Electrical Tech Note — 536

Biosystems &Agricultural Engineering Department
Michigan State University

Numbering Systems

Several types of numbering systems are used with digital equipment other than the decimal system. It is useful to the technician to understand the application of these numbering systems and know how to convert from one system to another. This Tech Note describes the types of numbering systems commonly used with digital electronic devices and how to convert numbers from one system to the other systems.

Decimal System: With this system each digit location can have up to ten values ranging from 0 to 9. When that digit location reaches 9 the next higher number results in a value of 1 being carried over to the next digit location to the left and the first location is reset to 0. This is called the base 10 numbering system and each location is a power of 10.

$10^n \ldots . 10^8, 10^7, 10^6, 10^5, 10^4, 10^3, 10^2, 10^1, 10^0$

Binary System: With this system each digit location can have only two values which are 0 and 1. When that digit location reaches 1 the next higher number results in a value of 1 being carried over to the next digit location and that location is reset to 0. This is called the base 2 numbering system and each location is a power of 2.

$2^n \ldots . 2^8, 2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0$

To convert a binary number to a decimal equivalent multiply the value in each digit location by the value of that digit according to the following example. Convert the 8 bit (one byte) binary number 0110 1101 to a decimal equivalent.

Place the binary number in the proper location as stated in the problem, multiply each digit by the value of that digit and sum the values. In this case the decimal equivalent is 109.

$2^7 = 128$	×	0	=	0
$2^6 = 64$	×	1	=	64
$2^5 = 32$	×	1	=	32
$2^4 = 16$	×	0	=	0
$2^3 = 8$	×	1	=	8
$2^2 = 4$	×	1	=	4
$2^1 = 2$	×	0	=	0
$2^0 = 1$	×	1	=	1

The next task is to be able to convert a decimal number into it's binary equivalent. There is an algorithm that works well for this conversion. Remember in this case that the base is 2. The algorithm is best explained using an example. Convert the decimal number 87 into a binary equivalent.

Divide the number by the base and write down the answer. If the division is not even there will be a remainder which for the binary system is only a 1. Take the answer from each division and repeat the division process until the value has been reduced to zero. The remainders will be the binary number starting at the top as the least significant digit which in the final number will be the digit to the right.

87	÷	2	=	43 and remainder 1	⇦ least significant digit
43	÷	2	=	21 and remainder 1	
21	÷	2	=	10 and remainder 1	
10	÷	2	=	5 and remainder 0	
5	÷	2	=	2 and remainder 1	
2	÷	2	=	1 and remainder 0	
1	÷	2	=	0 and remainder 1	

The 8 bit binary equivalent of 87 is 0101 0111. Since the value is transferred as an 8 bit number and the algorithm stopped at 7 digits, the most significant digit which is the one to the left is simply 0.

ASCII Code: Since digital electronic equipment must operate in binary it is necessary to be able to convert alphanumeric letters, numbers, and symbols into a binary code. The American Standard Code for Information Interchange (ASCII) uses seven bits to represent all letters, numbers and symbols on a standard keyboard plus a number of control functions. There are 128 ASCII codes, and since the codes can be communicated with one byte of information consisting of two 4-bit nibbles, they are frequently represented as two hexadecimal numbers (one hex number for each nibble). Most any digital electronics text will provide a table of ASCII codes or the table can easily be found on the internet.

Hexadecimal System: Writing instructions for a digital electronic device that operates in binary is a tedious task where mistakes can easily be made when looking at long strings of zeros and ones. Since a binary nibble of 4-bits is used to count up to sixteen (0 to 15), the task of communicating with electronic equipment can be simplified by using the hexadecimal system where one character represents four binary characters. This does create a problem where the two digit decimal numbers from 10 to 15 must be represented in some single digit form. In the hexadecimal system those numbers are simply represented by the letters of the alphabet as follows: 10 (A), 11 (B), 12 (C), 13 (D), 14 (E), and 15 (F).

With this system each digit location can have sixteen values which range from 0 to 15. When that digit location reaches 15 the next higher number results in a value of 1 being carried over to the next digit location to the left and the first location is reset to 0. This is called the base 16 numbering system and each location is a power of 16.

$$16^n \ldots .\ 16^8, 16^7, 16^6, 16^5, 16^4, 16^3, 16^2, 16^1, 16^0$$

To convert a hexadecimal number to a decimal equivalent multiply the value in each digit location by the value of that digit according to the following example. Convert the 3 digit hexadecimal number 1D7 to a decimal equivalent.

