Input Devices:

Keyboard:

Keyboard use switches and circuits to translate a person's keystrokes into a signal a computer can understand. In this article we will explore keyboard technology along with different key layouts, options and designs.

Keyboard Basics

A keyboard's primary function is to act as an input device. Using a keyboard, a person can type a document, use keystroke shortcuts, access menus, play games and perform a variety of other tasks. Keyboards can have different keys depending on the manufacturer, the operating system they're designed for, and whether they are attached to a desktop computer or part of a laptop. But for the most part, these keys, also called keycaps, are the same size and shape from keyboard to keyboard. They're also placed at a similar distance from one another in a similar pattern, no matter what language or alphabet the keys represent.

Most keyboards have between 80 and 110 keys, including:

Typing keys

A numeric keypad

Function keys

Control keys

The typing keys include the letters of the alphabet, generally laid out in the same pattern used for typewriters. According to legend, this layout, known as QWERTY for its first six letters, helped keep mechanical typewriters' metal arms from colliding and jamming as people typed. Some people question this story -- whether it's true or not, the QWERTY pattern had long been a standard by the time computer keyboards came around.

**Mouse**:

Every day of your computing life, you reach out for your **mouse**whenever you want to move your cursor or activate something. Your mouse senses your motion and your clicks and sends them to the computer so it can respond appropriately.

In this article we'll take the cover off of this important part of the human-machine interface and see exactly what makes it tick.

**Evolution of the Computer Mouse**

It is amazing how simple and effective a mouse is, and it is also amazing how long it took mice to become a part of everyday life. Given that people naturally point at things -- usually before they speak -- it is surprising that it took so long for a good pointing device to develop. Although originally conceived in the 1960s, a couple of decades passed before mice became mainstream.

In the beginning, there was no need to point because computers used crude interfaces like teletype machines or punch cards for data entry. The early text terminals did nothing more than emulate a teletype (using the screen to replace paper), so it was many years (well into the 1960s and early 1970s) before arrow keys were found on most terminals. Full screen editors were the first things to take real advantage of the cursor keys, and they offered humans the first way to point.

**Light pens** were used on a variety of machines as a pointing device for many years, and graphics tablets, joy sticks and various other devices were also popular in the 1970s. None of these really took off as the pointing device of choice, however.

When the mouse hit the scene -- attached to the Mac, it was an immediate success. There is something about it that is completely natural. Compared to a graphics tablet, mice are extremely inexpensive and they take up very little desk space. In the [PC](http://computer.howstuffworks.com/pc.htm) world, mice took longer to gain ground, mainly because of a lack of support in the [operating system](http://computer.howstuffworks.com/operating-system.htm). Once Windows 3.1 made Graphical User Interfaces (GUIs) a standard, the mouse became the PC-human interface of choice very quickly.

# Inside a Mouse

The main goal of any mouse is to translate the motion of your hand into signals that the computer can use. Let's take a look inside a track-ball mouse to see how it works:

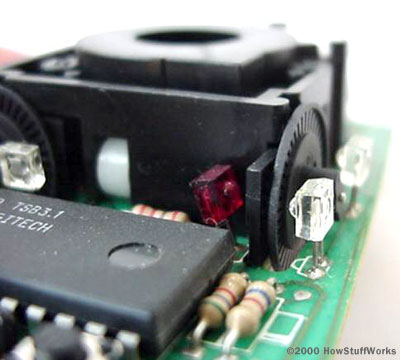
1. A **ball** inside the mouse touches the desktop and rolls when the mouse moves. **The underside of the mouse's logic board: The exposed portion of the ball touches the desktop.**
2. **Two rollers** inside the mouse touch the ball. One of the rollers is oriented so that it detects motion in the X direction, and the other is oriented 90 degrees to the first roller so it detects motion in the Y direction. When the ball rotates, one or both of these rollers rotate as well. The following image shows the two white rollers on this mouse:**The rollers that touch the ball and detect X and Y motion**
3. The rollers each connect to a **shaft**, and the shaft spins a **disk** with holes in it. When a roller rolls, its shaft and disk spin. The following image shows the disk: **A typical optical encoding disk: This disk has 36 holes around its outer edge.**
4. On either side of the disk there is an **infrared LED** and an **infrared sensor**. The holes in the disk break the beam of [light](http://science.howstuffworks.com/light.htm) coming from the LED so that the infrared sensor sees pulses of light. The rate of the pulsing is directly related to the speed of the mouse and the distance it travels. **A close-up of one of the optical encoders that track mouse motion: There is an infrared LED (clear) on one side of the disk and an infrared sensor (red) on the other.**
5. An **on-board processor chip** reads the pulses from the infrared sensors and turns them into binary data that the computer can understand. The chip sends the binary data to the computer through the mouse's cord.



