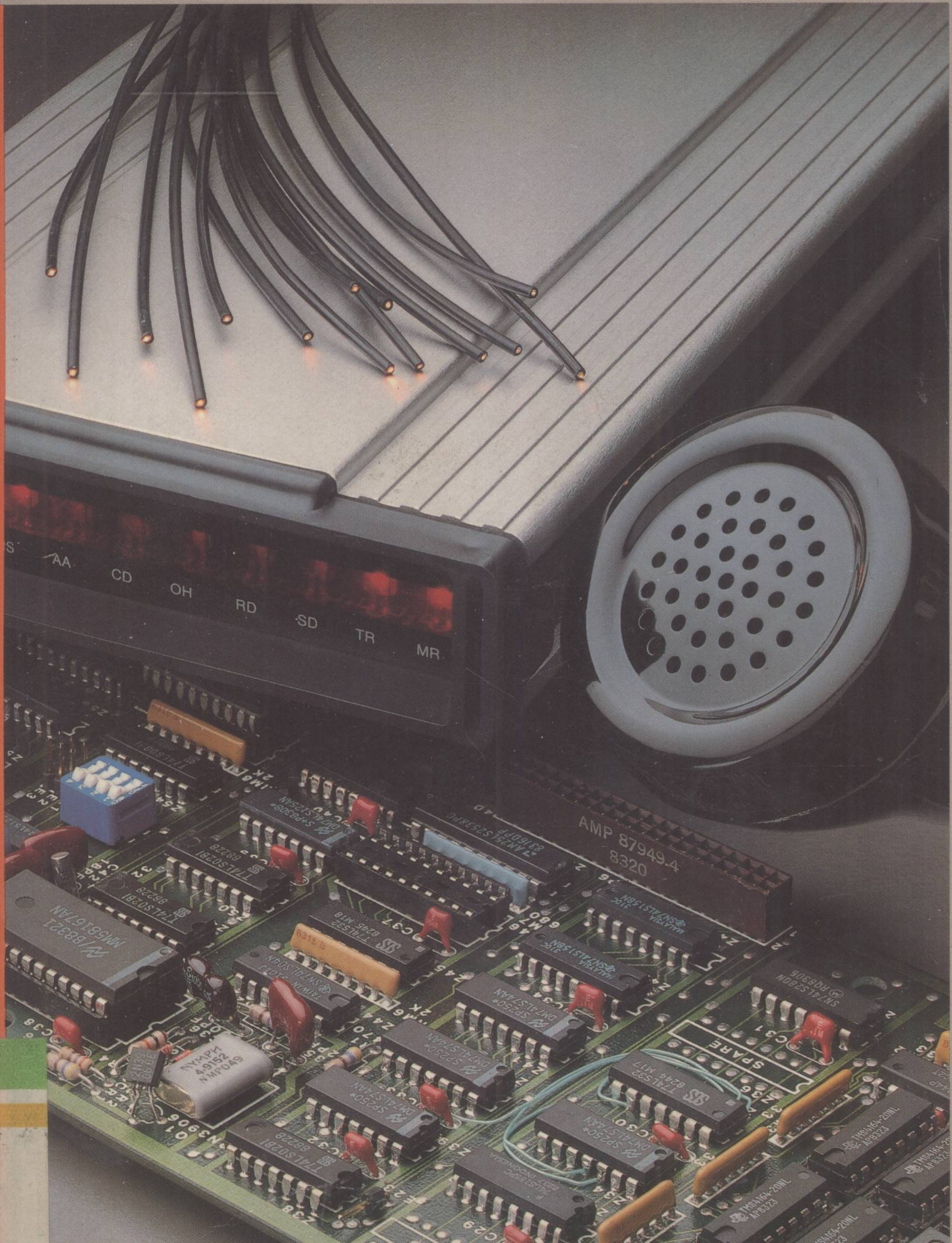
Understanding Data Communications



Data Communications — the transmission of words or symbols from a source to a destination — is no longer exclusive to the business world. Its pervasive impact can be felt everywhere. Learn the basic principles, and a whole lot more, in this fully illustrated, easy-to-understand format that's ideal for self-paced, individualized learning.

