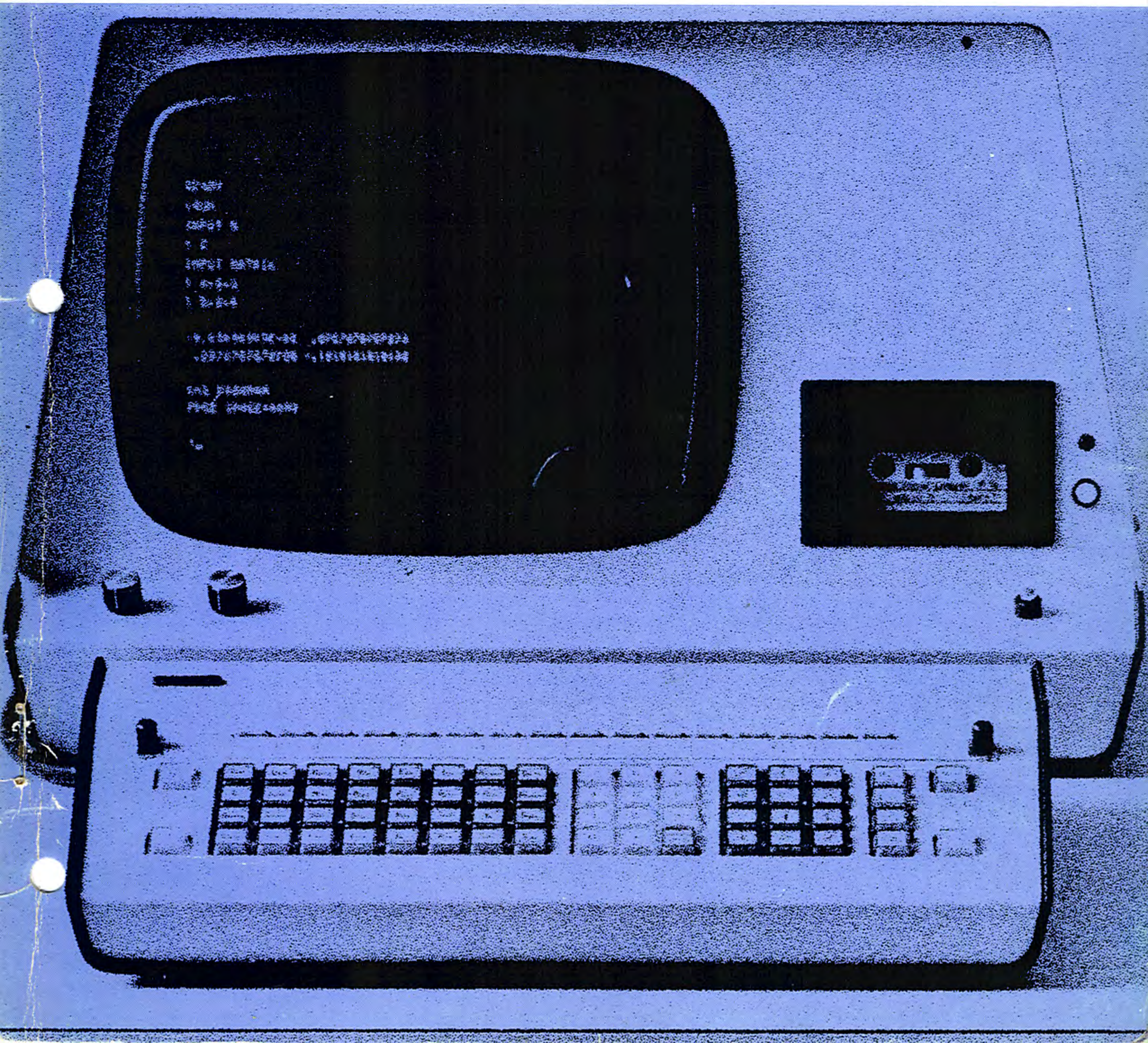
WANG

2227 TELECOMMUNICATIONS CONTROLLER REFERENCE MANUAL

SYSTEM 2200



2227

Telecommunications Controller Reference Manual

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

HOW TO USE THIS MANUAL

This manual provides answers to questions concerning the operation of the Model 2227 Telecommunications Controller. It is designed for users already familiar with the WANG System 2200 and its BASIC language.

Users not familiar with the System 2200 should read and understand the System 2200 A/B BASIC Programming Manual before proceeding with this manual.

The System 2200 A/B Reference manual serves as a supplement to this manual.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section I

General Information

INTRODUCTION

The Model 2227 Telecommunications Controller for the WANG System 2200 consists of a controller board (also called controller card) which plugs directly into the CPU (Central Processing Unit) chassis and a connector cable (length 12 feet or 3.6 meters). See Figure 1.

With the Model 2227 controller card/connector cable and a compatible modem (modulator/demodulator), either a data set or an acoustic coupler, the System 2200 can be used as a telecommunications terminal for data transmission over voice-grade telephone lines. If the System 2200 is connected to a Bell 103A, a Bell 103A3 (if auto-answer is required), or an equivalent modem, data transmission rates up to 300 baud are feasible. Using a Bell 202C or equivalent modem, data transmission rates up to 1200 baud are feasible.

Transmission rate is one of the switch selectable features of the Model 2227. Rates of 110, 150, 300, 600, or 1200 baud can be selected.

Over direct-connection lines, using a WANG Model 2227-N null modem or equivalent, data transmission rates up to 1200 baud are feasible for communications between two System 2200 Central Processing Units or between one System 2200 and a compatible computer.

Character formatting is another switch selectable feature on the Model 2227 controller board. Control over the following elements of character formats is provided: the number of data bits, parity checking, odd or even parity, and the number of stop bits.

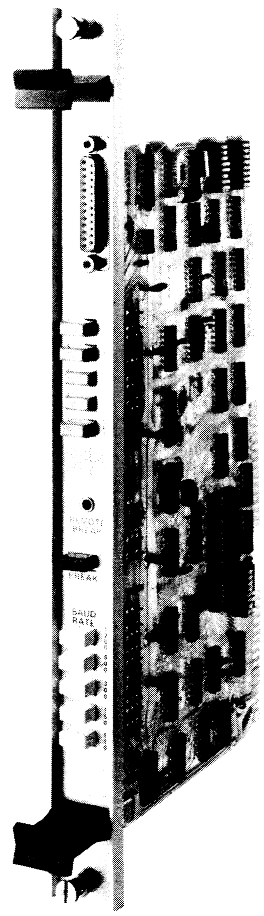


Figure 1. Model 2227 Telecommunications Controller

INSTALLATION

Call your WANG Service Representative. Request installation of the Model 2227 Telecommunications Controller.

MODEL 2227 FEATURES

Electrical Connection: RS-232-C

The EIA (Electronics Industry Association) standard specification defining the pin assignments and signals for "Interconnection of Data Terminal Equipment with a Communication Channel" is called RS-232. The Model 2227 Telecommunications Controller is designed to be compatible with the widely used revision of this standard, RS-232-C.

SECTION I – GENERAL INFORMATION

The controller card can receive a 25-pin RS-232-C male plug. Each end of the connector cable, provided with the board, contains one RS-232-C male plug. One end of the cable plugs into the Model 2227 board. The other end plugs into a Bell 103A modem or into any RS-232-C compatible data set or acoustic coupler (see Figures 2 and 3).

A list of the circuits and pin assignments in the Model 2227 connector is given in Table 1.

TABLE 1. MODEL 2227 CONNECTOR CABLE PIN ASSIGNMENTS

Pin	Circuit	I/O**	Function
1	AA	–	Protective Ground
2	BA	O	Transmitted Data
3	BB	I	Received Data
4	CA	O	Request to Send
5	––	–	––––
6	CC	I*	Data Set Ready
7	AB	–	Signal Ground
8	CF	I	Carrier Detector
9	––	–	––––
10	––	–	––––
11	––	–	––––
12	––	–	––––
13	––	–	––––
14	––	–	––––
15	––	–	––––
16	––	–	––––
17	––	–	––––
18	––	–	––––
19	––	–	––––
20	CD	O*	Data Terminal Ready
21	––	–	––––
22	––	–	––––
23	––	–	––––
24	––	–	––––
25	––	–	––––

* "Fail Safe" Circuit
**As seen by the Model 2227

Figure 2. The Bell 103A3 DATA-phone

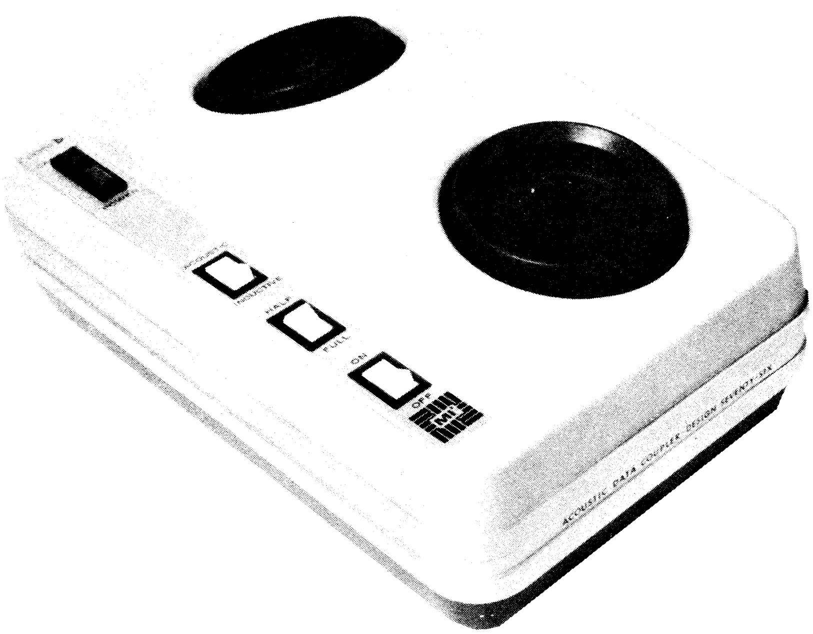


Figure 3. An Acoustic Data Coupler

SECTION I – GENERAL INFORMATION

Communication Mode: Duplex

Electronically, the Model 2227 is capable of full duplex transmission; however, the current method of program control limits data transmission to the half duplex mode. When using the Model 2227 and a modem equipped with a full/half duplex switch, set the modem switch in the full duplex position. The Bell 103A is full duplex automatically.

In communications, the duplex (i.e., full duplex) mode implies simultaneous two-way independent transmission between a pair of terminals. The half duplex mode implies alternate one-way-at-a-time independent transmission between terminals. A third mode, echo plex, implies simultaneous transmission of data over two routes: a) the normal route from transmitting to receiving terminal, and b) an "echo" route within the transmitting terminal.

Transmission Mode: Asynchronous

The Model 2227 controller board is designed to handle asynchronous (noncontinuous) transmissions only.

In asynchronous transmission, each character transmitted is synchronized individually by using a start bit and one or two stop bits to denote the beginning and the end of the train of bits corresponding to a character. On the other hand, in synchronous transmission, data is transmitted continuously with each terminal using substantially the same frequency and maintaining a desired phase relationship.

Transmission Rate: Push-button Selectable

The telecommunications controller card contains five "baud rate" buttons which provide switch selectable control of the data transmission rate. Available transmission rates are 110, 150, 300, 600, and 1200 baud.

If baud is defined as the reciprocal of the shortest signal element (usually one data-bit interval), then the baud rate equals the bit rate in synchronous transmission where each signal element is one data-bit long. In asynchronous transmission, baud rate equals the bit rate only if the STOP element is always one bit long. Often the STOP length used in asynchronous transmission is greater than one bit long (when data is transmitted from manual entry equipment such as keyboards, or when two stop bits are used instead of one), thereby making the baud rate differ from the bit rate.

Feasible transmission rates are:

a) 110, 150, or 300 baud over a voice-grade telephone line when the System 2200/2227 is connected to a Bell 103A compatible data set or equivalent,

b) 110, 150, 300, 600, or 1200 baud over a voice-grade telephone line when the System 2200/2227 is connected to a Bell 202C compatible data set or equivalent, and

c) 110, 150, 300, 600, or 1200 baud over a direct-connection line between two System 2200 Central Processing Units or between the System 2200 CPU and a compatible computer, if the connection is made via the WANG Model 2227-N null modem or equivalent.

Data transmission rates at the transmission and receiving terminals must agree. Furthermore, if a Bell 100 Series modem (or equivalent) is used at the transmitting terminal, a Bell 100 Series modem (or equivalent) must be used at the receiving terminal.

SECTION I – GENERAL INFORMATION

Character Format: Push-button Selectable

Five push-button switches on the telecommunications board control the character format of data transmissions. The labels on these buttons are NB1, NB2, PAR, OPS, and 1SB; their functions are explained in Tables 2 and 3.

In asynchronous transmission, each character is represented by a unique train of binary bits. A start bit marks the beginning of the character train. The format of the train is defined by specifying the following information:

the number of data bits;
whether or not a parity check is included;
if so, whether parity is odd or even; and
the number of stop bits.

> **NOTE:**
> *Character formats at the transmitting and receiving terminals must agree.*

Table 2 gives the range of character format information in general use for asynchronous data transmission and identifies the Model 2227 switches which facilitate character formatting.

TABLE 2. CHARACTER FORMATS IN ASYNCHRONOUS TRANSMISSION

Information	Quantity or Type	Model 2227 Capability	Model 2227 Switches
Start bit	1	Automatic	
Data bits	5, 6, 7, or 8	Switch Selectable	NB1, NB2
Parity	Odd, Even, or No	Switch Selectable	PAR, OPS
Stop bits	1 or 2	Switch Selectable	1SB

Table 3 contains a matrix of the Model 2227 push-button settings for control of character formats. In the table, the symbol D denotes the DOWN (i.e., depressed) position of a button. Similarly, U denotes the UP (i.e., released) position of a button.

The button configuration in the left column of Table 3 corresponds to the actual configuration of the character-format-buttons on the Model 2227 controller card. The matrix of positions for buttons NB1 and NB2 is printed on the controller board just below the NB1 button as an aid when setting these buttons for 5, 6, 7, or 8 data bits (see Figure 1).

TABLE 3. MODEL 2227 PUSH-BUTTON OPTIONS FOR CHARACTER FORMATTING

Buttons	Push-Button Options*	Function
1SB	D U 1 2	Number of Stop Bits
OPS	D U Odd Even	Odd Parity Selection
PAR	D U No Parity Parity	Parity Selection
NB2 NB1	D D U U D U D U 5 6 7 8	Number of Data Bits

* D = Down U = Up

Break Key and Remote Break Jack

The red BREAK key on the Model 2227 controller board, if depressed, sends a 200 millisecond spacing signal to the host CPU. The key is effective when the System 2200/2227 and the host CPU are connected to Bell 103A modems or equivalent. The host CPU interprets the break-signal as an interrupt and returns control to the receiving terminal.

The BREAK key is not effective when the System 2200/2227 and the host CPU are connected to Bell 202C modems or equivalent. The break-signal is interpreted by the host CPU as line noise or a parity error.

The Remote Break jack on the face plate of the Model 2227 controller board can be used to plug-in a Break key mounted on the keyboard for greater operator convenience. This key is not standard equipment for the System 2200.

Section II
Programming for the Model 2227

INTRODUCTION

Programming techniques for the System 2200 Model 2227 Telecommunications Controller are presented in this section. Included are descriptions of the following: BASIC language verbs used for telecommunications, address assignment, device selection, data verification, data transmission timing, multi-variable data messages, and message control. Some programmed examples are given. Each example describes a progressively more complex use of the telecommunications controller board.

VERBS USED FOR TELECOMMUNICATIONS

Each telecommunications controller (TC) unit in a System 2200 configuration is accessed for subsequent receiving operations (input), transmitting operations (output), or both receiving and transmitting operations, by using SELECT statements discussed later (see Device Selection). Receiving and transmitting operations cannot be programmed for simultaneous execution, but the operations can be interspersed.

Logic for data reception can be programmed using INPUT statements with numeric variables or alphanumeric string variables as arguments. BASIC language restrictions on legal names for variables and the maximum number of characters assignable to each type of variable are the same for telecommunications applications as for other applications of the System 2200. See the System 2200A/B BASIC Programming Manual.

An INPUT statement must contain at least one argument (often, only one is needed) to identify received data when it is stored in memory. In a multi-argument INPUT statement, each successive variable must be separated from the preceding variable by a comma. As an option, a literal string message can be included in the statement to supply the System 2200 operator with an explanation of the required input when the statement is executed. If included, the literal string must be enclosed in quotation marks, must follow the INPUT verb (with no comma between), and must be separated from the first or only variable in the statement by a comma.

A summary of the System 2200 processing procedure for INPUT statements is given in Appendix D. Users with special telecommunications requirements, such as data streams with more than 64 characters, should study the summary before writing programs. The INPUT-statement-processing-procedure provides the rationale for programming techniques employed in PRINT statements used for data transmission as well as INPUT statements used for data reception.

The KEYIN verb, available only in the System 2200B, can be used for data reception; but inherent limitations restrict its use. The KEYIN verb differs in several ways from the INPUT verb, available in the System 2200 A and B. The KEYIN verb can be used only in the Program Mode; the INPUT verb can be used in the Immediate Mode and the Program Mode. The KEYIN verb has no inherent display capability. The INPUT verb produces an echo of data being received in the buffer of the device last selected for INPUT-class operations; the INPUT-data-echo is printed on the CRT (or the device last selected for CO-class operations). See Appendix B and Chart B-1.

