DC-83-05 No. 3

Data Communications in Energy Management and Control Systems: Issues Affecting Standardization

H.M. Newman

ABSTRACT

One of the principal reasons for the incompatibility that currently exists between different vendors' EMCS computers and field interface hardware is the way in which information is communicated between the various system components. This paper reviews the aspects of data communications that contribute to this situation.

OVERVIEW

The effective operation of an energy management and control system (EMCS) depends on the ability of the various system components to communicate with one another rapidly and reliably. While communication speed is generally not as critical a concern in an EMCS as it is in many other process control applications, the need persists for data to be received error-free. As the number of sensors and control points increases, either the speed of data transmission must be increased or intelligence, i.e., computing capacity, must be more widely distributed throughout the system. In either situation, the overriding concerns are those of throughput and data integrity.

Three aspects of the data communications problem, taken together, represent the communications protocol and determine the degree to which components of one system may be interfaced to components of another system. They are (1) the physical connection, or link, and the methods of controlling the end-to-end error-free transmission of information along it, (2) the encoding of the message data, and (3) the content and structure of the messages themselves. These factors are analogous to sending an important piece of correspondence: (1) How should the message be packaged and conveyed? Should it be sent by surface mail or air express? How can the sender be certain that it arrived safely? (2) What alphabet or character set should be used? (3) What language or message structure should be employed? Many of the world's tongues use the same alphabet but are by no means mutually intelligible. This paper examines the significance of each of these factors with respect to EMCS communications.

H. Michael Newman, Senior Computer Systems Engineer,
Facilities Engineering, Cornell University, Ithaca, NY.

THE PHYSICAL LINK AND TRANSMISSION CONTROL

Regardless of how it is to be packaged and conveyed, all information in a digital computer system is represented by a string of binary digits or "bits." Each bit takes on a value of either 0 or 1. (For reasons dating back to Samuel Morse, a 0-bit is called a "space" while a 1-bit is called a "mark.") Bits may be arranged in a variety of ways to represent a specific piece of information, but the bits themselves may be transmitted in one of only two ways between communicating system elements: in parallel or serially.

Parallel Transmission

In parallel transmission, a group of bits is transmitted simultaneously by applying DC voltages to the conductors of a multiconductor cable corresponding to the current state of each binary digit. If the data port happened to employ TTL logic levels, for example, a 1-bit would be represented by 5 VDC, while a 0-bit would correspond to 0 VDC. In this system, there is one conductor for each data bit plus one or more conductors to control the timing signals which notify the receiver that valid data have been impressed on the data lines. In the general case, additional conductors are required to address the location from which or to which data are to be transmitted, although the data lines (or data bus) may double as address lines with some reduction in throughput. The prime advantage of such systems, especially when coupled with computers permitting direct memory access, is that extremely high data-transfer rates are attainable, on the order of millions of bits per second. The disadvantages are the relatively expensive multiconductor cable and the limited range, generally several thousand feet at most, arising from the effect of cable capacitance on the DC signaling voltage. As a result, parallel transmission techniques are generally limited to critical process control applications, where most of the computing is done centrally and response time is of the essence.

Serial Transmission

In the world of EMCS, however, where a time scale of minutes may be acceptable for most monitoring and/or control purposes, almost all data transmission is serial, meaning that data are transmitted one bit at a time. In this scheme, each bit is presented at the output of the computer or terminal as a DC voltage. (In the data communications field, computers and terminals are referred to as data terminal equipment or DTEs.) The specific value of the voltage that corresponds to a 0 or 1 varies. If the voltage is coupled directly to the transmission medium, one speaks of "DC," "baseband," or "digital" transmission. As mentioned above, cable capacitance has a deleterious effect on alternating DC waveforms and tends to smear all the 0's and 1's together. Thus, in the absence of strategically located repeaters to sample, reconstruct, and retransmit the data, direct digital, like parallel transmission, is limited to several thousand feet. As a result, the DC voltages emanating from the DTE, which represent the individual bits of data, are generally used to modulate some form of analog signal that serves as a carrier of the digital information. Such signaling is called "AC," "broadband," or "analog" and has the advantage that distance limitations are effectively eliminated. The device that performs the modulation or demodulation is referred to as a "modem." If the device simply amplifies the voltage signal from the DTE or converts it to a current loop signal, it is called a "line driver" and is, again, generally usable only over limited distances. Modems and line drivers are referred to generically as data communications equipment or DCEs (see figure 1).