**The logic section of a mouse is dominated by an encoder chip, a small processor that reads the pulses coming from the infrared sensors and turns them into bytes sent to the computer. You can also see the two buttons that detect clicks (on either side of the wire connector).**

In this **optomechanical**arrangement, the disk moves mechanically, and an optical system counts pulses of light. On this mouse, the ball is 21 mm in diameter. The roller is 7 mm in diameter. The encoding disk has 36 holes. So if the mouse moves 25.4 mm (1 inch), the encoder chip detects 41 pulses of light.

You might have noticed that each encoder disk has two infrared LEDs and two infrared sensors, one on each side of the disk (so there are four LED/sensor pairs inside a mouse). This arrangement allows the processor to detect the disk's **direction of rotation**. There is a piece of plastic with a small, precisely located hole that sits between the encoder disk and each infrared sensor. It is visible in this photo:



**A close-up of one of the optical encoders that track mouse motion: Note the piece of plastic between the infrared sensor (red) and the encoding disk.**

This piece of plastic provides a window through which the infrared sensor can "see." The window on one side of the disk is located slightly higher than it is on the other -- one-half the height of one of the holes in the encoder disk, to be exact. That difference causes the two infrared sensors to see pulses of light at slightly different times. There are times when one of the sensors will see a pulse of light when the other does not, and vice versa. [This page](http://www.4qdtec.com/meece.html) offers a nice explanation of how direction is determined.



**In this photo, you can see the sensor on the bottom of the mouse.**

# Optical Mice

Developed by Agilent Technologies and introduced to the world in late 1999, the optical mouse­ actually uses a tiny camera to take thousands of pictures every second.

Able to work on almost any surface without a mouse pad, most optical mice use a small, red [light-emitting diode](http://electronics.howstuffworks.com/led.htm) (LED) that bounces light off that surface onto a[complimentary metal-oxide semiconductor](http://electronics.howstuffworks.com/cameras-photography/digital/question362.htm) (CMOS) sensor. In addition to LEDs, a recent innovation are laser-based optical mice that detect more surface details compared to LED technology. This results in the ability to use a laser-based optical mouse on even more surfaces than an LED mouse.

Here's how the sensor and other parts of an optical mouse work together:

* The CMOS sensor sends each image to a digital signal processor (DSP) for analysis.
* The DSP detects patterns in the images and examines how the patterns have moved since the previous image.
* Based on the change in patterns over a sequence of images, the DSP determines how far the mouse has moved and sends the corresponding coordinates to the computer.
* The computer moves the cursor on the [screen](http://computer.howstuffworks.com/monitor.htm) based on the coordinates received from the mouse. This happens hundreds of times each second, making the cursor appear to move very smoothly.

Optical mice have several benefits over track-ball mice:

* No moving parts means less wear and a lower chance of failure.
* There's no way for dirt to get inside the mouse and interfere with the tracking sensors.
* Increased tracking resolution means a smoother response.

Tablet:

a **tablet PC** is a mobile computing device that's larger than a [smartphone](http://electronics.howstuffworks.com/smartphone.htm) or personal digital assistant. There's not a strict cutoff size for tablet devices -- the iPad line sports a screen size of just under 10 inches but other tablets can be larger or smaller. In general, if the computing device uses an on-screen interface and doesn't include a phone, it's a tablet.

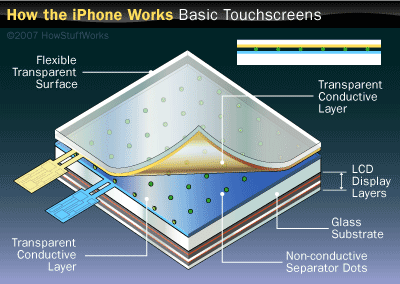
To confuse matters, some manufacturers produce hybrid devices that are part tablet, part laptop computer. The device might come with an attached keyboard --

The brain of a tablet is its [microprocessor](http://computer.howstuffworks.com/microprocessor.htm). Typically, tablets use smaller processors than full-fledged computers. This helps save on space and cuts down on heat generation. Heat is bad for computers -- it tends to cause mechanical failures.

Tablet computers typically draw power from a rechargeable battery. Battery life for tablets varies between models, with eight to 10 hours being the average. Some tablets will have replaceable batteries. But others, like Apple's [iPad](http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/ipad.htm) and iPad 2, don't allow you to switch out a battery without taking it to a store or voiding your warranty.