A KEYIN statement must contain one (and only one) alphanumeric variable followed by two line numbers, with commas after the variable and the first line number. When a KEYIN statement is processed during program execution, the system checks the buffer of the device last selected for INPUT-class operations. System action depends upon the input condition existing in the device. One of three conditions is possible:

a) Device not ready –– execution advances to the next statement in the program.

b) Device ready with character –– the character is transferred from the buffer to memory and stored in the specified alphanumeric variable; then execution branches to the *first line number* specified in the KEYIN statement.

c) Device ready with a special-function-code –– the special function code is stored in the specified alphanumeric variable; then execution branches to the *second line number* specified in the KEYIN statement. If the keyboard is the current input-device, the special function code is one of the codes HEX(00) through HEX(1F) corresponding to the 32 possible special-function-keys available for input. If the input channel of a TC board is the current input-device, the special function code is HEX(4D). The code HEX(4D) is supplied by the TC board when a parity error is detected in data being received.

KEYIN statements are well suited for reception of the single-character-messages (ENQ, ACK, or NAK) used for overall control of data-transfer between terminals. With the logic of a simple loop, KEYIN statements can be used to test input devices (i.e., the keyboard and each TC unit in a multi-unit telecommunications configuration) one at a time. If no control-character is waiting in the buffer of the last selected INPUT-class-device, another device is tested. Or, if a control-character is waiting, execution branches to the program line whose number matches the first specified line number in the KEYIN statement. See Examples 5 and 6.

KEYIN statements are not recommended for reception of multicharacter data streams. However, if desired, such statements can be used for data reception at low data transmission rates.

INPUT statements, by contrast, can be used for reception of multicharacter data streams at all data transmission rates available on the Model 2227 Telecommunications Controller. Furthermore, a multivariable INPUT statement can be used to receive messages consisting of more than the 64 character storage limit available for a single alphanumeric variable. However, long messages must contain data-separator-commas at intervals not exceeding 64 characters and a carriage-return at an interval not exceeding 191 characters. See Appendix D.

Logic for data transmission can be programmed using PRINT statements. Also, PRINTUSING statements with their companion Image statements can be used to supply commas in multivariable data transmissions requiring data-separator-commas in the output data stream.

Data transmission via PRINT statements is illustrated in Appendix D, Example D5. Data transmission via PRINTUSING statements is illustrated in Section 2 (see Multi-variable Data Messages).

The telecommunications capabilities of the System 2200/2227 can be extended by additional verbs, provided in Option 2 –– General I/O ROM.

ADDRESS ASSIGNMENT

The Model 2227 Telecommunications Controller is both an input and an output device for the System 2200. Therefore, each Model 2227 controller board is assigned two addresses, an input address and an output address, each of which must be unique within the System 2200 configuration to which the unit belongs.

The device address of every System 2200 peripheral unit, regardless of the category of the peripheral, is specified by a three-hexadecimal-digit code of the form XYY. The first hexadecimal digit (i.e., the leftmost or X digit) represents the Device Type. The last two hexadecimal digits (i.e., YY) represent the unique device address of the specific peripheral unit.

SECTION II – PROGRAMMING FOR THE MODEL 2227

A list of device address codes for the System 2200 peripherals is given in Appendix A. The list provides a standard for unique address assignments; however, other addresses may be assigned by customer request.

Three unique device addresses are reserved for the *input channel* of telecommunications controller boards; the addresses are 219, 21A, and 21B. Three unique addresses are reserved for the *output channel* of telecommunications controller boards; the addresses are 21D, 21E, and 21F.

The input-channel-address of a TC board is assigned by setting the 8-pole RCV switch located on the printed circuit board; the output-channel-address is assigned by setting the 8-pole XMT switch. The RCV and XMT switches are not visible after the controller board is mounted in the CPU. Address switches are set by the factory before shipment or by a WANG Service Representative when a controller board is installed. A label showing the addresses of the TC unit is placed on the controller board face plate. The label is visible for reference purposes since the correct addresses must be used when writing applications programs for the System 2200.

Normally, a System 2200/2227 configuration with one TC unit is installed with the address code 219 for the TC input-channel and address code 21D for the output-channel.

A system with two TC units is installed as follows: the address codes 219 (input) and 21D (output) are assigned to the "first" TC unit; the addresses 21A (input) and 21E (output) are assigned to the "second" TC unit. In a system with three TC units, the addresses for the "third" TC unit are 21B (input) and 21F (output). See Table 4.

TABLE 4. DEVICE ADDRESSES FOR TC UNITS IN A SYSTEM 2200/2227 WITH THREE MODEL 2227 BOARDS

Model 2227 TC Boards	Device Address Codes		Switch Settings Hexadecimal	
	Input Channel	Output Channel	RCV	XMT
Unit 1	219	21D	19	1D
Unit 2	21A	21E	1A	1E
Unit 3	21B	21F	1B	1F

DEVICE SELECTION

When the System 2200 is Master Initialized (power turned off and then on), the "primary devices" are selected automatically for input/output operations. A primary device is a device whose address is one of the five default addresses for the System 2200. The keyboard with address 001 and the CRT with address 005 are two of the primary devices.

The Model 2227 Telecommunications Controller is not a primary device. A SELECT statement must be executed by the system either in the Immediate Mode or the Program Mode before data can be transmitted via a TC unit.

See Appendix B of this manual. There, the primary devices are listed in Table B1, and the I/O operations and I/O class parameters for the System 2200 are identified in Chart B1. Also, the general format of the SELECT statement is discussed and examples are given.

After reading Appendix B, consider a sample System 2200 configuration consisting of the primary devices in Table B1 and three telecommunications units with the addresses shown in Table 4. A diagram of a system with three TC units is shown in Figure 4 (see Multi-Variable Data Messages), but no primary device model numbers or addresses appear in the diagram.

SECTION II – PROGRAMMING FOR THE MODEL 2227

For telecommunications applications, device selection is an important consideration whether a System 2200/2227 contains one TC unit or more. For example, in a triple-unit configuration, assume the following:

a) the system has been Master Initialized,
b) a program stored on tape has been loaded into memory,
c) the command RUN has been entered via the keyboard, and
d) the EXECUTE key has been touched.

If the first statement of the program in memory is

```
INPUT A$
```

the system recognizes the default address 001 (the keyboard) as the device to access for execution of INPUT-class operations (statements with the INPUT or the KEYIN verb). A question mark appears on the CRT to indicate the system is awaiting data. Until data is supplied via the keyboard, program execution ceases.

However, if the objective of the input statement in the program is data reception via TC Unit 1, the program should begin as follows:

```
10 SELECT INPUT 219
20 INPUT A$
```

When Line 10 is executed, address 219 (the input channel of TC Unit 1) becomes the current device for all subsequent INPUT-class operations. Then, when Line 20 is executed, a question mark appears on the CRT to indicate the system is awaiting data. Meanwhile, the system accesses the device with address 219 for data; and the keyboard is locked out. As data is received, an echo of the data appears on the CRT. (If no data-echo appears, the operator must touch the RESET key to regain control of the system.)

If the next line encountered is

```
30 INPUT B$
```

the system again accesses device 219 for data. The keyboard is ignored as a data source for all INPUT-class operations until after the statement

```
SELECT INPUT 001
```

is encountered in the program.

Similarly, TC Unit 3 does not become the data reception device until the system executes the statement:

```
SELECT INPUT 21B
```

Furthermore, no TC unit becomes a channel for data transmission until the system executes a SELECT statement with the I/O-class parameter PRINT followed by the address of the output channel of a TC unit. For example, to transmit data via TC Unit 1, in the sample configuration, the program must contain the statement:

```
SELECT PRINT 21D
```

Or, to transmit data via TC Unit 3, the statement:

```
SELECT PRINT 21F
```

must appear in the program.

Two-way data transfer (receiving and transmitting via the same TC unit) is programmable. For example, upon execution, the statement:

```
SELECT INPUT 219, PRINT 21D
```

assigns the TC Unit 1 input-channel-address as the device for all subsequent INPUT-class operations, and also assigns the TC Unit 1 output-channel-address as the device for all subsequent PRINT-class operations. Even though receiving and transmitting operations cannot be executed simultaneously, they can be interspersed by the program logic.

If CRT dialogue (messages and displays for an operator) is desired, a programmer must provide for reselection of the CRT as needed by including statements of the form

```
SELECT PRINT 005
```

in the program logic. (See examples at the end of Section 2.)

MULTI-VARIABLE DATA MESSAGES

Consider data transfer between two System 2200's called System I and System II. Let System I include a three-unit telecommunications configuration with device address codes identical to those in Table 4. Let System II be a one-unit telecommunications configuration with the standard input address code 219 and output address 21D. Let System I be linked to System II via Unit 3 as shown in Figure 4.

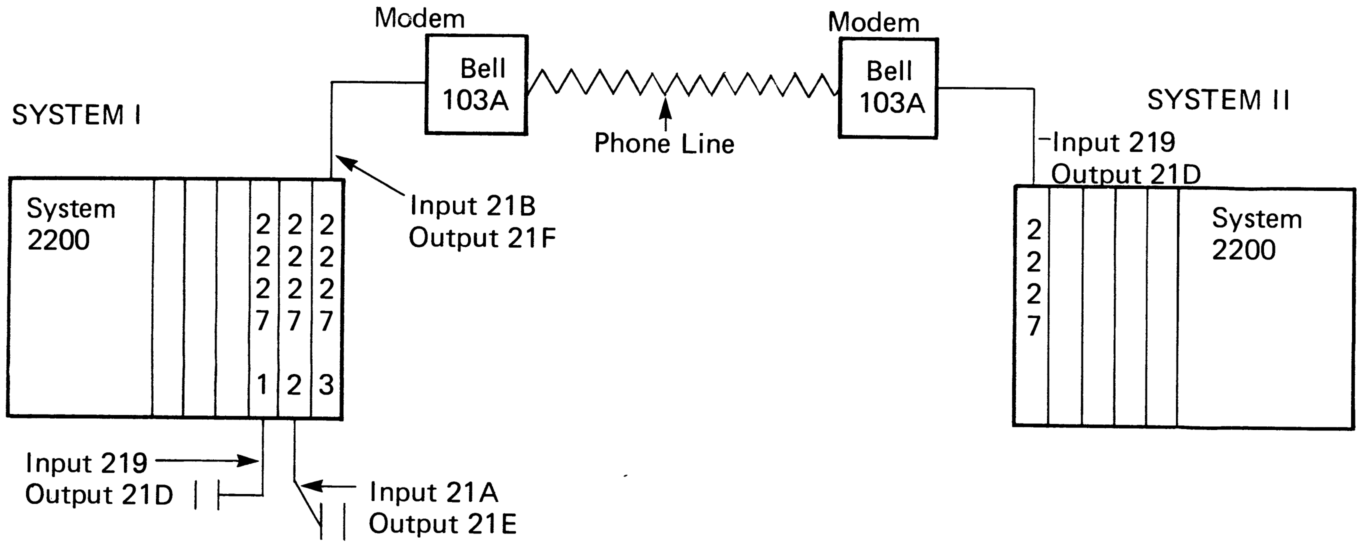


Figure 4 Data Transfer Between System I and System II

If data corresponding to more than one variable is to be transmitted, the program for the transmitting (sending) System 2200 must structure the data stream with variable separators (i.e., commas). For example, the program for System I to send data via Unit 3 contains the following statements:

```
70 SELECT PRINT 21F
80 PRINTUSING 90, A$, B, C, D$
90 %##### ,##### ,##### ,###
```

where a space preceding each comma serves as a pound-sign-field-delimiter in the image statement. Without a space before each comma, image statement 90 formats only one variable.

The program for System II to receive the above data stream includes the statements:

```
30 SELECT INPUT 219
40 INPUT A$, B, C, D$
```

DATA VERIFICATION

Parity formatting of transmitted data and parity checking of received data are push-button selectable features of the Model 2227 peripheral. Section 1 (see Character Formats) contains the push-button settings which correspond to available parity options.

Once the parity feature is activated, the parity of each character is verified as it is received. When a parity error is detected, a carriage return is issued immediately. The System 2200 processor special-function-indicator is set.

If an INPUT statement is being processed, a branch is made to the primed subroutine DEFFN '77; i.e., HEX(4D). However, if a KEYIN statement is being processed when the parity error is detected, the character received by the System 2200 is HEX(4D), the ASCII code for uppercase M. Then, program execution branches to the second line number specified in the KEYIN statement.

Of course, the receiving system program must contain a special-parity-error-function subroutine DEFFN '77 written to produce a time-delay greater than the length of time normally required to receive a line of data and, also, to request retransmission of the data line.

Sample DEFFN '77 and KEYIN default subroutines are included in Example 3 of this section.

TRANSMISSION TIMING

Depending upon the "baud rate" push-button setting on the controller card, the data transfer rate via the Model 2227 Telecommunications Controller is 110, 150, 300, 600, or 1200 baud.

Depending upon the setting of the five character-format-buttons, the total number of bits-per-character in the data is the *sum* of the following:

1 start bit;
5, 6, 7, or 8 data bits;
0 or 1 parity bit; and
1 or 2 stop bits.

Therefore, the *total* bits-per-character ranges from a minimum of seven to a maximum of twelve.

Knowing the data transfer rate in baud and the total bits-per-character in the format, the transmission timing is expressable in either characters-per-second (see Table 5) or milliseconds-per-character (see Table 6).

TABLE 5. TRANSMISSION TIMING IN CHARACTERS PER SECOND

DATA RATE IN BAUD \ TOTAL BITS PER CHARACTER	7	8	9	10	11	12
110	15.71	13.75	12.22	11.00	10.00	9.16
150	21.42	18.75	16.66	15.00	13.63	12.50
300	42.85	37.50	33.33	30.00	27.27	25.00
600	85.71	75.00	66.66	60.00	54.54	50.00
1200	171.42	150.00	133.33	120.00	109.09	100.00

TABLE 6. TRANSMISSION TIMING IN MILLISECONDS PER CHARACTER

DATA RATE IN BAUD \ TOTAL BITS PER CHARACTER	7	8	9	10	11	12
110	63.63	72.72	81.81	90.90	100.00	109.09
150	46.66	53.33	59.99	66.66	73.33	80.00
300	23.33	26.66	30.00	33.33	36.66	40.00
600	11.66	13.33	15.00	16.66	18.33	20.00
1200	5.83	6.66	7.50	8.33	9.16	10.00

OVERALL DATA TRANSMISSION CONTROL

Several examples presented at the end of Section 2 utilize the standard ASCII control characters for enquiry, acknowledge, and negative acknowledge (see Appendix C). These characters (ENQ, ACK, and NAK) provide overall control of data transfer between terminals. The ENQ code is issued when a terminal wishes to send a message. ACK is issued to acknowledge successful reception. NAK is issued to acknowledge unsuccessful reception. See flowchart in Figure 5 (page 26).

Table 7 contains the two-digit hexadecimal notation and equivalent seven-bit binary notation for these control characters.

TABLE 7. ASCII CHARACTERS FOR OVERALL TRANSMISSION CONTROL

CONTROL CHARACTERS	HEXADECIMAL NOTATION	7-BIT BINARY NOTATION
ENQ	05	0000101
ACK	06	0000110
NAK	15	0010101

Therefore, to transmit one of these characters using a System 2200-2227, the following statements *seem* appropriate:

a) SELECT PRINT XXX (length)

b) PRINT HEX (YY)

where XXX is the Model 2227 Telecommunications output device address and YY is the two-digit hexadecimal code for a control character from Table 7. The length, if given, is an integer (less than 256) specifying the desired carriage width; if omitted, the default length (64) is used.

However, output character codes with zero as the first hexadecimal digit do not increment the character-count made by the system for its end-of-line processing procedure; therefore, a PRINT statement with only an ENQ or ACK character produces a character count of zero for the line being processed. In such cases, no special end-of-line characters (i.e., Carriage Return, Line Feed, or Null) are added by the transmitting System 2200.

On the other hand, if the receiving System 2200 is using the INPUT verb to request data, the requested data stream (even a single character message) must terminate with one or more special end-of-line characters in order for the system to complete the processing procedure for the input statement. This requirement places a constraint on the PRINT statement when used to transmit an ENQ or ACK message. The statement must produce a Carriage Return and a Null character in addition to the ENQ or ACK character.

Accordingly, the examples which follow contain statements of the form

c) PRINT HEX (YY0D00)

where YY, as before, is the two-digit hexadecimal code from Table 7; 0D is the hexadecimal code for the Carriage Return character; and 00 is the code for the Null character (see Appendix C). The examples also contain the statement PRINT HEX (150D) for transmission of NAK which has a non-zero first hexadecimal digit. A BASIC statement line can contain any number of HEX codes. If more than one HEX code is included in sequence, the line can be written two different ways. Statement (c) above is an alternative to the following format:

d) PRINT HEX (YY); HEX (0D); HEX (00)

SECTION II – PROGRAMMING FOR THE MODEL 2227

SPECIAL EOM CHARACTER

Special EOM (end-of-message) microswitches, not visible when the Model 2227 board is mounted in the CPU, are set to HEX(11) which corresponds to the special X-ON character sometimes used to terminate data streams. With this setting, each X-ON character received by the Model 2227 board is translated into a carriage return character; i.e., HEX(0D).