By:
George E. Friend
John L. Fike
H. Charles Baker
John C. Bellamy

The Texas Instruments Learning Center

Understanding Data Communications

Written by: George E. Friend, M.S.
Visiting Industrial Professor of Electrical Engineering
Southern Methodist University
Staff Consultant, Texas Instruments Information Publishing Center

John L. Fike, Ph.D., P.E.
President, Communications Enterprises, Inc.
Adj. Assoc. Professor of Electrical Engineering
Southern Methodist University
Staff Consultant, Texas Instruments Information Publishing Center

H. Charles Baker, Ph.D., P.E.
Adj. Professor of Electrical Engineering
Southern Methodist University
Staff Consultant, Texas Instruments Information Publishing Center

John C. Bellamy, Ph.D.
Director of Switching Systems Development
United Technologies Corporation
Adj. Professor of Electrical Engineering
Southern Methodist University
Staff Consultant, Texas Instruments Information Publishing Center

with contributions by: Gerald Luecke, Managing Editor
Charles W. Battle, Editor
Texas Instruments Information Publishing Center

TEXAS INSTRUMENTS
P.O. BOX 225012, MS-54 • DALLAS, TEXAS 75265

This book was developed by:

The Staff of the Texas Instruments Information Publishing Center
P.O. Box 225012, MS-54
Dallas, Texas 75265

For marketing and distribution inquire to:

Orm F. Henning
Marketing Manager
P.O. Box 225012, MS-54
Dallas, Texas 75265

About the cover:

Represented on the cover are the components of modern day data communications: a telephone, a modem, optical fiber cables, integrated circuits, and the interconnecting printed wiring boards used to assemble the equipment.

Word Processing:

Betty Brown

Design and artwork by:

Plunk Design

ISBN 0-89512-158-1
Library of Congress Catalog Number: 84-50867

Preface

The communication of information of all kinds by means of binary signals – the ones and zeros that are used by computers – has gained such an important place in American society that it is fair to say that successful operation of the U.S. economy would be impossible without it. Information sent using data communications controls a major portion of the long distance telephone network, enables the rapid authorization of credit purchases and cashing of checks, and provides for the inventory and ordering of goods in stores of all kinds from fast-food franchises to photo finishers. Data communications technology allows the simultaneous printing in widely separated cities of national magazines and newspapers, whose entire text and picture copy is transmitted from a central source to the city in which they are to be printed on a daily basis.

A basic knowledge of such a pervasive force in our lives is as necessary for all of us as some knowledge of the workings of another transportation system – the automobile. We have attempted to cover the basic principles of data communications and to explain those areas of application which have a daily impact: communications between terminals and computers, including local area networks and packet networks; communication of telephone conversations and control of the telephone network; and the use of light rays and satellites for carrying the ever-increasing volume of data helping to make our lives more comfortable and sometimes a great deal more confusing.

This book is arranged somewhat like a textbook. Each chapter starts by saying what it covers, ends by saying what has been covered, and provides a short multiple-choice quiz on the chapter contents. Like other books in the series, this book builds understanding step-by-step. Try to master each chapter before going on to the next one.

If your objective is to obtain general information about the subject, the more detailed portions of the text and the quiz can be bypassed. This will allow you to absorb the information more easily and, hopefully, enjoy the style of presentation. If you need more basic information about the telephone system and telephone electronics, a companion book in the series titled *Understanding Telephone Electronics* should be helpful. For more detailed and more advanced information about data communications, refer to the books listed in the bibliography.

G.F.
J.F.
C.B.
J.B.

Table of Contents

An Overview of Data Communications

ABOUT THIS BOOK

This book is intended to explain how data communications systems and their various hardware and software components work. It is not intended to tell you how to build the equipment, nor will it tell you how to write computer programs to transmit data between two computers. What can be learned is a basis for understanding data communications systems in general, and some tips for setting up your own system to communicate between your personal or professional computer and another personal computer, a data base service, or an electronic bulletin board.

To understand data communications, it is necessary to have some understanding of the telephone channel because it is the medium used by most data communications systems to move information from one place to another. Therefore, part of this book is devoted to describing the public telephone network and the equipment used to interface between it and a computer.

ABOUT THIS CHAPTER

This chapter presents some history of data communications and a general description of a data communications system. Explanations of bits, bytes, two-state communications systems, and codes help lay the foundation for further discussions in the remainder of the book.

WHAT IS DATA COMMUNICATIONS?

Data communications is often referred to as computer communications due to the ever increasing use of computers and their support equipment.

Data communications is the process of communicating information in binary form between two points. Data communications is sometimes called computer communications because most of the information interchanged today is between computers, or between computers and their terminals, printers or other peripheral devices. The data may be as elementary as the binary symbols 1 and 0, or as complex as the characters represented by the keys on a typewriter keyboard. In any case, the characters or symbols represent information.

WHY IS DATA COMMUNICATIONS IMPORTANT?

