Communications Package for the RS-232-C

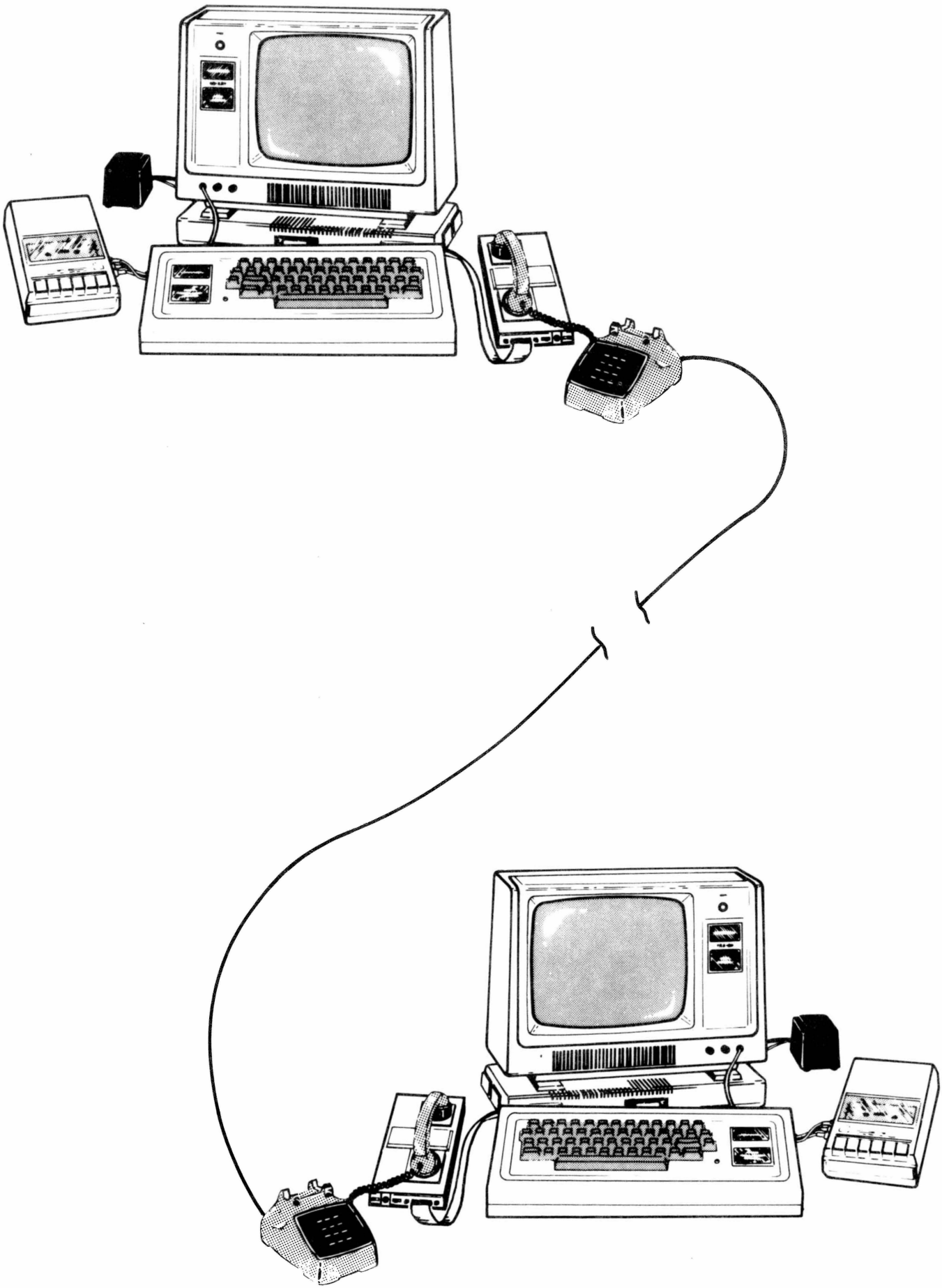
Radio Shack®

TRS-80 MICRO COMPUTER SYSTEM

SOFTWARE

Catalog Number 26-1146

CUSTOM MANUFACTURED IN THE USA BY RADIO SHACK A DIVISION OF TANDY CORPORATION



To Our Customers:

All computer equipment must be programmed before it can do anything useful. For the TRS-80, many different programs exist – to make your Computer play a game, calculate a math problem, or print a business report. These kinds of programs are called applications programs because they are used for a particular purpose or a specific application.

There are also more general kinds of programs available like TBUG that do not play a game or serve any one specific purpose. Instead, they provide general commands that you can use to make your Computer do many different specific things.

The Communications Package falls more in the general-purpose than the specific category. The programs in the package were written as general purpose "driver" programs for the RS-232-C hardware. As with any Computer hardware, the RS-232-C is useless without this kind of software to "drive" it.

The package was written with the particular idea of TRS-80 to TRS-80 communications. Two TRS-80's are linked through a telephone line so they can be at a distance from one another. Then, the communications take place.

One of the programs, Communications Program (COMPROG for short), allows you to transmit data to another TRS-80 or receive data from another system. The other portion HOST-TERM allows you to go on-line with another TRS-80 and operate the Host system from your Term Keyboard.

But the Communications Package is flexible and general purpose because the kind of information you send is entirely up to you. You can use COMPROG to send a new game program to your friend across town or an important business report to one of your branch offices or a personal letter to a friend. With HOST-TERM, you can build an online network of TRS-80's.

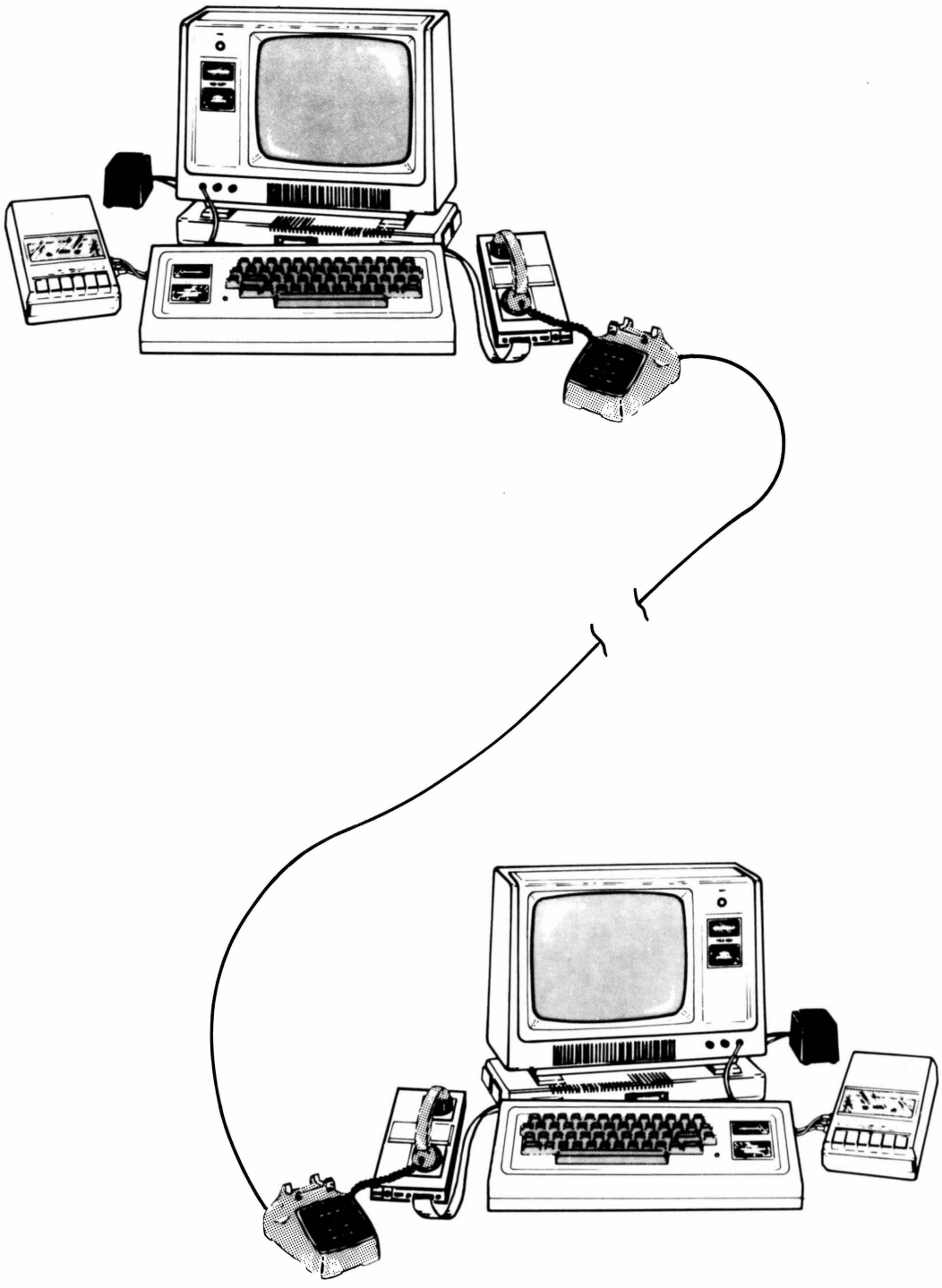
And, if you know enough about it, you can customize these programs to send from your TRS-80 to a different type of Computer. In this case, we provide some information to help you, but you are on your own in programming any special applications.

First Edition – 1979

Software Copyright Notice

Printed in the United States of America

HOST-TERM Program



The HOST-TERM Program

The Communications Package consists of two separate programs each designed to give you different capabilities with your RS-232-C hardware. The HOST-TERM program is quite easy to use and is described briefly in this first section. The Communications Program (COMPROG for short) is considerably more versatile and takes up all the rest of the manual.

The HOST-TERM portion is composed of two different programs – HOST and TERM. It requires two stations – one running each program.

At the Term Station, the CPU, memory and peripheral equipment are no longer being used. The Keyboard and Video are treated as if they were hooked up to the Host Computer instead. Other equipment is not used at all. There is only one version of TERM no matter what memory size your system has.

The HOST program has three versions for the three different memory sizes. HOST16 is for 16K systems; HOST32 is for 32K; and HOST48 is for 48K. The Host Computer can be controlled through its own Keyboard or the Term Station's Keyboard.

What You Need To Get Started

Both TRS-80's must have at least 16K RAM. However, they need not have identical memory sizes. You also both need the Expansion Interface with the RS-232-C Hardware Interface installed. (NOTE: The RS-232-C must be set to 300 Baud. See the next section for more information on this.)

Also, you both need the Radio Shack Telephone Interface II OR one of you can have the Telephone Interface I while the other has a Telephone Interface II. But, you can't both use a Telephone Interface I.

And, finally, you both need an ordinary telephone.

BAUD RATE	S6	S7	S8
1110	Closed	Closed	Closed
150	Closed	Closed	Open
300	Open	Closed	Closed
600	Open	Closed	Open
1200	Closed	Open	Closed
2400	Closed	Open	Open
4800	Open	Open	Closed
9600	Open	Open	Open

PARITY ENABLE	S4	STOP BITS	S5
Parity Enable	Closed	One Stop Bit	Closed
Parity Disable	Open	Two Stop Bits	Open

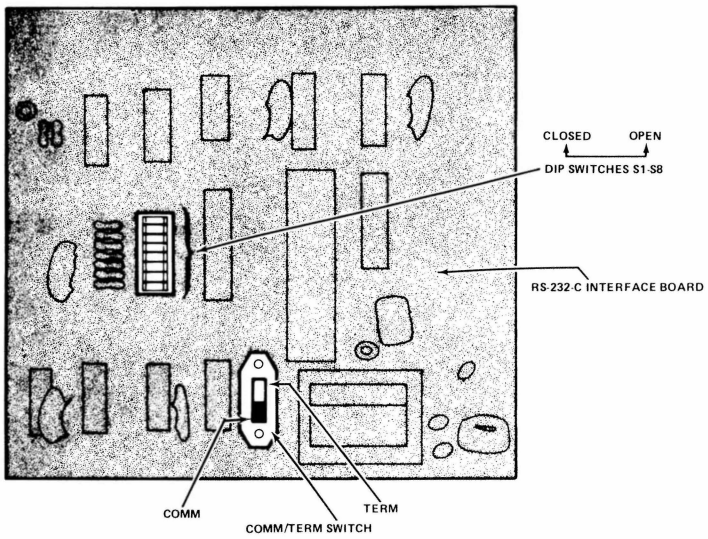
WORD LENGTH (Excluding parity bit)	S2	S3
5 Bit Word	Closed	Closed
6 Bit Word	Closed	Open
7 Bit Word	Open	Closed
8 Bit Word	Open	Open

PARITY SELECT	S1
Odd Parity	Closed
Even Parity	Open

Setting the RS-232-C

In data communications, problems will readily occur if you have incorrectly set the parity, baud rate, stop bits or word length. Neither HOST nor TERM allows you to change these settings through software. Both programs use the settings on the RS-232-C hardware.

So, both stations must set the switches on the RS-232-C Board the same, following the chart on page 4. The drawing below shows the switches on the RS-232-C board.



Using acoustic coupler type modems like the Telephone Interface II and TRS-80's, you should set the switches according to the chart below. The Baud Rate should be 300, one Stop Bit, Parity disabled and an eight-bit Word Length.