One of the few standards that both exists and is frequently employed in EMCS serial data transmission prescribes the interface between the DTE and the DCE. It is the Electronic Industries Association's Recommended Standard 232-C. RS-232-C prescribes the voltage levels (-15 VDC = binary 1, +15 VDC = binary 0) and the control line conventions for the "handshaking" between DTE and DCE. It has nothing to do with the physical link between DCEs, the nature of the codes to be employed, the structure of messages, or even the cable connectors. (The so-called "RS-232 connector" is a de facto standard that has nothing whatever to do with the recommendations of the EIA.) In short, the statement that a EMCS computer is "RS-232 compatible" merely implies that standard data communications equipment may be used, not that the equipment can communicate in any meaningful way with other RS-232 compatible equipment.

Modulation and the Transmission Medium

Three properties of a sinusoidal AC signal can be varied: its amplitude, its frequency, and the phase between successive cycles. A modem may employ any one or a combination of these techniques (see figure 2). The most commonly employed technique for relatively low-speed modems (1200-1800 bits per second) is to vary the frequency of an audio tone. Such a technique is called "frequency shift keying" or FSK. It has the advantage that the resulting signal can be transmitted on commercial voice-grade telephone lines or inexpensive pairs of wires. The higher the speed of transmission, the greater the bandwidth of the transmission medium must be. Fourier analysis shows that the bandwidth required is proportional to the number of state transitions, e.g., from high frequency to low frequency or vice versa in the case of FSK. The number of transitions per second is called the "baud rate." If each transition represents a bit changing from 0 to 1 or from 1 to 0, the baud rate is synonymous with the transmission rate expressed in bits per second. Clever modem manufacturers, however, have devised methods of holding down the baud rate, and hence the bandwidth, while increasing the transmission rate in bits per second. This is done by multistate modulation techniques. For example, instead of sending an audio signal with just two frequencies, four frequencies might be sent, each frequency correlating to the state of not one but two bits in the modulating data stream. In such a case, the number of bits per second transmitted is equal to twice the baud rate. Another commercially available modem uses 12 different phase transitions, 4 of which are also amplitude modulated. This gives 16 different states, just the proper number to represent all the possible combinations of four data bits. Running at a transition or baud rate of 2400, it thus conveys information at the rate of 9600 bits per second (see figure 3). The trade-off is that such modems require more sophisticated circuitry and are therefore more expensive than simple low-speed, bi-state modems.

The selection of the physical transmission medium is determined by the frequency and bandwidth of the signal to be transmitted. For audio signals, ordinary twisted wire pairs are commonly used. If the transmission is half duplex, i.e., if data are to be sent in only one direction at a time, a single wire pair may suffice. (This could be an ordinary leased or dial-up telephone line.) If the transmission is to be duplex, i.e., data may flow to and from the field at the same time, two pairs may be required or different carrier frequencies may be used for the transmitted and received data. If radio frequency signals are being modulated, coaxial cables may be used or the signals may be sent through the air. Power line carrier systems transmit radio frequency data along existing in-house electrical wiring. At the upper end of the frequency spectrum, modulated light signals may be sent via fiber optic cables or through space.

Control of the Physical Link

In order for data to be successfully sent and received, several requirements must be met. First, the receiver has to know when to sample the condition of the transmission line in order to have the best chance of detecting a valid bit; second, the receiver has to have some way of knowing where a message starts and where it ends. A third requirment, which will be discussed shortly, is that the receiver ought to be able to know if the message was correctly received. The first two requirements are both aspects of transmitter-receiver synchronization and are taken care of by various forms of message framing. The third requirement is handled by various error detection techniques.

Message Framing. The most common form of serial data transmission in EMCS systems is called "start-stop" or "asynchronous." This transmission method was originally devised for use with ordinary, "dumb" terminals, where data are sent only when a keyboard key is struck. Consequently, the "message" in this case is normally a single character, by convention five to eight bits long, and the framing is accomplished by a single bit at the beginning of the character and one or possibly more bits at the end. In order for the start bit to be detected, the line is maintained at the mark, or logical 1, condition between characters that may be sent asynchronously, i.e., at random intervals. When the receiver senses a 1 to 0 transition, it waits for one-half bit time and then begins sampling bits. The delay is intended to insure that the sampling occurs at the center of each bit. After the requisite number of bits has been sampled, the receiver looks for a stop bit, which must be a 1 (see figure 4).