It is important for us to understand data communications because of its significance in today's world. Data communications is commonly used in the world of business and it is being used more and more in homes as well. Whether it is the transmission of bank account information from a central computer to a convenient electronic teller machine, or the downloading of a video game from a computer bulletin board to a home computer's memory, data communications is becoming an integral part of our daily activities.

THE FIRST DATA COMMUNICATIONS SYSTEMS

Modern data communications involves the use of electrical or electronic apparatus to communicate information in the form of symbols and characters between two points. Since electricity, radio waves, and light waves are all forms of electromagnetic energy, it is stretching things only a bit to say that early forms of communication, such as the puffs of smoke from the Indian's signal fire (*Figure 1-1*), or reflections of sunlight from a hand-held mirror, also were forms of the same type of data communications. To carry this idea further, think of the puffs of smoke as discrete symbols, just as the symbols used in today's communications systems are discrete.

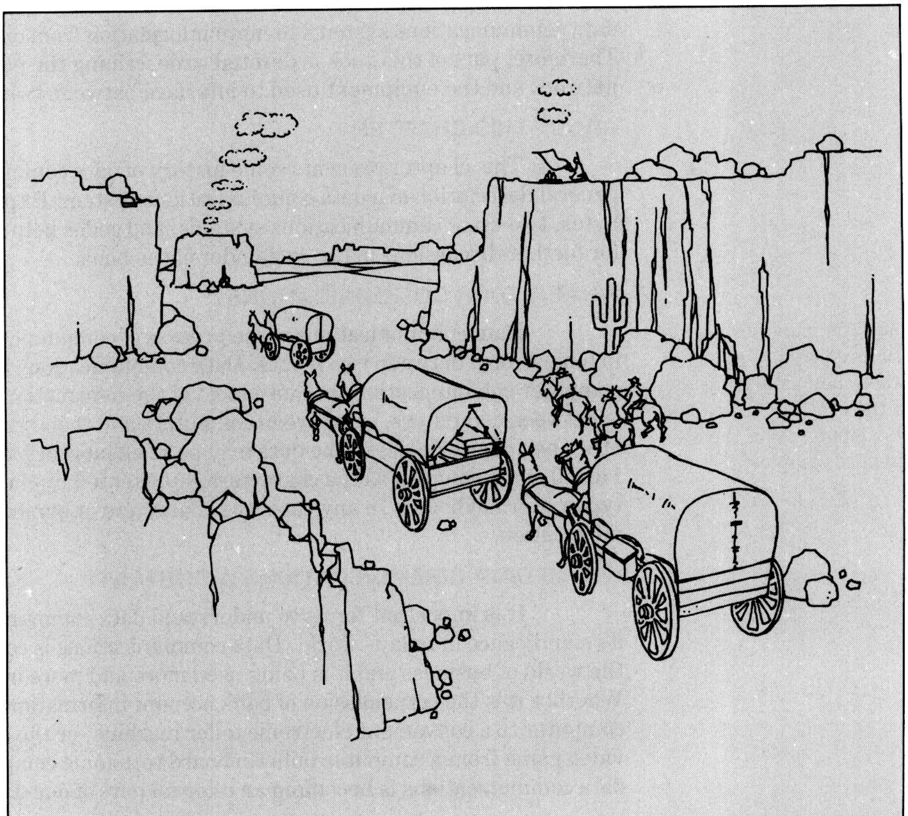
Early Uses of Electricity

Originally, data communications depended on codes transmitted by visual systems such as mirrors, flags, and smoke. Electrical data communication systems transmit codes by switching electrical current.

The discovery and harnessing of electricity introduced many new possibilities for communications codes beyond the smoke signals, mirrors, signal flags and lanterns in use in the eighteenth and nineteenth centuries.

One of the early proposals, submitted to a Scottish magazine in 1753, was simple but had profound implications for hardware. This idea was to run 26 parallel wires from town to town, one wire for each letter of the alphabet. A Swiss inventor built a prototype system based on this 26-wire principle, but the technology of wire making eliminated that idea from serious use.

Figure 1-1. Looks Like Trouble Ahead



The first data communications did not rely on electricity.

In 1833, Carl Friedrich Gauss used a code based on a 5 by 5 matrix of 25 letters (I and J were combined) to send messages by deflecting a needle from one to five times, right or left. The first set of deflections indicated the row; the second the column.

The Telegraph

The first notable development in data communications occurred in the nineteenth century when an American, Samuel F. B. Morse, invented the electric telegraph. Although other inventors had worked on the idea of using electricity to communicate, Morse's invention was by far the most important because he coupled the human mind (intelligence) with the communications equipment.

Samuel F.B. Morse perfected the telegraph, the first mass data communications system based on electric power.