Switch	S8	S7	S6	S5	S4	S3	S2	S1
Baud Rate=300	CLOSED	CLOSED	OPEN					
One stop bit and parity disabled				CLOSED	OPEN			
Word length = 8 bits of data						OPEN	OPEN	
Even parity								OPEN

Refer to the RS-232-C Interface Manual, Catalog Number 26-1145 for more information on setting up this board.

The programs only read the hardware switches once when they are first executed. So, if you change the board after the program has executed, you'll have to rerun the program to pick up the new switch settings.

Loading Instructions

Now that you have the right equipment, you need to load the program.

Cassette Operation

Using the program under LEVEL II BASIC, you must reserve high memory by declaring a memory size when you initialize BASIC. Use the size from the chart below that corresponds to the program you will be running.

	Memory Size	Entry Point
HOST16 (for 16K systems)	32249	32250
HOST32 (for 32K systems)	48633	48634
HOST48 (for 48K systems)	65017	65018
TERM (for all size systems)	32255	32256

This address should be typed in response to the MEMORY SIZE? prompt on your Video when you first power up under LEVEL II.

For example, use **32255** if you're running TERM. Responses you type are in black.

```
MEMORY SIZE? 32255    (ENTER)
```

Then, use the SYSTEM command to load the HOST or TERM program from cassette tape. Make sure the recorder is on and the volume is set correctly and the correct version of the program is in the machine.

Type in as follows:

```
>SYSTEM      ENTER
*? H         ENTER
```

The HOST Program will load.

OR

```
>SYSTEM      ENTER
*? T         ENTER
```

The TERM Program will load.

When the tape starts loading, two asterisks (one of them flashing) will appear in the upper right of the Video Display. When it is finished, the SYSTEM prompt will appear. Type / **ENTER** and the program will execute.

An error during loading will generate the character "C" in the upper right of the Video. In this case, rewind the tape and start over. You may need to alter the volume setting to load the tape.

If you exit from HOST or TERM and want to re-execute it, type SYSTEM **ENTER**. This will return the SYSTEM prompt. Then, type / followed by the entry point from the chart above. Press **ENTER** and the program will re-execute.

Diskette Operation

If you have TRSDOS, you should create a disk file for the HOST and TERM programs. Then, you can execute them from the diskette under TRSDOS.

Use the TAPEDISK utility program to create two CMD-type disk files – one named HOST and the other TERM. (You can use any valid TRSDOS filespec you wish.) First, make sure your recorder is properly set up.

C : Cassette
E : End.

```
DOS READY
TAPEDISK      ENTER
? H      ENTER
? F HOST/CMD: d aaaa bbbb cccc ENTER
```

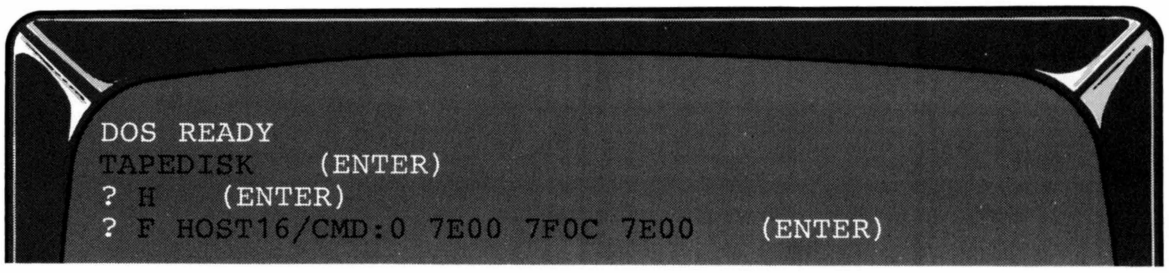
where d is the TRSDOS drive spec, aaaa is the appropriate beginning address from the chart below for your memory size, bbbb is the ending address corresponding to your memory size, and cccc is the entry address for your machine.

TRSDOS

	Beginning	Ending	Entry
HOST16	7E00	7F0C	7E00
HOST32	BE00	BF0C	BE00
HOST48	FE00	FF0C	FE00
TERM	7E00	7F03	7E00 (16K-Version)

Low Mem
Offset = - 11250
21006 ≙ 520E
21264 ≙ 5310

For example, if your TRS-80 has 16K:



Once you save the TERM program and the proper version of the HOST program you can execute either one under TRSDOS by typing in the program name and pressing **ENTER**.

It will then load into the appropriate hex address and execute.

	Hex	Dec	New HM
TERM	7E00	32256 / 65256	FEE8
	7F03	32514 / 65514	FFEA

Offset = 33 000

Using the Telephone Interface

The Radio Shack Telephone Interface II has a switch setting labeled ORIGINATE/ANSWER. Each station must set this switch to opposite settings. It does not matter which is set to ORIGINATE and which to ANSWER, as long as they are opposite. By convention, the station that calls on the phone first uses the ORIGINATE setting, and the other station uses ANSWER.

NOTE: If one of you has the Telephone Interface I, you will notice that it does not have this switch setting. This is because the Telephone Interface I is an ORIGINATE only unit. In this case, whoever uses the Telephone Interface II must set it to ANSWER, so they will be opposite.

Also, there is a DUPLEX switch on both models that must be set at FULL at both stations.

Using the Program

Now that the program is running, all you have to do is establish a telephone connection between the two systems.

Both of you place your receivers in the Telephone Interface. (Make sure the Interfaces are properly connected, and the switches are set.)

Now any commands entered through the Term Keyboard will execute on the Host Computer. Any data that is entered on the Term Keyboard will be input to the Host System. Screen displays will be shown on the Host Video and sent out to the Term Station also. However, no Graphics characters (codes 128-191) will be transmitted for display.

If the Host System is running under TRSDOS, you can enter any valid TRSDOS command from the Term Station to execute at the Host Station. This includes executing BASIC and then running a BASIC program. (Be sure to protect memory by answering the **MEMORY SIZE?** prompt according to the chart on page 6.) If a program requires input, either the Host or Term Station can supply it.

If the Host Station is running under LEVEL II, you can run any BASIC program from the Term Station. Again, you can supply input at the Host or Term Station. It will be just as if you were running LEVEL II with two Keyboards and two Videos hooked up to the Host TRS-80.

You can also run machine language programs in both cases if they do not use memory required by HOST. (See page 6 and 7 for HOST memory requirements.)

In addition to the full range of TRSDOS and LEVEL II commands, there are a few special control codes you can execute from the Term Station by typing **SHIFT**, ↓, and the correct control letter from the chart below.

CODE	MEANING
Control B	Same as the BREAK key
Control C	Same as the BREAK key
Control D	Same as the BREAK key
Control E	Not used by TRS-80
Control F	Not used by TRS-80
Control G	Return from term program to LEVEL II or TRSDOS
Control Q	Identifies Term Station as a TRS-80

At the Term Station, you should enter Control Q (**SHIFT**, ↓ , Q pressed at the same time) in order to identify yourself to the Host System. This will allow you to receive **PRINT AT** output on your Video correctly. You can enter this at any time.

These programs can be used on other systems, but this may require changes in the program. Only the experienced programmer should attempt this.

There are many applications for programs like HOST and TERM. Here are a few ideas to get your imagination started.

Running an Accounting Package. The main branch of a business could run the HOST Program under TRSDOS and a branch office could run TERM. Execute an Accounting Program at the Host Station. Then, the branch office can enter transactions during the day. Several branches can be connected but only one can input at a time. The data will be displayed at all branches.

Running an Online Data Base. A central Host System can maintain disk files and a Term Station from another department or a branch office can access this file for information. The stored information can be account records, an information retrieval system, a library catalog, or any kind of file system.

Playing a Game with Friends. With one of you running HOST and the other TERM, you can play Computer games with each other. Both of you can input. However, Graphics characters will not be displayed at the Term Station.

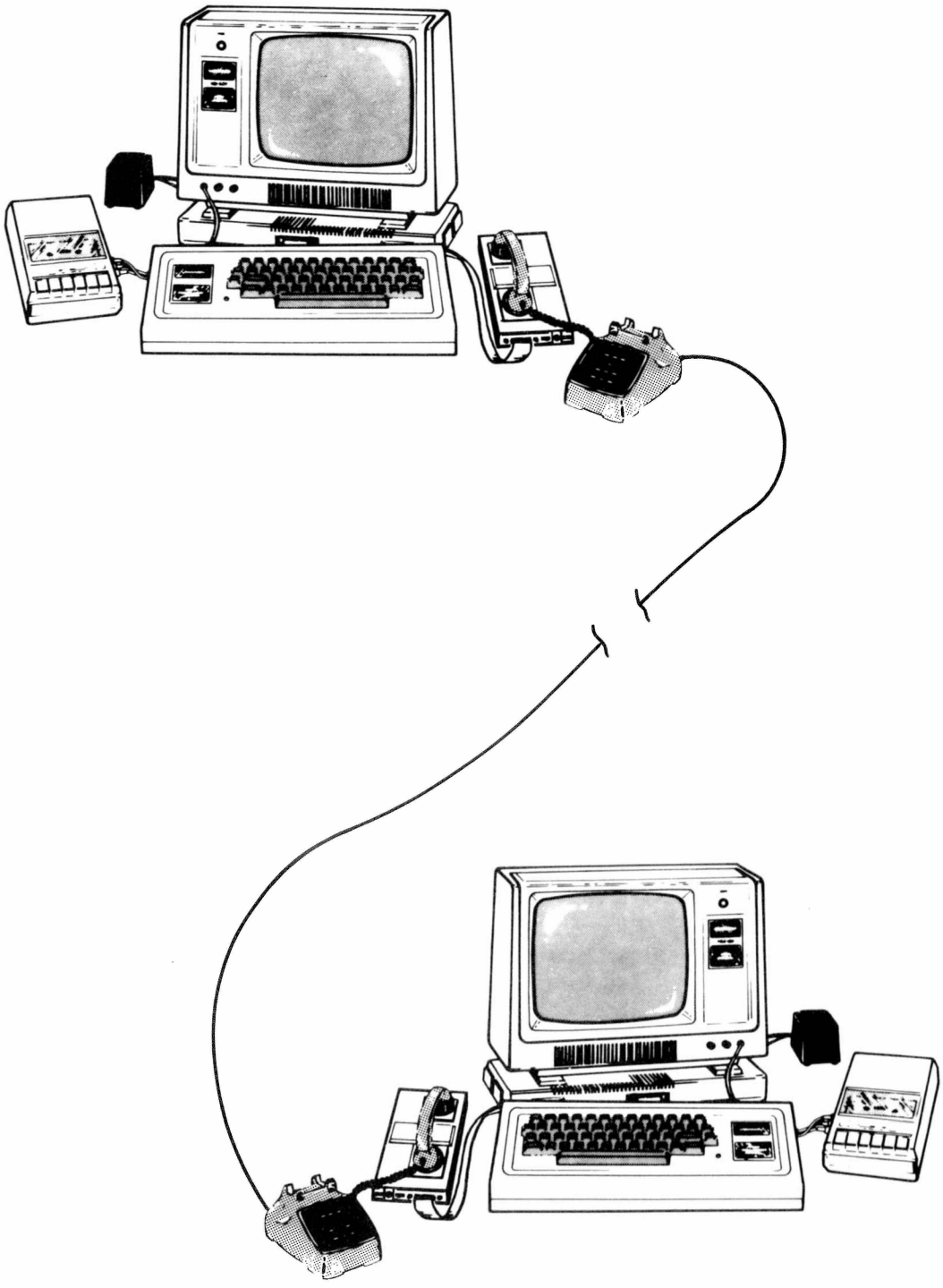
Running a Bulletin Board, Newsletter, or Advertising. The Host Station can run a program in an infinite loop that will print out a Newsletter or Bulletin Board of information or Advertising. Then, any Term Station can call up and read the display on his own Video.

In applications such as this, you will not want the Term Station to be able to interrupt your Display Program with a BREAK command. To disable this interrupt recognition POKE the correct address from the chart below with 138 at the beginning of your Display Program.

HOST16	**POKE 32453,138**
HOST32	**POKE-16699,138**
HOST48	**POKE-315,138**

At the end of your program POKE the same address with 01 to re-enable the BREAK recognition.

Communications Program



Quick Operating Instructions

Imagine that you've just gotten a new game program that you'd like to share with your friend across town who has his own TRS-80.

At best, this means that you must copy your program on a cassette or diskette and drive your car across town to load the program into his machine. Then, he will have to produce his copy from this.

If your friend lives out of town instead of just across town, it complicates matters even more.

But, if you both have the Communications Program, the solution to this problem is easy. The Communications Program will allow you to automatically send information stored in your TRS-80 to another TRS-80.

The next part of this manual provides quick operating instructions for two popular applications of the program – sending or receiving a BASIC program and sending or receiving a disk file. These two typical applications are described step-by-step so you can get immediate results. It's intended for those of you who will primarily work with only these two, and it leaves out the detailed operating instructions for a broader set of uses.

The more extensive uses for this program and details on how they are carried out follow.

