Step By Step

When it is necessary to measure linear or angular movement by means of an optical sensor, binary coding proves fallible, so the Gray code was devised

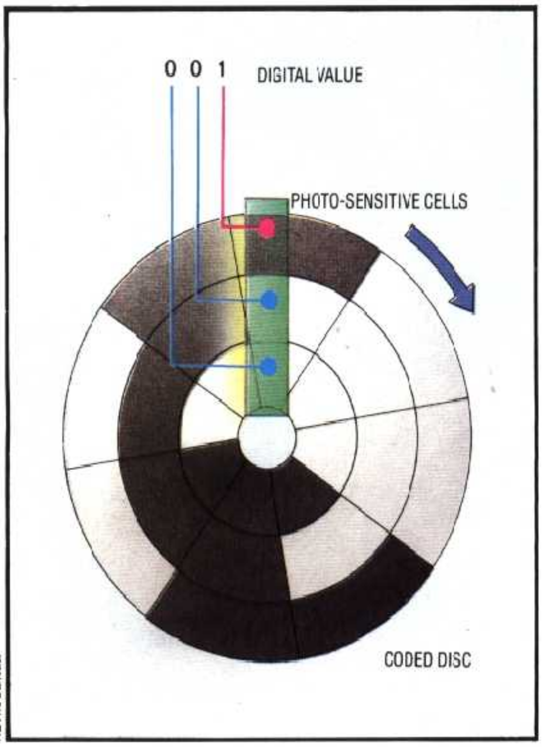
Odd One Out
This table shows the binary and Gray code equivalents to decimal 0 to 15.

Decimal	Binary	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

There are many tasks in which the physical position of a movable object must be precisely determined and relayed to a computer. In robotics, for example, the computer must be aware of the positions and orientations of all its limbs; and in computer-controlled machine tooling, the position of the milling table must be accurately determined. How then can a position be converted into a binary value for computer processing?

One method is to use an analogue system. This could involve connecting the moving device to a variable resistance and then feeding the resulting voltage into an analogue-to-digital convertor (or directly into the analogue port if your computer has one). However, this is not a very accurate system, and the mechanical parts are prone to wear and tear.

The alternative is to print a binary code onto the moving device and read this directly into the computer. The code is usually printed as a pattern of black and white blocks along the top of the device, and is read by means of a light source shining onto a pattern in conjunction with a line of photo-sensitive cells, each responsible for one digit in the binary pattern. As the workpiece moves, the pattern under the light cells changes and this gives a binary output that defines the device's position. In addition to linear arrangements, radial patterns are also used to encode angular movement, such as that of a joint in a robot arm.



Angle Of View
The angular position of a device can be read into a computer by means of a disc on which a code is printed. A light shines on the device, and a line of photo-sensitive cells detect the pattern. A parallel digital signal is produced that changes as the device moves. The problem with using binary as the code is that if the disc stops on the join between two values, a meaningless result can be produced. Gray code overcomes this problem

Problems arise, however, when the device moves from one binary code to another — and particularly if it should come to rest halfway between two of them. The printing is accurate to only a finite tolerance, and when the device stops on a join between two codes, the light cells could pick either of them to read. If the workpiece comes to rest with the light cells shining on the join between binary position 11 (1011) and 12 (1100), for example, then it is only the most significant bit (that is, the leftmost 1) that can be relied upon to give the correct output, whereas the other three light cells will have conflicting values to read. There are certain situations when all the bits change, such as the join between binary seven (0111) and eight (1000), so the minor inaccuracies in printing could produce incorrect readings on all the cells. The result could be a totally false value for the position, and the computer would have no way of knowing that this had happened. The consequences could be disastrous.

What is needed therefore, is an alternative counting system to binary, where only one bit changes with each increment. This will mean that only one bit can be in doubt on any join, and the output can be in error by no more than one position. This alternative code is called the Gray code, and is uniquely determined by these requirements: moving from one value to the next, only a single bit changes, and this should be the rightmost bit that will still result in a unique pattern. So if we start with 0000, as with binary, the number one will be represented by 0001. However, to represent two, we must change the second bit to get 0011. For three it is now possible to change the first bit, and get 0010. Note how this differs from the binary sequence for the same numbers: 0000, 0001, 0010, 0011.

The panel shows this process extended to the equivalent of decimal 15, together with the binary equivalents for reference. As an exercise you might like to work out the Gray code values beyond this.

Computers could be designed to perform arithmetic and function internally in Gray code, but this would be inefficient and unnecessary. Some means is therefore needed to convert from Gray to binary, which could be done in hardware or software.