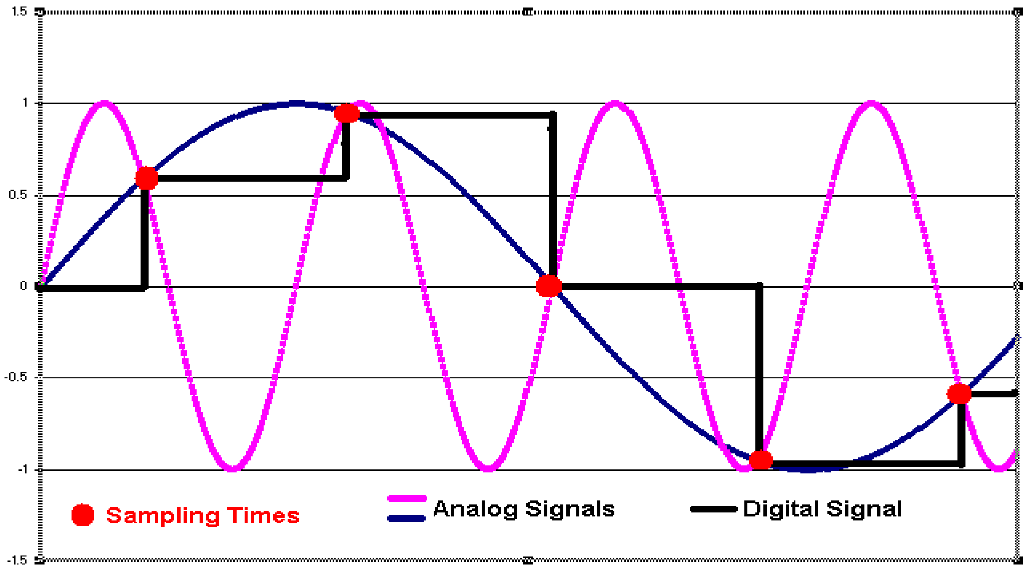


Figure 5.8: Example of Aliasing

A practical means of preventing aliasing is to place a low-pass filter before the input of the ADC. This filter will block any signal frequencies greater than the practical limit. The ADC circuitry will be prevented from seeing any excessive frequencies and thus will not try to digitize them. It is generally considered better that such frequencies go unconverted than to have them be "aliased" and appear in the output as false signals.

Digital synthesizers create sounds by working with digital representations of sound waves, in the form of binary numbers. These representations can then be manipulated in a variety of mathematical ways (different equations can produce effects such as fades and echoes), but in order to hear the results of these changes, or to hear a reproduction of any sound that has been digitally recorded, the numbers have to be passed through a digital-to-analog converter (DAC). The DAC changes the numeric representations of the sound waves to equivalent electrical (analog) voltages. These voltages become sound after being fed into an amplifier and out of a speaker system.

Of course, DACs are useful for many applications besides music, and there are numerous ways to affect Digital to Analog conversion. In the world of industrial control, the most common method is Pulse-Width Modulation (PWM). PWM is used to control DC motor speed, electric heater current, lighting levels and many other things.

PWM works by dividing time into a series of “on” and “off” pulses. The relative width of the two determines the output level. Figure 5.9 illustrates the idea. Suppose we have a motor we want to run at ½ speed. If it runs at full speed with Vin = Vmax, then it will run at ½ speed at Vin = Vmax/2. But in a digital world, Vmax/2 is not convenient to create directly. Instead, we approximate it by turning on the power for half of the time and turning off the power for half of the time. Then the average voltage is Vmax/2.