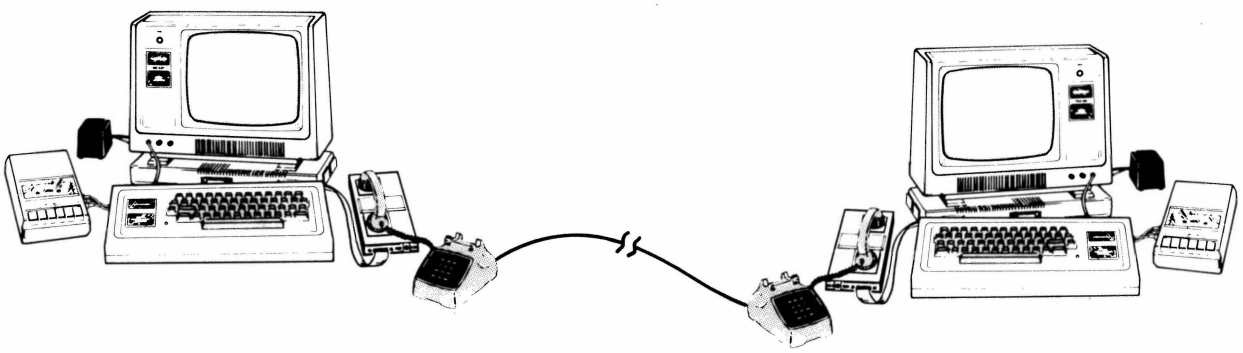
What You Need To Get Started

Both you and your friend will need similar TRS-80 equipment. Both TRS-80's must have at least 16K RAM. (However, they need not have identical memory sizes.) You also both need the Expansion Interface with the RS-232-C Hardware Interface installed. (NOTE: The RS-232-C must be set to 300 Baud. See page 5 for more information on this.)

Also, you both need the Radio Shack Telephone Interface II OR one of you can have the Telephone Interface I while the other has a Telephone Interface II. But, you can't both use a Telephone Interface I.

And, finally, you both need an ordinary telephone.

Receive Station **Send Station**



Of course, you both have to have the Communications Program. There are two versions of the program: one runs with LEVEL II BASIC and the other with TRSDOS; use whichever matches your system. There are also three different options within each version corresponding to the memory size of your TRS-80.

Loading Instructions

Now that you have the right equipment, you need to load the program.

LEVEL II

Using the LEVEL II version, you must reserve high memory for the program by declaring a memory size when you initialize BASIC. Use the beginning memory address minus one (the memory size in the chart below) that matches the total RAM you have.

LEVEL II BASIC

	Beginning	Ending	Entry	Memory Size
16K Memory	28618	32187	30229	28617
32K Memory	45002	48571	46613	45001
48K Memory	61386	64955	62997	61385

This address should be typed in response to the **MEMORY SIZE?** prompt on your Video when you first power up under LEVEL II. For example, use **28617** if you have 16K memory.

```
MEMORY SIZE? 28617 ENTER
```

Then, use the SYSTEM command to load the Communications Program from cassette tape. Make sure the recorder is on and the volume is set correctly.

Key in as follows:

```
>SYSTEM ENTER

*? C ENTER
```

The tape will begin to load, as indicated by the flashing asterisks on the Video Display. When it's finished, the SYSTEM prompt will appear. Type / **ENTER** and the program will execute, entering the Monitor mode of the Communications Program first.

An error during loading with generate the character "C" in the upper right of the Video. In this case, rewind the tape and start over. You may need to alter the volume setting to load the tape.

If you exit from the program after it has started running, type SYSTEM **ENTER**. This will return the SYSTEM prompt. Then, type / followed by the entry point of the program from the chart above. Press **ENTER** and the program will re-execute.

TRSDOS

If you have TRSDOS, you will probably want to create a disk file for the Communications Program. Then you can execute it from a diskette under TRSDOS. For this, you will use the TRSDOS version of the Communications Program.

Use the TAPEDISK utility program to create a CMD-type disk file named COMPROG (or any other valid TRSDOS name you wish). Make sure your recorder is set up properly.

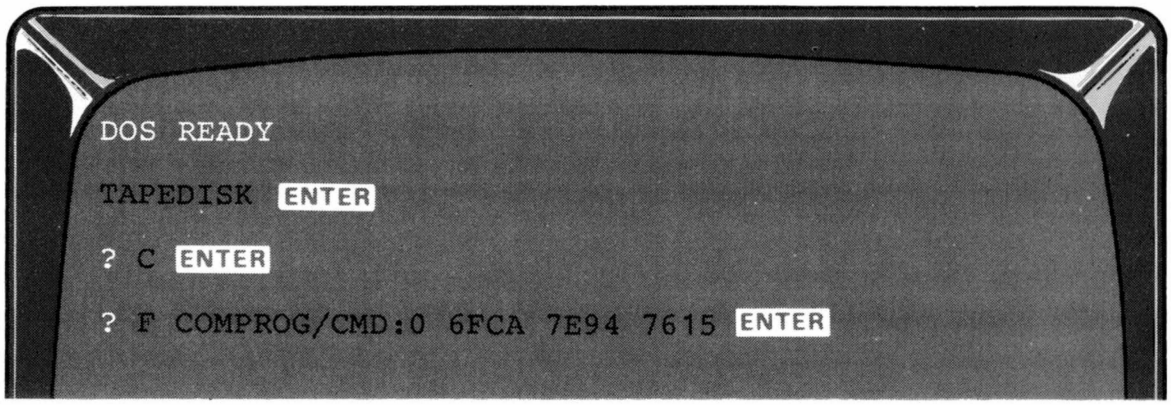
```
DOS READY

TAPEDISK ENTER

?C ENTER

?F COMPROG/CMD: d aaaa bbbb cccc ENTER
```

where *d* is a TRSDOS drive spec, *aaaa* is the appropriate beginning address for your memory size, *bbbb* is the ending address corresponding to your memory size, and *cccc* is the entry address for your machine. All are given in the following chart.

TRSDOS

	Beginning	Ending	Entry
16K Memory	6FCA (28618)	7E94 (32404)	7615 (30229)
32K Memory	AFCA (45002)	BE94 (48788)	B615 (46613)
48K Memory	EFCA (61386)	FE94 (65172)	F615 (62997)

For example, if your TRS-80 has 16K:



Responses you type are in black.

You may now execute the program whenever you wish under TRSDOS by typing the program name.

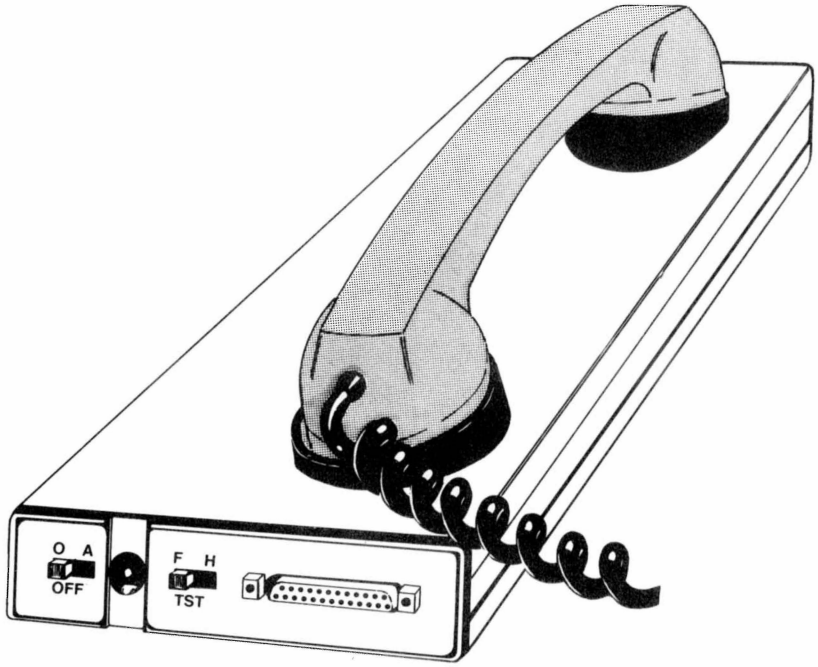
It will then load into the appropriate hex address and execute bringing you into the Monitor mode of the Communications Program.

Using The Telephone Interface

The Radio Shack Telephone Interface II has a switch setting labeled ORIGINATE/ANSWER. Each station must set this switch to opposite settings. It does not matter which is set to ORIGINATE and which to ANSWER, as long as they are opposite. By convention, the station that calls on the phone first uses the ORIGINATE setting, and the other station uses ANSWER. Both stations can send and receive data regardless of these settings.

NOTE: If one of you has the Telephone Interface I, you will notice that it does not have the switch setting. This is because it is automatically set to ORIGINATE only. In this case, whoever uses the Telephone Interface II must set it to ANSWER, so they will be opposite.

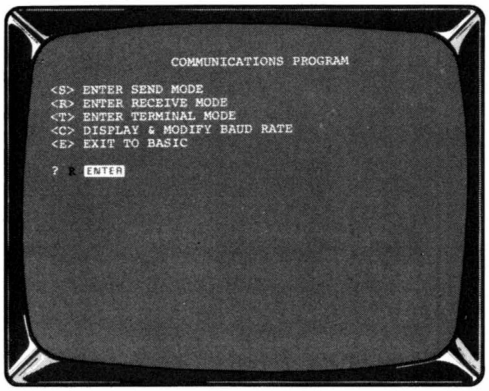
Also, there is a DUPLEX switch on both models that must be set to FULL at both stations.



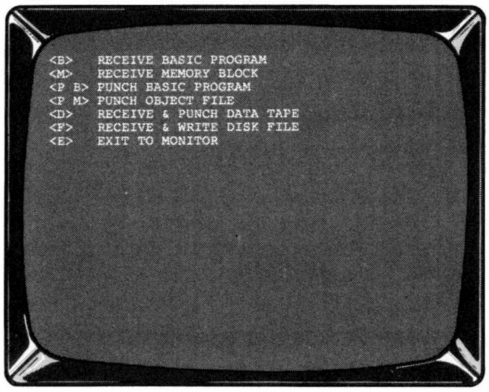
Sending A BASIC Program

When you first load the Communications Program, you automatically enter the Monitor mode, and the following will be displayed at both ends:

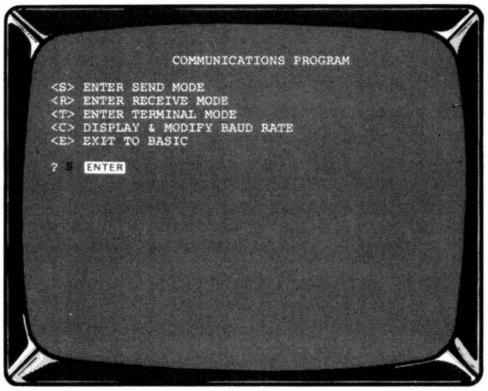
Receive Station



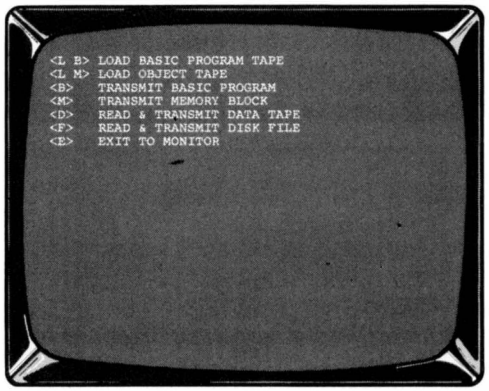
1. Type R **ENTER** . The following will replace the original display.



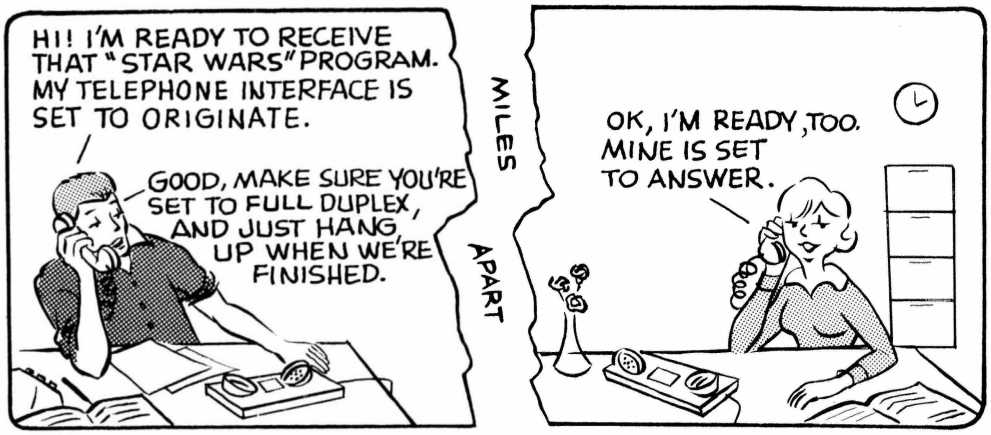
Send Station



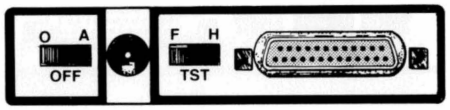
Type S **ENTER** .The following will replace the original display.

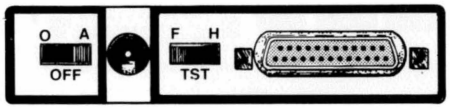


2. One end telephones the other. It doesn't matter who does this. If you are calling long distance, keep an eye on the time so you won't be surprised by next month's telephone bill.

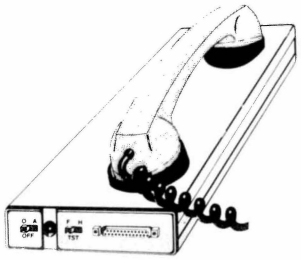


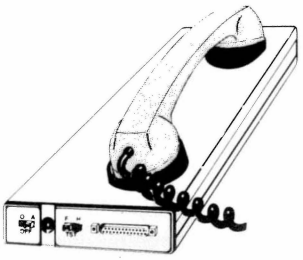
3. Make sure both your Telephone Interfaces are set to FULL DUPLEX, and one is set to ORIGINATE while the other is set to ANSWER.