Place the hexadecimal number in the proper location as stated in the problem, multiply each digit by the value of that digit and sum the values. In this case the decimal equivalent is 471.

$16^3 = 4096$	×	0	=	0	
$16^2 = 256$	×	1	=	256	
$16^1 = 16$	×	D (13)	=	208	
$16^0 = 1$	×	7	=	7	(sum is 471)

The next task is to be able to convert a decimal number into it's hexadecimal equivalent. There is an algorithm that works well for this conversion. Remember in this case that the base is 16. The algorithm is best explained using an example. Convert the decimal number 390 into a hex equivalent.

Divide the number by the base and write down the answer. If the division is not even there will be a remainder which for the hexadecimal system will be a number from 1 to 15. Take the answer from each division and repeat the division process until the value has been reduced to zero. The remainders will be the hexadecimal number starting at the top as the least significant digit which in the final number will be the digit to the right.

390	÷	16	=	24 and remainder 6	⇦ least significant digit
24	÷	16	=	1 and remainder 8	
1	÷	16	=	0 and remainder 1	

The 3 digit hex equivalent of the decimal number 390 is hex 186. The most significant digit which is the one to the left is 1 in this case.

Octal System: Some electronic equipment operates in groups of three binary bits which means there are eight possible numbers ranging from 0 to 7. When the octal system is used one octal digit represents three binary bits. With this system each digit location can have eight values which range from 0 to 7. When that digit location reaches 7 the next higher number results in a value of 1 being carried over to the next digit location to the left and the first location is reset to 0. This is called the base 8 numbering system and each location is a power of 8.

$8^n \ldots . \; 8^8, 8^7, 8^6, 8^5, 8^4, 8^3, 8^2, 8^1, 8^0$

To convert an octal number to a decimal equivalent multiply the value in each digit location by the value of that digit according to the following example. Convert the 3 digit octal number 437 to a decimal equivalent.

Place the octal number in the proper location as stated in the problem, multiply each digit by the value of that digit and sum the values. In this case the decimal equivalent is 287.

$16^3 = 512$	×	0	=	0	
$16^2 = 64$	×	4	=	256	
$16^1 = 8$	×	3	=	24	
$16^0 = 1$	×	7	=	7	(sum is 287)

The next task is to be able to convert a decimal number into it's octal equivalent. There is an algorithm that works well for this conversion. Remember in this case that the base is 8. The algorithm is best explained using an example. Convert the decimal number 388 into an octal equivalent.

Divide the number by the base and write down the answer. If the division is not even there will be a remainder which for the octal system will be a number from 1 to 7. Take the answer from each division and repeat the division process until the value has been reduced to zero. The remainders will be the octal number starting at the top as the least significant digit which in the final number will be the digit to the right.

388 ÷	8	=	48 and remainder 4	⇦ least significant digit
48 ÷	8	=	6 and remainder 0	
6 ÷	8	=	0 and remainder 6	

The 3 digit hex equivalent of the decimal number 388 is octal 604. The most significant digit which is the one to the left is 6 in this case.

Binary Coded Decimal (BCD): A digital device will crunch numbers in binary, but when sending numbers to a decimal readout device four lines or bits are required, but only 10 (0 to 9) digits are needed not 16 (0 to 15). A special form of binary that uses four bits that only count up to 9 then resets back to zero is called binary coded decimal (BCD). Each decimal digit is represented by a packet of 4 bits with this highest number as 9. For example the number 952 is represented by four packets of 4 bits each which are 1001 0101 0010.

Gray Code: The most common use of this system is where an encoder on a turning shaft provides a digital code that specifies the angle of movement. If a four digit code is used there will be 16 possible increments which divided into 360° means there will be 22½° per digit. This angle is cut in half for each additional digit. For this example only four digits will be used, but in actual applications more digits are frequently used.

The gray code looks like binary, but it is different. Encoders used in the field have been prone to errors when using a binary numbering system. With the gray code only one digit changes as the numbers advance. For example the binary number for 3 is 0011. The binary number for 4 is 0100. Note that three of the digits changed value simply by changing from 3 to 4. In the gray code only one digit is permitted to change as the numbers progress. In the gray code the number 3 is represented by 0010 and the number 4 is represented by 0110. Note that only one digit changed. The entire gray code for the numbers 0 through 15 are represented as follows:

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

Conclusion: It is important to learn how the basic numbering systems used in electronic computing operate and how to convert from one system to another. In these examples all conversions were between decimal and the other numbering system. Sometimes it is easy to make a direct conversion from one to the other, and sometimes it is easier to first convert to decimal, then convert to the other system.