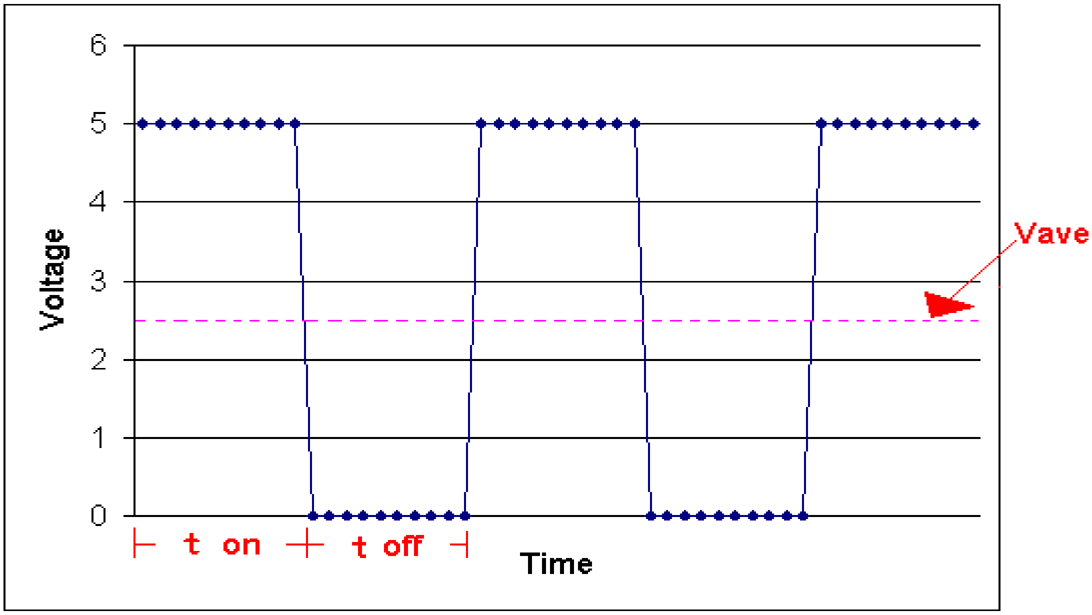


Figure 5.9: Illustration of Pulse Width Modulation

To make the motor run at ¾ speed, we make ton = 3 toff. By controlling the ratio of ton to toff, we can generate any Vave between 0 and Vmax. If the switching frequencies are very high, devices like motors, heaters and lamps don't even "see" the on/off cycles. Fortunately, digital electronics are so fast that high switching rates are easy to obtain

Control Structure

Now that you have a basic understanding of how a processor communicates with sensors and actuators, we can discuss how to use that communication system to control the mechanical devices that are so important to our lives. A machine simply operates when and how it is told to operate. People must decide how and when the machine should operate and then design the systems that control the machine's function to suit those parameters.

Closed-loop, or continual control systems require some special consideration during the design phase. Sometimes, an exact setpoint can be nearly impossible to maintain. For example, if your furnace were set to 72 degrees F, it would be extremely difficult for the furnace to maintain such an exact temperature. Opening a door to another room or a person walking past the thermostat could cause the temperature to vary above or below the setpoint. The furnace would be shutting on and off constantly, trying to keep the temperature at the desired level. To prevent the wear and tear that is associated with the frequent starting and stopping of machinery, deadband is introduced into the control structure. Deadband is simply a range of values surrounding the setpoint that allows some leeway in the control system. In the furnace system example, with the setpoint still at 72F, deadband would allow for maybe plus or minus 1 degree before reporting a change to the processor. For example, the furnace would remain on until the feedback from the thermostat reported a temperature of 73F and once the furnace was off, it would remain off until the feedback reported a temperature of 71F. Many systems in which the output is on/off contain deadband to prevent rapid cycling.

Scaling

As discussed in the section on ADC, computers can only deal with digital signals. Most of the variables that are measured and manipulated are analog signals. In order to design a control system that can be governed by an analog signal, you must understand the relationship of the analog signal to its digital equivalent. You must be able to tell the computer, in the form of a program, exactly what that relationship is so that the controller can output a variable in a unit that we can understand. It would be extremely inconvenient to walk over to a thermometer to check the temperature only to find that it read 1.65 Volts or to try to sing a song in the key of 440 Hertz.