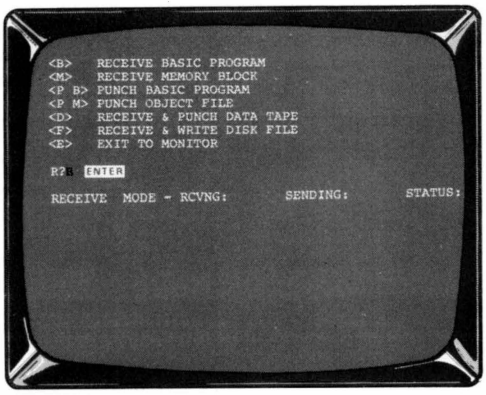
4. Both place the telephone handset into the interface.

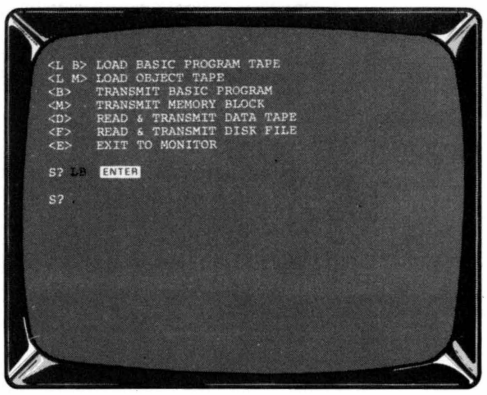




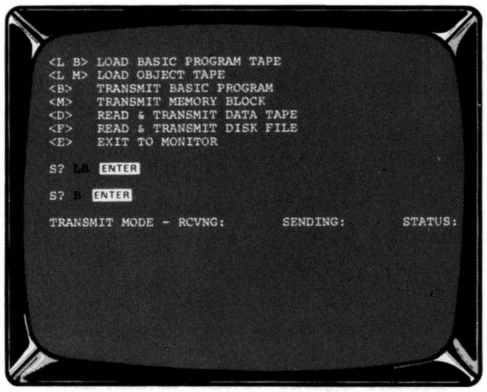
5. The Receive station should set up first by typing **B** **ENTER** . The following will appear below the option list and will reflect the status of the transmission while you are receiving.

While the Receive end is setting up, the Send station can load the BASIC program into memory. (If the program is already in memory wait a few seconds before sending so the Receive end will be ready.) First, turn on the cassette recorder and wind it down to the beginning of the BASIC program. (The BASIC program should have been saved using the CSAVE format.) Type **LB** **ENTER**. The Send prompt, **S?** , will reappear automatically when loading is complete.

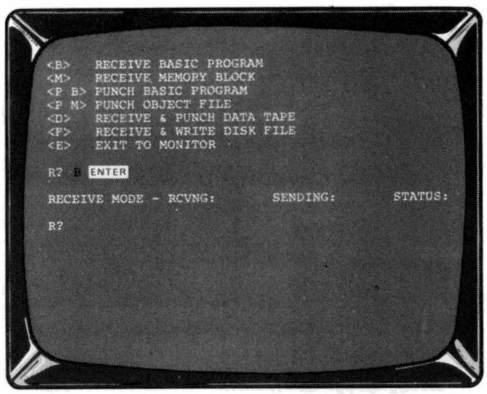


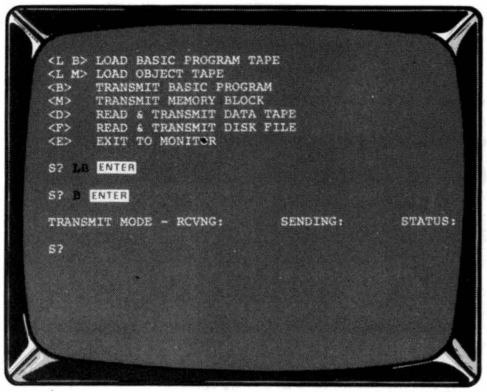


6. Type **B** **ENTER** and the program will be sent on a block-by-block basis. The status message will be displayed during the transmission.

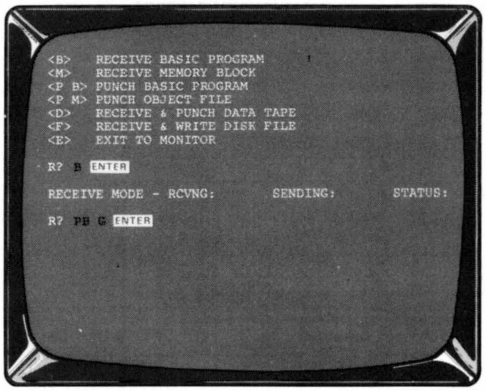


7. When complete the prompts will return at each end.





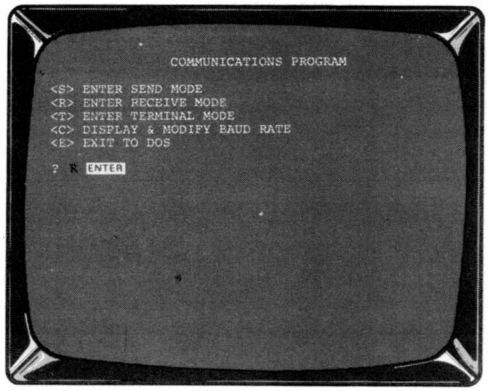
8. The Receive station can save the program on cassette in the CSAVE format. Type in **PB** *x* **ENTER** where the *x* is any one-letter file name you wish. When recording is complete the **R?** prompt will return.

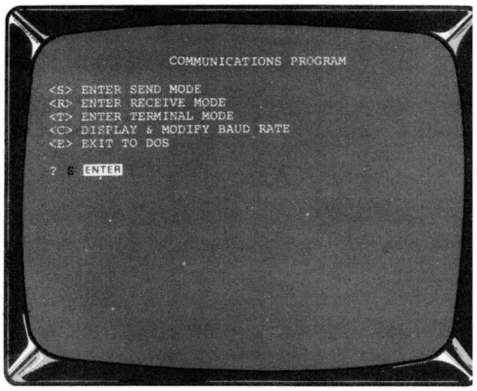


9. You can both pick up the receive when the transmission is over and have a conversation, send another program (either station can do this), or hang up.

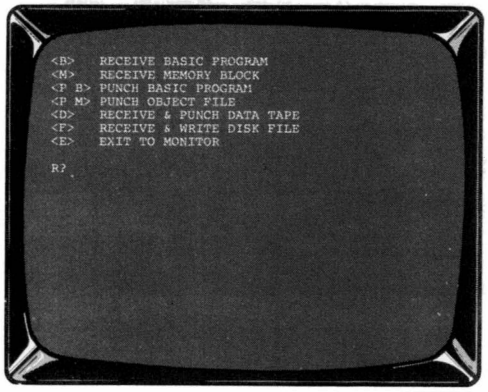
Sending A Disk File

You can send a disk file only if *both* of you have the TRSDOS version of the Communications Program. When you load the program as before, the Monitor mode will be displayed first.

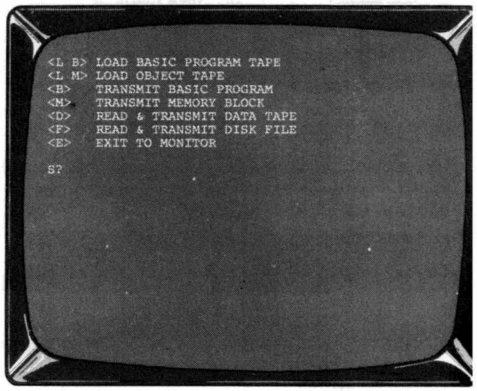




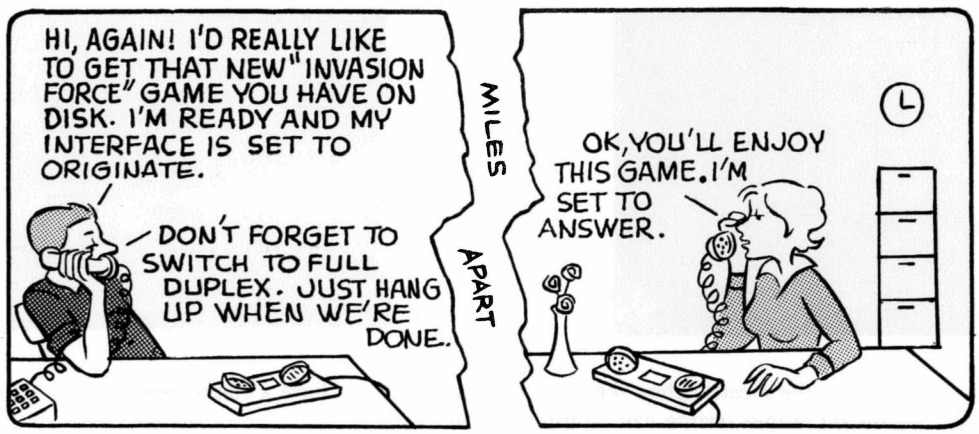
1. Type **R** **ENTER**.



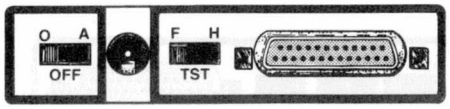
Type **S** **ENTER**.

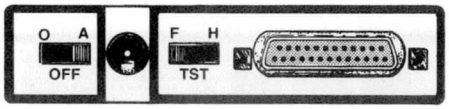


2. One station calls the other. It does not matter which does the calling. And don't forget that long distance bill.

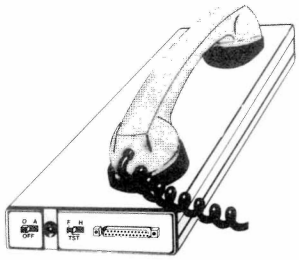


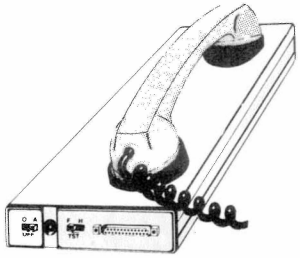
3. Both you and your friend should set your Telephone Interfaces to FULL DUPLEX, and one to ORIGINATE and the other to ANSWER.



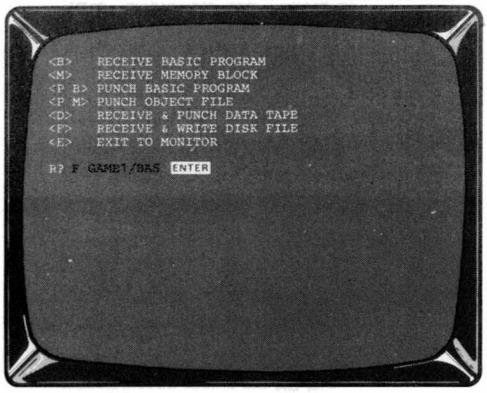


4. Both place the telephone handset into the interface.





5. The Receive station should set up first as always. Type **F** followed by the valid TRSDOS file specification (filename, followed by optional extension, password and drive spec) that you want to name the file.

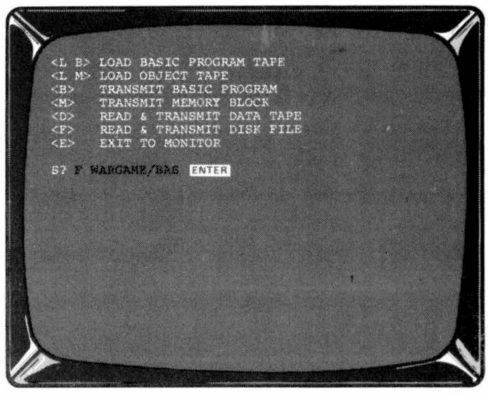


Here, the password is not specified. The BASIC program named GAME2/BAS will be written on Drive 2.

NOTE: The file name here is not (and need not be) the same as the name of the file sent from your friend. You can name the file whatever you want to.

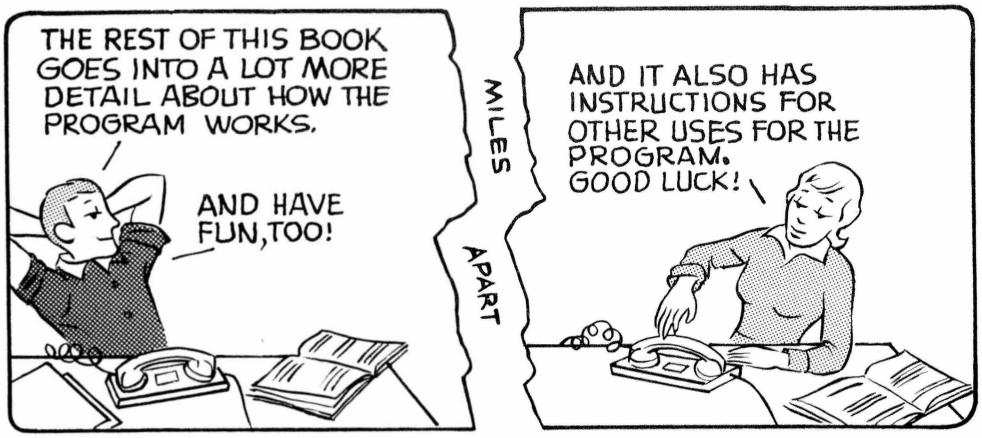
6. The Send station waits a few seconds while Receive is getting ready. Then, type **F** followed by the valid TRSDOS file specification.