A basic telegraph system is diagrammed in *Figure 1-2*. When the telegraph key at station A was depressed, current flowed through the system and the armature at station B was attracted to the coil, clicking as it struck the stop. When the key was released, it opened the electrical circuit and the armature of the sounder was forced to its open position by a spring, striking the other stop with a slightly different click. Thus, the telegraph sounder had two distinctive clicks. If the time between successive clicks of the sounder was short, it represented a dot; if longer, a dash. Morse developed a code, similar to the one shown in *Figure 1-3*, to represent characters by a series of these dots and dashes. The transmitting operator converted the characters in the words of a message to be sent into a series of dots and dashes. The receiving operator interpreted those dots and dashes as characters; thus, the information was transmitted from point A to point B.

Morse first developed the telegraph in 1832, but it wasn't until much later that he successfully demonstrated its use. The best-known demonstration took place in 1844, when Morse transmitted over a wire from Washington to Baltimore the message, "What hath God wrought!"

The telegraph system was the first electrically based communications system to connect the east and west coasts of the United States, and both sides of the Atlantic.

Since mail delivery by the Pony Express was the typical means of communication before the telegraph, the telegraph quickly became a success because of its much greater speed. The equipment was simple and rugged; the key and the sounder each contained only one moving part. Both the system's strength and weakness (and its only real complexity) was the human mind — the transmitting and receiving operators. By the time of the Civil War, a telegraph line spanned the continent, crossing the prairies and deserts to connect California with the rest of the United States. It was from this historic technological breakthrough that the Western Union Telegraph Company made its legacy and indeed was named; their line connected the *West* with the *Union*. By the time the telephone was invented some thirty years later, the telegraph industry was large and prosperous, with many companies providing service to almost every city and town in the United States. In 1866, the telegraph connected the nations of the world with the laying of the trans-Atlantic cable between the United States and France.