In order to establish a relationship between variables, you have to know the range of each variable. Once the ranges of the variables are known, you have to know the range of the processor. After you know the range of the processor, you can set up a relationship between the original variable and the variable that is output from the processor. Luckily, all of the relationships that you will deal with in MAE 211 result in simple linear equations. An example of the scaling process for a temperature sensor is illustrated in Figure 5.10.

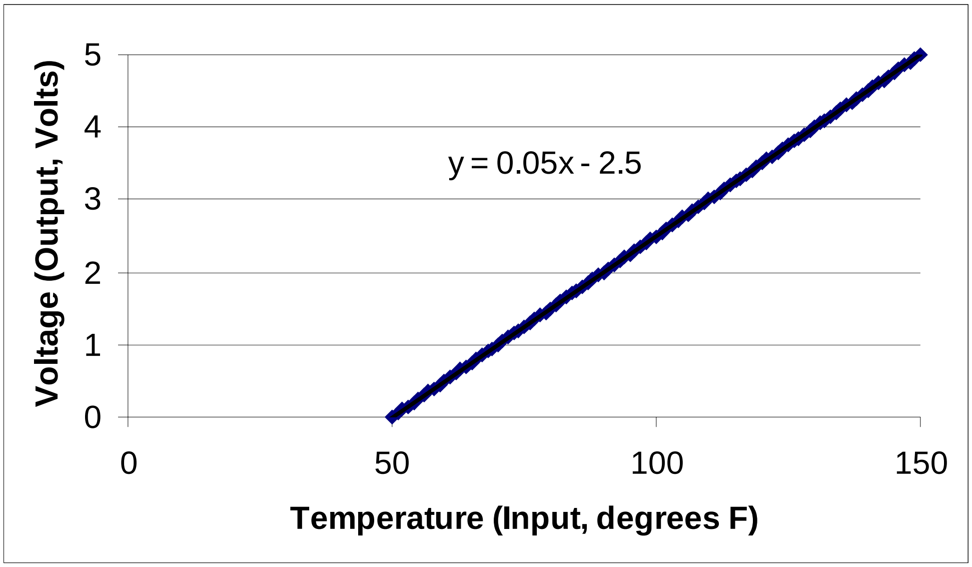
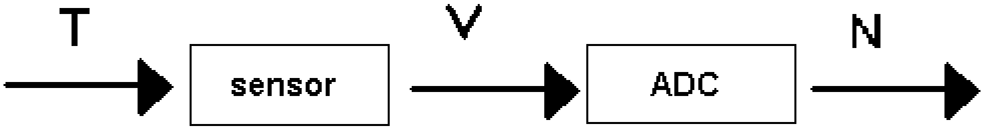


Figure 5.10: Linear relationship between temperature input and voltage output

Suppose we have a temperature sensor that coverts temperature (T) to a voltage (V) as shown in Figure 0000. If T=50F, the sensor outputs V=0volts. If T=150F, the sensor outputs V=5volts. Between 50F and 150F, V is linearly proportional to T. Now when the voltage from the sensor is input to the HandyBoard, the ADC converts it to a number from 0-255. Thus, when we "read" the sensor input, the result is a number (N) that we must be able to convert back to the temperature.



From algebra we can find V(T).
V = mT + b
m is the slope = ΔV / ΔT = (5-0) / (150-50) = 0.05
V = (0.05)*T + b ; at T=50, V=0 so:
0 = (0.05)*(50)+b
-2.5 = b and
V = 0.05T – 2.5

Likewise, we can find N as a function of V. Don't worry about quantization for this step– treat the ADC as a linear device. (See Figure 5.11)