Other variations on the asynch scheme are possible. One EMCS vendor uses 3 start bits, a 010 sequence, followed by 22 data and error-checking bits. The key concept is that no particular timing relationship exists between one message and the next.

It should be clear that although asynch communications may permit sufficiently rapid data transmission for most EMCS applications, it does not make very efficient use of the line: at least two, and possibly more, of the transmitted bits carry no information.

A more efficient transmission method is known as synchronous. Because the transmitter and receiver are kept in constant synchronization with each other, the necessity for individual character framing using start and stop bits is eliminated and data can be sent in long, continuous streams. There are two general classes of synchronous protocols: character-oriented and bit-oriented. In character-oriented protocols, synchronization between messages is maintained by the repeated transmission of a specific synch character. Message framing, however, is accomplished by other special characters such as a start-of-text or an end-of-text character. In bit-oriented protocols, such as High-level Data Link Control (HDLC) recommended by the International Telecommunications Union, synchronization and framing are both accomplished by a unique bit string called a "flag": 01111110.

An interesting problem, which occurs to some degree in all forms of data communication, is how to achieve data transparency, i.e., how can one send an end-of-text character, for example, as part of a message without causing the receiver to think that the end of the message has arrived? Solutions to this problem are discussed in the references listed in the bibliography.

Error Detection. Data transmission can only be considered successful if the message is received error-free. This is particularly the case in EMCS, where a garbled message could potentially cause a critical building system to be disabled at an inopportune moment or an erroneous alarm to be produced. The techniques range from extremely simple to highly sophisticated, and the amount of overhead, i.e., the additional number of bits transmitted, tends to increase with the sophistication of the scheme.

One of the simplest, most widespread error detection methods makes use of the property of bit strings known as "parity." A string of bits is said to be of even parity if the total number of 1-bits present is an even number, e.g., 0,2,4,6, etc. If the number of 1-bits present is an odd number, e.g., 1,3,4,7, etc., the string is said to be of odd parity. In mark or space parity, the parity bit in a given string is always 1 or 0 respectively. It is clear that parity checking alone can only detect a very limited range of errors, namely, those where an odd number of bits have been altered by transmission interference in the case of even or odd parity checking or where the parity bit itself has been affected in the case of mark or space parity checking. The use of a single parity bit tagged onto a bit string, generally representing a single character, is known as a "vertical redundancy check" (VRC). An extension and improvement of the VRC concept involves generating a parity bit for a string of bits corresponding to a given position in a series of fixed-length message blocks. Considering figure 5, suppose that each message consists of a total of eight bits, seven data bits and the VRC parity bit. By mutual agreement between sender and receiver, it is decided that after five such messages, an additional message will be sent consisting of seven error-checking bits, one corresponding to the parity of the five bits in bit position 1, one corresponding to the five bits in bit position 2, etc. In addition, the block check character, or BCC as it is called, will also have its own parity bit. This technique for generating a BCC is called a "longitudinal redundancy check" (LRC). The combination of VRC and LRC is far more effective in detecting errors than the VRC parity check by itself.

Another even more powerful technique makes use of a cyclical redundancy code (CRC). This code, which is appended to the transmitted message, is generated by subjecting the message data stream to a specific set of mathematical operations that are performed in the identical manner by the receiver. If the codes match, the message is considered to have been correctly received. Virtually any degree of reliability may be attained by increasing the number of bits in the error code. The trade-off is to keep the number as small as possible while attaining data integrity that is as high as possible. The details of CRC generation are taken up in the references. The main point is that each vendor is free to use one of the more standard sets of mathematical operations for generating the CRC code, or one of his own invention, thus leading to the probability of incompatibility with other vendors' equipment.

What happens when a parity error, framing error (no stop bit, end-of-text, or flag), VRC/LRC, or CRC error is detected? This, again, depends on the vendor. The system may be "smart" enough to automatically request that the data be retransmitted or the hardware may simply notify the software that a certain type of error has occurred, in which case the software must decide what action to take. (A "break" or "attention" character is actually transmitted from a terminal by intentionally introducing a framing error. The break key simply causes the line to go to the space, or 0, condition until it is released or for some fixed but relatively long time. Thus the receiver fails to see the binary 1 stop bit at the appropriate time and notifies the computer's operating system.)

