

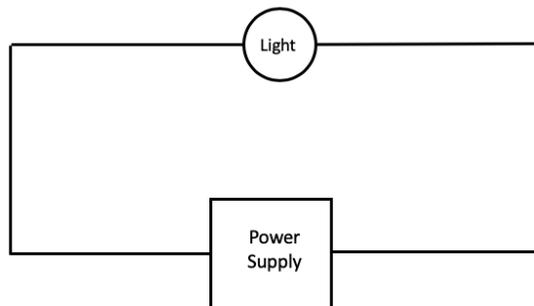
Switches

How can a switch make a light or motor work?

Switches work by mechanically connecting a power wire to turn a motor on, or disconnecting a power wire to stop a motor. Essentially a switch mechanically switches the power on or off, hence the name switch.

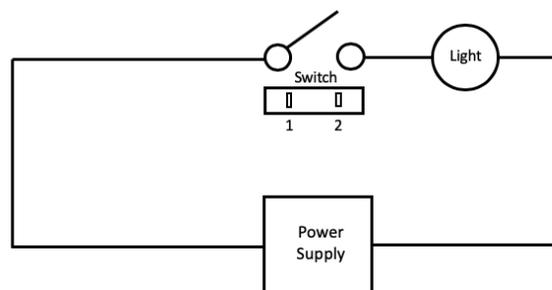
Imagine a simple electric circuit: a battery connected to a light. In this example, power comes out of the positive side of the battery and enters the light via a wire. The power comes out of the light and returns to the battery via another wire. This set-up is called a loop. As long as this loop is connected and the battery is charged, the light turns on. This simple circuit can be built with a light, battery and some wire. If the wire between the battery and the bulb were removed, the light would turn off because power can no longer get to or from the bulb.

Simple Electric Circuit



A switch mechanically disconnects the wire from the light. One type of switch is called a Single Pole Single Throw (SPST). An SPST has one wire (or Pole) that can be connected or disconnected to only one other wire (the Throw). An SPST is a simple On-Off switch. Typically, an SPST switch has two terminals (connections) – one for input (1) and the other for output (2). In one position, the switch is “Open” and no connection is made because no power can flow to the device. In the second position, the switch is “Closed” so that the two terminals are connected and power can flow to the device.

Single Pole Single Throw (SPST) Switch



Another type of switch is called a Single Pole Double Throw (SPDT). It is Single Pole because it has one wire (the Pole) that can be switched between two connections (the Throw). These types of switches usually have three terminals, the center being the Common (C) or Pole, and the other two designated as either Normally Open (NO) or Normally Closed (NC). An SPDT switch can be used to power two different devices such as a red light and a green light (Figure A). A push button type of SPDT (Figure B) is used to power a device when the button is depressed (by connecting the power supply to NC). The device will not receive power if the button is left alone. NO designates the terminal that will not be connected to the battery when the button is not pushed, i.e., this circuit is open, whereas NC designates the terminal that will be connected when the button is pushed, i.e., the part of the circuit that is closed. The push buttons on the SeaPerch controller work in a manner similar to an SPDT switch.

Figure A: Single Pole Double Throw (SPDT) Switch

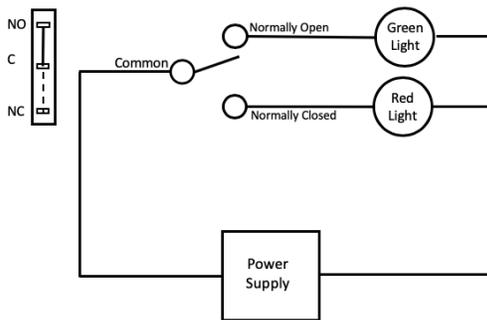
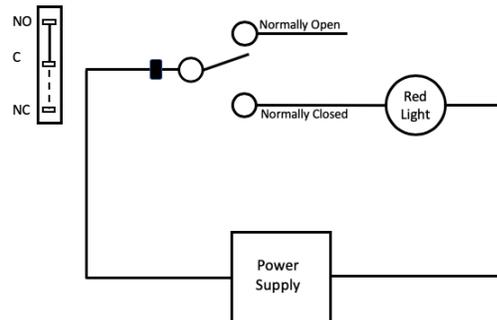
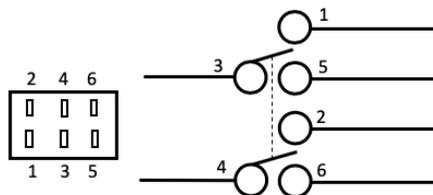


