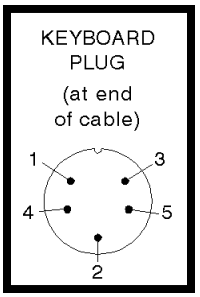
No. 57, Aug 1998

How Keyboards Work

- by Stan Kaplan, WB9RQR
105 Martin Drive
Port Washington, WI 53074-9654
(414) 284-9346
WB9RQR @ N9PBY.EN63BI.WI.USA.NA
skaplan@mcw.edu

Keyboards are neat devices. They represent a way for us humans to use our most dexterous appendages to tell a machine what we want. Just about everyone uses one, but few really know how they work. Lets see if I can do a decent job of explaining it to you.

First, pull the keyboard plug out of your computer and take a look at its end. Just about all IBM PC, XT, AT and compatible keyboards sport the same connector, a 5-pin DIN plug. Only a few brands and the IBM PS/2 use a slightly different, 6-pin plug. Now look at the keyboard itself. Yours has between 83 and 102 keys, and each key is simply a SPST switch. Now, fellow hams, it should be clear that 5 wires cannot directly carry the on/off state of 100 or so SPST switches. If they were wired directly, as you or I would wire the on/off switch of a power supply, the keyboard cable (about 3/8 inch in diameter) would need to be expanded to nearly 8 inches in diameter! Obviously, there must be some other way that the state of the 100 switches is sent to the computer.



The secret is that each keyboard is itself a miniature computer. Inside each keyboard case is a microprocessor chip whose job it is to look at the state of each key every now and then. Every now and then to a keyboard microprocessor means every couple of microseconds - close to a million times per second! If it finds, during scanning of the keys, that one has been depressed or released, it then generates a one-byte (8 bits) "scan code". It sends this code down the cable to the computer as serial data - much like a single file line of people walking down a narrow hallway.

Well, if this is all true, then the microprocessor should need only one or two (signal and ground) wires to send stuff to the computer. Here is the assignment of the five pins:

PIN	FUNCTION
1	Keyboard clock signal to the computer
2	Data line to the computer
3	Unused
4	Ground
5	+5 volt supply to the keyboard from the computer

See? Pin 2 carries the data and pin 4 is the ground. Pin 5 provides juice to the keyboard from the computer, so that the microprocessor feels well nourished and the LEDs will work. The only other useful wire is connected to pin 1. That's a sort of metronome line - tick tock, tick tock - from the clock circuit in the keyboard, so the

computer and keyboard can be in synchrony when data is passed. Ever look inside a keyboard? There is a crystal in there, along with the microprocessor. The crystal controls the beat rate of the clock. Neat as peaches!

Now, what about this "scan code" business? How exactly does that work? When you press, for example, the letter A key, the keyboard microprocessor detects that you have done so. It sends the code 1Eh (one-E; the h stands for hexadecimal notation) down the wire attached to pin 2. When you let go of the key, it also detects that but sends a different code, 9Eh, down the wire. The message is received by another microprocessor in the computer called the keyboard controller chip. The keyboard controller then tells the CPU (386, 486 or pentium chip) that a keyboard event has occurred by sending the CPU a hardware interrupt. The CPU pauses from whatever it was doing and tells the ROM (Read Only Memory) BIOS (Basic In/Out System) about it. The BIOS then reads the scan code and takes action depending upon what key was pressed, or if it was released, and what else is going on at the time. If, for example, you are at a DOS prompt and you pressed and released the letter A, it echoes the letter to the screen and also puts a copy into a buffer for temporary storage.

Whew! A lot of things going on when you simply press or release a key! But now you can understand why, when you hold a key down, the character is repeated. The keyboard controller chip knows you have pressed a key, but that you haven't released it yet. It tells that to it's buddy the CPU, who tells it to the BIOS, and the BIOS knows that it is supposed to repeat characters until the key is released. When you finally release the key, the BIOS gets the new scan code message and it quits sending characters to the screen.

There are some variations on this scheme. When you press and hold the Shift, Ctrl or Alt key, a unique scan code is generated and sent but the BIOS takes no action until you press another key, for example the letter A. When you finally press the letter A, the A's 1Eh is preceded by the unique scan code and sent to the keyboard buffer. Unique means unique. The left shift generates 2Ah when pressed and AAh when released, while the right shift key sends 36h on make and B6h on break. You see, your computer really does know the difference when you press the left Alt key versus the right! Only a few key pairs are the same. Pressing the 5-key at the top of the keyboard generates the same scan code as pressing the 5 on the number pad (provided the Num Lock is on). In other words, the computer can't tell the difference when you press either of these two keys. But it sure can when you press the left Alt versus the right, or the left Ctrl versus the right.

Some other keys change their scan codes completely if the shift key is pressed and held first. For example, Num Lock is E0h 35h when pressed, but AAh if the shift key is held down first. On release, Num Lock sends E0h B5h, but E0h B5h 2Ah if the shift key is held down first. Still other keys change their code if a Ctrl key is held down first, and send an even different code if preceded by holding down an Alt key. Probably the weirdest scan code belongs to the Pause key, a key rarely used by anyone. It sends out a long string (E1h 1Dh E1h 9Dh C5h) when pressed, and nothing at all when released. Also pretty strange from a use point of view is the Sys Req key, added in 1984 for some function that never panned out. But, it is still there today, with a scan code all its own. By the way, you can assign your own special functions to the weird keys, in DOS and even Windows 95.

The keyboard is truly an ingenious device. Today's models are very reliable and the prices have plummeted. You can get a fully functional model for about $10, making it (unfortunately) another disposable item. Many find their way into landfills, and will probably be cussed by future landfill miners for breaking their equipment.

After this exploration of keyboards, let me leave you with a prediction. These devices are clearly on their way out. When you buy a new computer a few years from now, it will not come with a keyboard, just a microphone. You will speak your commands, or just point on the screen from across the room with a gadget like a laser pointer. That technology is already with us. And, of course, the computer will speak when a reply is warranted, just like Hal in 2001. A few years after that, you won't need a microphone or a pointing device. The computer will be tuned into your brainwaves, so you will just need to think your commands. Happy computing!