THE CODING OF DATA

Once the questions of compatibility involving the physical link between computer and field hardware have been resolved, the next concern is the method by which data are to be encoded, in effect, what alphabet is to be used? Here again there are several possibilities and a complete lack of standardization. One possibility is to use a character-oriented encoding scheme. Here each command or piece of data is represented by a string of characters, each of which has its own binary representation. The most widely used character code is the American Standard Code for Information Interchange (ASCII). ASCII is a seven-bit code and can therefore represent 128 various alphanumeric and control characters, the latter traditionally reserved for the operation of terminal devices by signaling the need for carriage returns, line feeds, and the like. The problem with using ASCII, for example, is that each character takes up at least seven bits, not counting possible parity or start-stop bits. For example, to send the ASCII representation of the decimal number 100 requires a minumum of 21 bits, 7 for each character. The alternative to character-oriented encoding is bit-oriented encoding, where one deals with the representation of data in binary format. The decimal value 100 can now be represented by just seven powers-of-two binary digits, 1100100, or the operational status of a fan or pump can be represented by a single bit, e.g., 0 for ON, 1 for OFF. There is thus a great advantage in terms of throughput in using the more compact bit-oriented data representation. Unfortunately, no two vendors configure their bit streams in quite the same way.

MESSAGE STRUCTURE AND THE CONTROL OF THE NETWORK

The bottom line in data communications is to send and receive intelligible messages. What are the components of such messages in the case of EMCS networks? The answer depends critically on the way in which the network is configured in terms of communications and how the monitoring and control functions are distributed, i.e., how much computing "intelligence" is located where. Some insight into the first issue, that of communications control, can be gained from a consideration of generalized data processing networks, which can be described as vertically or horizontally organized. This distinction is made on the basis of where the bulk of the computing power resides. If one computer predominates, the network is vertically organized and this computer directs the communications. Such networks generally use the master-slave concept: the master station determines unilaterally whose turn it is to speak and no station speaks until spoken to. If the computing power is more uniformly distributed, one control possibility is to use a variation on the master-slave scheme in which each station takes a turn at being the master, i.e., in charge of the network. One common method by which authority is passed from one station to the next is referred to as token passing. The station currently in possession of the "token" is the master. When it has conducted whatever communication it wishes, it simply passes the token, that is the control authority, to the next station in line. Another type of horizontal network is the contention network. Again, each station is of equal rank, but any station may attempt to gain control of the link any time it has something to say. The main problem with contention networks is resolving which station is to gain control of the communications link when more than one station transmits simultaneously. Recovery techniques from such collisions are discussed in the references.

Vertical organization is representative of traditional, supervisory central control systems (see figure 6). In such a system, the field equipment is likely to be totally passive and to contain little, if any, local intelligence. Messages in such a system are likely, therefore, to contain only four types of elements: (1) address information- to which piece of equipment is the message being sent?; (2) command information- what is the equipment supposed to do upon receipt of the message?; (3) data- the results of readings taken by field instrumentation or the conditions to be established by the remote equipment, such as new set points, new start/stop settings, and so on; and (4) some type of error-checking code, as discussed above.

A more horizontally organized system is represented by the case where a central system, again operating in a supervisory mode, communicates with a network of stand-alone, direct digital control microcomputers (see figure 7). In such an arrangement, address, command, and error-checking information would still be required, but the data might consist of binary-coded machine instructions being downline loaded to the DDC computer. The data from the DDC system might be the value of a given parameter, such as a set point or alarm limit, the current value of a temperature or the status of a piece of HVAC equipment, or merely the notification that nothing has changed since the last communication.

It is no exaggeration, regrettably, to state that there is absolutely no standardization of message format in the EMCS industry. Although for a given system architecture and distribution of function between the various components, any vendor's message must contain essentially the same information, there are no fixed rules for determining the order in which the data are sent or the organization of any particular element. Consider, for example, the address information. At a minimum, the address must contain the identifier of the field multiplexer and the point to be interrogated or controlled or for which a new strategy or set point is being downline loaded. The designer's choice for the number of bits allocated to each part of the address influences the rate at which messages may be sent as well as the ease of system expansion: if many bits are used for the address field, overall transmission time will be increased but more addresses, that is more field hardware, can be accommodated.