The status message will appear. NOTE: If you specify a disk drive as part of the TRSDOS spec, the program will look for the file only on the drive you specify. Otherwise, it starts at Drive 0 and searches until it finds the file with the proper name. Any passwords must be specified as part of the name before a file can be found.



7. At the end of the transmission the prompts will return to both stations. The file is saved automatically at the Receive station during transmission. You can remove the telephone and have a conversation, or send another file or hang up.

Any kind of disk file (object, BASIC program, or data) can be sent. For example, you can type a letter and save it as a disk file and transmit this file. If the Receive end names it with a TXT extension, he can use the TRSDOS LIST- command later on to list it on his Video Display.



General Information

The first two sections of this part of the manual present a general description of the RS-232-C Communications Package. They'll give you a "forest" to keep sight of . . . because, in later sections, we'll start looking at the "trees". And we don't want you to feel too lost lost in the woods.

After reading the third section, you'll be able to use all aspects of the package, and section four will give you some helpful hints on troubleshooting while also going into greater detail on the theory. Finally, there's the appendix with related information and reference charts.

Enough said about the manual – On to the system!

The Communications Program is a machine-language program on cassette tape that will let you use your TRS-80 (if it's properly equipped) to send and receive data to and from remote locations where similar equipment is set up. In other words, your Computer can "talk" to another Computer located somewhere else.

Required equipment is the TRS-80 with at least 16K Ram, Expansion Interface with RS-232-C, a Radio Shack Telephone Interface II, an ordinary telephone and the Communications Program. Of course, the remote station must have similar equipment although both of you needn't have the same memory size.

(NOTE: You can both use a Telephone Interface II, or one can use a Telephone Interface I while the other has the Telephone Interface II. But both of you cannot use the Telephone Interface I.)

The data is sent over telephone lines (Be sure to watch the time if you are transmitting long distance!) using all of the required equipment.

The RS-232-C is a hardware device that you can add to your TRS-80 Expansion Interface to provide a parallel-to-serial interface. It converts eight bits of data received from the CPU in parallel (all at one time) into eight bits sent out to the modem serially (one bit at a time) and vice versa. (For more information on this, refer to the **TRS-80 RS-232-C Interface Manual**, Catalog Number 26-1145.)

The acoustic coupler, Radio Shack Telephone Interface II, is a hardware interface device that transforms a digital electronic signal (which can be one of two possible states) into a corresponding analog electronic signal (a frequency-modulated carrier) that can be sent over a telephone line and vice versa.

Using this equipment, your Computer can change data, first to eight-bit serial form and then to analog. In this form, it can be transmitted over an ordinary telephone to a distant receiver. There it is a converted back, first to eight-bit serial and then to eight-bit parallel, so it can be input to the other Computer.

There are two versions of the program: one runs with LEVEL II BASIC and the other with TRSDOS, so you should use whichever matches your system. There are also three different options within each version corresponding to the memory size of your TRS-80.

Naturally, there are some possible substitutions for required equipment. For example, although Radio Shack cannot guarantee other manufacturers' so-called RS-232-compatible equipment, there are other acoustic-coupler modems available. Also, different kinds of modems (although more expensive) may be used.

It's also possible to make direct connections between the two Computers without using modems or telephones at all, if the distances involved permit direct-wire hook-ups.

And finally, in some applications, Computers other than the TRS-80 may be involved. For example, the TRS-80 may be linked to a larger communications network (if you make prior arrangements with the network).

In many of these applications, especially the last where another type of computer is involved, special programming may be necessary. Although such programming is not provided by Radio Shack, there is information in the appendix to help with such a project. (Again, the **TRS-80 RS-232-C Interface Manual** would be useful in these special applications.)

The Program And You

The Communications Program consists of four major command modes, each of which in turn consists of several individual commands. These control the transfer of data and perform related tasks.

Data can be transmitted as:

1. Disk files (if you use the TRSDOS version)
2. BASIC programs on tape recorded with the CSAVE command
3. Data tapes recorded with the PRINT# command
4. Object programs on tape recorded with the Editor/Assembler, TBUG, or purchased as SYSTEM tapes
5. BASIC programs in memory
6. Blocks of memory in RAM or
7. Keyboard input (in Terminal mode only)

The related tasks include:

1. Error checking and error recovery functions
2. Data access functions to retrieve data you wish to transmit and
3. Exit routines that allow you to move from one mode or set of commands to another

When you enter a command mode, a list of the commands for that level will be displayed at the top of the screen.

Each mode also has its own unique prompt character which will be displayed whenever the Computer is waiting to receive a command. An error message **INVALID COMMAND** will be displayed if you use a command that is not appropriate for the level you are on.

Once you enter a given mode, the system will remain there until you instruct it to change to another mode using an exit command.

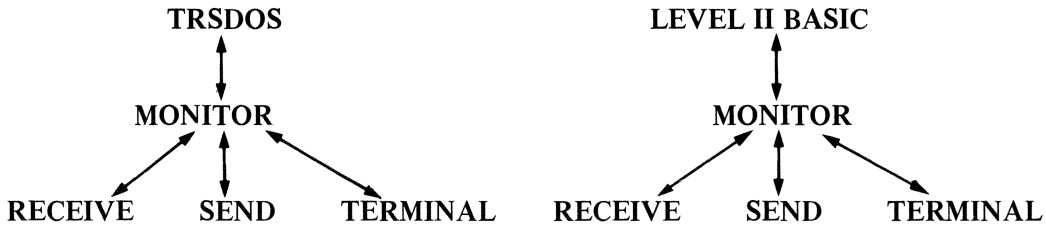
The four major modes of operation are: **Monitor**, **Receive**, **Transmit**, and **Terminal**.

When you first execute the program, you will enter **Monitor** mode. From here, you can access the **Receive**, **Send** or **Terminal** modes; or you can exit to TRSDOS or LEVEL II.

In the **Terminal** mode, you can only transmit data through the Keyboard or receive data on your Video. There are no commands on this level except an exit command that returns you to **Monitor** mode.

In the **Receive** and **Send** modes, you can transmit or receive data from disk, tape, or memory, but you cannot communicate using the Keyboard. To exit either of these modes, you must return to the **Monitor** mode.

The following diagrams illustrate the various enter/exit possibilities for TRSDOS and LEVEL II operation.



By now, you should have a rough idea of:
what your Communications Package will do
what equipment you will need
what types of data can be transferred.

Furthermore, you should have an idea of how you will operate this system by typing in short commands to execute these functions.

And you're probably getting plenty of good ideas of what you can use this system for – seriously and for fun.

If you need to review these basic ideas now, reread these sections before going on to the detailed operation instructions.

Operating The System

When you are in any one of the several possible modes, the program will display a prompt while it awaits a command from you.

A command consists of a single-letter operator code followed, in some cases, by an operand. The operator code specifies the function to be performed, and the operand (if required) qualifies this by specifying the data to be operated on. In other words, the operator code tells the Computer what to do, and the operand tells what to do it with. Operands can be disk file names, BASIC or machine level tape file names or hexadecimal addresses.

To enter a command, type the appropriate key and press **ENTER**.

Spaces typed between operator codes and operands are ignored. However, blanks are not allowed *within* an operand. Disk file names must follow the TRSDOS file specification format (filename, optional extension, password and drive spec). Memory addresses are four hexadecimal digits with no blanks separating them. Tape file names must follow the SYSTEM or CSAVE format.

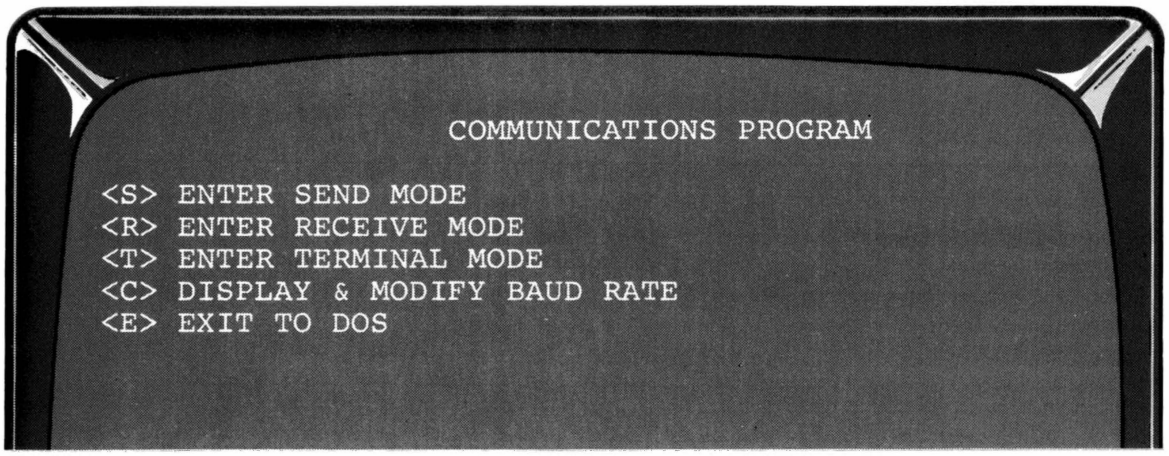
Now let's start using the program. The best way for you to use this part of the manual is to sit down at your Keyboard and try out the examples as they are given.

In all these examples, the response you type in will be highlighted in black. First, follow the loading instructions found earlier, page 15-16.

Monitor Mode

When you first execute the program, or when you exit from another mode, you will enter the Monitor command level. It is only used to enter one of the three communications modes. No communications can take place at this level.

The commands list will appear at the top of the Video and will be followed by the unique prompt ?



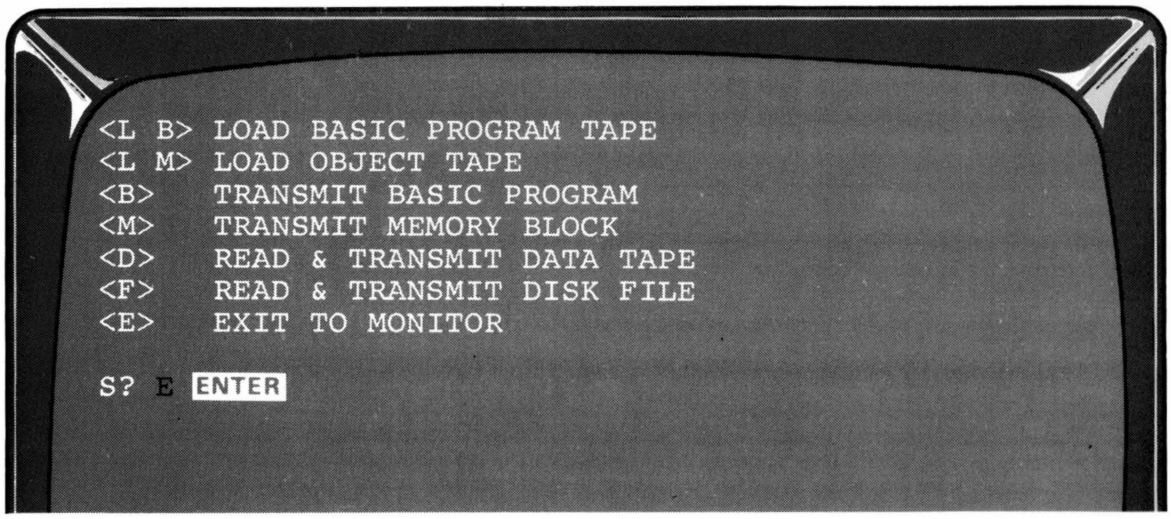
This will remain on the screen until it is scrolled off the top by later entries. In the LEVEL II version, **<E> EXIT TO BASIC** will be displayed instead of the last line above.

None of the commands at this level require any special operands so you need only enter the single-letter key for the mode you wish.

For example:

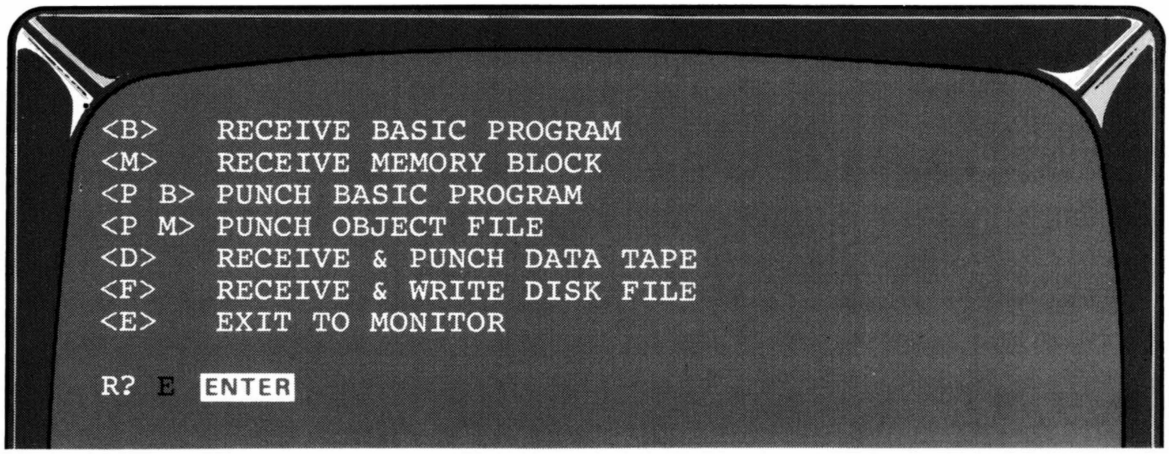
?S **ENTER**

This command will put you in the Send mode, and the Monitor display will be replaced by the Send display.