This substitution of HEX(0D) for HEX(11) is an important feature. As mentioned previously, a data stream requested by the INPUT statement/verb *must* terminate with a carriage return character for the System 2200 to complete the processing procedure associated with INPUT statements.

Teletype® messages are terminated with an X-ON character. Also, the WANG Computer Systems' time-sharing computer WYLBUR (see Example 7) terminates interrogational messages with a HEX(11) character after the question mark in a data stream.

By special request, the EOM microswitches can be set to a different hexadecimal representation. However, the customer requesting such a change should realize that a carriage return character is substituted always for the character set on the EOM microswitches. If the EOM switches correspond to a character ordinarily imbedded in a data stream, difficulty arises; a carriage return replaces the imbedded character, and the INPUT statement is processed prematurely.

EXAMPLES

Several examples of programming for the System 2200 Model 2227 Telecommunications Controller are given in outline form. Program statements are presented in the left-hand column with explanations in the right-hand column. Flowcharts accompany most of the examples. The examples are guides-to-programming rather than working models. The list below indicates the topics covered by the examples.

Example 1:	Minimum statements for data transmission.
Example 2:	Minimum statements for data reception.
Example 3:	Data reception including ENQ, ACK, and NAK for overall control; KEYIN to poll data sources; INPUT to receive data; and parity checking.
Example 4:	Data transfer test for matching the settings of character format and rate buttons on a receiving System 2200-2227 to the unknown settings on a sending System 2200-2227.
Example 5:	Two-way multi-line data transfer with CRT dialogue, ENQ/ACK/NAK control, and no parity.
Example 6:	Two-way multi-line data transfer with CRT dialogue, ENQ/ACK/NAK control, and parity checking including logging of errors.
Example 7:	Data transfer between the System 2200-2227 and a time-sharing computer terminal.
Example 8:	Teletype terminal simulation.

Example 1: Minimum statements for data transmission.

STATEMENTS	REMARKS
10 DIM T$64	Dimension string variable T$ to 64.
20 T$ = " "	Clear variable T$.
30 SELECT PRINT 21D	Select output device Model 2227.
40 T$="ABCDE"	Assign data to variabie.
50 PRINT T$	Transmit data via Model 2227.*

*The data-line to be transmitted contains at least one character with a non-zero first-digit hexadecimal code. Therefore, the end-of-line processing procedure of the sending System 2200 automatically terminates the data stream with a Carriage Return character. Thus, the data stream is suitable for the receiving System 2200 processing procedure initiated by the INPUT statement (line 40) in Example 2.

® Registered Trademark, Teletype Corporation

Example 2: Minimum statements for data reception.

STATEMENTS	REMARKS
10 DIM R$64	Dimension string variable R$ to 64.
20 R$ = " "	Clear variable R$
30 SELECT INPUT 219	Select input device Model 2227.
40 INPUT R$	Receive data via Model 2227.

Example 3: Data reception including ENQ, ACK, and NAK for overall control; KEYIN to poll data sources; INPUT to receive data; and parity checking.

The KEYIN verb is available only on the System 2200-B. This is an outline, not a complete program (see flowchart in Figure 6).

STATEMENTS	REMARKS
10 DIM R$1,K$1	Dimension variables R$ and K$.
20 R$,K$=" "	Clear variables.
30 SELECT INPUT 219	Select TC (Model 2227) for input.
40 KEYIN R$,100,700	Accept control byte, if any (see Note 1).
50 SELECT INPUT 001	Select keyboard for input.
60 KEYIN K$,300,300	Accept keyboard input, if any.
70 GOTO 30	Continue polling procedure.
100 SELECT PRINT 21D	Select TC for output.
110 IF R$=HEX(05) THEN 140	If ENQ, go to 140.
120 PRINT HEX(150D)	Send NAK to originator.
130 GOTO 20	Branch to polling procedure.
140 PRINT HEX(060D00)	Send ACK.
150 INPUT "DATA",R	Receive data via TC.
160 PRINT HEX(060D00)	Send ACK.
170 GOSUB 500	Process data.
180 GOTO 20	Branch to polling procedure.
300 . . .	Start logic for keyboard input.
.	
380 GOTO 20	Branch to polling procedure.
500 . . .	Begin subroutine to process data received via TC.
.	
580 RETURN	End of subroutine.
700 FOR Z=1 TO 1000	KEYIN parity-error default subroutine (see line 40).
710 NEXT Z	Produce time-delay.
720 SELECT PRINT 21D	
730 GOTO 120	Send NAK to originator.
900 DEFFN'77	INPUT parity-error default subroutine (see line 50).
910 FOR Z=1 TO 2000	Time-delay for maximum data interval.

(continue)

Example 3 (continued)

STATEMENTS	REMARKS
920 NEXT Z	
930 SELECT PRINT 21D	
940 PRINT HEX(150D)	Send NAK to originator.
950 RETURN	Request retransmission of data (line 150).

Note 1. If no character is ready to come in from the TC buffer, program execution continues at the next statement (i.e., 50). If a character is ready, execution continues at the first line number of the KEYIN statement (i.e., 100). If a parity error is detected, the character is not read into the system; execution continues at the second line number of the KEYIN statement.

Example 4: Data transfer test for matching the settings of character format and rate buttons on a receiving System 2200-2227 to the unknown settings on a sending System 2200-2227.

This two-part program is not an example of an operational system; no control byte transmission or parity-error-logic is included. However, since "baud rate" and character format switch settings can be changed during execution, the program serves as a "trial and error" procedure for matching the push-button settings on the controller board of a receiving System 2200 to the unknown settings on the board of a sending System 2200. A flowchart is presented in Figure 7.

OPERATING INSTRUCTIONS FOR EXAMPLE 4

1) Load the two-part program in each System 2200.
2) Set the Model 2227 push button switches, using any feasible combination of settings.
3) On the keyboard of the sending System 2200, touch the RUN key and then the EXECUTE key.
4) On the keyboard of the receiving System 2200, touch RUN and key in 110, then touch EXECUTE.
5) When the CRT of the sending system displays the message LINE TO TRANSMIT and a question mark appears on the next line of the screen, type in a line of test data followed by a Carriage Return.
6) Watch the CRT of the receiving system for a display of the test data as it is received repetitively.
7) Since good data transfer occurs only when the switches on the sending and receiving Model 2227 controller boards are set identically, test the match by changing the settings of the baud switches and character format switches on the receiving system boards until data reception is satisfactory.

PROGRAM LOGIC FOR EXAMPLE 4

STATEMENTS	REMARKS
10 REM 2200 TO 2200 VIA 2227	
20 REM LOGIC FOR SENDING 2200	
30 DIM T$64	Dimension variable T$ to 64.
40 T$=" "	Clear variable.
50 PRINT "LINE TO TRANSMIT"	Display instruction on CRT.
60 INPUT T$	Request data line.
70 SELECT PRINT 21D	Select TC (Model 2227) for output.
80 PRINT T$	Transmit data line.
90 GOTO 80	Loop to retransmit. (See Note 2.)
110 REM LOGIC FOR RECEIVING 2200	

(continue)

Example 4 (continued)

STATEMENTS	REMARKS
120 REM KEY FUNCTION 0 TO DISPLAY STATUS	
130 DIM R$64,A$64	Dimension variables A$ and R$.
140 SELECT INPUT 219	Select TC for input.
150 INPUT A$	Receive data once to set sync.
160 A$=" "	Clear variable A$.
170 INPUT A$	Receive data (assign to A$).
180 R$=" "	Clear variable R$.
190 INPUT R$	Receive data again (assign to R$).
200 C=C+1	Increment reception counter C.
210 IF R$=A$ THEN 180	Verify received data and loop if identical.
220 E=E+1	If not identical, increment error counter E.
230 DEFFN'0	Key 0 special function.
240 SELECT PRINT 005	Select CRT for display.
250 PRINTUSING 270,C,E	Print reception/error summary.
260 GOTO 140	Loop to resync. (See Note 2)
270 %RECEPTIONS=#,###,### ERRORS=###,###	Output format for C and E.

Note 2. To interrupt the loop, touch the BREAK key on the Model 2227 controller board.

Example 5: Two-way multi-line data transfer with CRT dialogue, ENQ/ACK/NAK control, and no parity.

This program includes logic for multi-line data transmission with CRT dialogue and overall control of data transmission using ENQ, ACK, and NAK. For simplicity, parity-error subroutines are omitted.

A simplified flowchart presenting the interaction of the sending and receiving systems is given in Figure 8a. A more detailed flowchart is given in Figure 8b.

OPERATING INSTRUCTIONS FOR EXAMPLE 5

1) Load the program in each System 2200.
2) Identically set the Model 2227 push buttons (no parity).
3) Touch the RUN key followed by the EXECUTE key, in each system.
4) Now, in the system wishing to initiate message transmission, touch the RESET key. Then depress Special Function Key 1.
5) When the CRT displays the message READY TO TRANSMIT** KEY IN DATA, followed on the next line by a question mark, enter a data line (maximum 64 characters).
6) When the message reappears after each data line is accepted, continue to transmit additional data, or
7) terminate transmission by touching the EXECUTE key without supplying a data line.

PROGRAM LOGIC FOR EXAMPLE 5

STATEMENTS	REMARKS
100 REM 2200-2200 VIA 2227, CRT DIALOGUE	
110 REM ENQ/ACK/NAK CONTROL	
120 DIM R$64,T$64	Dimension variables.
130 DEFFN'0	Key 0 special function.
150 SELECT PRINT 005	Select CRT for output.
160 PRINT "IDLE WAIT"	Display remark.
170 SELECT INPUT 219	Select TC for input.
180 R$=" "	Clear variable.
190 INPUT R$	Receive data via TC.
200 R=LEN(R$)	Save data length.
210 IF R <> 1 THEN 150	Loop if R ≠ 1.
220 IF R$=HEX(06) THEN 300	If ACK, go to 300.
230 IF R$ <> HEX(05) THEN 150	If not ENQ, go to 150.
240 SELECT PRINT 005	If ENQ, select CRT.
250 PRINT "WAITING FOR EXPECTED DATA"	Display remark.
260 SELECT PRINT 21D	Select TC for output.
270 PRINT HEX(060D00)	Send ACK.
280 SELECT PRINT 005	Select CRT.
290 GOTO 170	Loop for new input.
300 SELECT PRINT 005	ACK received, select CRT.
310 PRINT "READY TO TRANSMIT**KEY IN DATA"	Display remark.
320 SELECT INPUT 001	Select keyboard for input.
330 T$=" "	Clear variable.
340 INPUT T$	Request KB data.
350 SELECT PRINT 21D	Select TC for output.
360 IF T$=" "THEN 150	If no data (CR only), loop to idle.
370 PRINT T$	Send data.
380 GOTO 300	Loop for new KB input.
390 DEFFN'1	Key 1 special function.
400 SELECT PRINT 21D	Select TC for output.
410 PRINT HEX(050D00)	Send ENQ.
420 SELECT PRINT 005	Select CRT.
430 PRINT "WAITING FOR RESPONSE"	Display remark.
440 GOTO 170	Loop to wait.
450 DEFFN'15	Key 15 special function.
455 SELECT PRINT 005	Select CRT.
460 PRINT HEX(03)	Clear CRT display.
470 PRINT "KEY 0 RESTORE IDLE WAIT"	Display Key Picture.
480 PRINT "KEY 1 INITIATE MESSAGE"	(i.e., define special function keys).
490 PRINT "KEY 15 KEY PICTURE"	
500 GOTO 150	Return to 150.

Example 6: Two-way multi-line data transfer with CRT dialogue, ENQ/ACK/NAK control and parity checking including logging of errors.

Using ENQ, ACK, and NAK for overall message control, this program permits multi-line transfer of data between two systems at transmission rates up to 1200 baud. A parity-error default subroutine introduces a time-delay each time a parity error is detected and then requests retransmission of the data line. Parity errors are logged. Control messages and data messages are monitored with CRT displays.

The program is loaded in each System 2200. One system becomes the originator; the other becomes the receiver. The flowchart is presented in four parts (see Figures 9a, 9b, 9c, and 9d).

Flowchart Part 1 for Example 6 shows the logic for the receiving System 2200. Flowchart Part 2 shows the logic for the sending System 2200 (i.e., the originator). Flowchart Part 3 contains the reception loop in Column 1, the transmission loop in Column 2, and the parity-error-processing procedure in Column 3. Flowchart Part 4 contains the subroutines for transmitting and for receiving the enquiry code ENQ.

OPERATING INSTRUCTIONS FOR EXAMPLE 6

1) Load the program in each System 2200.
2) Identically set the Model 2227 push buttons.
3) Touch the RUN key followed by the EXECUTE key in each system and watch for the display "RECEIVE ENQ?"
4) Choose one System 2200 to be the originator. In this system,
   a) touch the RESET key, then
   b) depress Special Function Key 16.
5) Watch for the display "READY TO RECEIVE ** TYPE MESSAGE LINE".
6) Type a data line (maximum 64 characters) and then touch the EXECUTE key.

These instructions initiate a testing and display loop.

PROGRAM LOGIC FOR EXAMPLE 6

STATEMENTS	REMARKS
10 REM 2200-2200 VIA 2227	
12 REM PARITY CHECKING	
14 REM ENQ/ACK/NAK CONTROL	
16 REM ENQ="05" ACK="06" NAK="15"	
20 DIM R$64,K$64,T$64:GOTO 870	Dimension variable/go to line 870.
50 REM ** RECEPTION VERIFICATION SUB-ROUTINE	Subroutine (Receive/verify).
60 SELECT PRINT 005:SELECT INPUT 219	Select CRT for output/TC for input.
70 R$=" "	Clear variable.
80 PRINT "WAITING FOR TELECOM DATA"	Display remark.
90 INPUT R$	Receive data via TC.
100 PRINT "SENDING ACK"	Display remark.
110 SELECT PRINT 21D	Select TC for output.
120 PRINT HEX(060D00)	Send ACK.
130 SELECT PRINT 005:RETURN	Select CRT/return.
190 REM **TRANSMISSION/VERIFICATION	Subroutine (transmit/verify).
200 SELECT PRINT 005	CRT for output.

(continue)

Example 6 (continued)

STATEMENTS	REMARKS
210 SELECT INPUT 219	TC for input.
220 R$=" "	Clear variable.
230 PRINT "TRANSMITTING TELECOM DATA"	Display remark.
240 SELECT PRINT 21D	TC for output.
250 PRINT T$	Send data (see line 990).
260 SELECT PRINT 005	CRT for output.
270 PRINT "WAITING ACKNOWLEDGE"	Display remark.
280 INPUT R$	Receive control byte via TC.
290 IF STR(R$,1,1) HEX(06)THEN 130	If ACK, go to 130.
300 IF STR(R$,1,1) <> HEX(15)THEN 340	If not NAK, go to 340.
310 PRINT "RE";	NAK received, display remark.
320 C1=C1+1	Increment error count.
330 GOTO 200	Loop to 200.
340 PRINT "ACK/NAK MISTAKEN SENDING NAK"	Display remark (see line 300).
350 SELECT PRINT 21D	TC for output.
360 PRINT HEX(15)	Send NAK.
370 GOTO 130	Branch/complete subroutine.
380 REM TRANSMIT ENQ	Subroutine (send ENQ).
390 SELECT PRINT 005	CRT for output.
400 PRINT "TRANSMITTING ENQ."	Display remark.
410 SELECT PRINT 21D	TC for output.
420 PRINT HEX(050D00)	Send ENQ.
430 RETURN	
440 REM RECEIVE ENQ	Subroutine (receive ENQ).
450 SELECT PRINT 005	CRT for output.
460 PRINT "RECEIVE ENQ."	Display remark.
470 R$=" "	Clear variable.
480 SELECT INPUT 219	TC for input.
490 INPUT R$	Request data via TC.
500 RETURN	
510REM **SENDER LOOP	Sending logic(see 1080).
520 T$=K$	Assign KB data to T$.
530 GOSUB 200	Go to transmit-subroutine.
540 GOSUB 450	Go to subroutine 450.
550 IF STR(R$,1,1) <> HEX(05)THEN 640	If not ENQ, go to 640.
560 C=C+1	Increment transmission count.
570 SELECT PRINT 005	CRT for output.
580 PRINTUSING 830,C,C1	Display C and C1 in format of 830.
590 PRINT HEX(01)	Cursor home.
592 IF C>1 THEN 600	If C>1, go to 600.
594 PRINT HEX(03)	Clear screen and cursor home.
600 PRINT "ORIGINATOR"	Display remark.
620 PRINT K$	Display K$ data.
630 GOTO 520	Return to 520.
640 IF STR(R$,1,1)=HEX(15)THEN 680	If NAK, go to 680.