One vendor uses five bits for the field multiplexer address and six bits for the point address. Thus each channel on this system can support 32 multiplexers, each of which can accommodate 64 points. This may be adequate or it may not.

In the case of the command code, the number of bits allocated determines the number of commands that may be issued. The width of the data field may influence the maximum resolution that can be obtained in reading an analog sensor. An eight-bit field, for example, permits a resolution of 1/256, or approximately .4%, of the analog span. Whether this is adequate depends on the application. In the metering of energy flows in chilled water systems, where differential temperature calculations must be performed, a field width, and hence resolution, of 10 or 11 bits is generally required for meaningful results.

Is there, in light of all these possibilities for physical link arrangements, data encoding, and message organization and content, any reasonable chance that any meaningful degree of compatibility will ever be achieved? The author thinks there is.

CONCLUSIONS

The computerized EMCS industry, it is worth remembering, has been with us now for only about 10 years. In that time an incredible evolution of computer technology has been witnessed, which, it is fair to say, has taxed, if not outstripped, our capacity to meaningfully employ it (not just in the EMCS field). The present situation in EMCS data communications is symptomatic of this plight. There are, as this paper has attempted to show, a vast number of competing alternatives at each step of the way. The net result is the inability of the system components of the various vendors to communicate. But there is hope! As time goes on, the problem that EMCSs are supposed to solve is becoming more well-defined and, as a result, the hardware and software solutions as well.

Consider the case of distributed, direct digital controllers linked to a central supervisory computer (an arrangement many believe to represent the state of the art). While the most common objection to standardization is that it could stifle a vendor's creativity in solving a given problem, the adoption of distributed architectures means that each vendor still has the opportunity to fashion the most effective and energy-efficient control algorithms independent, for the first time, of the communications issue. In other words, because the remote microprocessors already know what they have to do, there is less need to communicate with them in the first place. Moreover, the designer can say in advance, and therefore standardize, what the communication needs are likely to be. Messages are likely to be copies of programs being downline loaded block by block, requests for current values of remotely maintained constants or variables, or simply requests to know how things are going. The host system doesn't need to know, and probably doesn't particularly care, how the remote system is carrying out its work (using its proprietary software), only that it is!

It is also conceivable, given the explosive evolution of microprocessor technology, that inexpensive protocol converters will someday be available to take one vendor's codes and message formats and convert them to another's, thus opening up the entire industry to the benefits, at least from the consumer's point of view, of healthy competition. (Code converters of a similar nature have stimulated just this sort of competition in other areas of the data communications industry.)

Ultimately, standards will evolve in the EMCS industry just as they have in other technological areas: either one vendor will come to dominate the market, whose methods will thus become de facto standards, or all the vendors will decide it is in their common interest to provide a degree of compatibility, basing their marketing instead on the ability of their system to outperform those of the competition where it really counts- in the mechanical room!

In the meantime, it is still up to the individual EMCS owner to become as knowledgeable as possible about the pros and cons of each vendor's system and to select the one most nearly fulfilling the requirements of his particular situation, both now and in the immediate future.

BIBLIOGRAPHY

Davies, D.W., et al. 1979. Computer networks and their protocols. Chicester: John Wiley and Sons.

Freeman, R.L. 1980. Telecommunication system engineering. New York: John Wiley and Sons.

Martin, J. 1981. Computer networks and distributed processing. Englewood Cliffs, NJ: Prentice-Hall.

Martin, J. 1976. Telecommunications and the computer, 2d ed. Englewood Cliffs, NJ: Prentice-Hall.

McNamara, J.E. 1977. Technical aspects of data communications. Maynard, MA: Digital Equipment Corp.

Roden, M.S. 1982. Digital and data communication systems. Englewood Cliffs, NJ: Prentice-Hall.

Terminals and communications handbook. 1981. Maynard, MA: Digital Equipment Corp.