Depending on the manufacturer, a tablet computer may be underpowered on purpose. Computer CPUs execute commands in clock cycles. The more clock cycles a CPU runs per second, the more instructions it can process. Some tablets have **underclocked** processors, meaning the CPU is set to run fewer instructions per second than it's capable of executing. The reason for making a CPU underperform on purpose is to reduce heat production and conserve battery life.

 Page



**The Apple iPhone uses a capicitive touch-screen interface, as do many tablet computers.**

# Touch Screens and Tablets

There are two basic methods of creating [touch screens](http://computer.howstuffworks.com/question716.htm) for tablet devices: resistive screens and capacitive screens. Manufacturers have to choose between the two -- they don't work together.

**Resistive** systems detect a touch on a screen through pressure. Tablets that require a stylus often use resistive screens. But how does it work?

Resistive systems have a layer of resistive material and another layer of conductive material. Spacers hold the two layers apart. When the tablet is on, an electric current runs through both layers. If you put pressure on the screen, it causes the two layers to come into contact with one another. This changes the electrical field for those two layers.

Imagine you own such a tablet and you've decided you want to activate a game. You use your stylus to tap the game icon on your tablet's screen. The pressure from your touch causes the two layers in the resistive system to touch, changing the electric field. A microchip inside the tablet interprets this change in the field and translates it into coordinates on the screen. The tablet's CPU takes these coordinates and maps them against its [operating system](http://computer.howstuffworks.com/operating-system.htm). The CPU determines that you have activated the app and launches it for you.

Resistive screens can be susceptible to damage. If you use too much pressure, you may cause the resistive and conductive layers to be in constant contact. This will cause the tablet to misinterpret commands. Resistive screens also tend to have poorer resolution than capacitive screens.

A **capacitive** system also detects changes in electrical fields but doesn't rely on pressure. A capacitive system includes a layer of material that stores an electrical charge. When you touch a conductive material to this screen, some of that electrical charge transfers over to whatever is touching it. But the material must be conductive or the device won't register a touch. In other words, you can use anything to touch a resistive screen to register a charge but only conductive material will work on a capacitive system.

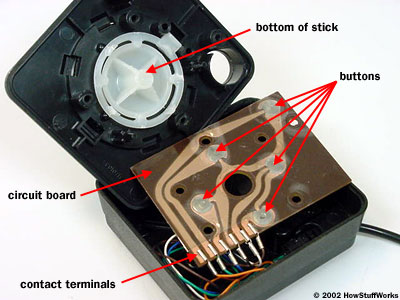
Capacitive systems tend to be more robust than resistive systems since you don't have to press down as hard to register a touch. They also tend to have a higher resolution than resistive systems.

Joystick



Joysticks pull off a really neat trick. They take something entirely physical -- the movement of your hand -- and **translate** it into something entirely mathematical -- a string of ones and zeros (the language of computers). With a good joystick, the translation is so flawless that you completely forget about it. When you're really engaged in a game, you feel like you're interacting with the virtual world directly.

In this article, we'll find out how several common joystick designs handle this translation. As we'll see, the technology has evolved a great deal from the first game console designs to the sophisticated "force feedback" models available today.



**An early Atari joystick**

# The Simplest System: Design

The basic idea of a joystick is to translate the movement of a plastic stick into electronic information a[computer](http://computer.howstuffworks.com/pc.htm) can process. Joysticks are used in all kinds of machines, including [F-15 fighter jets](http://science.howstuffworks.com/f-15.htm),[backhoes](http://science.howstuffworks.com/transport/engines-equipment/backhoe-loader.htm) and **wheelchairs**. In this article, we'll be focusing on computer joysticks, but the same principles apply to other sorts of joysticks.

The various joystick technologies differ mainly in how much information they pass on. The simplest joystick design, used in many early [game consoles](http://electronics.howstuffworks.com/video-game.htm), is just a specialized **electrical switch**.

This basic design consists of a stick that is attached to a plastic base with a flexible rubber sheath. The base houses a **circuit board** that sits directly underneath the stick. The circuit board is made up of several "printed wires," which connect to several **contact terminals**. Ordinary wires extend from these contact points to the computer.

The printed wires form a simple electrical circuit made up of several smaller circuits. The circuits just carry electricity from one contact point to another. When the joystick is in the neutral position -- when you're not pushing one way or another -- all but one of the individual circuits are broken. The conductive material in each wire doesn't quite connect, so the circuit can't conduct electricity.

Each broken section is covered with a simple plastic button containing a tiny metal disc. When you move the stick in any direction, it pushes down on one of these buttons, pressing the conductive metal disc against the circuit board. This **closes the circuit** -- it completes the connection between the two wire sections. When the circuit is closed, electricity can flow down a wire from the computer (or game console), through the printed wire, and to another wire leading back to the computer.