(continue)

Example 6 (continued)

STATEMENTS	REMARKS
650 PRINT "MESSAGE=NAK"	Display remark.
660 T$=HEX(15)	Assign NAK to T$.
670 GOTO 530	Return to 530.
680 PRINT "NAK REC'D RETRANSMIT LAST DATA"	Display remark.
690 GOTO 520	Return to 520.
700 REM **RECEIVER LOOP	Receiving logic (see 1010).
710 SELECT PRINT 005	CRT for output.
720 PRINT HEX(01)	Cursor home.
722 IF E>1 THEN 730	If E > 1, go to 730.
724 PRINT HEX(03)	Clear screen and cursor home.
730 PRINT "RECEIVER"	Display remark.
740 GOSUB 60	Go to subroutine 60.
750 IF STR(R$,1,1)=HEX(15)THEN 810	If NAK, go to 810.
760 E=E+1	Increment reception counter.
770 SELECT PRINT 005	CRT for output.
780 PRINTUSING 840,E,E1	Display E & E1 values in format of 840.
790 GOSUB 390:REM SEND ENQ.	Go to subroutine 390.
800 GOTO 710	Return to 710.
810 E1=E1+1	Increase error count (see 750).
820 GOTO 770	Return to 770.
830 %TOTAL TRANSMISSIONS O.K.= #,###,### REPEATS=#,###,###	
840 %TOTAL RECEPTIONS O.K.= #,###,### REPEATS=#,###,###	
870 DEFFN'15	Key 15 special function.
880 SELECT PRINT 005	CRT for output.
890 PRINT HEX(03);	Clear screen.
892 PRINT "2200-2200 TELECOMMUNICATIONS EXERCISE"	
894 PRINT "TO OPERATE FIRST KEY RESET";	
896 PRINT "THEN KEY DESIRED SPECIAL FUNCTION"	Display information.
900 PRINT "KEY 15 DISPLAY OF S.F. KEYS"	
910 PRINT "KEY 16 START COMMUNICATIONS– TYPE IN MESSAGE"	
940 GOSUB 450 :REM WAIT FOR ENQ	Go to subroutine 450.
950 IF STR(R$,1,1)=HEX(05) THEN 990	If ENQ, go to 990 (see Note 3).
960 SELECT PRINT 21D	TC for output.
970 PRINT "KEY RESET THEN S.F. KEY 16 FOR ENQ."	Send instruction.
980 GOTO 940	Branch to 940.
990 T$="READY TO RECEIVE**TYPE MESSAGE"	Assign message to T$.
1000 GOSUB 200	Go to subroutine 200.
1010 GOTO 940	Branch to 940.

(continue)

Example 6 (continued)

STATEMENTS	REMARKS
1020 DEFFN'16	Key 16 special function.
1030 GOSUB 390:REM SEND ENQ	Go to subroutine 390.
1040 GOSUB 60:REM RECEIVE RESPONSE	Go to subroutine 60.
1050 SELECT INPUT 001	Keyboard for input.
1060 K$=" "	Clear variable.
1070 INPUT K$	Request input via KB.
1080 GOTO 520	Branch to 520.
1090 DEFFN'77	Parity error S.F. '77.
1100 SELECT PRINT 005	CRT for output.
1110 PRINT :PRINT :PRINT :PRINT	Space down four lines.
1120 PRINT "PARITY ERROR"	Display remark.
1130 FOR Z=1 TO 2000:NEXT Z	Create time-delay.
1140 PRINT "SENDING NAK"	Display remark.
1150 E1=E1+1	Increase error count.
1160 SELECT PRINT 21D	TC for output.
1170 PRINT HEX(15)	Send NAK.
1180 R$=" "	Clear variable.
1190 SELECT PRINT 005	CRT for output.
1200 PRINT "WAITING FOR RETRANSMISSION"	Display remark.
1210 RETURN	

Note 3. In general, the string function STR lets a programmer extract, examine, compare, or replace a specified portion of a particular alphanumeric string. The first element of the function is the string variable name. The second element denotes the position in the string of the starting character of the portion to be operated upon. The third element, if given, denotes the number of consecutive characters to be included in the designated portion. If no third element is given in the string function, all the remaining characters of the string beginning with the designated starting character are used. In line 950, the string function translates as follows: starting with the first character and including only that one character of R$, compare this character to HEX(05).

Example 7: Data transfer between the System 2200-2227 and a time-sharing computer terminal.

This conversational mode program sends a hook-up message GTWX (initiated by depressing Special Function Key 16) and then branches to a simple interactive loop. In this loop, if messages from the time-sharing computer terminal end with a question mark, the System 2200 program requests data via the keyboard for transmission to the computer. The operator can respond by typing in a message or by depressing Special Function Key 2 which transmits a canned message.

With simple modifications, the program listed here can be adapted to time-sharing services other than WYLBUR. A flowchart is presented in Figure 10.

OPERATING INSTRUCTIONS FOR EXAMPLE 7

1) Set appropriate switches on the Model 2227 board.
2) Load the program into the System 2200.
3) Dial the time-sharing service.
4) Wait for the "ready" signal on the Bell 103A modem (or equivalent).
5) On the System 2200 keyboard, touch the RUN key, followed by the EXECUTE key.

(continued)

Example 7 (continued)

PROGRAM LOGIC FOR EXAMPLE 7

STATEMENTS	REMARKS
10 REM TELECOM PKG #1 B$ IS CHECK CODE	
20 DIM B$1,A$64,T$1:GOTO 100	Dimension variables/go to setup.
.	
.	
22 DEFFN'14	Special Function Key 14.
24 SELECT INPUT 219:SELECT PRINT 005	TC for input/CRT for output.
26 A$=" " : T=1	Clear A$. Assign 1 to T.
28 PRINT "USING KEYIN FOR TELECOM INPUT"	Display remark.
30 KEYIN T$,32,32:GOTO 30	See Note 4.
32 HEXPRINT T$;:IF T$=HEX(11) THEN 30	See Note 5.
34 STR(A$,T,1)=T$:T=T+1	See Note 6.
36 IF T$<>HEX(0D) THEN 30	If T$ ≠ CR only, go to 30.
38 PRINT :PRINT A$	Display A$.
40 IF STR(A$,LEN(A$),1)<>B$ THEN 22	See Note 7.
42 SELECT PRINT 21D	TC for output.
44 GOTO 56	Go to 56.
48 DEFFN'3	Special Function Key 3.
50 SELECT INPUT 219	TC for input.
52 A$=" ":INPUT "T",A$	Clear A$. Request data via TC.
54 IF STR(A$,LEN(A$),1)<>B$ THEN 50	See Note 7.
56 DEFFN'1	Special Function Key 1.
58 SELECT INPUT 001	Keyboard for input.
60 A$=" ":INPUT "K",A$	Clear A$. Request data via KB.
62 IF A$<>"ATTN" THEN 70	If data not ATTN, go to 70.
64 GOSUB 310:GOTO 50	If ATTN, delay. Go to 50.
70 PRINT A$:GOTO 50	Send A$. Go to 50.
100 DEFFN'15	Special Function Key 15.
102 SELECT PRINT 005(64)	CRT for output (64 char/line).
104 PRINT HEX(03)	Clear screen.
110 PRINT "KEY 16 SIGN-ON TO TERMINAL"	
120 PRINT "KEY 0 ATTENTION"	
130 PRINT "KEY 1 TYPE IN MESSAGE FOR TRANSMISSION"	
140 PRINT "KEY 2 CANNED MESSAGE FOR TRANSMISSION"	
145 PRINT "KEY 3 ACCEPT DATA OVER TELECOM LINE"	Display information.
148 PRINT "KEY 14 USING KEYIN VERB FOR TELECOM INPUT"	
150 PRINT "KEY 15 PICTURE OF FUNCTION KEYS"	
160 STOP "KEY APPROPRIATE FUNCTION"	Stop/display instruction.

(continue)

Example 7 (continued)

STATEMENTS	REMARKS
200 DEFFN'16	Special Function Key 16.
202 SELECT PRINT 005	CRT for output.
204 PRINT "SIGN-ON PROCEDURE"	Display remark.
210 B$=HEX(3F)	Assign HEX(3F) to B$.
212 SELECT PRINT 21D	TC for output.
220 PRINT "G";:GOSUB 300	Send G, then delay.
230 PRINT "T";:GOSUB 300	Send T, then delay.
240 PRINT "W";:GOSUB 300	Send W, then delay.
260 PRINT "X";:GOSUB 300	Send X, then delay.
270 PRINT	Send CR effectively.
280 GOTO 50	Go to 50.
300 REM CREATE DELAY	Subroutine (delay).
310 FOR I=1 TO 100:NEXT I	Create delay.
320 RETURN	
400 DEFFN'0	Special Function Key 0.
410 REM SEND ESCAPE	
420 SELECT PRINT 21D	TC for output.
430 PRINT HEX(1B)	Send control character ESC.
440 RETURN	
500 DEFFN'2	Special Function Key 2.
510 SELECT PRINT 005	CRT for output.
520 PRINT "CANNED MESSAGE TRANSMISSION="	Display remark.
540 A$="TEST ORIGINATING FROM THE WANG SYSTEM 2200 IN TEWKSBURY, MASS."	
550 PRINT A$	Display message (assigned in line 540).
560 SELECT PRINT 21D	TC for output.
570 GOTO 70	Go to 70 (Send A$).

Note 4. Here KEYIN is used to request data via the TC for assignment to the string variable T$. If no character is ready to come in from the TC buffer, program execution continues at the next statement (i.e., GOTO 30). If a character is ready, execution continues at the first line number of the KEYIN statement (i.e., 32). See the System 2200 A/B Reference Manual for an explanation of the purpose of the second line number, if different.

Note 5a. The NEXPRINT statement prints on the current output device (in this case, the CRT) the value of the alpha variable in hexadecimal notation.

Note 5b. If T$ = HEX(11), execution returns to line 30. If not equal, execution continues at line 34.

Note 6. One character of A$, the one whose position in the string is numerically equal to the current value of T, is replaced by the character currently assigned to T$ (see Reference Manual). Afterwards, according to the next statement on the line, the value of T is incremented.

Note 7. One character of A$ (the one whose position in the string is numerically equal to LEN(A$); i.e., the length of A$) is compared to the character currently assigned to B$. In line 210, which is executed before this point is reached, B$ is assigned the value HEX(3F) which is the code for a question mark. Therefore, at this point, the last character of the string variable A$ (i.e., the last character of the data received via the TC and assigned to A$) is compared to HEX (3F). Now, if the received data does not end with a question mark, program execution returns to the designated line number (either 50 or 22) to request additional data via the TC. On the other hand, if the message from the computer terminal does end with a question mark, execution continues at line 56. There, keyboard data is requested for transmission to the time-sharing computer.

Example 8: Teletype terminal simulation.

With the program below, the System 2200-B Model 2227 becomes a Teletype-like terminal.

```
10 DIM T$1,K$1:GOTO 70:REM SYSTEM 2200-B/2227 TELETYPE-LIKE TERMINAL
20 SELECT INPUT 001:KEYIN K$,30,60:SELECT INPUT 219:KEYIN T$,40,50:GOTO 20
30 SELECT PRINT 005:PRINT K$;:SELECT PRINT 21D:PRINT K$;:GOTO 20
40 SELECT PRINT 005:PRINT T$;:GOTO 20
50 SELECT PRINT 005:HEXPRINT T$;:PRINT "PARITY/FRAMING":GOTO 20
60 SELECT PRINT 005:IF K$=HEX(00)THEN 62:GOTO 30
62 PRINT "ESCAPE":SELECT PRINT 21D:PRINT HEX(1B):GOTO 20
70 SELECT PRINT 005:PRINT HEX(03),"SYSTEM 2200-B/2227 AS TELETYPE-LIKE TERMINAL"
100 PRINT "S.F. KEY 0 (ESCAPE) SENDS HEX(1B)"
110 PRINT "S.F. KEYS 1-31 SEND EQUIV. HEX CODE (USUALLY NON-PRINTABLE)."
150 PRINT
160 PRINT " **NON-DISPLAY OF TYPED CHARACTER SIGNIFIES:"
170 PRINT " 1). T.C.LINE WAS NOT AVAILABLE FOR SENDING OF PRIOR CHARACTER"
180 PRINT
190 PRINT " **IF T.C. LINE DROPS PROBABLE DISPLAY, THE CONDITION IS '4D=PARITY/F
  RAMING' "
200 PRINT
205 PRINT "USING TELEPHONE AT BELL 103-A3 MODEM OR EQUIV. FOR CALL"
210 PRINT "TO SIGN ON TO HOST COMPUTER--"
220 PRINT "CALL AREA-EXCHANGE-NUMBER"
225 PRINT "AT TONE, DEPRESS DATA BUTTON"
230 PRINT "THEN TYPE-IN/RECEIVE REQUIRED HANDSHAKE PROTOCOL."
300 GOTO 20
```

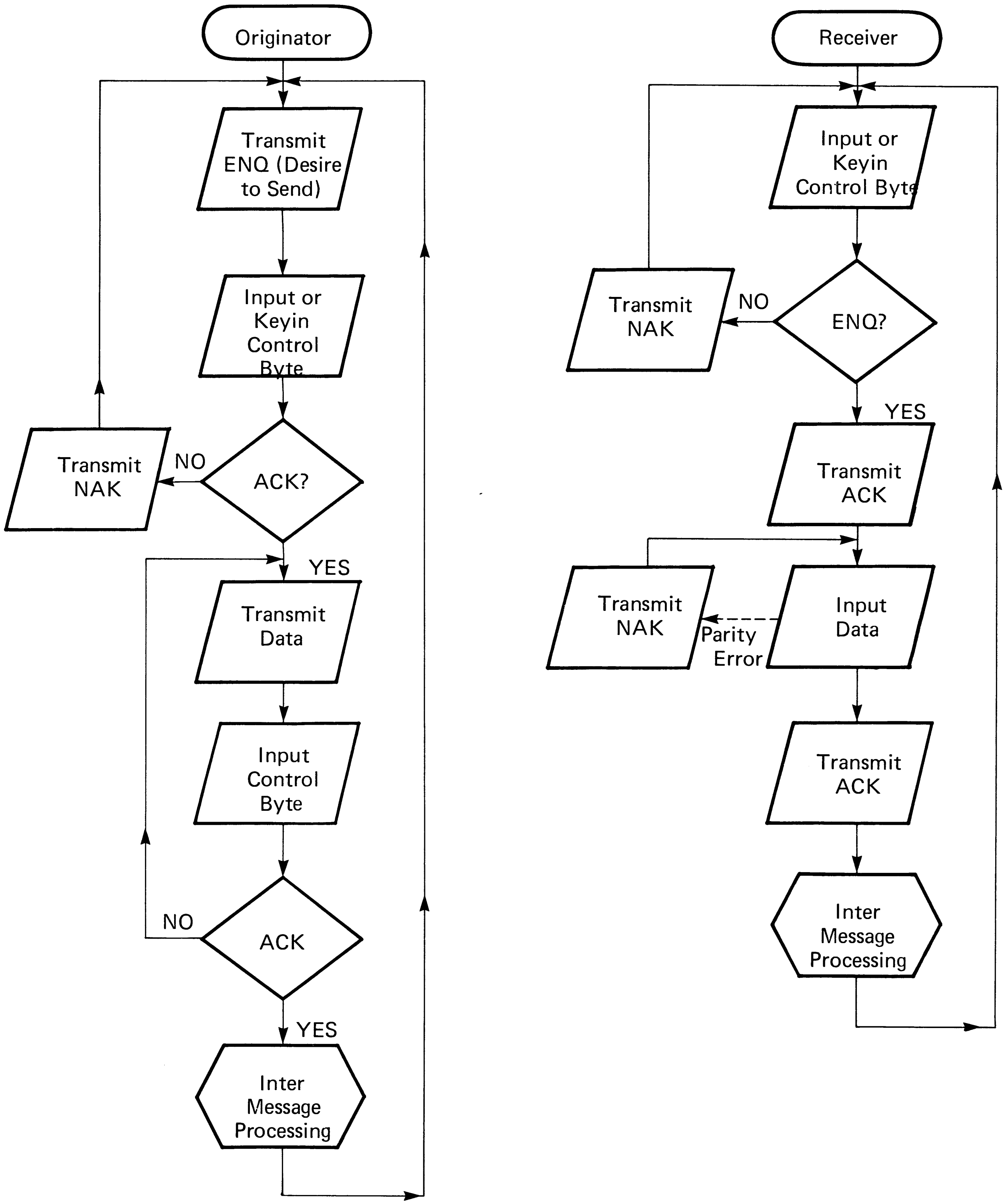


Figure 5. Flowchart for Overall Data Transmission Control Using Control Bytes ENQ, ACK, and NAK

