KEYBOARDS

The QWERTY keyboard is perhaps the most common device used to input data to a computer. Regardless of high-speed serial links or high-density drives, almost all information is inputted using a keyboard. Although voice recognition as a form of computer input has been highly touted, it is far from being used as a major input source for the majority of computer users. There have been many variations suggested for keyboards. Several new ergonomic designs have arisen that have had some success where the keyboard is curved to better suit the position of the hands and wrists. There have also been suggestions to rearrange the keys so that the most commonly used keys would be directly under the resting positions of the fingers over the keyboard. To date, there has been no serious consideration of an alternative keyboard arrangement in the keyboard industry. Therefore, it seems certain that the standard QWERTY keyboard will be the de facto standard for keyboards with the most significant variations being that of ergonomic design.

As a historical matter, there have been keyboards that have had other keypad placements other than the QWERTY system. In fact, typists who trained on these systems could achieve significantly higher typing speeds due to the fact that the most commonly used keys were placed underneath the resting position of the typist's fingers. On a QWERTY keyboard, the letters "e" and "i" are on the row of keys above the resting position of the fingers. This arrangement was originally proposed to purposefully slow down the typist as the old mechanical systems could not keep up. In other words, two or more letters would physically get stuck, requiring the operator to stop and clear the jammed keys.

Keyboards have, nonetheless, evolved to contain many new features that add flexibility and reliability when compared with standard mechanical keyboards. As the vast majority of keyboards for modern computers consist of a microcontroller and a matrix of switches, a discussion of keyboards will begin at that point. Figure 1 is offered to demonstrate how a microcontroller and a switch matrix forms a keyboard. In this example a program continuously scans the keyboard matrix. One of the outputs is driven to a logic 0 state and the other three outputs are put in an inactive state, that is, driving no level at all. The four input lines are scanned to check for a logic level. If no keys are pressed, then all of the inputs will read a logic 1. Each column of the matrix is tested in order. If a key is pressed, the corresponding input will read a logic 0, indicating a key closure. For example, assume that key "c2r1" is depressed. The only case where any of the inputs will receive a logic 0 signal is when output B is active low and the input level of input 1 is being checked. In all other cases, all