The **E ENTER** command returns you to Monitor mode. The display will also change back.

Now, try typing R **ENTER** . The program will enter Receive mode, and the following will be displayed:



Again, you can return to Monitor by typing E **ENTER** .

Did you notice what prompts were used for Send and Receive modes? Try entering these modes again and returning to Monitor, and see how their prompts differ from each other and from the Monitor prompt.

Remember, you cannot go directly to Receive from Send or vice versa. You must first exit to Monitor to enter the mode you wish.

The other three Monitor commands are a little different from these first two. The Terminal mode, for example, does not display a list of commands because it has no commands. To enter it, first type T **ENTER** .

Then, a prompt will appear on the next line. To enter Terminal mode, you must specify three control parameters by answering three such prompts.

```
?T ENTER
BAUD=XXXX? ENTER
PARITY=AAAA? ENTER
STOP BITS=X? ENTER
```

A three or four digit Baud rate will appear in place of the *x*s next to the baud rate; either **EVEN**, **ODD**, or **NONE** will appear in place of the *a*s; and either **1** or **2** will appear in place of the *x* next to the Stop Bits.

These allow you to set values to control the way your data is transmitted. There will be more about them in the section on the Terminal mode. For now, just hit the **ENTER** key. This will cause the value to default to whatever it is currently so no change is made.

Go ahead and try as shown above. Surprised? Well, if your screen winked at you and returned immediately to Monitor mode, don't worry. Your Computer has not suddenly become mischievous.

It actually did enter Terminal mode when it blinked; but, at this command level, the program checks your modem. If you are not already hooked up by telephone to a remote station and receiving a carrier signal from them, you will automatically return to Monitor. Of course, this only takes a short time, and you will see it as a blink.

After you have made a connection with a remote site and are receiving a carrier signal from them, you can stay in Terminal mode until you command an exit back to Monitor.

The next command, **C**, will display the Baud rate and allow you to change it if you wish. As soon as you set this, you will return to Monitor mode automatically.

```
?C ENTER
BAUD=9600? 30 ENTER
```

For now just press **ENTER** and the Monitor prompt will return on the next line.

Finally, if you type **E** **ENTER** you will exit to TRSDOS or LEVEL II. If you are running the TRSDOS version, the message **DOS READY** will be displayed, and from here you can execute any regular TRSDOS commands. If you are running under LEVEL II, you will exit back to BASIC.

Now, if you have already tried this command, you may realize that you can no longer access Monitor mode or any other command level without first re-executing the program. Go ahead and try it if you haven't already, and practice rerunning the program. You will return to Monitor mode as soon as you do this.

NOTE: Under Level II, the program will remain in memory and you can execute it by typing **SYSTEM** **ENTER** followed by */ entry point* **ENTER** . Use the entry point from the chart given in the Load Instructions, page 15-16.

Terminal Mode

At this point, it would be very helpful for you to start consulting the flow charts in Appendix C. They diagram the options you can take at any level and the steps to implement any of these options. The text explains these steps and includes examples. You might want to use this chart later for reference.

First, call the remote station on the telephone and decide on procedures to follow and settings to use for Baud rates, Parity, and Stop Bits. Connect the telephone receiver to the modem, making sure the modems are set opposite – one to ORIGINATE and the other to ANSWER. Both should be set at HALF DUPLEX.

In the following examples, suggested values that correspond to a typical application will be used.

Now, type T **ENTER** to display and change existing Baud rates, Parity and number of Stop Bits one at a time as agreed upon.

By entering values for these three parameters, you can change the control specification without changing the RS-232-C hardware switches. All of this is done while in Monitor mode.

```
?T ENTER
BAUD=9600? 300 ENTER
PARITY=NONE? EVEN ENTER
STOP BITS=2? 1 ENTER
```

The Baud rate is the number of bits per second you will transmit and should be set to reflect the speed at which your modem can transmit. The Radio Shack Telpehone Interface II (and other acoustic-coupler type modems) are capable of transmission up to 300 Baud.

Other types of modems, capable of a faster transfer rate, require special telephone equipment and are a good deal more expensive. (See Appendix A for information to help you adapt the program to other equipment.)

If you do not wish to change the Baud rate, press **ENTER** .

If you do want to make a change, key in a value from the following choices – 11Ø, 15Ø, 3ØØ, 6ØØ, 12ØØ, 24ØØ, 48ØØ, or 96ØØ. The existing Baud rate will be displayed in its entirety. However, to make a change, you need type only the first two digits and press **ENTER** , because the program only checks the first two digits. Anything you type after this is insignificant.

For example,

BAUD=2400? 300 ENTER , BAUD=2400? 30 ENTER , BAUD=2400? 30XY ENTE

all result in the same change, resetting the Baud rate to correspond to a modem capable of 3ØØ Baud.

If you type in a value that is not acceptable, the existing Baud rate will be displayed again on the next line until you enter an appropriate response.

Both stations, local and remote, must use the same Baud setting. Their modems must be capable of this same transfer rate.

Once you answer appropriately, the second parameter appears on the next line, allowing you to set the Parity.

```
BAUD=9600? 30 ENTER
PARITY=NONE?
```

You have four choices here:

1. NONE ENTER
2. EVEN ENTER
3. ODD ENTER or
4. simply press ENTER if you do not wish to change the existing value.

As in the case of the Baud rate, the program only checks the first character. So, you may enter any string of characters as long as it begins with N, E, or O.

```
PARITY=NONE? ZAP ENTER
PARITY=NONE? EACH ENTER
```

In the example above, ZAP will not be accepted. The question will be displayed again until you enter an acceptable response. EACH will set the Parity to even.

If the Parity is set to none, all eight bits will be used to represent data. In case Parity is set to even or odd, only seven bits represent data and the eighth bit is used for Parity checking.

You are allowed to set Parity only in the Terminal mode. This is because, in the Terminal mode, data only from the keyboard can be transmitted and only seven bits are necessary to represent each keyboard character. Since the eighth bit is extra, it can be used for Parity checking.

In the Send and Receive modes, data is not transmitted from the Keyboard but from disk files, tapes or memory. Here, all eight bits are used to represent the data, so Parity is automatically turned off when you enter these modes. Other error checks are made in these modes.

If Parity is not set, no character error checking is done in the Terminal mode.

When it is set, to even or odd, data is checked for error control. Before transmitting the data, the system checks whether the number of bits turned on in the first seven is even or odd. If the Parity is set even and the number of bits turned "on" is even, then the eighth bit is turned off. If the number of "on"-bits is odd, the eighth bit is turned on, so that the number of "on"-bits in the entire byte is always even.

If the Parity is set to odd, the eighth bit is set for the opposite conditions, so that the byte always has an odd number of bits set.

At the Receive end, the first seven bits are again tested for Parity. Then, the eighth bit is verified. If it is not turned on or off properly, errors in the data are caught. A graphics bar is displayed next to the character that did not check on the Receive end.

When in the Terminal mode, both local and remote stations must have Parity set the same. This should be agreed upon in advance.

After the Parity is set, the number of Stop Bits is displayed. (See the **TRS-80 RS-232-C Interface Manual** for an explanation of the Stop Bit in Data Communications.) The Stop Bit is used in asynchronous data transfer as with the RS-232-C. It signals the Receive end to seek the Start Bit of the next byte since this may not immediately follow the one just received.

You may type 1 **ENTER** , 2 **ENTER** or simply press **ENTER** if you do not wish to modify the current value. Any other response will be ignored and the question repeated.

```
STOP BITS=2? R ENTER
STOP BITS=2? 1 ENTER
```

Both locations must use the same value, and only the Terminal mode allows this choice in setting. With Send or Receive, it is automatically set to one.

Once you supply all three of these parameters, you enter the Terminal mode. The screen clears and there is no display. Now you can transmit data from your Keyboard to the other station's Video on an item-by-item basis, one byte at a time. Since each byte represents one Keyboard character, each letter is transmitted as you type it.

This means that you both can't be typing characters at the same time, or the screen display will be a jumbled and unreadable mess.

If you want the data that you type to be displayed on your Video, as well as being transmitted, you should set the DUPLEX switch on your modem to HALF. (Other acoustic couplers have a similar setting.) This switch must be set to FULL DUPLEX before you can transmit in Send or Receive modes. Both stations should have the ORIGINATE/ANSWER switch set oppositely.

Once you finish entering messages, you can return to Monitor mode by holding down three keys simultaneously – SHIFT ↓ and X. There are no other commands in the Terminal mode, and this is the only way to exit. You do not need to press ENTER

The examples above illustrate the use of Terminal Mode between two TRS-80's. However, a more practical application of Terminal Mode is between your TRS-80 and another computer such as an Online System. You'll have to do some special programming to interface your TRS-80 to the other system. Keep in mind the above information about Baud rates, Parity and Stop Bits. Also, you might find the Technical Information Section and Appendix A helpful.

C Command In Monitor Mode

As explained earlier, the Parity and Stop Bits are automatically set as soon as you enter Send or Receive modes. Data is transmitted serially with all eight bits used for data (no Parity) and one Stop Bit. However, you still need to make sure the Baud rate is correct for the modem you are using.

This should be done while still in the Monitor mode using the C command. The value must be the same for both the Send and Receive stations and must correspond to the capability of the modems used.

```
?C ENTER
BAUD=9600? 30 ENTER
```

The above illustrates resetting the Baud rate to 300 to be used with the Radio Shack Telephone Interface II.

Send And Receive Modes

This section includes descriptions of both Send and Receive modes. Since they are always used together, this approach will allow you to get an idea of what it is like to actually use them.

Before entering either of these modes:

1. Make sure your modem is set to FULL DUPLEX and ORIGINATE or ANSWER.
2. Call the answer station to decide on procedures to follow in case of error, the Baud rate to use, and the data to be transmitted.
3. Make sure the Baud rate is set the same for both stations and at a level the modems are capable of handling (using the C command).
4. Make sure that any other equipment you will be using is set up (cassettes, disks).

In transmitting data using these modes, the Receive end must set up first to make sure he receives all the data being sent from the beginning. In the examples, the display for the Receive station will be shown first.

In the following example, a block of memory is to be transmitted at 3ØØ Baud. Both stations are already in Monitor mode, and both modems are set to FULL DUPLEX. The calling station is set to ORIGINATE, and the other station is set to ANSWER.

```
?C ENTER
BAUD=2400? 30 ENTER
```

Both stations should enter the above.

```
?R ENTER
```

When the Receive station enters the above, the screen will clear and the Receive mode display will appear.

```
R? M 6000 ENTER
```

This readys the program to receive data.

After waiting the agreed upon time, the Send station can begin transmission.

```
? S ENTER
```

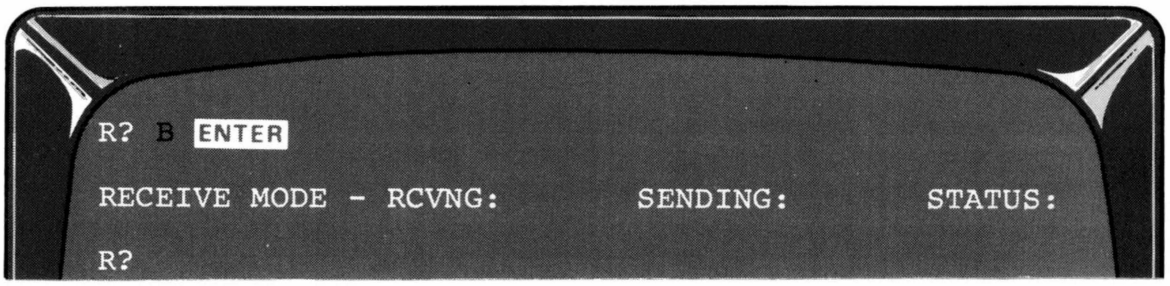
The above will allow you to enter the Send mode to transmit. At this point, you can send the block of memory by keying the appropriate command. If the data is not already in memory, it can be loaded using other commands.

If for some reason you need to terminate a transmission before it is complete, type **C** or **X**. You do not need to press **ENTER** . This can be done during any kind of data transfer by either station. It will return you to Send or Receive mode. In addition, **C** will signal the other station.

In the rest of this section, various types of data transfers will be discussed with the commands used to accomplish them.

Transmitting A Basic Program

Receive Station



The above message appears and remains during the transmission reflecting the status of the transmission and any errors that occur.

When the transmission is complete, the **R?** prompt returns, and the BASIC program is residing in memory. It can be saved on cassette tape in the CSAVE format using the **PB** command.

```
R? PB X ENTER

R?
```

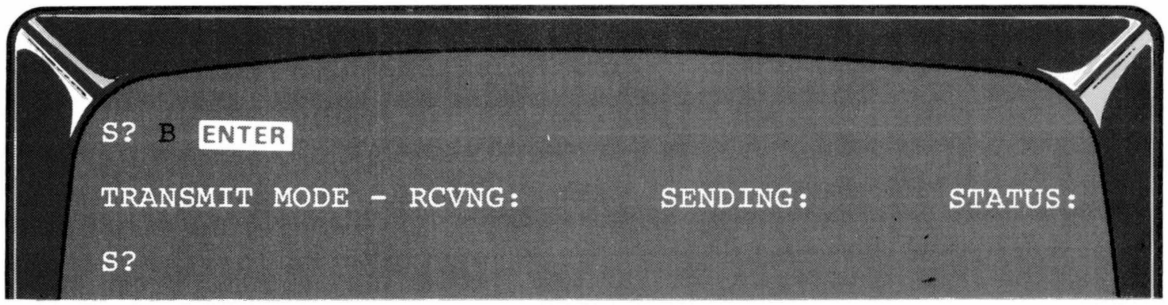
Make sure you turn the cassette recorder on first. *x* is the filename. It can be any one-letter name you wish.