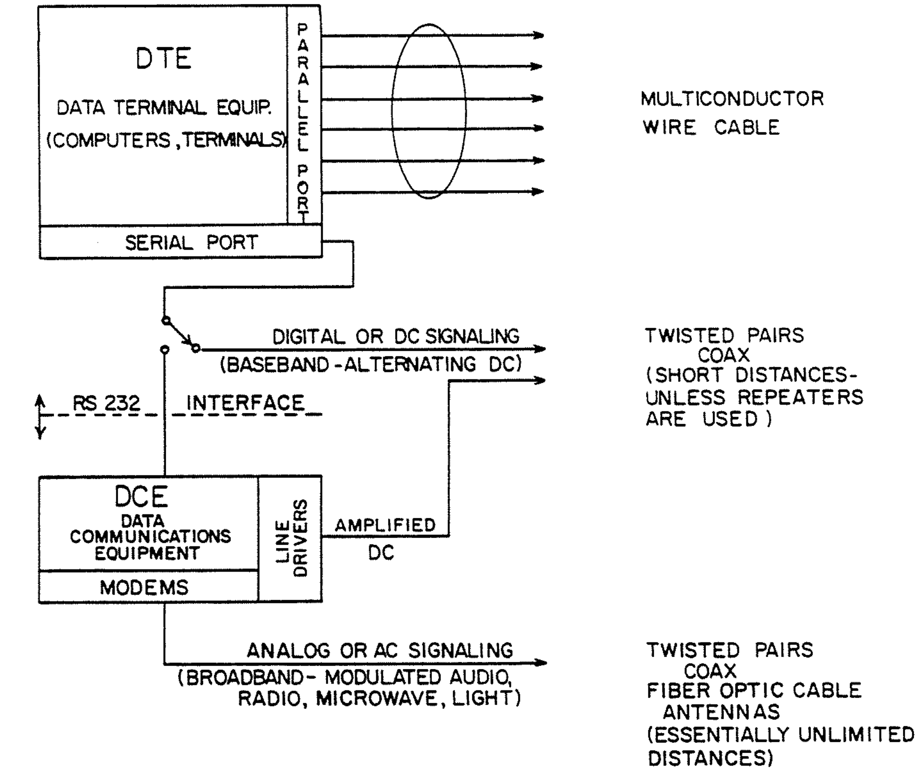

Figure 1. The physical link. Although the possible configurations are numerous, they all come down to: parallel or serial; digital or analog.

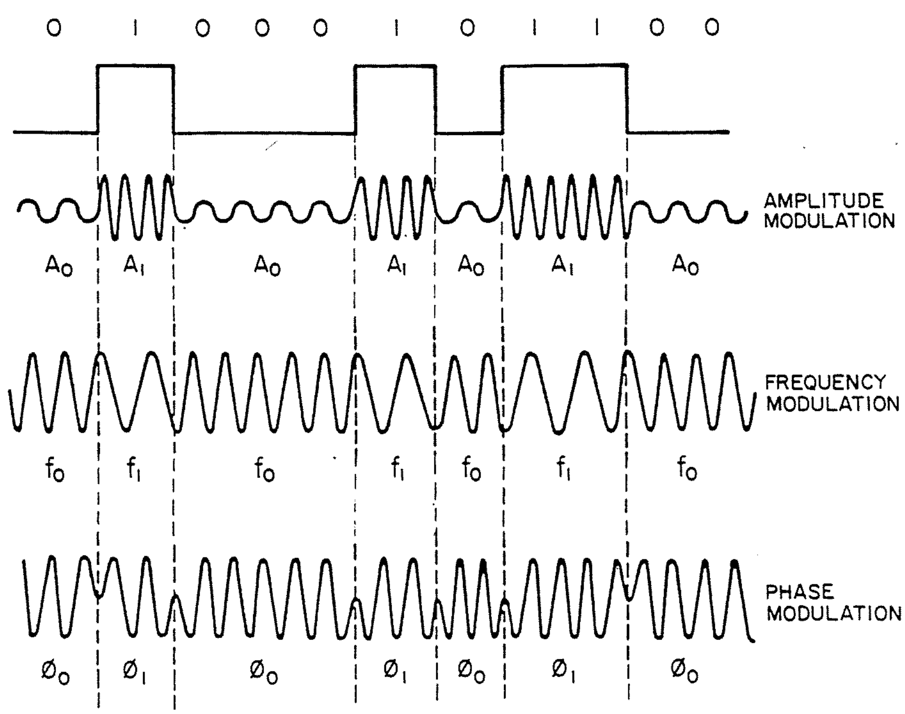

Figure 2. The amplitude, frequency and phase of a sinusoidal carrier wave can be modulated to transport binary data. The base frequency may range from that of an audio signal to that of light.