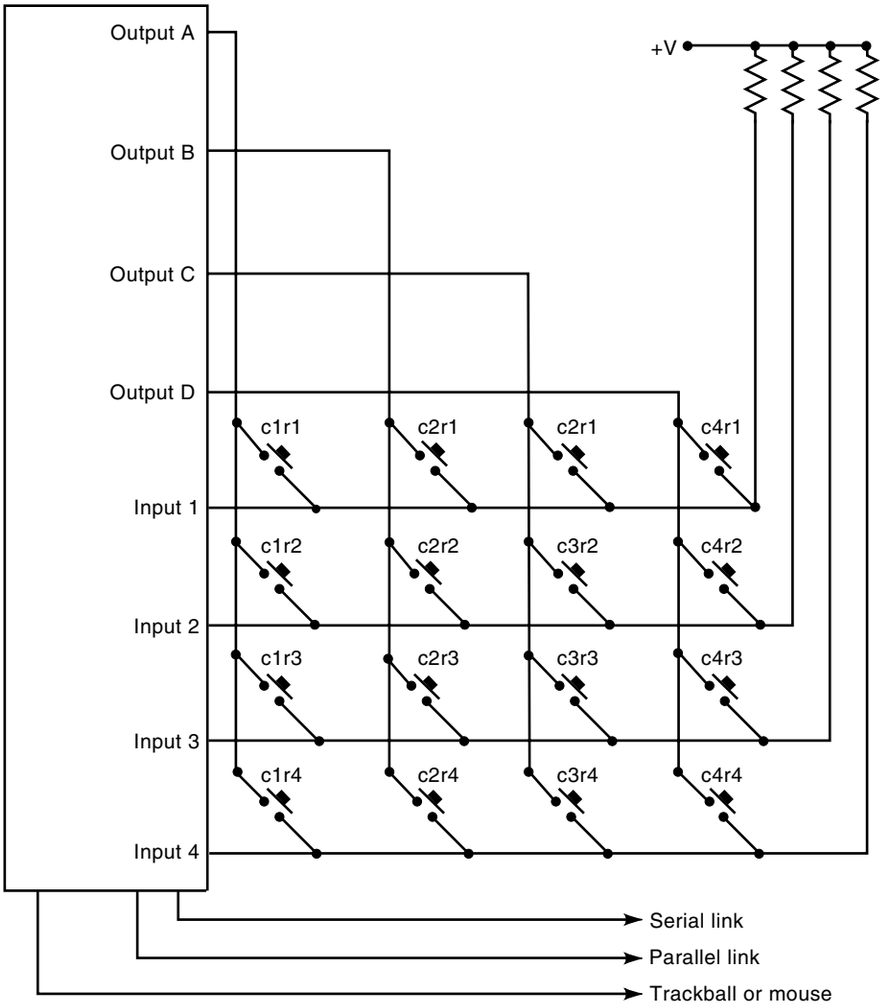

Figure 1. Typical keyboard with microcontroller.

inputs will receive a logic 1. Once a key is found to be pressed, the scanning program can jump to the appropriate routine to take some further action. Once a key is found to be depressed, the program can take a specific action, such as outputting the ASCII code for the key either in serial or parallel form.

There are other considerations with electronic keyboards. For instance, it is very common for a fast typist to have more that one key depressed at a time. In fact, three or four keys may be depressed simultaneously. Therefore, it is important for keyboard controllers to keep a history file of what keys have been depressed but have not been released. A second recognition of a key must not be allowed to occur until the key has been released. If all four keys in row 1 are pressed, input 1 will always be driven with a logic 0 due to the fact that one of the four outputs A, B, C, or D will be low and thus drive input 1 low. It is necessary that the scanning routing individually check each key in each row to check for multiple closures and discard any results that indicate that the key has not been released since the last closure.

Debouncing is another problem with keyboards. As a key is pressed, there is a very brief time when the actual key switch will go through some mechanical bouncing. It is very possible for a key to be closed on one scan, open on the next, and closed on the next scan. This problem is generally handled by requiring that the key be detected as closed for a certain time period, perhaps 50 ms. This debounce time can be very variable based on the actual type of mechanical switch being used. The debounce time must be empirically determined by actual measurement.

As a practical matter, the manufacture of keyboards is extremely cost competitive. Generally, the most economical device that can perform the required tasks will be chosen for a design. The implication is that microcontrollers without serial ports will be used and a UART function will be implemented with a software routine.

There are many new features that have been added to basic keyboards. For instance, infrared links between the keyboard and the computer have been implemented so that a physical connection does not have to be maintained. However, the basic function and operation of the keyboard is still the same as before.

Because all modern computer keyboards are controlled by some type of intelligent device, essentially any type of function can be performed. For instance, multiple keyboard fonts can be maintained. Programmable sequences can be performed for function keys. A sequence of bytes can be sent to

the computer for special keys. However, the basic function of the keyboard and the way it is implemented, that is, a scanning routine checking for key closures, remains fundamental.

BIBLIOGRAPHY

1. S. W. Hobday, The Maltron keyboards, *Colloquium on Interfaces—The Leading Edge,* IEE, London, UK; 1996.
2. R. W. Soukoreff and I. S. Mackenzie, Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard, *Behaviour Inf. Technol.* **14**: (6), 370–379, 1995.
3. J. Noyes, QWERTY—the immortal keyboard, *Comput. Control Eng. J.,* **9**: (3), 117–122, 1998.
4. R. Blake, Core ASICs in mass-market designs, *Electronic Product Design,* **14**: (10), 61–62, 1993.
5. I. Gilad and S. Harel, Keyboard operations in a negative slope design, *Design Computing Syst.: Cognitive Considerations, Proc. 7th Int. Conf. Human-Computer Interaction* (HCI International '97). Amsterdam, Netherlands: Elsevier, 1997, pp. 611–614.

C. MELEAR
Motorola

KILN, CEMENT. See CEMENT INDUSTRY.

KINEMATICS, ROBOT. See ROBOT KINEMATICS.