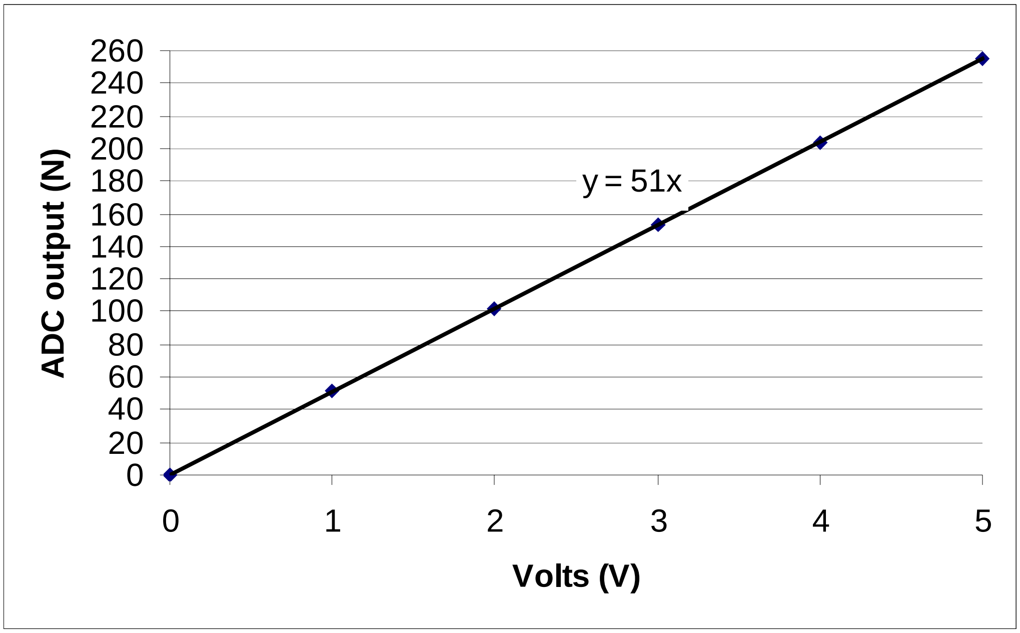


Figure 5.11: Relationship between input Volts and ADC output

N = mV + b
m = (255-0)/(5-0) = 51
since N=0 at V=0, b=0 ; therefore, N = 51V

And from above, V = 0.05T – 2.5 so:
N = 51*(0.05T-2.5)
But we know N and want to find T – so solve for T:
T = (N +127.5)/ 2.55 = 0.392N + 50

So, if we want to display the measured temperature on the HandyBoard screen, we can use this simple equation to calculate it.

How does quantization error affect this system? Let's look. Recall that N can only be an integer. Suppose N=1: then,
T = (0.392)*(1) +(50) = 50.4

Now suppose N=2
T = (0.392)*(2) +(50) = 50.8
For N=3,
T = (0.392)*(3) +(50) = 51.2

You can see the pattern (an accuracy of +/- 0.2 degrees) in this way or by using the formula for quantization error:
q = (0.5)* (Vmax-Vmin)/ (2N – 1) = 0.5*(5/255) = 0.01Volts or +/- 0.2%

Over the sensor's range of 100 degrees (150F-50F) this means that we can only measure the temperature to within +/- 0.2 degrees. For most purposes, this will be adequate, but if it were not, we would need to use an ADC with a higher resolution – perhaps 10 bits instead of 8.

Controls Homework Set

Show all work ***and provide*** ***proper units*** ***with answers where appropriate. Round answers to*** ***3 significant figures***

1.) Explain what is meant by 8bit, 10bit, or 12bit when applied to Analog-to-Digital converters. How are they different from each other? What are the pros and cons of having more bits?

2.) An 8-bit A/D has an input range of 0-5V. If an input voltage of 1.77 V is applied to the A/D converter, what is the converter output?

3.) An 8-bit A/D operates on an input range of 0-1V. If its output reading is 89, what is the input voltage?

4.) What is meant by the "quantization error" of an A/D converter, and what factors affect it?

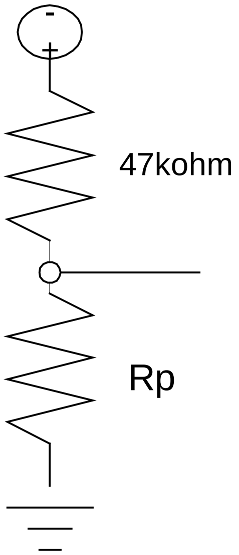
5.) A 10-bit A/D converter is operating over the input range of –10V to 10V. What is the quantization error (resolution) in volts for this converter?