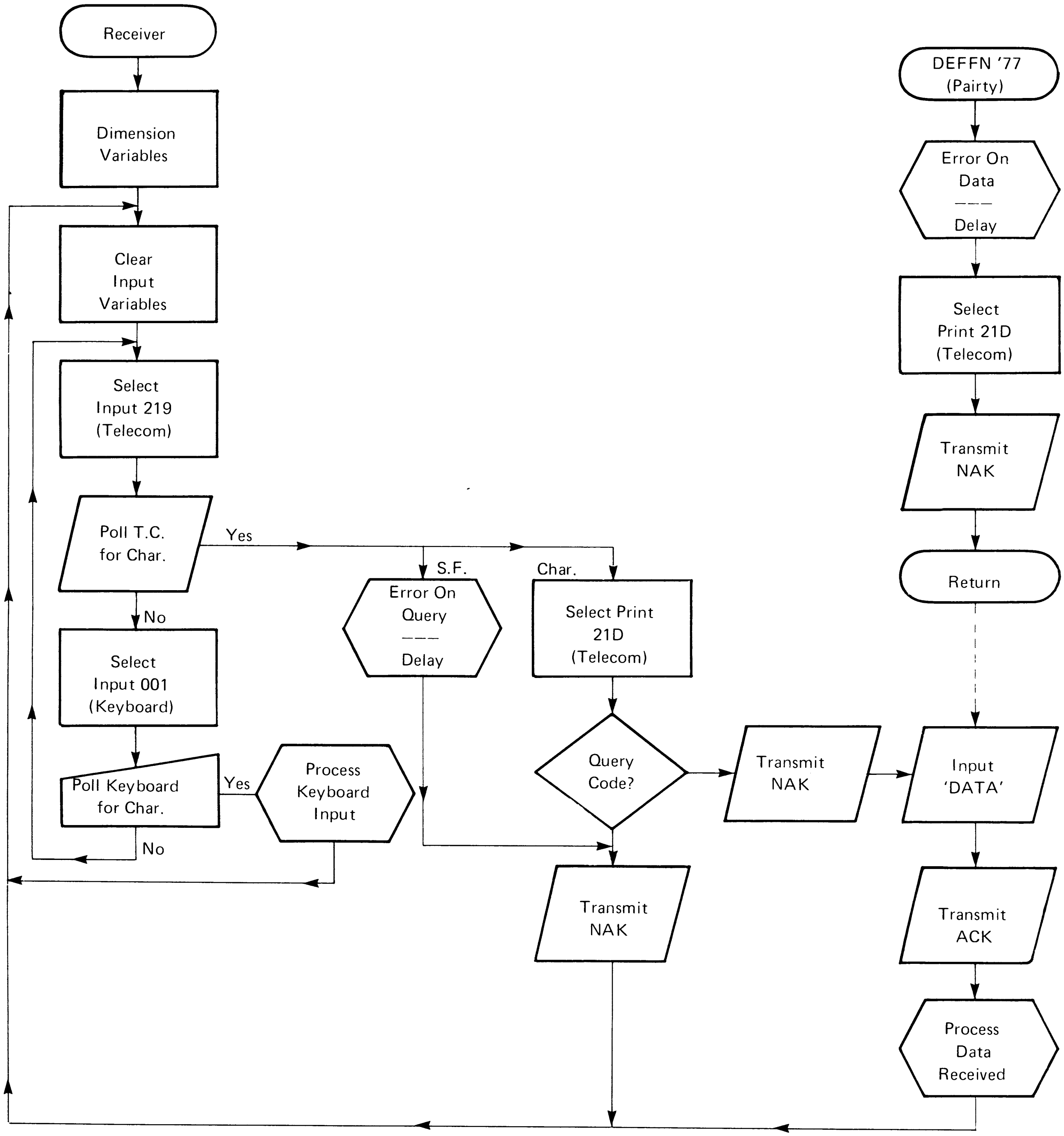


Figure 6. Flowchart for Example 3

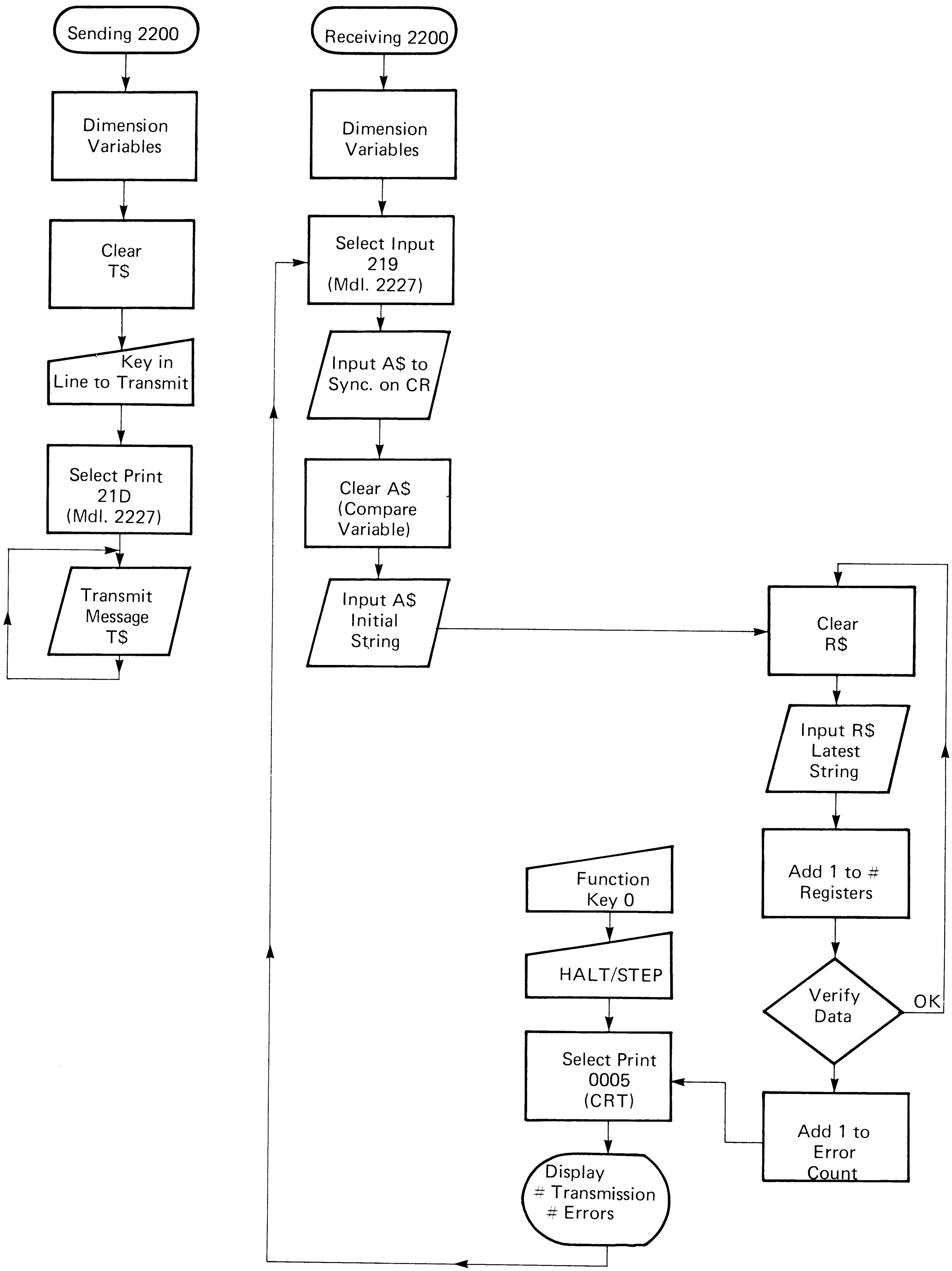


Figure 7. Flowchart for Example 4

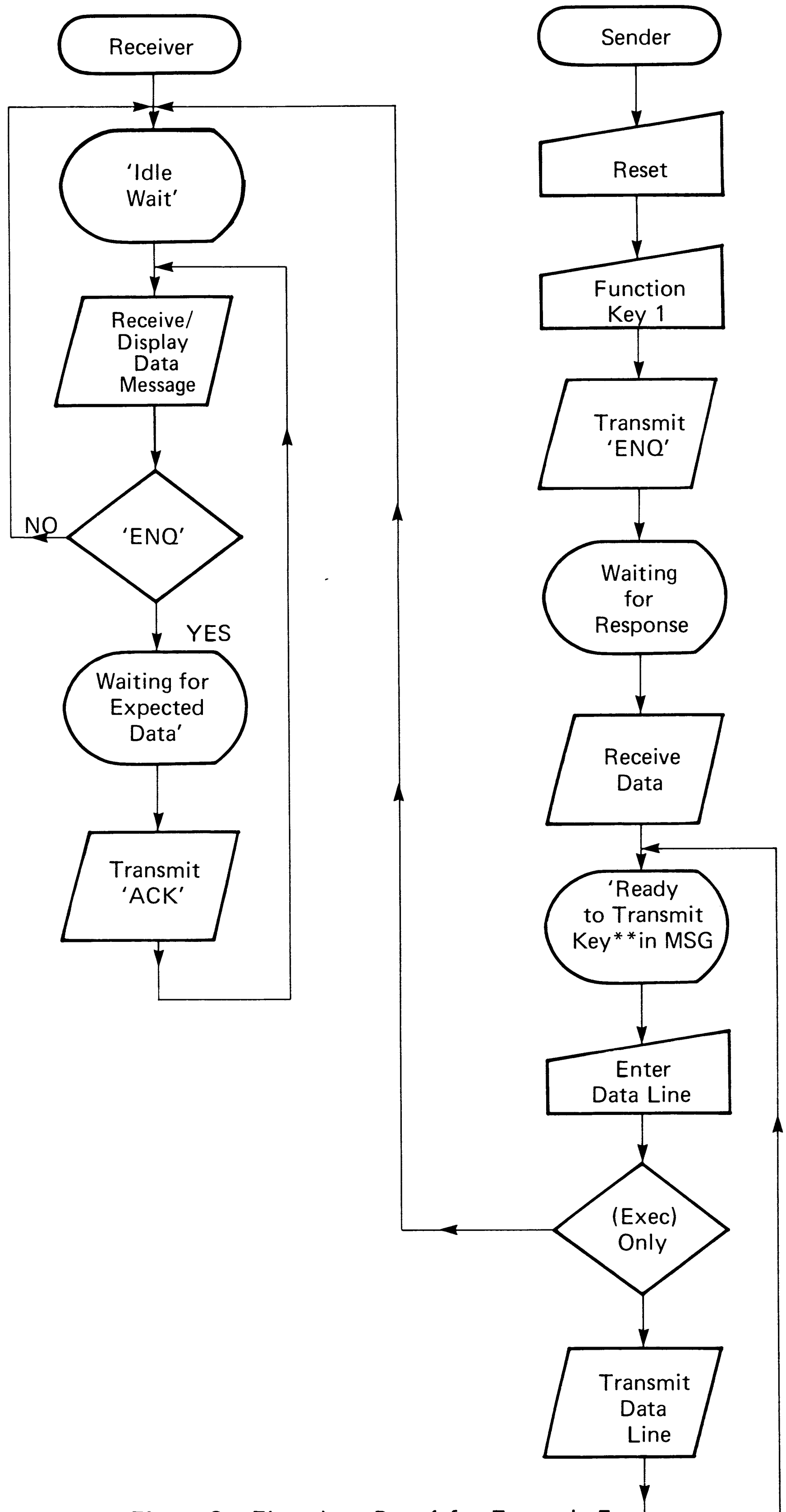


Figure 8a. Flowchart Part 1 for Example 5

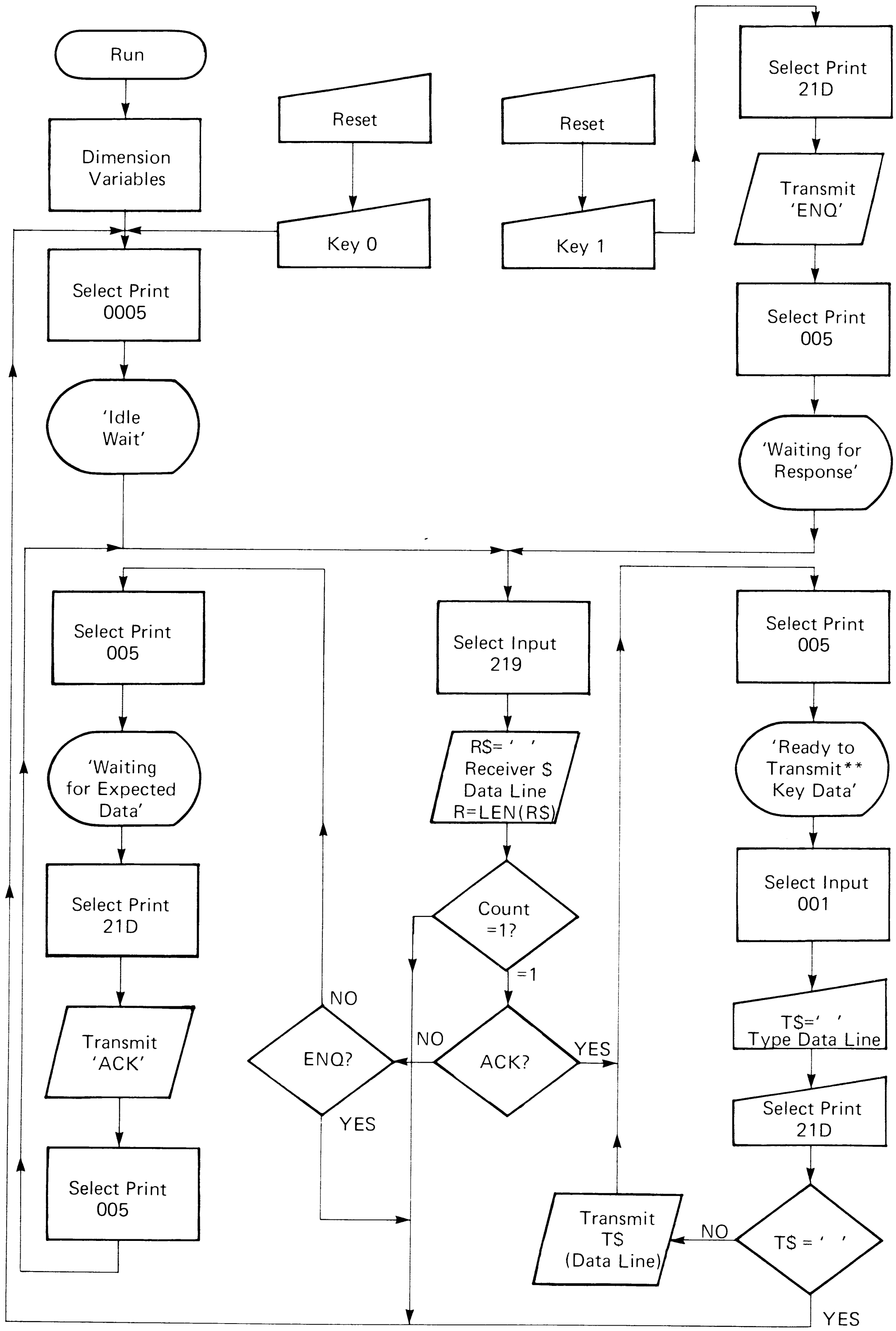


Fig. 8b Flowchart Part 2 for Example 5

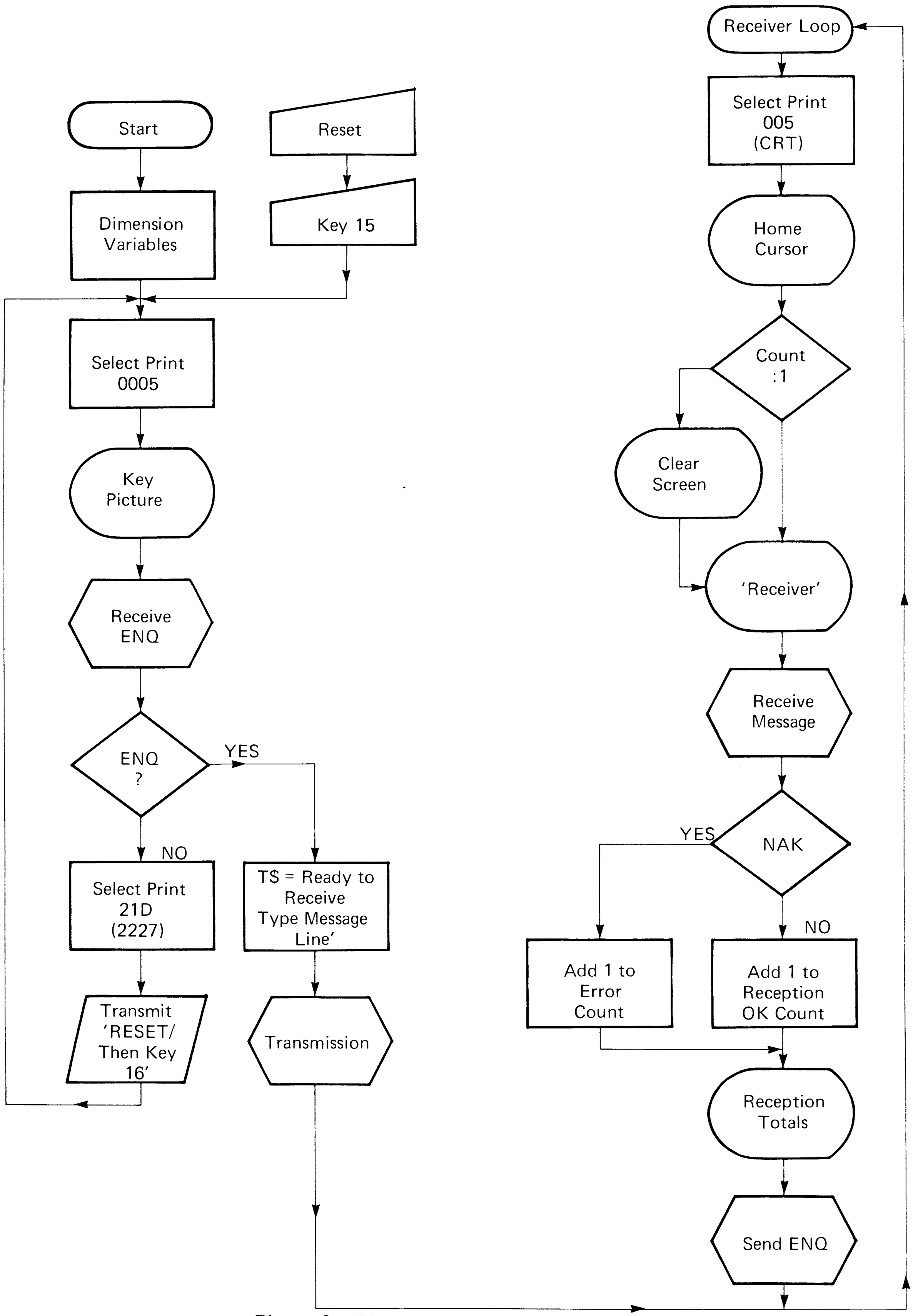


Figure 9a. Flowchart Part 1 for Example 6

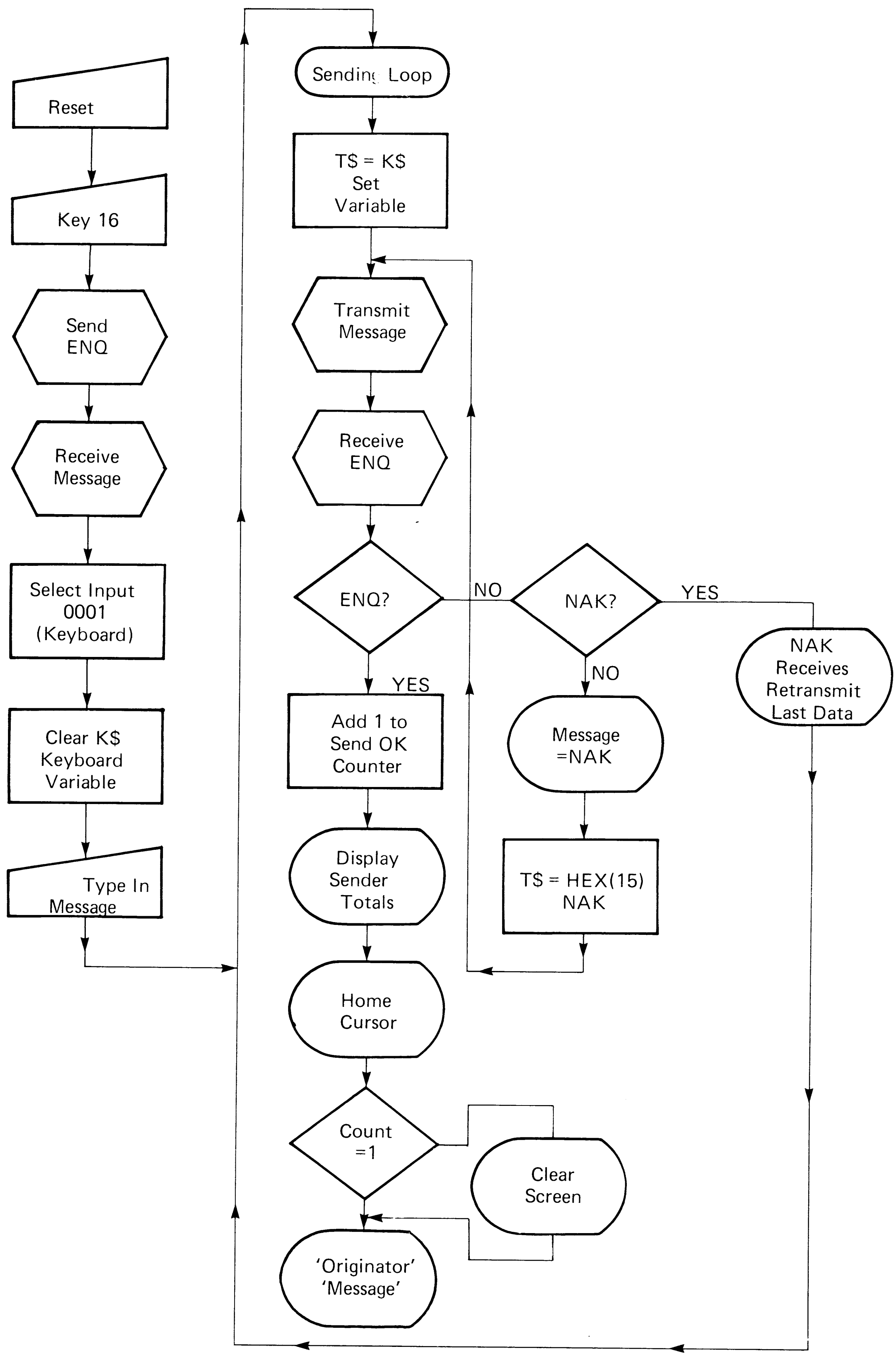


Figure 9b. Flowchart Part 2 for Example 6

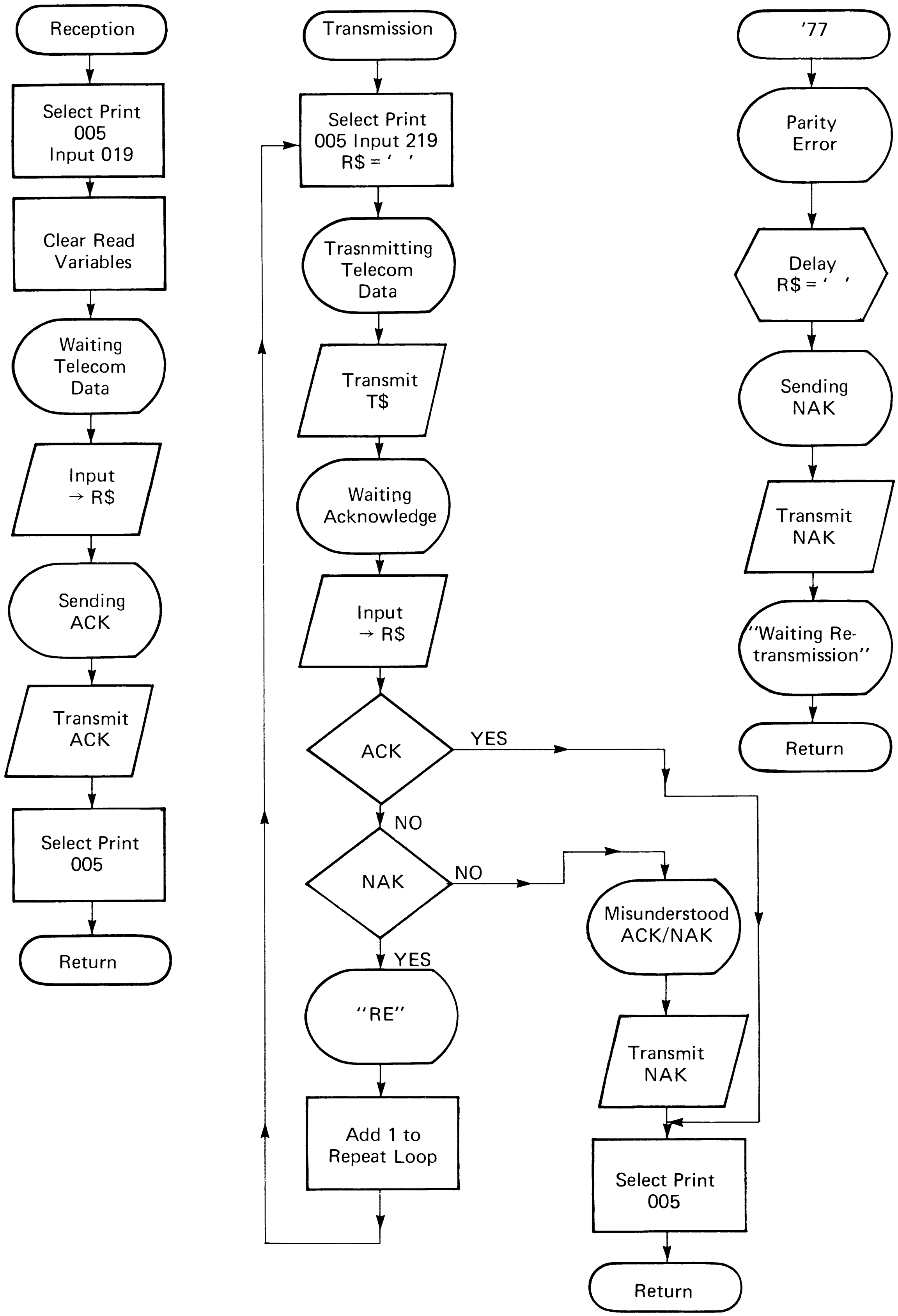


Figure 9c. Flowchart Part 3 for Example 6

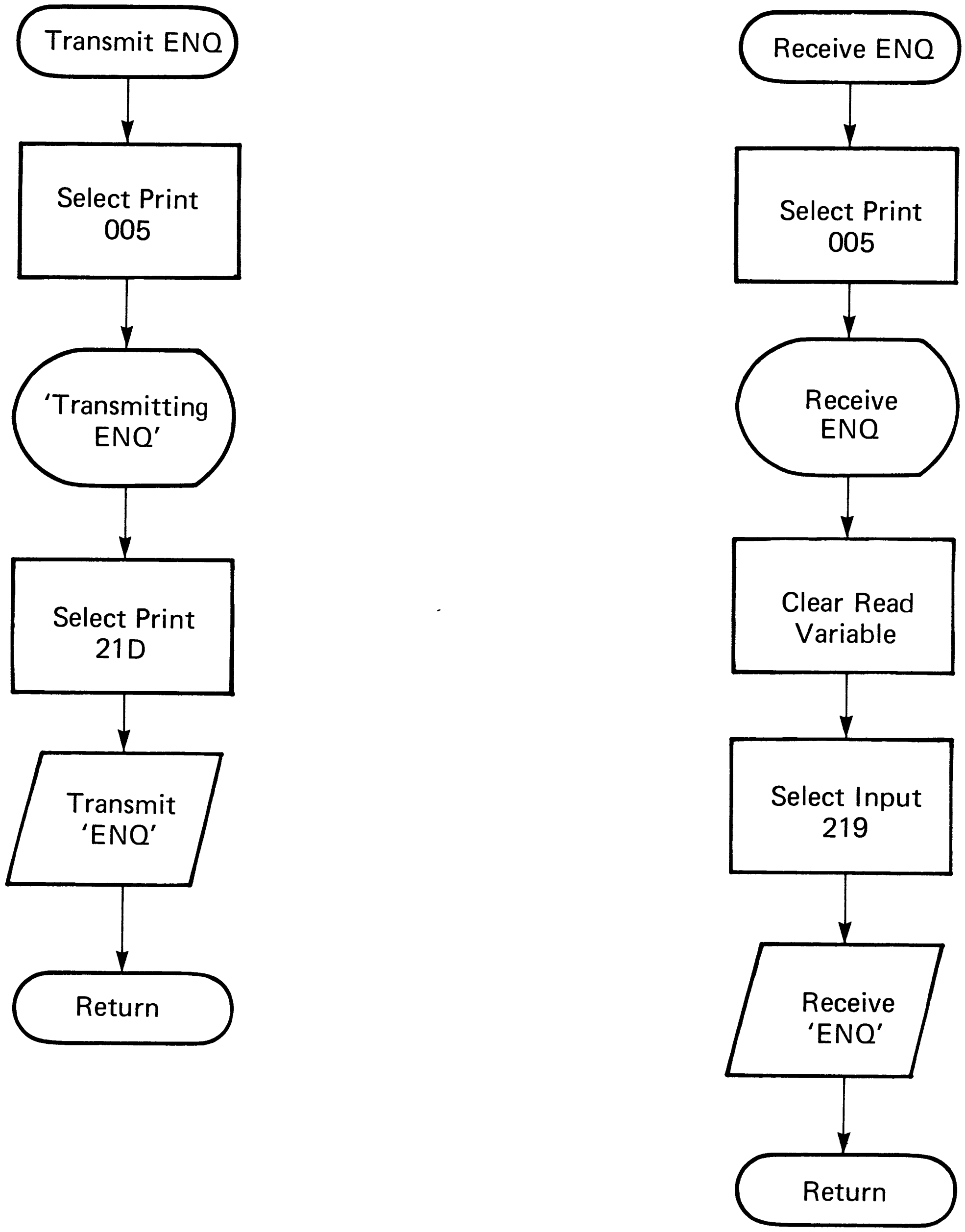


Fig. 9d Flowchart Part 4 for Example 6

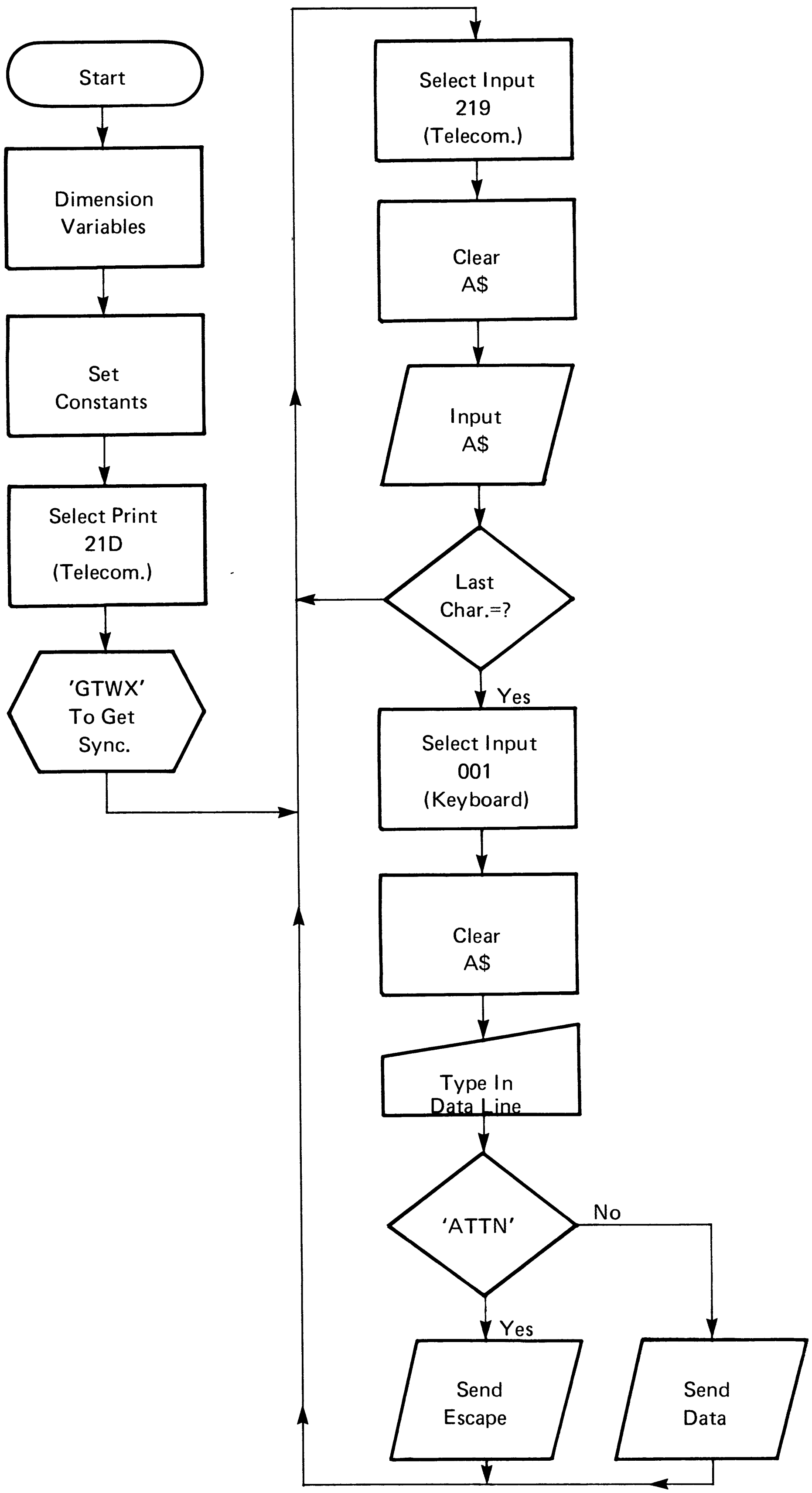


Figure 10. Flowchart for Example 7

APPENDIX A — Device Addresses for System 2200 Peripherals

I/O DEVICE CATEGORIES	DEVICE ADDRESSES*
KEYBOARDS (2215, 2222)	001, 002, 003, 004
CRT (2216)	005, 006, 007, 008
TAPE CASSETTE DRIVES (2217, 2218)	10A, 10B, 10C, 10D, 10E, 10F
LINE PRINTERS (2221, 2231) (2261)	215, 216
OUTPUT WRITER (2201)	211, 212
THERMAL PRINTER (2241)	215, 216 (Revised 3/7/74)