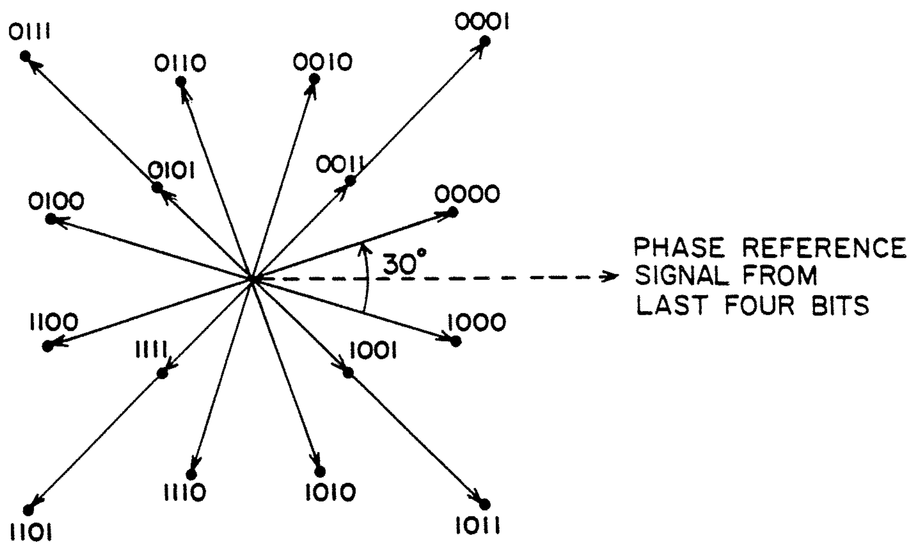

Figure 3. By combining phase and amplitude modulation, 16 discrete carrier wave conditions can be made to represent the 16 possible configurations of 4 bits. If the state of the carrier wave is changed at the rate of 2400 transitions per second, i.e., 2400 baud, the effective data transmission rate is 9600 bits per second.

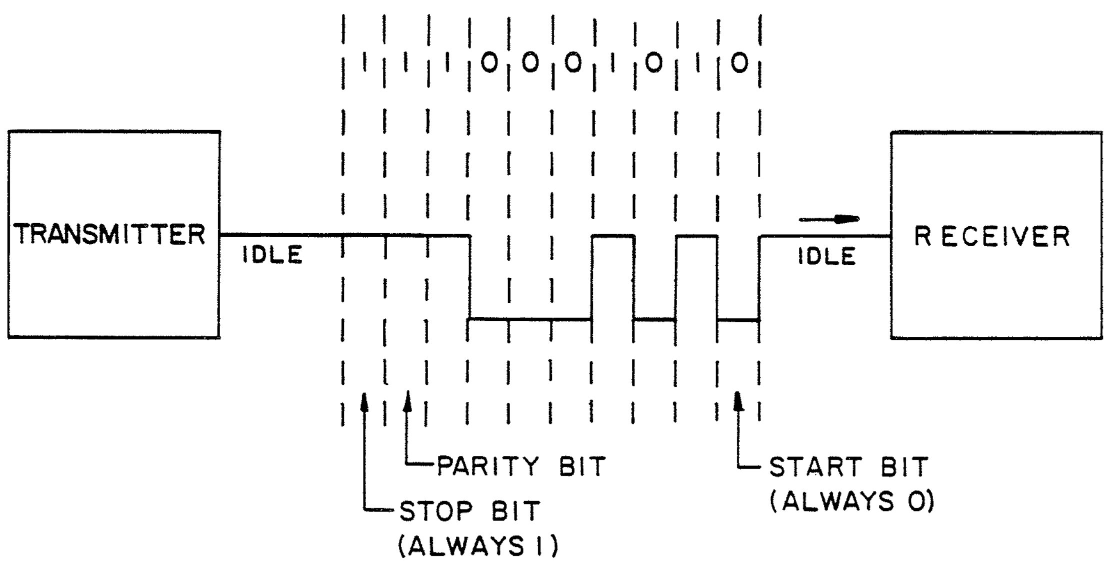

Figure 4. In asynchronous communication, each small group of bits, generally 5 to 8, represent an entire "message"--complete with framing and error-checking bits. The string shown could represent an even parity ASCII 'E'.