Figure 1-2.
Basic Telegraph System

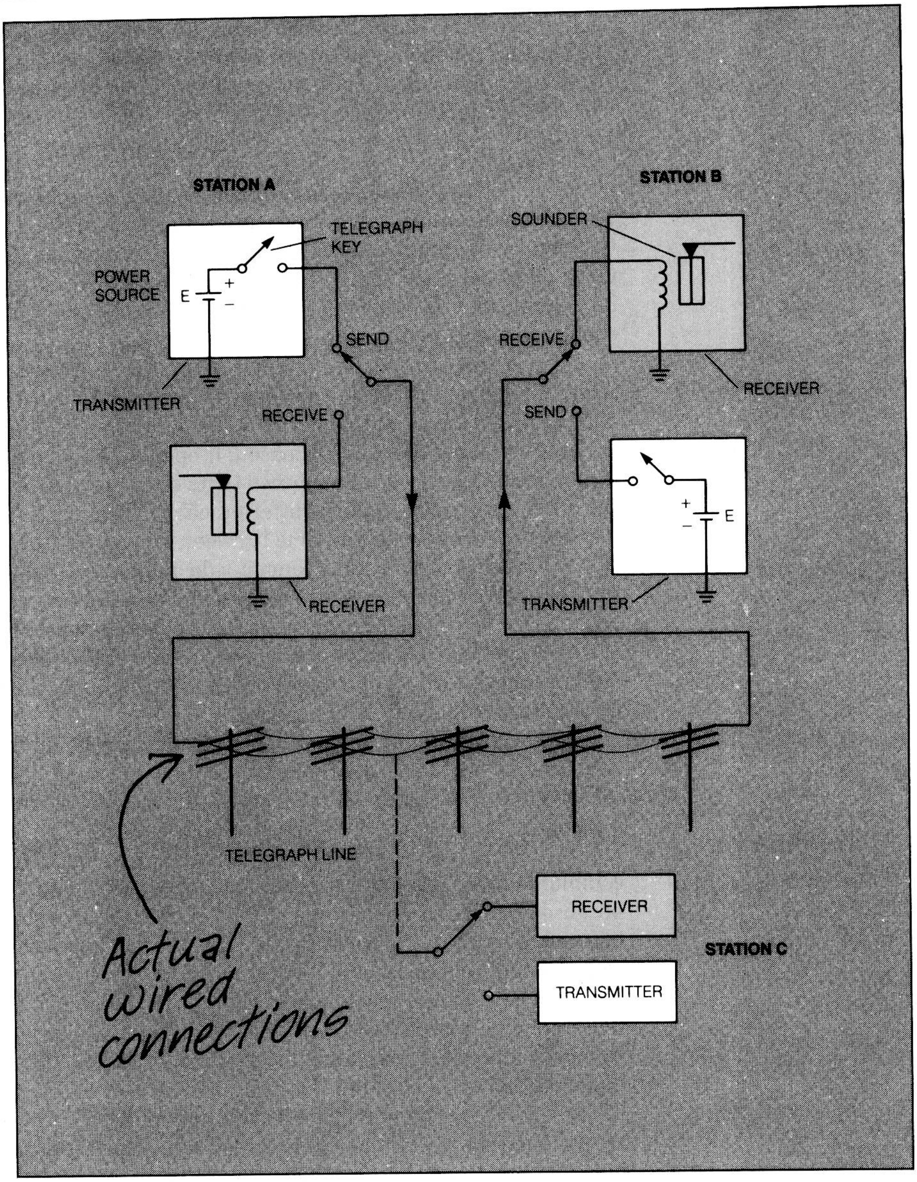


**Figure 1-3.
International Morse Code**

Much of Morse terminology is still in use today.

TELEGRAPH CHARACTERS

	Morse		Morse
A	•—	T	—
B	—•••	U	••—
C	—•—•	V	•••—
D	—••	W	•——
E	•	X	—••—
F	••—•	Y	—•——
G	——•	Z	——••
H	••••	,	——••——
I	••	.	•—•—•—
J	•———	1	•————
K	—•—	2	••———
L	•—••	3	•••——
M	——	4	••••—
N	—•	5	•••••
O	———	6	—••••
P	•——•	7	——•••
Q	——•—	8	———••
R	•—•	9	————•
S	•••	0	—————

TWO-STATE COMMUNICATIONS SYSTEMS

Much of the telegraph's terminology and principles of operation are still used today. The most important of these is the two-state communications system.

The importance of the Morse telegraph is not just historical; it illustrates the simplicity of a complete data communications system. Much of the terminology that developed around the Morse system still is in use today. For example, consider the terms "mark" and "space." If a device were arranged so that paper continually moved under a pen attached to the telegraph sounder armature, then a mark would be made on the paper when the armature was attracted to the coil. Thus, we could refer to the state of current flow in the line as the marking state and the state of no current flow in the line as the spacing state. Worldwide standards for data communications today still use the terms mark and space, with the idle condition on the transmission channel called the marking condition.

The wire ("telegraph channel") between the operators, therefore, is in one of two states; either current is flowing, or it is not. This illustrates a simple idea which has been repeated over and over again in the development of data communications systems. A two-state communications system is the simplest, the easiest to build, and the most reliable. The two states can be On and Off (as in the telegraph), or Plus and Minus (with current flowing in opposite directions), or Light and Dark (like turning a flashlight on and off to send code), or 1 and 0 (the concept used in the computer), or some other design with only two possible values. A two-state or two-valued system is referred to as a binary system.

BITS AND BYTES

The 0 and the 1 are the symbols of the binary system. A binary digit is commonly referred to as a bit. The individual line changes in digital data (like the mark and space) are called bits and each bit is assigned a value of 0 or 1.

Transmission codes in the binary system are made up of digits called bits. Each bit can have one of two possible states (high or low, on or off). Several bits combined in a uniform group are called a byte.

The binary system uses positional notation, just as the familiar decimal system does, except that each position has only two possible values, rather than ten. For example, the number 345.27 in decimal means three hundreds plus four tens plus five ones plus two tenths plus seven hundredths. The *weight* of each position is a power of ten, thus:

$$\begin{aligned}345.27 &= (3\times100) + (4\times10) + (5\times1) + (2\times1/10) + (7\times1/100)\\ &= 3\times10^2 + 4\times10^1 + 5\times10^0 + 2\times10^{-1} + 7\times10^{-2}\end{aligned}$$

Note the use of positive and negative exponents, and remember the definition that *any* number raised to the zero power is equal to 1. In the decimal system, as shown by the preceding example, ten is the *base*, and the weight of each digit position is a power of ten.

Similarly, we can define powers of two in the binary positional system, where the weight of each position is two times the weight of the one to the immediate right. A table of powers of two can be quite useful when dealing with binary numbers because most of us don't want to memorize them. *Table 1-1* is such a table for powers of two up to 2^8 which will be sufficient for this book.