PLOTTERS (2202, 2212, 2232)	413, 414
DISK DRIVES (2230-1, -2, -3) (2240-1, -2) (2242, 2243)	310, 320, 330
MARK SENSE (MANUAL) CARD READER (2214)	517
HOPPER FEED CARD READERS (2234, 2244)	629, (029)
PUNCHED TAPE READER (2203)	618
TELETYPE (2207)	019, 01A, 01B INPUT 01D, 01E, 01F OUTPUT
TELETYPE TAPE UNITS	41D, 41E, 41F
TELECOMMUNICATIONS (2227)	219, 21A, 21B INPUT 21D, 21E, 21F OUTPUT
PARALLEL I/O INTERFACE (2250)	23A, 23C, 23E INPUT 23B, 23D, 23F OUTPUT (Revised 3/7/74)
BCD INPUT INTERFACE (2252)	25A, 25B, 25C, 25D, 25E, 25F

*In some cases, more than one device address is listed for each device category. Unless otherwise noted, each peripheral device is assigned a unique address; device addresses are assigned sequentially. Therefore, if a System 2200 has only one device of a particular category (such as a tape drive), it is set up with the first device address listed (10A in the case of the tape drive). If it has two drives, they are set up with device addresses 10A and 10B. Each device address is printed on the interface card which controls that device.