As soon as the recording is complete, the **R?** prompt appears, and you can receive more data or exit to the Monitor mode. If you exit to BASIC, your program will still be in memory and can be executed or edited.

Send Station

```
S? LB ENTER

S?
```

Here you load a cassette tape with a BASIC program on it in CSAVE format. The recorder must already be on and wound down to the beginning of the file. **S?** reappears when loading is complete.



The BASIC program is now being sent from memory on a block-by-block basis. When complete, the **S?** prompt will return, allowing you to exit to Monitor or send more data.

If you have entered the program from LEVEL II and you already had a BASIC program in memory, you can transmit this program immediately. Also, once you have finished your transmission, you can exit to LEVEL II and execute or edit your program.

The only time your BASIC program in memory will be destroyed is if:

- you transmit a data tape after the BASIC program and write over the program;
- you load an object file into memory;
- you receive a data tape;
- you receive a block of memory; or
- you receive a BASIC program.

In these cases, the memory storing the BASIC program will be written over by the other data. If you then try to transmit the BASIC program, the error message **NO DATA FOUND** will be displayed.

Transmitting Object Files

The M command lets you transmit a block of memory specified by a four-digit hex starting and ending address on a block-by-block basis.

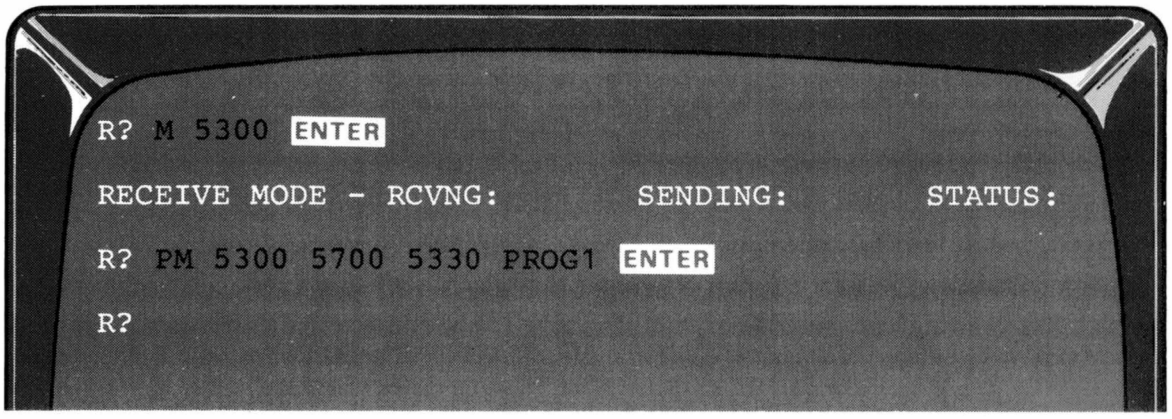
A machine language program stored as a SYSTEM tape (called an object file) can be read from tape into memory and then transmitted using the M command.

Receive Station

Before you try this function, you must be careful to keep one thing in mind. You must not write over BASIC or TRSDOS in low RAM locations, and you must not write over the Communications Program in high RAM locations. The hex assignments at both the Receive and Send stations must not be below 42E9 (for LEVEL II) or 52ØØ (for TRSDOS). Nor can the assignments be above 6FCA, AFCA, or EFCA for 16K, 32K, and 48K respectively.

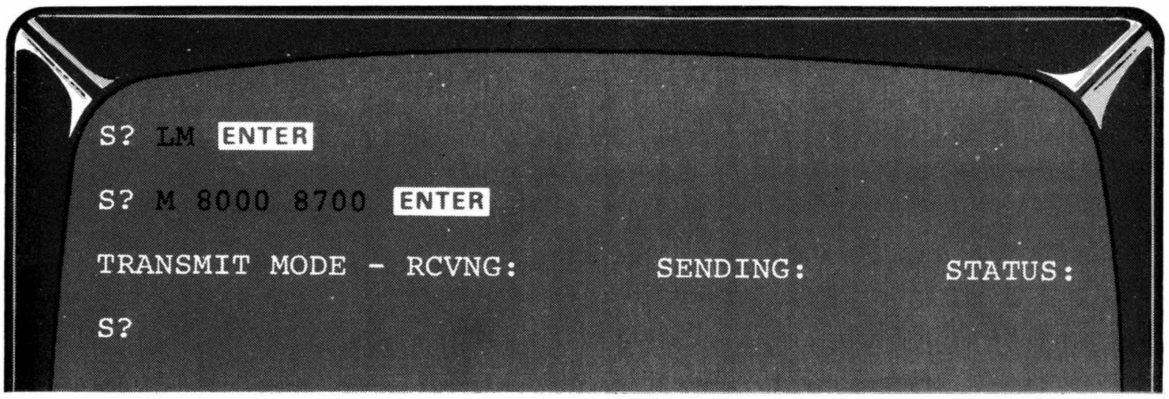
When transmitting a BASIC program (as in the previous section), the program is automatically stored in locations above TRSDOS or LEVEL II. However, if your program is extremely long, it might write over the Communications Program. In this case, the entire program cannot be transmitted from memory at one time, unless it can be sent as a disk file under the TRSDOS version.

To receive a memory block, the Receive station need only specify the four-digit starting address. No ending address is required. After reciving the data, you may then record the block as a SYSTEM tape using the PM command. You will need the ending address and entry point both of which you can get from the Send station.



Data is received starting at hex address 53ØØ. This can be entered without spaces between operands, but there can be no spaces within each operand. When the data is received, the R? prompt returns, and the memory from 53ØØ to 57ØØ is recorded as a SYSTEM tape with the file name PROG1. The ending and entry addresses should be obtained from the Send station before the transmission. When the recording is complete, the R? prompt returns.

Send Station



In this example, an object file is loaded from tape to memory. The memory addresses do not need to be specified since they are on the beginning of the tape. Just be sure they will not interfere with BASIC, TRSDOS, or the Communications Program. The cassette must be ready and positioned at the beginning of the file before loading.

Notice that the memory locations you send from and the memory locations that you receive at need not be the same. This will allow a machine with a larger RAM to transmit to one with a smaller memory size. Of course, in this case, you will have to adjust the ending address and entry point on the Receive end for recording.

Transmitting Data Tapes

A cassette tape produced by the PRINT# command in LEVEL II can be transmitted using the D command. These data tapes consist of blocks of data – each block produced by a separate PRINT# statement. Each block has a header byte (A5 in hex) followed by data followed by an ending byte (ØD in hex).

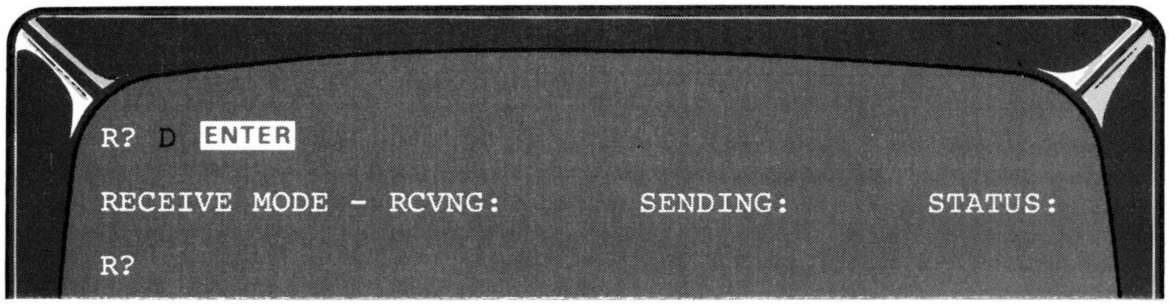
The program will read one block, transmit it, read another block, transmit it, etc., until it reads blank tape for five seconds.

At the Receive end, data is received and recorded one block at a time until transmission is complete.

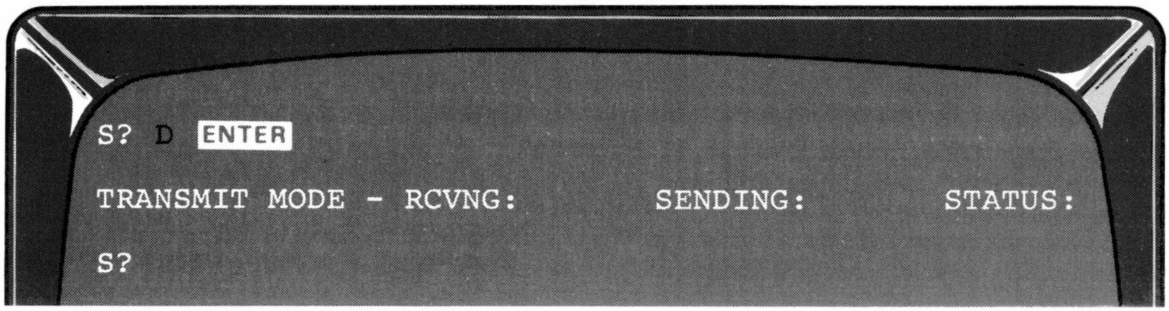
You should have cassettes ready and waiting on both ends before entering this command.

Since there is no end-of-file marker on data tapes, the transmission will continue until there is a time-out at the Send station. This occurs when five seconds pass with no data being read from the tape. If you want to send a data file in this way, be sure it is the only file on that side of the tape. Otherwise, the next file will be transmitted as well.

Receive Station



Send Station



Transmitting A Disk File

This option is only available on the TRSDOS version of the Communications Program. Any disk file (object, data, or BASIC program) can be transmitted. For example, a disk file storing a letter you have keyed in can be transmitted. It can then be displayed and read (or even printed if equipment is available) at the Receive station using the TRSDOS command LIST. Such a file should be named using the TXT optional extension.

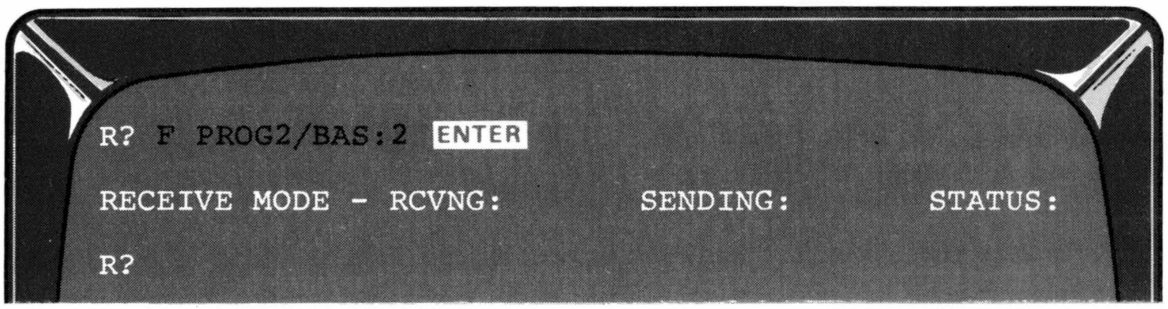
The Send station must specify the disk file name he wishes to transmit according to TRSDOS specifications (filename followed by optional extension, password, and drive spec).

On the Receive end, you may specify any name you wish for the disk file. It need not be the same as the name that the Send station uses.

Disk files are not loaded into memory before transmission. At the Send station, one sector is read from disk and transmitted. At the Receive end, this sector is read and written to disk. This is repeated until the entire file is transmitted.

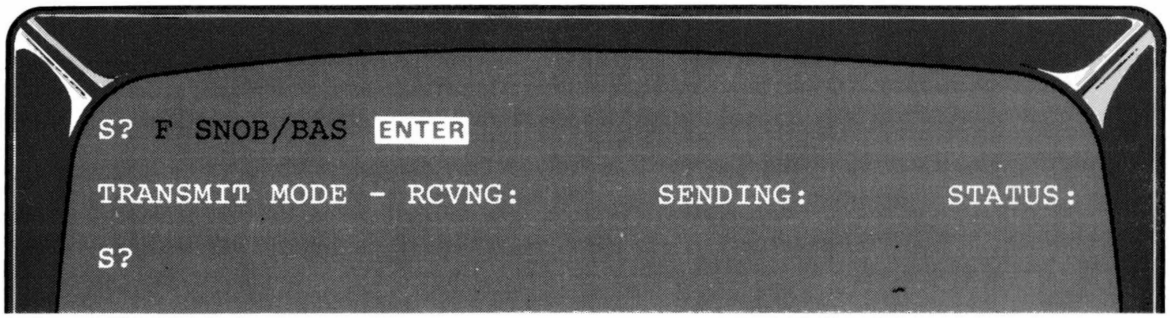
If a disk fault error occurs during a disk read or write, the transmission is terminated, leaving an incomplete file on the Receive end. As soon as you exit to TRSDOS, this incomplete file should be KILLed.