Table 1-1. Powers of Two

Power of Two	Positional Weight	9-Bit Binary Number
2^0	1	000000001
2^1	2	000000010
2^2	4	000000100
2^3	8	000001000
2^4	16	000010000
2^5	32	000100000
2^6	64	001000000
2^7	128	010000000
2^8	256	100000000

Using this table makes it easy to convert a binary number to its decimal equivalent. For example;

$$\begin{aligned} 1101001 &= 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0 \\ &= 64 + 32 + 0 + 8 + 0 + 0 + 1 \\ &= 105 \text{ decimal} \end{aligned}$$

Computers are more efficient if their internal paths and registers are some power of two in length (two bits, four bits, eight bits, sixteen bits, etc.). A commonly used grouping is eight bits and this 8-bit group is called a byte or an octet. From *Table 1-1*, you can see that a byte can represent 2^8 or 256 unique sequences of 1s and 0s.

COMMUNICATIONS CODES

The use of simplified and standardized binary codes allowed information to be encoded and decoded by mechanical or electrical means, and made it possible to automate data communications.

A common characteristic of data communications systems is the use of an intelligent device to convert a character or symbol into coded form and vice versa. In the Morse system, the intelligent "devices" were the telegraph operators who converted the characters into dots and dashes.

Skilled telegraph operators were always in short supply and the work was difficult and exhausting, so an electrical or mechanical means of coding the characters was needed. However, it was essentially impossible to automate the transmitting and receiving operations because of the varying duration of the dots and dashes in the Morse code, and the fact that the codes for the characters were made up of different quantities of dots and dashes. Therefore, a code that had an equal number of equal duration signaling elements for each character was needed.

Some Definitions

A code is a previously agreed upon set of meanings that define a given set of symbols and characters; in data communications these codes are built into the equipment.

Before we go any further, let's establish the following definitions in order to discuss communication codes:

Codes: Standard (agreed-upon-in-advance) meanings between signaling elements and characters. The key idea is *standard*. The codes used in data communications systems are already defined and the code set is built into the equipment. About the only time the user might need to deal with codes is when interfacing two machines (computers, printers, etc.) from different manufacturers.

Characters are the letters, signs and symbols on an input device's keyboard; some of these do not print, but are used to control the system.

Characters: The letters, numerals, space, punctuation marks, and other signs and symbols on a keyboard. (Remember that the space character is just as important as any other even though we tend to think of it as "nothing" or "blank.") For example, A 7# is a sequence of four characters. Communication systems also use control characters which do not print, but they also must be coded. Some of these (such as Carriage Return or Tab) may be on a keyboard, but many are not.

Symbols are the representations of characters that are transmitted over transmission lines. It is easier to design and build machines to recognize symbols rather than characters.

Signaling Elements: Something that is sent over a transmission channel and used to represent a character. The dots and dashes (or marks and spaces) of the Morse code are signaling elements, as are the ones and zeros in this sequence:

0100000111001000110101110110010110001001

This is the way A 7# might look when transmitted between a personal computer and another computer or printer.

The definitions for characters and signaling elements illustrate why machines and people need different ways to represent information. People quickly and reliably recognize printed characters by their distinctive shapes, but that is difficult and expensive for a machine to do. On the other hand, machines can easily handle long strings of two-valued signaling elements such as marks and spaces or ones and zeros, but that is hard for a person to accomplish with any accuracy.

Baudot Code

Emile Baudot developed one of the more successful codes suited for machine encoding and decoding. However, it was limited because it used only five symbols.

As stated above, the Morse code was unsuitable for machine encoding and decoding because of the problems caused by the varying lengths of the codes for the characters. When, early in the twentieth century, interest developed in replacing human telegraph operators with machines, several suitable codes already existed. The most prominent of these had been invented in the 1870s by a Frenchman named Emile Baudot. Since Baudot's code, similar to the one shown in *Figure 1-4*, used the same number of signaling elements (marks and spaces) to represent each character, it was better suited to machine encoding and decoding. Unfortunately, the number of signaling elements was limited to five by problems in timing the electromechanical devices. The 5-bit code could generate only 32 possible combinations, fewer than was needed to represent the 26 characters of the alphabet, the 10 decimal digits, the punctuation marks, and the space character. In addition, Baudot used two shift-control characters — the letters (LTRS) shift and the figures (FIGS) shift — to permit the code set to represent all the characters which seemed necessary at the time. The shift codes do not represent printable characters, but select one of two character sets, each composed of 26 to 28 characters. Receipt of the letters shift code causes all following codes to be interpreted as letters of the alphabet; receipt of the figures shift code causes all the following characters to be interpreted as numerals and punctuation marks. Notice that the LTRS and FIGS codes, as well as the other control codes and the space character, always have the same interpretation no matter which shift mode the machine is currently in. Although Baudot's invention did not immediately revolutionize telegraphy (because of the difficulty that human operators had in sending equal length codes), it did provide a basis for the later development of the teleprinter.