APPENDIX B – Device Selection for System 2200 I/O Operations

Five address codes are designated as Primary Device Addresses in the System 2200 (see Table B-1). Devices with these default addresses are "selected" automatically whenever the system is Master Initialized (i.e., power is turned off and then on again).

TABLE B-1. PRIMARY DEVICE ADDRESSES FOR THE SYSTEM 2200

DEFAULT ADDRESS	I/O DEVICE CATEGORY	MODEL NUMBER	SELECT Statement I/O Class Parameter
001	Keyboards	2215 or 2222	CI (console input) INPUT
005	CRT	2216	CO (console output) PRINT LIST
10A	Tape drives	2217 or 2218	TAPE
310	Disks	2230 or 2240	DISK
413	Plotters	2202, 2212, or 2232	PLOT

Two or more model numbers in Table B-1 correspond to one default address in several cases. These models should be considered as "candidates" for the primary device address. Since device addresses within a configuration are unique, only one keyboard or one tape drive can be the primary device for a particular device category.

Input/output operations for the System 2200 are grouped in eight classes designated by the I/O "class parameters": CI, INPUT, CO, PRINT, LIST, TAPE, DISK, and PLOT. Chart B-1 identifies the I/O operations in each class, and Table B-1 gives the default device address for these classes of I/O operations.

To change the device address for particular I/O operations, use the SELECT statement. This statement has other uses; but, in every case, it requires a "select parameter" (see the Reference Manual). When selecting I/O peripheral devices, keep in mind that the select-statement-parameter is two-fold. It consists of an I/O class parameter and a three-digit device address.

For example, the statement

SELECT PRINT 215

instructs the system to access the Line Printer with address 215 for all subsequent output from Program Mode PRINT and PRINTUSING statements.

To reselect the CRT for this output, use the statement

SELECT PRINT 005

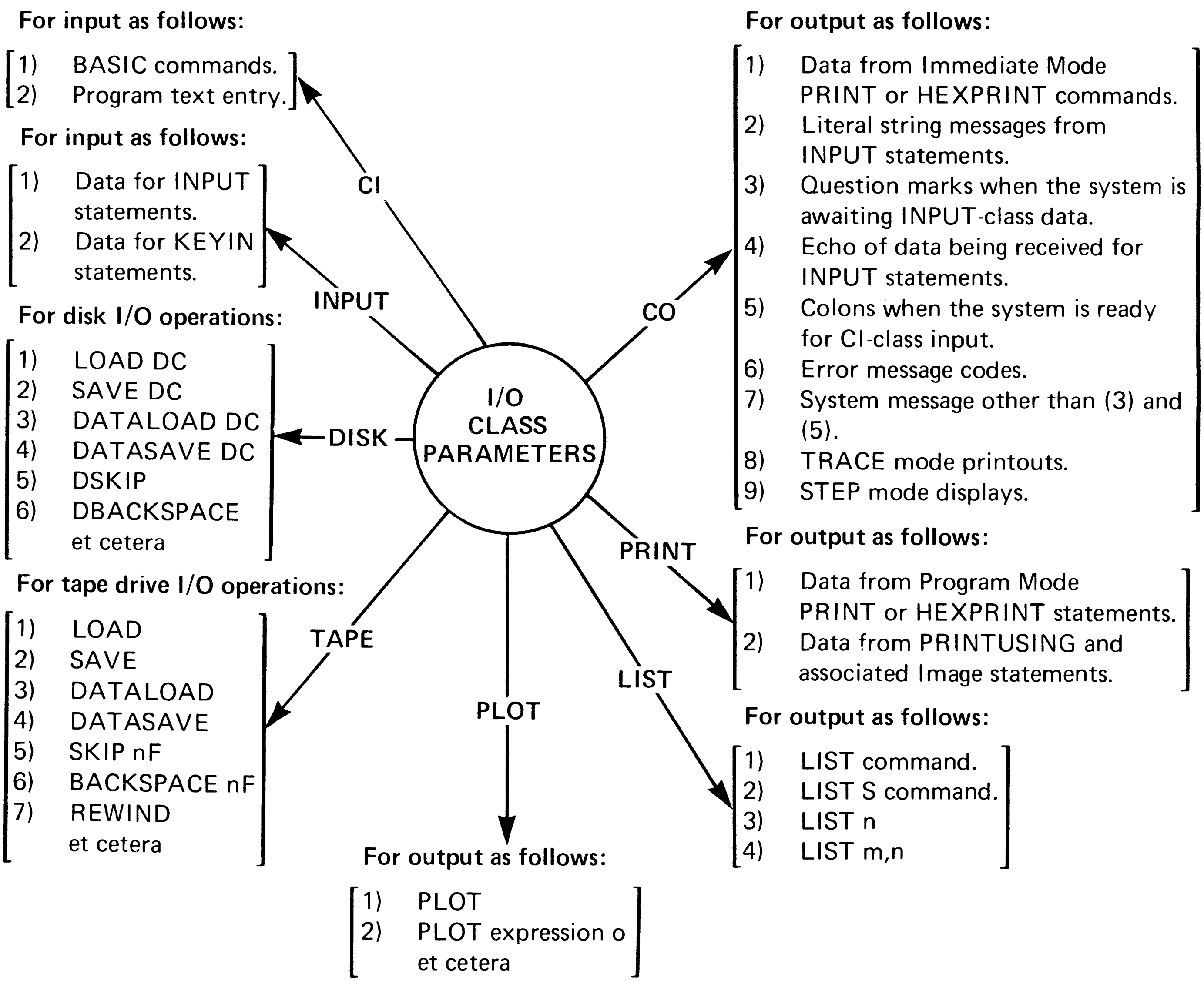
or Master Initialize the system if the memory can be cleared at this point.

Two or more devices performing different functions can be selected in one statement by using commas as device separators. For example,

SELECT LIST 215, PRINT 211, TAPE 10C

The SELECT verb "assigns" the specified device address to the specified I/O class parameter. Using a select-statement is analogous to setting an I/O-class rotor switch which includes the device-address-options for that class. All subsequent I/O operations in the I/O class are "switched" to the designated device until the system encounters another select-statement for that I/O class.

Chart B-1 I/O Operations and Class Parameters for the System 2200



NOTE:

This chart identifies the input/output operations in the eight System 2200 I/O classes. A class parameter (CI, INPUT, CO, PRINT, LIST, TAPE, DISK, or PLOT) with a device address is used in SELECT statements to change or reselect a device for I/O operations. The default address for CI and INPUT operations is 001 (the keyboard). The default address for CO, PRINT, and LIST operations is 005 (the CRT).

APPENDIX C – ASCII CONTROL AND GRAPHIC CHARACTERS IN HEXADECIMAL AND BINARY NOTATION

FORMATS:

HEXADECIMAL CODES: HEX (a_1 a_2)

7-BIT BINARY CODES: (b_7 b_6 b_5 b_4 b_3 b_2 b_1)

THREE BITS (HIGH ORDER) b_7 b_6 b_5 — 7-BIT BINARY NOTATION		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
FIRST HEX DIGIT (HIGH ORDER) HEX a_1		0	1	2	3	4	5	6	7
SECOND HEX DIGIT (LOW ORDER) a_2	**FOUR BITS (LOW ORDER)** b_4 b_3 b_2 b_1	**CHARACTERS**							
0	0 0 0 0	NUL	DLE	SP	0	@	P	`	p
1	0 0 0 1	SOH	DC1	!	1	A	Q	a	q
2	0 0 1 0	STX	DC2	"	2	B	R	b	r
3	0 0 1 1	ETX	DC3	#	3	C	S	c	s
4	0 1 0 0	EOT	DC4	$	4	D	T	d	t
5	0 1 0 1	ENQ	NAK	%	5	E	U	e	u
6	0 1 1 0	ACK	SYN	¢	6	F	V	f	v
7	0 1 1 1	BEL	ETB	' (apos.)	7	G	W	g	w
8	1 0 0 0	BS	CAN	(	8	H	X	h	x
9	1 0 0 1	HT	EM	)	9	I	Y	i	y
A	1 0 1 0	LF	SUB	*	:	J	Z	j	z
B	1 0 1 1	VT	ESC	+	;	K	[	k	{
C	1 1 0 0	FF	FS	, (comma)	<	L	\	l	|
D	1 1 0 1	CR	GS	— (dash)	=	M	]	m	}
E	1 1 1 0	SO	RS	. (period)	>	N	↑	n	~
F	1 1 1 1	SI	US	/	?	O	_ (under-line)	o	DEL

LEGEND FOR ASCII CONTROL CHARACTERS

NUL	Null	DLE	Data Link Escape
SOH	Start of Heading	DC1	Device Control 1
STX	Start of Text	DC2	Device Control 2
ETX	End of Text	DC3	Device Control 3
EOT	End of Transmission	DC4	Device Control 4
ENQ	Enquiry	NAK	Negative Acknowledge
ACK	Acknowledge	SYN	Synchronous Idle
BEL	Bell (audible or attention signal)	ETB	End of Transmission Block
BS	Backspace	CAN	Cancel
HT	Horizontal Tabulation (punched card skip)	EM	End of Medium
LF	Line Feed	SUB	Substitute
VT	Vertical Tabulation	ESC	Escape
FF	Form Feed	FS	File Separator
CR	Carriage Return	GS	Group Separator
SO	Shift Out	RS	Record Separator
SI	Shift In	US	Unit Separator
		DEL	Delete

APPENDIX D – INPUT STATEMENT PROCESSING PROCEDURE

When a program is run and the System 2200 encounters an INPUT statement, the processing procedure includes the following events:

1. If the INPUT statement contains a literal string message, the message is displayed or printed on the last selected CO (console output) device, or the CRT (the default CO device).
2. The last selected device for INPUT-class operations is enabled (i.e., the keyboard, the Model 2227 TC board, the Model 2252, the Model 2250, et cetera).
3. A question mark, followed by a space, is displayed or printed on the CO device (after the literal string message, if any) to indicate the system is awaiting data.
4. When data is received via the input device, an echo of each character is sent to the CO device; the echo begins on the line containing the question mark and consists of the printable symbols available on the CO device in use.
5. Meanwhile, the characters received via the input device are held in a buffer.
6. If no carriage-return-character is received before 192 characters reach the buffer, the processing procedure is interrupted as follows:
   a) an automatic error message code (↑ERR 45) is displayed on the CO device,
   b) the data in the buffer is ignored (not transferred to memory), and
   c) steps (2) through (5) are repeated until a carriage-return-character reaches the buffer.
7. The carriage-return-character (which has no printable symbol but is echoed as a carriage-return/line feed) is interpreted by the system as the signal to begin processing the data in the buffer. Meanwhile, no additional characters are read into the CPU buffer.
8. When processing of buffered data begins, commas and double-quotation-marks are given special significance depending upon their relative positions in the data stream with respect to each other. Usually, a comma is interpreted as a separator of data to be assigned to one variable in a multiargument INPUT statement. However, a comma is considered part of the data to be assigned to a variable if the comma is embedded in a set of characters embedded, in turn, in a properly placed pair of double-quotation-mark characters. To be properly placed, as the beginning and the end of data to be assigned to an alphanumeric variable, the first double-quotation mark character must be the first nonblank character in the data stream (or in the portion of the stream following a comma which has been recognized as a data separator) and the second double-quotation-mark character must be followed by a comma or the carriage-return. See examples D1 through D5.
9. As soon as the first data separator in the buffered data is encountered, the system checks the first variable named in the INPUT statement. The transfer procedure depends upon whether the variable is numeric or alphanumeric.

   Legal characters for a numeric variable consist of an algebraic sign (plus or minus), a decimal point, up to 13 decimal digits (0 through 9), and also the letter E (used in floating point formats) if followed by a plus, minus, or no sign and one or two digits. If an illegal character is encountered, transfer of the buffered data is not made; an error message is issued; and execution returns to Step 2.

   For an alphanumeric variable, all ASCII control and graphic characters are legal. However, the maximum storage space allotted to each alphanumeric variable is determined by the default dimension (16 characters) unless a DIM statement specifies a different dimension (any number from 1 to the limit 64). If the data to be transferred consists of more characters than the allowable maximum, only those characters which fill the storage location are transferred; the additional characters up to the separator are ignored. When the data to be transferred contains fewer characters than the allowable maximum, all the characters up to the separator are transferred to the storage location; and one of two possibilities occurs. If the alphanumeric variable is named explicitly in the argument list, trailing space characters; i.e., HEX(20), are added until the storage location is filled. If the alphanumeric variable is named implicitly by a STR function in the argument list, no HEX(20) characters are transferred. Also, no data is transferred to an alphanumeric variable location if one of the following conditions exists:

a) The first nonblank character is a double-quotation-mark, but no second double-quotation-mark character precedes the carriage-return character.

b) The first nonblank character is a double-quotation-mark, and a second double-quotation-mark character is present; but no data separator (either a comma or the carriage-return) immediately follows the second double-quotation-mark character.

When data is not transferred, an error message is issued; and the INPUT statement processing procedure returns to Step 2.

10. Once data is transferred to the memory location reserved for the first variable specified in the INPUT statement, the transfer process continues for each subsequent group of characters identified by a data-separator-comma in the data stream and each subsequent variable in the INPUT statement. As soon as an error condition arises, an error message is issued; any remaining data in the buffer is ignored. However, data already transferred to storage locations for some of the variables in the INPUT statement is not affected. Execution returns to Step 2. New data read into the buffer is processed after another carriage-return character is received. Processing and assignment of new data begins with the variable for which the last error condition existed. (Processing does not return to the first variable in the INPUT statement.)

11. If the system encounters a carriage-return character (rather than a comma) in the buffered data before all the variables in an INPUT statement are assigned data, execution returns to Step 2. New data read into the buffer is processed beginning with the variable following the last one to which data was assigned.

12. Any HEX(00) and HEX(7F) codes in the data stream are ignored.

13. The HEX(08) code is interpreted as a BACKSPACE character which erases the previous character in the data stream before transferring the data to memory.

14. The HEX(5C) code is interpreted as a LINE ERASE instruction.

15. The HEX(2C) code for a comma is interpreted as a data-separator unless embedded in alphanumeric data enclosed in quotation marks.

16. The HEX(0D) code for a carriage-return is interpreted as a data-separator always. Furthermore, the HEX(0D) character is required before the system initiates the micro-code processing procedure for an INPUT statement.

To illustrate how commas and double-quotation-marks are interpreted by the System 2200, four brief examples are included here. Identical input is used for each example.

No DIM statement or SELECT statement is included in any of the two-line programs in Examples D1 through D4. If the programs are entered and run after the system has been Master Initialized, the keyboard is selected automatically as the input device; and the dimension for each alphanumeric variable is 16 characters (the default value).

The PRINT statement is needed in each example to find out what characters are stored in memory for each variable named in the INPUT statement. However, the PRINT statement presents the data in a zoned or a packed format depending upon whether a comma or a semicolon follows each variable.

Thus, the PRINT statements in Examples D1 through D4 illustrate how partial control can be exercised over multivariable data streams transmitted via a Model 2227 Telecommunications Controller. With semicolons in the PRINT statement, trailing spaces are omitted. However, the examples also show the requirement to supply data-separator-commas and double-quotation-marks needed to embed a comma in a literal string.

PRINTUSING and image statements provide one method of supplying data-separator-commas and quotation marks in a data stream. However, both types of characters can be transmitted via PRINT statements also, if their proper HEX codes are used. See Example D5.

Examples D1 through D5 are presented here as hardcopy of the CRT display when the simple programs are entered, run, and given input especially prepared to demonstrate the System 2200 INPUT-statement-processing-procedure. To duplicate any example, first enter the program text (i.e., Lines 10 and 20). Then enter the command RUN by touching the RUN key followed by the EXECUTE key which produces no echo on the screen. After the question mark appears to indicate the system is awaiting data, enter the input shown in the fourth line of the hardcopy. The output shown in the fifth line appears automatically.

Example D1

```
:10 INPUT A$,B$
:20 PRINT A$,B$
:RUN
? "AB,CD",M"PQ,RS"
AB,CD           M"PQ
```

Remarks:

The first comma in the input data stream is considered by the system to be part of the literal string intended for storage in the variable A$ memory location. The second comma is recognized as a data separator and is not stored. The first double-quotation-mark of the next pair of double-quote characters is not the first nonblank character after the data-separator-comma preceding the letter M. Therefore, the expression "PQ,RS" is not recognized as a literal string. Instead, M"PQ is assigned to B$ because the comma after the letter Q is recognized as another data separator. Since the INPUT statement requests data for only two variables the remaining data RS" is ignored. Output appears in zoned format because a comma is used after A$ in the PRINT statement.

Example D2

```
:10 INPUT A$,B$,C$
:20 PRINT A$,B$,C$
:RUN
? "AB,CD",M"PQ,RS"
AB,CD           M"PQ            RS"
```

Remarks:

By adding a third variable to the INPUT (and PRINT) statements in Example D2, the input data previously ignored in Example D2 is stored in C$.

Example D3

```
:10 INPUT A$,B$
:20 PRINT A$;B$
:RUN
? "AB,CD",M"PQ,RS"
AB,CDM"PQ
```

Remarks:

By using a semicolon in Line 20, instead of the comma used in Example D1, the output appears in packed rather than zoned format.