Figure B: SPDT Push Button Switch

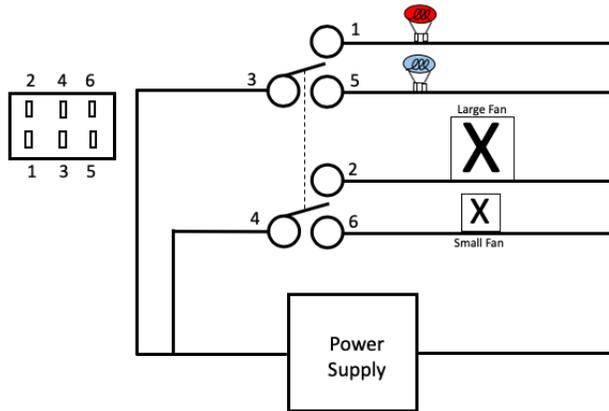


A third category of switches is called Double Pole Double Throw (DPDT). This means that there are two poles, each of which has two throws. It is easiest to think of a DPDT as two SPDT switches in one. A DPDT switch can turn multiple independent circuits on or off at the same time. A DPDT switch has six terminals with two of the terminals acting as the common, two NO terminals and two NC terminals.

Double Pole Double Throw (DPDT) Switch



Consider this problem: A business has a small cooling fan to run on cool days and a larger cooling fan for hot days. The business wants an indicator light to signal which fan is being used – a red light for the large fan and a blue light for the small fan. Using a DPDT switch, Figure C shows one possible way to wire the switch. *Note: One fan or the other is always running as there is no way to disconnect both fans and lights as long as power is connected to the switch.*



Switches seem pretty useful. What do switches have to do with computers?

Everything! Computers use nothing more than millions and millions of switches to remember what a photo looks like or what music sounds like, even if the computer is turned off and unplugged. The computer has an amazing ability to code information in a language called **binary**. Binary is essentially switch language for on or off. Computers operate in binary, meaning they store data and perform calculations using only zeros and ones. A single binary digit can only represent either zero (0) or one (1). However, multiple binary digits can be used to represent large numbers and perform complex functions. Everything on earth can be represented in this form. For example, the number 24 is a number written using standard decimal notation (also called Base-10). If you convert this to binary (called Base-2) you will see that it is equal to 11000.

How is 24 the same as 11000?

24 is a number in our decimal system. The decimal system works by indicating with the numbers 0 - 9 how many sets of 1's, 10's, 100's, 1000's, and so on, there are in any given number. So for 24, there are no 1000's, no 100's, two 10's, and four 1's. In the decimal system, each number ends in one more 0 than the previous number.

Another way to write numbers uses Base-10 notation. Each number in Base-10 comes from taking 10 (the base), and multiplying it by itself either zero times (not multiplied by zero which would equal zero), then 1 time, then 2 times, then 3 times, and so on. This is read as 10 to the 0th power, 10 to the 1st power, 10 to the 2nd power, and 10 to the 3rd power. In mathematical notation, it looks like:

$10^0, 10^1, 10^2, 10^3$, which is equal to 1, 10, 100, and 1000

Base-10					
100,000	10,000	1,000	100	10	1
10^5	10^4	10^3	10^2	10^1	10^0
hundred thousands	ten thousands	thousands	hundreds	tens	ones

If the base number (10) were replaced with a 2, binary or Base-2 is the result, or:

$2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6$ which equals 1, 2, 4, 8, 16, and 32