**Figure 1-4.
Five-Level Teleprinter Code**

Code Signals ● Denotes positive current							LTRS Shift	FIGS Shift	
								CCITT Standard International Telegraph Alphabet No. 2 Used for Telex	North American Teletype Commercial Keyboard
Start	1	2	3	4	5	Stop			
	●	●				●	A	—	—
	●			●	●	●	B	?	?
		●	●	●		●	C	:	:
	●			●		●	D	Who are you?	$
	●					●	E	3	3
	●		●	●		●	F	Note 1	!
		●		●	●	●	G	Note 1	&
			●		●	●	H	Note 1	#
		●	●			●	I	8	8
	●	●		●		●	J	Bell	Bell
	●	●	●	●		●	K	(	(
		●			●	●	L	)	)
			●	●	●	●	M	.	.
			●	●		●	N	,	,
				●	●	●	O	9	9
		●	●		●	●	P	0	0
	●	●	●		●	●	Q	1	1
		●		●		●	R	4	4
	●		●			●	S	'	'
					●	●	T	5	5
	●	●	●			●	U	7	7
		●	●	●	●	●	V	=	;
	●	●			●	●	W	2	2
	●		●	●	●	●	X	/	/
	●		●		●	●	Y	6	6
	●				●	●	Z	+	"
						●	Blank		
	●	●	●	●	●	●	Letters shift (LTRS)		
	●	●		●	●	●	Figures shift (FIGS)		
			●			●	Space		
				●		●	Carriage return		
		●				●	Line feed		

Notes:
1. Not allocated internationally; available to each county for internal use.

Modern Codes

There was a need for new codes that could represent all characters, be able to check for errors, and leave room for further expansion.

The Baudot code and variations of it were the backbone of communications for almost half a century, but they clearly left much to be desired. The newspaper industry found the lack of differentiation between upper and lower case letters to be a problem and devised a six-level code to designate the difference between upper and lower case letters. This was just one example of a general need; modern communications required a code that could represent all printable characters and still leave room for error checking. The code had to permit decoding without reliance on correct reception of previous transmissions, and also had to permit decoding by machine. Perhaps most important of all, the new code needed to be expandable.

During the 1960s, a number of data transmission codes were developed. Most of these have fallen by the wayside, leaving three predominant codes: a single 5-bit code (CCITT International Alphabet No. 2) still used for telex transmission; the Extended Binary-Coded-Decimal Interchange Code (EBCDIC, pronounced "eb-see-dik") developed by IBM and primarily used for synchronous communication in systems attached to large mainframe computers; and the American Standard Code for Information Interchange (ASCII, pronounced "as-key") code defined by the American National Standards Institute (ANSI) in the United States and by the International Standards Organization (ISO) worldwide.

EBCDIC

EBCDIC is a modern code using 8-bits to represent 256 characters. It was developed by IBM to provide a standard code for its own products.

When a clear need arises for standardization, standards can come into existence in two ways. A single manufacturer (especially a dominant one) can define a standard for its own products, and the rest of the industry may follow. This is what IBM[1] did. It created the EBCDIC 8-bit code, allowing 256 characters to be represented. The world would probably be better off if EBCDIC had become the standard, because it has enough unique characters to allow almost any representation. However, only IBM and firms who build IBM-compatible equipment adopted EBCDIC. Since EBCDIC is primarily used in large IBM-compatible computing systems, it will not be used in most of the examples in this book.

ASCII

The ANSI developed the 7-bit ASCII data communications code that is in general use today. It can represent 128 characters.

The other, more common method of creating a standard is through a committee, which serves as a forum for examination of needs, discussion between interested parties, and compromise. This process produced the ASCII 7-bit code, formally known as ANSI standard X3.4-1977. ASCII can represent 128 characters, but not all of them represent printed symbols. Included in the character set are all the letters of the English alphabet (both uppercase and lowercase), the numerals 0-9, punctuation marks, and many symbols. This standard code set is used in virtually all small computers and their peripherals, as well as in large computers in most of the world.

[1]IBM is a trademark of International Business Machines Corporation.

Figure 1-5 shows the ASCII characters and their associated codes. Compare this chart to the Morse code in *Figure 1-3* and the 5-bit teleprinter code in *Figure 1-4*. The Morse code had a varying number of elements (dots and dashes) for each character, and was quite restricted — only letters, numerals, and a few punctuation marks. The 5-bit code had a constant number of elements and a few more special characters, but still couldn't distinguish between upper case and lower case letters.

The ASCII format is arranged so that lower case letters can be changed to upper case by changing only one bit. Bits 4 through 1 of the numeric characters also are the BCD value of the number.