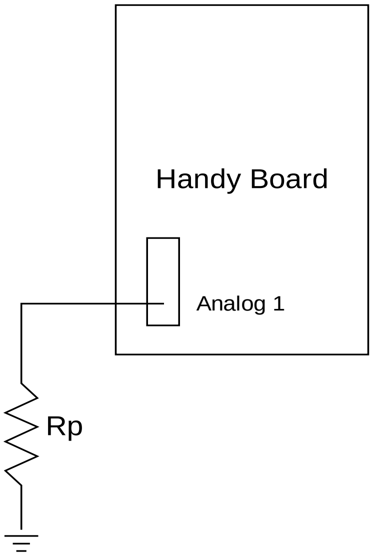
6.) Explain the difference between open loop and closed loop control systems. Give two examples of each system that are not listed in the text.

7.) What is deadband and why is it used in a control system? Are there systems in which including deadband would not be appropriate?

8.) Explain in your own words the phenomenon of aliasing. What is it and how does it occur?

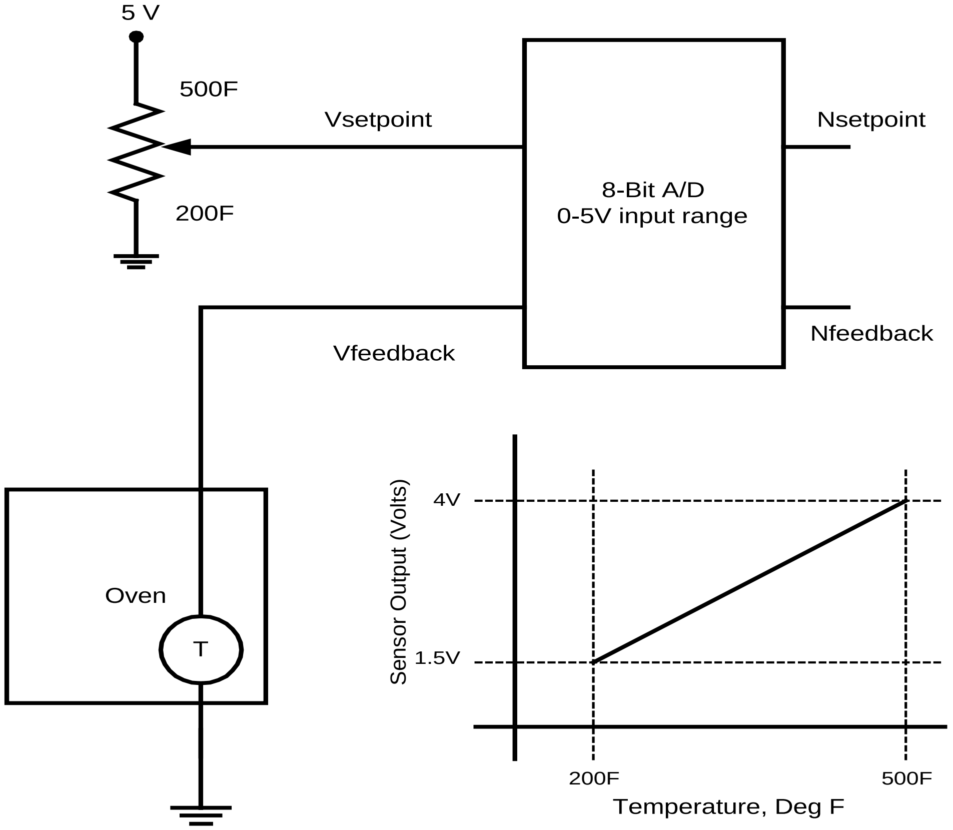
9.) Light is shining on a photoresistor, Rp, so that its resistance is 10kΩ. If the resistor is connected to analog input 1 on the HandyBoard, what will be the value of x in the command x=analog (1)? Recall that the HB inputs all have a 47kΩ pull-up resistor to +5V.