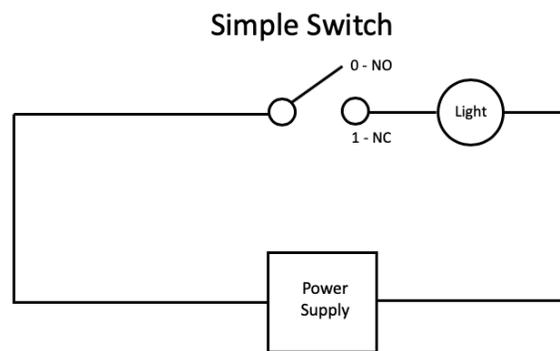
Base-2						
64	32	16	8	4	2	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
sixty-fours	thirty-twos	sixteens	eights	fours	twos	ones

Understanding this and applying the same theory of how to represent 24 in Base-10, we can obtain 24 in Base-2. There are no 64's or 32's in the number 24, so they are skipped. There is one 16 in 24. When 16 is subtracted from 24, there is a remainder of 8. Check to see whether there are any 8's in the remainder; there is one, which leaves us with a remainder of 0. Now even though there is no remainder left, the remaining numbers are checked. Are there any 4's in 0? No. Are there any 2's in 0? No. What about 1's? No. In the end, the number "24" contains one 16, one 8, zero 4's, zero 2's, and zero 1's, or 11000 in binary.

"24" in Binary					
2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	0
	1	1	0	0	0

How does a computer use binary numbers?

To answer this, it is essential to understand a few things. First, look at the number 11000. Notice that there are only 1's and 0's. This is the key concept to understand. Any number imaginable can be written as a binary number with only 1's and 0's. In the electronic world, the number 1 is equivalent to "on," and 0 is equivalent to "off." Look at the power switch on the back of your computer, or the power switch of any appliance. One (1) indicates that the device is powered on, and zero (0) means it is not. What does a number like 11000 mean? How about "on, on, off, off, off?" This concept describes exactly how switches work – Normally Open / Normally Closed – a binary process. In a simple switch, NO would indicate that electricity was not flowing, meaning the light would be off; and NC would indicate that electricity was flowing, meaning the light would be on.



A computer remembers the number 24 by turning the first two switches on and the next three off, thus creating a 5 switch pattern -"on, on, off, off, off". This is the basic way a computer works.

How does a computer remember a picture when there are no numbers in it?

A picture is actually a collection of millions of numbers in a few categories. These categories include color, intensity, and position. To understand this, a discussion about how light works is helpful. There are three primary colors of light: red, green, and blue. Look closely at a TV screen and thousands of little red, green and blue dots become visible. The screen blends these three colors to make the full spectrum of colors you see on the screen. Red, green and blue light combine together at full power create white light. Light works differently than paint, which if red, green and blue paint were mixed, would result in a brown-black hue.) The computer is able to turn each of the three lights on fully or partially as needed to create any color. It does this by assigning each of the three colors a numerical value between 1 and 256 (or 1 and 100000000 in binary) that is proportional to the required brightness. To make the color yellow, red and green lights are turned on to full power, and blue to about 1/3 power. So yellow would be represented in [red][green][blue] as:

[256][256][100] in decimal, or in binary, [100000000][100000000][1100100]

The only other parameter is the location on the screen where the yellow dot is to appear. This location is indicated by counting the number of dots or pixels from the top right corner of the screen. To locate a yellow dot one inch right and one inch down from the top right corner of the screen, with a screen having 300 pixels per inch, the computer annotation is represented by the number of pixels right and number of pixels down [pixels right], [pixels down], and then the numerical values for [red][green][blue]. The full numerical command for this one yellow dot would be:

[300] [300] [256] [256] [100] in decimal or

[100101100] [100101100] [100000000] [100000000] [1100100] in binary

One yellow dot on a screen requires 43 switches (one for each binary number). Every pixel on a screen is controlled by this process. Computers process these types of calculations and many more complex ones in order to work, but understanding how switches work is the basis to understanding computers.