ASCII not only has upper and lower case, but also has a great deal of regularity that may not be readily apparent. For example, to convert any upper case alphabetic character (A through Z) to lower case, it is only necessary to change bit 6 from zero to one! Another feature is that bits 4 thorough 1 of the numeric characters (0, 1, 2, 3, 4, 5, 6, 7, 8 and 9) are the binary-coded-decimal (BCD) value of the character. Another advantage of ASCII is that 128 different characters can be represented by the seven bits used in the code, rather than only 32 characters from the 5-bit code.

Figure 1-5. American Standard Code for Information Interchange (ASCII)

Bit Positions: 4	3	2	1	7: 0 6: 0 5: 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	0	0	0	**NUL**	**DLE**	SP	0	@	P	`	p
0	0	0	1	**SOH**	**DC1**	!	1	A	Q	a	q
0	0	1	0	**STX**	**DC2**	"	2	B	R	b	r
0	0	1	1	**ETX**	**DC3**	#	3	C	S	c	s
0	1	0	0	**EOT**	**DC4**	$	4	D	T	d	t
0	1	0	1	**ENQ**	**NAK**	%	5	E	U	e	u
0	1	1	0	**ACK**	**SYN**	&	6	F	V	f	v
0	1	1	1	**BEL**	**ETB**	'	7	G	W	g	w
1	0	0	0	**BS**	**CAN**	(	8	H	X	h	x
1	0	0	1	**HT**	**EM**	)	9	I	Y	i	y
1	0	1	0	**LF**	**SUB**	*	:	J	Z	j	z
1	0	1	1	**VT**	**ESC**	+	;	K	[	k	{
1	1	0	0	**FF**	**FS**	,	<	L		l	|
1	1	0	1	**CR**	**GS**	–	=	M	]	m	}
1	1	1	0	**SO**	**RS**	.	>	N	∧	n	~
1	1	1	1	**SI**	**US**	/	?	O	—	o	DEL

Sample of Control Characters (Bold)
STX = Start of text
EOT = End of transmission
CR = Carriage return
HT = Horizontal tabulation

Examples:
1000011 = C
0110011 = 3
1010000 = P
0110000 = 0 (Zero)
0100000 = SP (space)

Non-Printable Control Characters

ASCII also has several control codes which, in addition to their defined control functions, have been used by some manufacturers to represent specialized functions.

The two left-most character columns in the chart of *Figure 1-5* represent the nonprinting control characters which may be used to control the operation of the receiving device. For example, the control codes for carriage return (CR) and linefeed (LF), which are commonly used on a typewriter, are shown. Other control codes include formfeed, bell, horizontal tab and vertical tab. These control codes were designed for printing or display devices, although some manufacturers have used the control codes for all manner of special functions. Also, some codes control how a receiving device will interpret subsequent codes in a multiple character function or command. Two shift characters called Shift In (SI) and Shift Out (SO) are used to shift between ASCII and character sets other than those used in English. ANSI standards X3.41-1974 and X3.64-1979 expand the definition of the escape (ESC) control code for even greater flexibility. Other control codes delimit text, such as start of text (STX) and end of text (ETX). These codes are used primarily in block or synchronous data transmission, and you'll see more of them in later chapters.

Escape Character

Escape sequences are code sequences made up of noncontrol characters that are to be interpreted as control codes.

The escape (ESC) character designates that the codes that follow have special meaning. Characters received in an escape sequence are not interpreted as printing characters, but as control information to extend the range of the "standard" character set by allowing other definitions. The escape character has the effect of making all character codes available for control of a device. Graphics characters, foreign language character sets, and special applications sets have been developed which are accessible via escape character sequences; thus, they permit a much richer variety of displayed symbols than is possible with any single code.

The CRT terminal probably has benefited most by the escape sequence. The serial communications link to these terminals is the same as for a teleprinter, and ordinarily, any characters received via this channel are displayed on the terminal screen as expected. But the people who developed the ASCII standard did not forsee (thus did not make provision for) capabilities for character and line deletion, and display enhancements such as inverse video, underlining, and blinking that are available on the CRT terminal. Unfortunately, little standardization of these sequences existed until the ANSI X3.64 standard came out in 1979. Before then, and without standardization, designers felt free to exercise their creativity. For example, one major feature now found on most video display terminals is absolute cursor positioning. The computer can send a command to the terminal that will place the cursor anywhere on the screen. This capability is important for many types of form-filling operations. Unfortunately, there are almost as many escape sequences to do cursor positioning as there are terminal manufacturers. Even different models in a manufacturer's line may use different escape sequences to do the same thing.