10.) What is the possible range values of Rp in ohms when the HandyBoard reads 200 on analog 1? HINT: This deals with quantization error.

11.) You are designing a control system for an oven based on the Handy Board. The setpoint will be read from a potentiometer similar to the HB's ***knob( )*** that is graduated from 200F to 500F as shown in the sketch below. A special resistor called a thermistor will sense the temperature in the oven. The resistance of the thermistor changes with its temperature. You have already designed a clever circuit which causes the voltage "output" from the thermistor to be as shown in the graph. You must now figure out how to scale the setpoint input and the feedback voltage so they represent the same thing—temperature. Both the setpoint voltage Vsetpoint and the feedback voltage Vfeedback will go to the HB's analog inputs.



Find two equations:

a. **Find the equation that gives the setpoint temperature, given the A/D output from the setpoint knob, Nsetpoint**

b. **Find the equation that gives the actual temperature in the oven, given the A/D output from the sensor, Nfeedback. NOTE that you do not need to find one to find the other—they are independent**.

NOTE: You do not need to find one to find the other – they are independent.

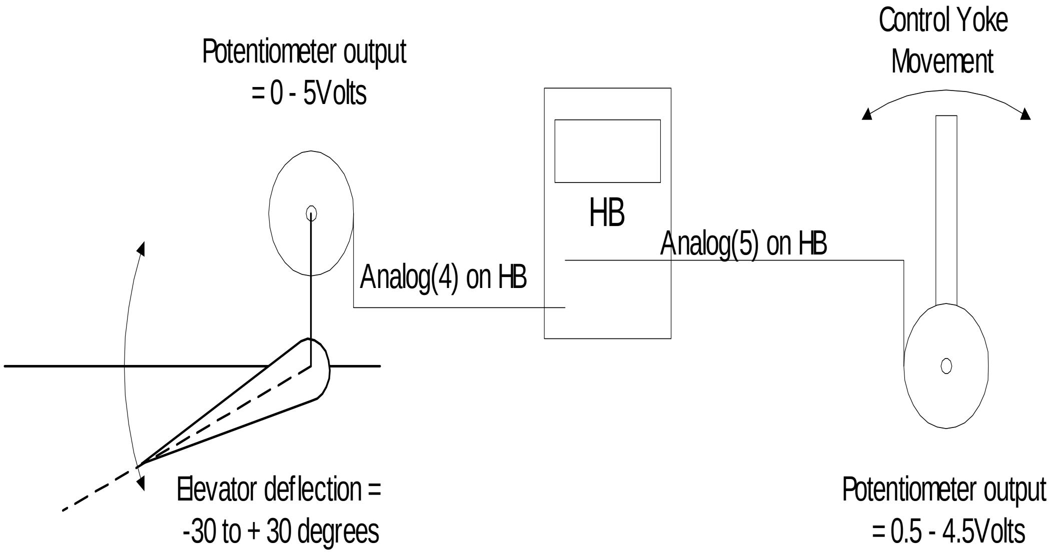
12.) Suppose you connect your Handy Board to an LM34 Temperature Sensor. The output of the sensor is 0.01 Volt per F. If the temperature in the room is 93F what will be the reading from the A/D converter on the HB?

13.) The relationship between Celsius and Fahrenheit degrees for measuring temperature is linear. Find an equation relating °C and °F if 0 °C is equal to 32°F and 100 °C is equal to 212°F. Show all work and draw a graph to illustrate your answer.

14.) The Kelvin scale for measuring temperature is obtained by adding 273 to the Celsius temperature. Write an equation that relates the voltage output of an LM34 to the temperature in both Celsius and Kelvin.