Vocabulary

Base-2 Number System: A number system (also called binary system) that uses only zeros and ones to symbolize any number. The system is based on the number 2 to an “n”th power.

Base-10 Number System: A number system (also called the decimal system) using bundles of 10 that are represented by integers from 0 – 9 and the position of the integer determines how many 1’s, 10’s, 100’s, 1000’s, etc. the integer represents.

Binary Code: A computer language based on a Base-2 number system in which there are only two possible states, off and on, symbolized by 0 or 1.

Electrical Circuit: A simple circuit is a path or line through which an electric current flows.

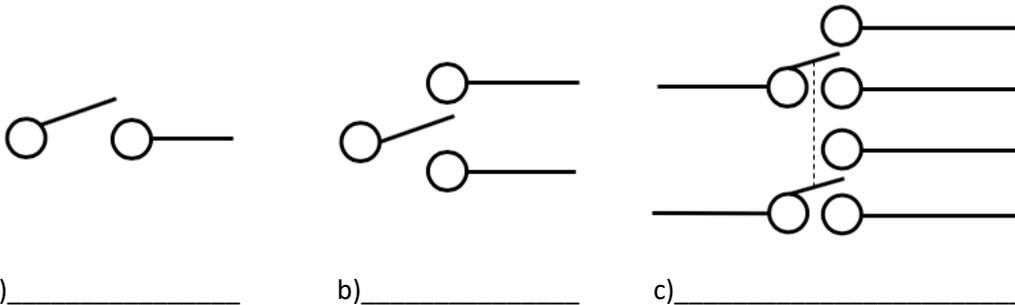
Pixel: A pixel, pel, or picture element is a physical point, or the smallest controllable element of a picture represented on a screen.

Switch: A device for making and breaking the connection in an electric circuit.

Terminal: Electrical connection points on a switch.

Test Your Knowledge

1. Write the number 231 in binary code.
2. Translate the number 11010 into Base-10.
3. In your own words, describe how an electrical switch is constructed.
4. Identify the following types of switches, according to the number of “poles” and “throws” each switch has:



Resources:

Videos:

Electricity – Explaining Electric Circuits: <https://www.youtube.com/watch?v=9yLPMhHE1hQ>

Explaining an Electrical Circuit: <https://www.youtube.com/watch?v=VnnpLaKsqGU>

The Learning Circuit – Switches: <https://www.youtube.com/watch?v=qTJdZsx7nvU>

The Learning Circuit – Poles, Throws, and Relays: <https://www.youtube.com/watch?v=ZSImeE-MCTc>

How Does a Switch Work?: <https://www.youtube.com/watch?v=iLbI7OKLO3M>

Why Do Computers Use 1’s and 0’s? Binary and Transistors Explained:
<https://www.youtube.com/watch?v=Xpk67YzOn5w>

See How Computers Add Numbers: <https://www.youtube.com/watch?v=VBDoT8o4q00>

Intro to Binary Numbers: <https://www.youtube.com/watch?v=b7pOcU1xMks>

Khan Academy – Introduction to Number Systems and Binary:
<https://www.khanacademy.org/math/algebra-home/alg-intro-to-algebra/algebra-alternate-number-bases/v/number-systems-introduction>

Binary Basics for Kids: <https://www.youtube.com/watch?v=2Z03p3yhGmk>

Binary Numbers for Kids: <https://www.youtube.com/watch?v=hvteVokz7jE>

Articles:

TRY Engineering: Electric Switches: <https://tryengineering.org/wp-content/uploads/elecswitch.pdf>

TRY Engineering: Flashlights and Batteries: <https://tryengineering.org/wp-content/uploads/flashlight.pdf>

MiniScience.com: Simple Electric Circuit: <https://old.miniscience.com/kits/KITSEC/index.html>

Hands-on Activity: Bulbs & Batteries in a Row:
https://www.teachengineering.org/activities/view/cub_electricity_lesson05_activity1