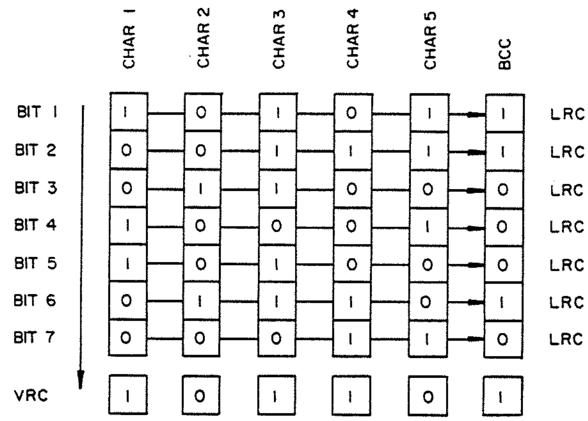

Figure 5. The concept of VRC/LRC. If even parity were in effect, the VRC parity bits and the LRC Block Check Character would be as shown.

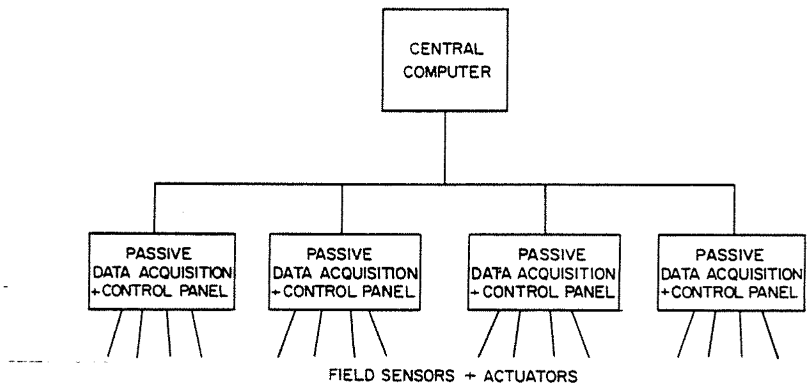

Figure 6. A vertically organized supervisory EMCS. All the computing is done centrally and the computer governs all communication, monitoring and control functions.

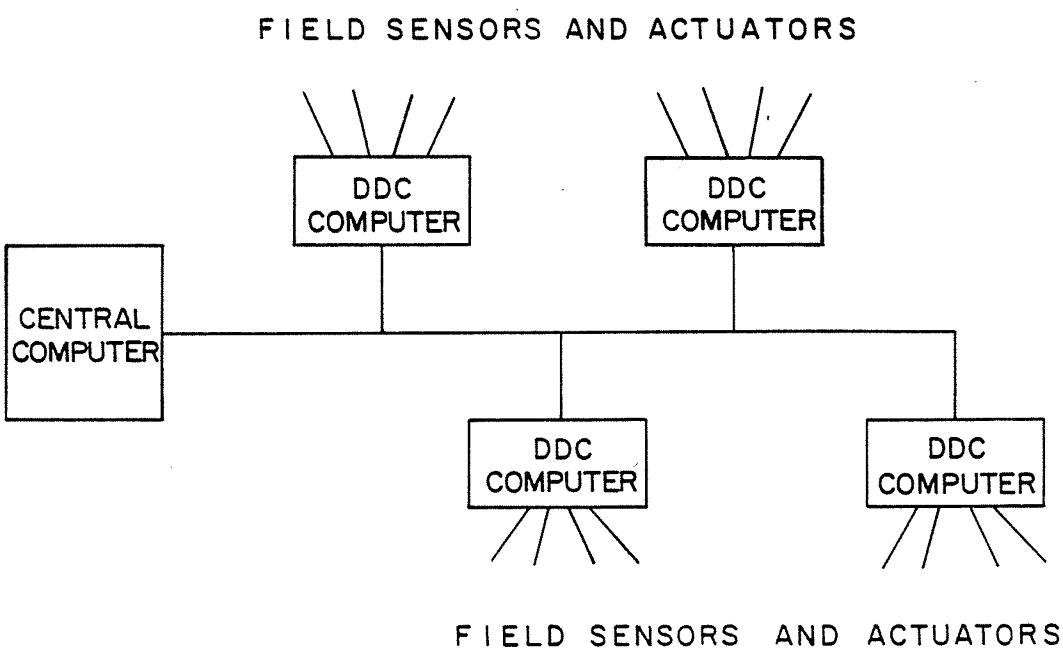

Figure 7. A horizontally organized distributed EMCS. The Direct Digital Control computers perform both closed loop and energy management functions, communicating with the central computer only when polled or when an alarmable change of state occurs