15.) You are to design a control system for a fan. The fan will be used to control airflow in and out of a kiln. When the kiln needs to warm up, the fan is to run clockwise, blowing warm air from the combustion process into the holding area. When the kiln needs to cool, the fan must run counter clockwise in order to blow the hot air out of the holding area and up the exhaust. While the kiln is at the desired holding temperature, the fan should not run. The fan speed should be a function of temperature whereas the larger the difference between the setpoint and the actual temperature, the faster the fan should run. The maximum range of the fan controls should be plus or minus 100 °F from the setpoint. Deadband should encompass plus or minus 5°F from the setpoint. Full speed clockwise and counterclockwise is equivalent to positive 24V and negative 24V respectively. Derive the equations for the functions relating the fan speed (percent of full power) and the temperature. Show all work and draw graphs to illustrate your answers. Explain how pulse width modulation could be used to control the fan and write the equation that could be used to control the fan speed with PWM. Hint: The equation will be a function of time on and time off.

16.) A simplified "fly by wire" controller is shown in the schematic below. The pilot moves the control yoke, which is attached to a potentiometer. The potentiometer puts out a voltage which can range from 0.5 V when the yoke is all the way back, to 4.5 V, when the yoke is all the way forward. The signal goes to a Handy Board. The HB drives a motor that controls the position of the elevators, which have a range of +/- 30 degrees. A potentiometer is connected to the elevators, and provides a signal of 5V when the elevators are at +30 degrees and 0V when they are at –30 degrees.



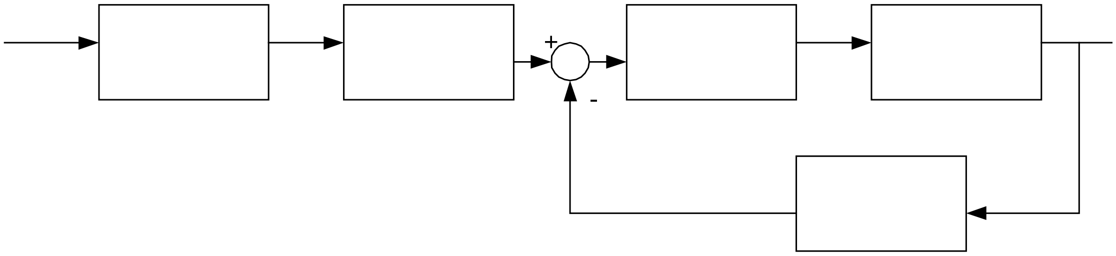
In your control code, you will read the elevator pot by writing something like:

E = analog(4);

a.) Find a scaling function Deflection = f(E) that will calculate the elevator deflection in degrees so it can be displayed in the cockpit.

b.) Assume that the position of the control yoke is the desired position for the elevators, i.e. yolk back=30° up, etc. Find the scaling function to relate the HB's ADC output to the desired elevator position in degrees.

c.) The diagram below is a control block diagram for the system. Label the boxes and signals to show where the yoke, motor, HB, potentiometers, etc. fit in the diagram. You may add boxes if you wish, and remember that some part(s) of the diagram are software, not hardware.



d.) The error signal is $\varepsilon = \theta_d - \theta_a$ where θ_d = desired position and θ_a = actual position. Assume a motor control similar to the HB with motor(x,100) = full speed CW (elevator up) and motor(x,-100) = full speed CCW(elevator down). Derive a control algorithm so that:

- $-0.5° \leq \varepsilon \leq 0.5° \Rightarrow$ motor off
- $0.5° \leq \varepsilon \leq 15°$ & $-0.5° \geq \varepsilon \geq -15° \Rightarrow$ motor speed proportional to ε
- $|\varepsilon| \geq 15° \Rightarrow$ motor speed = max speed

e.) Create a logic flow diagram showing how you would use the signals from the pilot and the elevators to perform closed loop control of the elevator drive motor and display the actual elevator position on the screen at all times. Do not worry about putting deadband into your controller yet. You do not need to include the actual scaling equations in the flow chart, but you do need to show where in the program they would be.