Example D4

```
:10 INPUT A$,B$,C$
:20 PRINT A$;B$;C$
:RUN
? "AB,CD",M"PQ,RS"
AB,CDM"PQRS"
```

Remarks:

By using semicolons in Line 20, instead of the commas used in Example D2, the output appears in packed format.

Example D5

```
:10 INPUT A$,B$,C
:20 PRINT HEX(22);A$;HEX(22);HEX(2C);B$;HEX(2C);C
:RUN
? "AB,CD",XYZ,125.3
"AB,CD",XYZ, 125.3
```

Remarks:

By using the code HEX(22) for each double-quotation-mark and HEX(2C) for each data-separator-comma, the PRINT statement in Line 20 structures a data stream which duplicates the input stream (except for the leading space allotted for a sign when the numeric input is assigned to the numeric variable C). A trailing space is allotted also when the numeric value is printed.

APPENDIX E – MODEL 2227 SPECIFICATIONS

Physical Dimensions

Length . . . . 14 in. (35 cm)
Depth . . . . 6 in. (15 cm)
Width . . . . 1 in. (2.5 cm)

Power Requirements

Supplied by the System 2200 CPU.

Electrical Connection

RS-232-C compatible.
A 25-pin RS-232-C female connector is mounted on the Model 2227 face plate.

Cable

A 12-foot (3.6 meter) cable with 25-pin RS-232-C male connectors on each end is provided with the board. (See Appendix F for other cables available from WANG and configurations possible using the Model 2227-N Null Modem.)

Transmission

Asynchronous.
Switch Selectable Rates: 110, 150, 300, 600, or 1200 baud.

Bit Format

1 Start Bit.
5, 6, 7, or 8 Data Bits – Switch Selectable.
1 or 2 Stop Bits – Switch Selectable.
Even, Odd, or No Parity – Switch Selectable.

Communication Mode

Full Duplex – electronically.
The Model 2227 controller board is designed for full duplex transmission; however, under System 2200 program control,input/output operations can be interspersed but not simultaneous.

APPENDIX F – MODEL 2227-N NULL MODEM AND SYSTEM 2200/2227/2227-N CONFIGURATIONS

The WANG Model 2227-N Null Modem is a double connector module which switches certain pin assignments in the RS-232-C connector mounted on the Model 2227 Telecommunications Controller. The dimensions of the Model 2227-N module are:

length = 3 inches (7.6 cm)
width = 2 inches (5.1 cm)
height = 1 inch (2.5 cm)

The 12-foot (3.6 meter) cable supplied with the Model 2227 Telecommunications Controller board is equipped with two RS-232-C male connectors, one on each end of the cable. For data transfer to or from a System 2200/2227 *via telephone lines,* one end of the cable is plugged into the female connector on the Model 2227 board; the other end of the cable is plugged into the connector on either a Bell 100 Series or a Bell 200 Series modem or a modem equivalent to one of these modems. On the other hand, for data transfer to or from a System 2200/2227 *via direct connection lines,* one end of the standard 12-foot cable is plugged into the Model 2227 board; the other end of the standard cable is plugged into a Model 2227-N Null Modem (or equivalent modem) or into a Model 2227 extension cable.

Model 2227 extension cables are available in two lengths; a 25-foot (7.6 meter) cable or a 50-foot (15.2 meter) cable. The extension cables are equipped with an RS-232-C female connector on one end and a male connector on the other end. The distance from the System 2200/2227 to the Model 2227-N Null Modem can be extended by using extension cables as shown in Figures F-1 and F-2. However, for proper transmission of data, total cabling distances should not exceed the maximum values shown. The maximum cabling distance for direct hookup of one System 2200/2227 to another System 2200/2227 via a null modem is 124 feet (37.8 meters). The maximum cabling distance for direct hookup of a System 2200/2227 to other equipment conforming to EIA standards is 50 feet (15.2 meters).

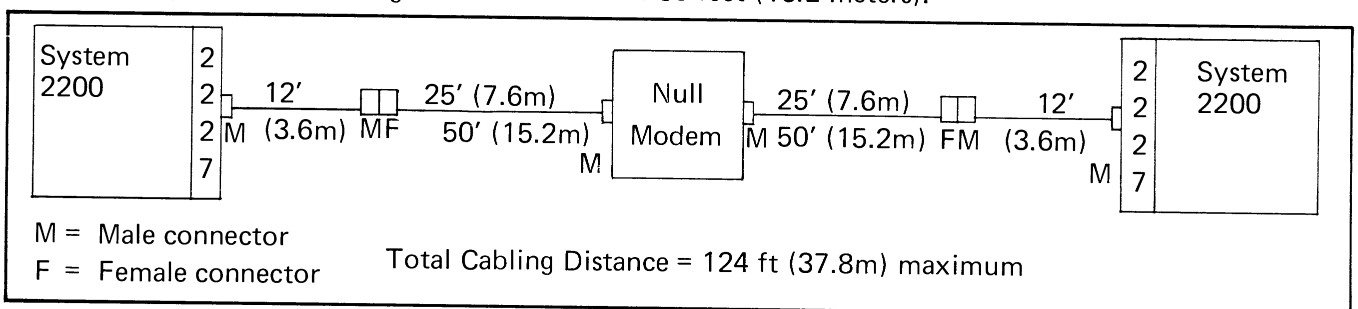


Figure F-1 Extension Cable Configuration for System 2200/2227 to System 2200/2227 Hookup

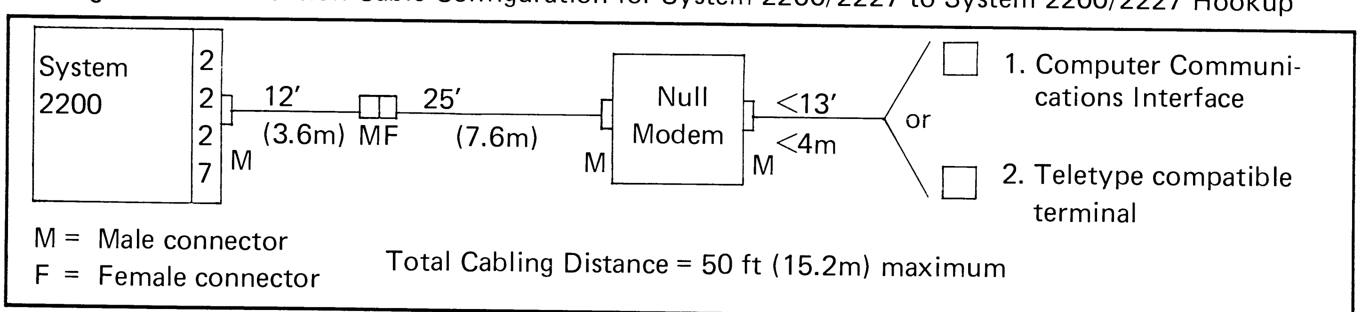


Figure F-2 Extension Cable Configuration for System 2200/2227 Hookup to Other Equipment Conforming to EIA Standards

With the Null Modem, the following configurations are possible for the System 2200/2227:

1. Direct connection of two System 2200/2227's up to 124 feet (27.8m) apart.
2. Direct connection to a computer which already has a cable and an RS-232-C connector capable of plugging into a Bell 103-A or 202-C data set.
3. Direct connection to a Teletype compatible terminal. The System 2200/2227/2227-N configuration can be used if there is no requirement for entering HALT/STEP or RESET from the terminal.

APPENDIX G – TELECOMMUNICATIONS GLOSSARY

ASCII	-	American National Standard Code for Information Interchange. A code for 128 control and data characters, with each character represented by 7-bits (8-bits including a parity check bit).
asynchronous	-	Occurring without a regular time relationship. Transmission in which each character is synchronized individually, usually by using start elements and stop elements.
baud	-	A unit of signaling speed whose definition must be specified. If one baud is defined as the reciprocal of one data bit interval, then a data transfer rate of 110 baud is equivalent to a data-bit-signal-interval of 9.09 milliseconds. Conversion of the data-rate-in-baud to characters-per-second or to milliseconds-per-character is dependent not only upon the number of data bits per character but also upon the type of transmission (synchronous or asynchronous).*
duplex	-	Simultaneous two-way independent transmission between terminals.
modem	-	**Mod**ulator-**dem**odulator. A device which modulates and demodulates signals transmitted over communication facilities.
parity	-	The state of being odd or even, used as the basis of a method of detecting errors in binary-coded data. (A parity check bit makes the sum of all the binary digits, including the parity bit, either always odd or always even.)
synchronous	-	Occurring with a regular or predictable time relationship.

* In asynchronous transmission, where each character is synchronized individually by using a start element and a stop element (in addition to a specified number of data-bit-elements), the start element is usually a fixed length signal equivalent to one data-bit interval. The stop element is often a variable length signal, as in the case of transmission via a keyboard.

For System 2200/2227 applications, the minimum length stop element occurs during steady data flow when the stop element is equivalent to one or two data-bit-intervals, depending on the setting of the 1SB push button (Down=1 stop bit, Up=2 stop bits). Therefore, if the character format of the data flow consists of the following:

7 data bits
1 parity bit
1 start bit
2 stop bits

total 11 bits per character

and the data transfer rate is set for 110 baud (9.09 milliseconds per data-bit), the maximum number of characters per second is 10 (see Table 5).

APPENDIX H – WANG SYSTEM 2200/2227 TELECOMMUNICATIONS CHECKLIST

1. **Is the transmission asynchronous?**

   The answer must be 'yes'; synchronous transmission is not possible with the System 2200/2227.

2. **Is the transmission rate 110, 150, 300, 600, or 1200 baud?**

   Sending and receiving rates must match. Depress the appropriate baud rate switch on the Model 2227 board.

   Check the feasible rates below.
   a) Over telephone lines, from (or to) a System 2200/2227 connected to a Bell 103A modem or equivalent: 110, 150, or 300 baud.
   b) Over telephone lines, from (or to) a System 2200/2227 connected to a Bell 202C modem or equivalent: 110, 150, 300, 600, or 1200 baud.
   c) Over direct connection lines, from (or to) a System 2200/2227 connected to the WANG Model 2227-N null modem or equivalent: 110, 150, 300, 600, or 1200 baud.

3. **What is the character set of the other systems?**

   Hopefully, ASCII. The System 2200 uses the ASCII character set consisting of 7-bit coded control and graphic characters (8 bits including the parity check bit). A different character set requires conversion either in the System 2200 or the other system.

4. **What is the character format of the other system?**

   The Model 2227 Telecommunications Controller can interact with most character formats; e.g., a typical timesharing service uses one start bit, seven data bits, even parity, and two stop bits.

   Set switches, as required, for the following items:
   a) Is there 1 start bit? (No switch. This is standard.)
   b) Are there 5, 6, 7, or 8 data bits? (Set switches NB1 and NB2 according to instructions printed on the Model 2227 face plate.)
   c) Is there parity or no parity? (Set switch PAR: DOWN for parity; UP for no parity.)
   d) If parity, is it odd or even? (Set switch OPS: DOWN for odd parity; UP for even parity.)
   e) Are there 1 or 2 stop bits? (Set switch 1SB: DOWN for 1 stop bit; UP for 2 stop bits.)

5. **What modem is connected to the System 2200? To the other system?**

   a) Is the modem a Bell 103A3, a Bell 202C, or equivalent to one of these? If so, determine the appropriate dialing procedure for hookup to a host CPU (which must be using an equivalent level modem).
   b) Is the modem an acoustic coupler? If so, set the FULL/HALF duplex switch in the 'FULL' position. Then, initiate the phone call for hookup to a host CPU.

6. **What is the protocol of the System being contacted?**

6.1 What telephone number should be called?
For 110 baud:
For 300 baud:
To ask questions:
Of whom? Name:

6.2 What is the normal event sequence immediately after the carrier is established?

a) Does the host CPU send a code to the System 2200 first? If so, what is the code? Is it a single character or a whole text line? (If communicating under program control, the BASIC language receiving variables must be dimensioned properly.) What response from the System 2200 is required by the host?

b) Does the WANG System 2200 transmit first? If so, what code is required? For example: Send a single character 'ESCAPE' code; i.e., HEX(1B), with no carriage-return character. Or, send the code 'GTWX', where several baud rates are valid. Or, send a sign-on message with a specified format.

6.3 What is the end-of-message (EOM) code; i.e., what is the termination code for a text line and/or a query line?

a) Is it a carriage-return-character; i.e., HEX(0D)?

b) Is it a Teletype X-ON character; i.e., HEX(11)?

c) Is it another character? If so, what code?

If the EOM code of the host CPU is not HEX(0D), the internal EOM switch on the Model 2227 printed circuit board may require resetting; e.g., to HEX(11) or other code. Ask your WANG Service Representative. The EOM switch automatically converts the code set on the switch to HEX(0D) required by the System 2200.

6.4 What message or code is sent by the host system when it wants the System 2200 to respond?

a) Is it a text line followed by the Teletype character X-ON; i.e., HEX(11)?

b) Is it a line number only, followed by a delay? (The System 2200B can be programmed for this procedure.)

7. **Is there a sample run on file to demonstrate a successful connection with the system being contacted?**
If not, document a sample run as soon as a successful hookup is made. Indicate what codes, messages, and control characters should be received and sent.

INDEX

NOTES:

EQUIPMENT MAINTENANCE

GUARANTEE

The Model 2227 Telecommunications Controller is guaranteed from defects in materials and workmanship for a period of ninety days (one year for State and Federal Governments).

MAINTENANCE

It is recommended that the Model 2227 be serviced annually. A Maintenance Agreement is available to automatically assure this servicing. If no Maintenance Agreement is acquired, any servicing must be arranged for by the customer. A Maintenance Agreement protects your investment and offers the following benefits:

Preventive Maintenance: Annually your Model 2227 is inspected for worn parts, lubricated, cleaned and updated with engineering changes, if any. Preventive maintenance minimizes "downtime" by anticipating repairs before they are necessary.

Fixed Annual Cost: When you buy a Maintenance Agreement, you issue only one purchase order for service for an entire year and receive one annual billing, or, more frequent billing, if desired.

Further information regarding Maintenance Agreements can be acquired from your local Sales Service Office.

> **NOTE:**
> *Wang Laboratories, Inc. will not guarantee or honor Maintenance Agreements for any equipment modified by the user. Damage to equipment incurred as a result of user modifications will be the financial responsibility of the user.*

To help us to provide you with the best manuals possible, please make your comments and suggestions concerning this publication on the form below. Then detach, fold, tape closed and mail to us. All comments and suggestions become the property of Wang Laboratories, Inc. For a reply, be sure to include your name and address. Your cooperation is appreciated.

TITLE OF MANUAL:

COMMENTS:

Fold

Fold

(Please tape. Postal regulations prohibit the use of staples.)

WANG

Fold

FIRST CLASS
PERMIT NO. 16
Tewksbury, Mass.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

– POSTAGE WILL BE PAID BY –

WANG LABORATORIES, INC.
836 NORTH STREET
TEWKSBURY, MASSACHUSETTS 01876

Attention: Marketing Department

Fold

Cut along dotted line.

Printed in U.S.A.

WANG LABORATORIES (CANADA) LTD.
49 Valleybrook Drive
Don Mills, Ontario M3B 2S6
TELEPHONE (416) 449-2175
Telex: 069-66546

WANG EUROPE, S.A.
Buurtweg 13
9412 Ottergem
Belgium
TELEPHONE 053/74514
Telex: 26077

WANG ELECTRONICS LTD.
1 Olympic Way, 4th Floor
Wembley Park,
Middlesex, England
TELEPHONE 01/903/6755
Telex: 923498

WANG FRANCE S.A.R.L.
47, Rue de la Chapelle
Paris 18, France
TELEPHONE 203.27.94 or 203.25.94
Telex: 68958

WANG LABORATORIES GMBH
Moselstrasse 4
6000 Frankfurt AM Main
West Germany
TELEPHONE (0611) 252061
Telex: 04-16246

WANG SKANDINAVISKA AB
Fredsgatan 17, Box 122
S-172 23 Sundbyberg 1, Sweden
TELEPHONE 08-98-1245
Telex: 11498

WANG NEDERLAND B.V.
Damstraat 2
Utrecht, Netherlands
(030) 93-09-47
Telex: 47579

WANG PACIFIC LTD.
902-3, Wong House
26-30 Des Voeux Road, West
Hong Kong
TELEPHONE 5-435229
Telex: HX4879

WANG INDUSTRIAL CO., LTD.
110-118 Kuang-Fu N. Road
Taipei, Taiwan
Republic of China
TELEPHONE 784181-3
Telex: 21713

WANG GESELLSCHAFT M.B.H.
Formanekgasse 12-14
A-1190 Vienna, Austria
TELEPHONE 36.60.652
Telex: 74640

WANG COMPUTER PTY. LTD.
25 Bridge Street
Pymble, NSW 2073
Australia
TELEPHONE 449-6388

WANG DO BRAZIL COMPUTADORES LTDA.
Rua Barao de Lucena No. 32
Bota Fogo
Rio de Janeiro, Brazil
TELEPHONE 246 7959

WANG INTERNATIONAL TRADE, INC.
836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-4111
TWX 710-343-6769
TELEX 94-7421

WANG COMPUTER SERVICES
836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-4111
TWX 710-343-6769
TELEX 94-7421

24 Mill Street
Arlington, Massachusetts 02174
TELEPHONE (617) 648-8550

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

Printed in U.S.A.
700-3288A
6-74-2M
Price $10.00