Receive Station



Here a file will be written on Drive 2 named PROG2/BAS with no passwords. If you do not specify a drive, the program will write on the lowest-numbered unprotected drive it locates.

Send Station



Here the file named SNOB/BAS is transmitted. Notice it does not have the same name used at the Receive end.

If you specify a disk drive number, the program will look for the file only on that drive. Otherwise, it starts at Drive Ø and searches until it finds the first file with the proper name. Any optional passwords you used when naming the file must be specified.

Technical Information

The information in this section will be helpful to you in tracing errors and in understanding the status messages from the system during a transmission. Although it is difficult to isolate every problem, the major conditions that can cause errors are included here.

There are basically three kinds of errors that can occur – operator, hardware, or software. The Communications Program has been thoroughly tested to eliminate software errors.

There are many kinds of operator errors. From some of them, you can recover with relatively little loss in time. From others, you will have to reload the program and start over.

For example, you might type in the **E** command in Monitor mode by mistake, and you will exit the program. To recover, you will have to re-execute the program and start from scratch.

But, if you type in the response that is illegal for the mode you are in, the prompt will be repeated until you answer correctly. A minimum of lost time occurs.

If you try to transmit a file without first turning on your cassette, the message **NO DATA FOUND** will be displayed, and you will have to try again.

If your transmission is OK, but you find after it has started that you are sending the wrong file, you can end it by holding down the **C** or the **X** key. You may have to hold the key for a few seconds until the system recognizes it.

Pressing **C** will kill the transmission and send a message to the other station. Pressing **X** only kills the transmission and does not attempt to notify the other station.

Hardware errors can be the result of a faulty diskette, a malfunctioning disk drive, a poor telephone connection, a bad RS-232-C interface, or a bad modem. These errors are usually handled by the system. Most of them cause the transmission to end.

Before entering any communications mode, you should set up procedures that both you and the remote station will follow in case of error. For example, both of you could pick up the receiver and wait until the other station does the same whenever you detect an error.

In the Terminal mode, Parity can be checked to detect errors. If a Parity error is found, a graphics bar is displayed next to the character in which the error occurred. If the incorrect byte that is transferred does not represent a keyboard character, a vertical bar appears.

In the Receive or Send modes, other techniques are used to detect errors. In addition to transmitting data, the Send station transmits certain control bytes and the Receive station checks these and compares them to what they should be.

A part of each block that is transmitted is a two-digit code that tells the Receive end to interpret what follows as data and not as a control message.

Following this is the byte count which consists of four hex digits and is the number of bytes of data that follow. The Receive station uses this to determine the number of bytes it will get before the check sum.

Following the actual data block is the two-byte check sum. The Receive end calculates the check-sum itself on the data and compares the result with this two-byte amount calculated at the Transmit end. (See Appendix A for more information.)

After receiving each block, the Receive end transmits control messages to the Send station telling it whether or not each block was received without error.

In some cases, if there was an error, the Send station will try to re-transmit the same block. In other cases, the transmission will terminate and both stations will be returned to their prompt. These error messages will be displayed in the status message that remains on the video during transmission.

```
RECEIVE  MODE - RCVNG:        SENDING:        STATUS:MOD.FLT
** TRANSACTION ABORTED **

TRANSMIT MODE - RCVNG:        SENDING:DSK.FLTSTATUS:MOD.FLT
** TRANSACTION ABORTED **
```

In addition to error messages, the Receive and Send stations transmit control messages to one another. Once a block is received without error, the Receive station sends a wait acknowledge message (WAK) to the Send station. The Send station will then wait before sending another block to give the Receive station time to place the block just received.

These control messages are also displayed in the status message during transmission:

```
TRANSMIT MODE - RCVNG:        SENDING:BLOCK   STATUS:

RECEIVE  MODE - RCVNG:BLOCK   SENDING:WAK     STATUS:
```

Here, the Receive station is getting a block and sending a wait acknowledge control message back to the Send station.

Error Messages

Message	Meaning
DSK FLT	• Disk read or write error. Transmission will end, returning both stations to their prompts.
MODEM FLT	• The phone is off the coupler so no carrier is detected, • The phones have been disconnected (Lift receiver and listen for tone), • The phone connection is so bad that no messages can get through, • Both modems are set to ORIGINATE or both to ANSWER (Check switch settings.), • Modem is set to HALF-DUPLEX at one or both stations (Check switch settings.), or • Modem is powered off or cable is disconnected. In all cases, transmission will end, returning both stations to their prompts.
COMM FLT	• Unspecified communication fault.
ILL RQST	• Meaningless request.
CHK SUM FLT	• An error in the data. The check-sum as computed by the Receive station does not agree with the amount calculated by the Send station. The same block is re-transmitted.
BC FLT	• The byte count received does not agree with its ones-complement check-sum computed by the Receive station. Transmission will end, returning both stations to their prompts.
???	• Unknown message has been received.

If no error message occurs, you may still be suspicious that no communications are taking place. (There may be no status messages either.) Remember, it takes nine seconds to transfer a 256-byte block at 3ØØ Baud, so after this time, if nothing happens, you may have a problem.

1. The Baud rate may not be set the same at both stations.
2. The telephone connection may be too noisy for any communications.
3. There may be a problem with the RS-232-C hardware.

If you suspect such a hangup, you may cancel the transmission from

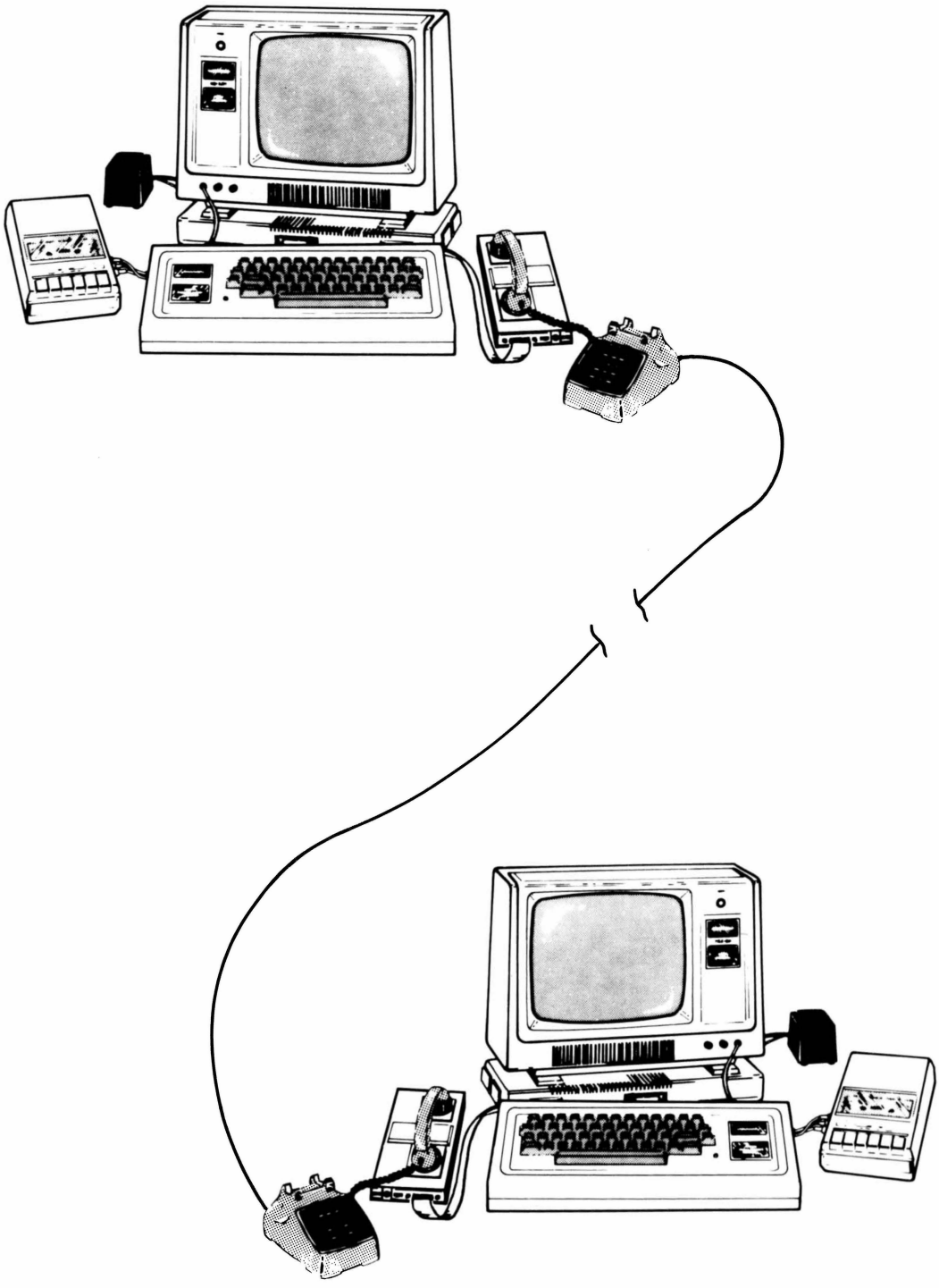
Control Messages

Message	Meaning
ENQ	• Enquiry. Are you there?
ACK	• Unconditional acknolwedge. The Receive station can send this to let the Send station know another block can be transmitted.
BLOCK	• A block is being sent or received.
WAK	• Wait acknowledge. The Receive station can send this to let the Send station know that it got the block but wait before sending another.
EOF	• End-of-file. The Send station transmits this at the end of the data to indicate a successful transmission.
NAK	• Negative acknowledge. The Receive end can transmit this if it detects certain errors in the data (such as a check-sum error) so the Send station can try re-transmitting the block.
CANC	• Cancel. The transmission has been terminated for some reason.
OK COMPL	• This is displayed in the status message at the Send station when the block has been transmitted without error (when the Send station receives the ACK message).

either station. Hold down the C or X key until the system recognizes it and kills the transmission. C will notify the other station before ending the transmission, so use it only if you can get through the line. X should be used if the system is hung up, and no messages can get through.

If you detect a problem, but cannot isolate the cause, you might try reloading the Communcications Program and starting over. The flow-chart in Appendix C might also be useful in troubleshooting.

Appendix



A/Custom Applications for COMPROG

Although Radio Shack cannot guarantee other manufacturers' equipment and cannot supply any special programming that may be necessary to support it, the following items are provided, without obligations to Radio Shack, to facilitate such work.

1. There is much valuable reference information in the **TRS-80 RS-232-C Interface Manual**, and it should be used.

2. It is possible to make direct RS-232-C connections in case telephones and modems are not used. In this case, it is necessary to solder pins 2Ø, 6, 5 and 8 (DTR, DSR, CTS, and CD respectively) together on the DB25 connector at both Interface Boxes.

This makes the RS-232-C think there is a modem connected and turned on (DSR), receiving a carrier tone (CD), and clear to send (CTS). Refer to the **TRS-80 RS-232-C Interface Manual** for illustrations.

3. The Send and Receive modes are made up of two major routines (RCV and XMIT) which are called by the Data Access routines at both station

The Receive station operator enters a command first to receive a block of data. This causes the data access routine to set the A register to Ø, and to put the buffer address in the HL register pair. RCV is then called and awaits the block.

At the Send station, the A register is set to Ø, the HL register pair is given the buffer address, and the BC register pair stores the block size.

If the communications link and equipment check out as OK, transmission begins, each block prefaced with a small header indicating whether it is a block of data or a control message and the number of bytes contained in the data. The data field follows this header and is followed by a check-sum which RCV can compare to the check it calculates on the data field.

Both stations transmit status or control messages during the communications which are displayed on the video in the status message. Refer to the Technical Information section for the meanings of these messages.

After successfully receiving a block, RCV sends XMIT a WAK (wait acknowledge control message). This means that the data has been received OK but RCV is not ready for another block yet.

This allows RCV the time to tell the data access routine that a block has been received and to give its size (from the BC register pair). The data access routines then place the block appropriately (disk, tape, or memory) and recall RCV which sends ACK to XMIT.

Dialog continues in this way until all the data is transferred. Then, XMIT sends the E0F control message to RCV and control is returned to the data access routines at both ends.

4. Formats for blocks of data follow:

7E ØØ	*byte count*	*byte count check*	*data*	*check sum*

a. The 7E in hex is a synchronizing byte that indicates something follows.
b. The ØØ is the code that indicates a block of data follows and not a control message. (See below for control message codes.)
c. The byte count (two bytes) is the number of data bytes.
d. The byte count check is a ones complement check-sum of the byte count.
e. The data consists of an arbitrary block of eight-bit data bytes.
f. The check-sum is a 16-bit (two byte) cumulative addition of all the data bytes only.

5. There are a number of control messages all of the following format:

7E *Code*

The codes are listed below:

Code	Mnemonic
ØØ	BLOCK
Ø1	ENQ
Ø2	ACK
Ø3	WAK
Ø4	EOF
Ø5	NAK
Ø6	CANC
Ø7	MODEM FLT
Ø8	COMM FLT
Ø9	ILL RQST
1Ø	???
11	DSK FLT
12	CHK SUM FLT
13	BC FLT
14	OK COMPL

The meanings of these are discussed in the Technical Information section.

6. Three types of program access are required to use the RS-232-C hardware:

a. Initialize where, first, the UART is reset, second, the Baud Rate Generator is set to the desired Baud rate, and, third, the UART and Modem Control Register is set to the proper number of bits/word, Parity or not, Parity type, modem control, etc.

b. Receive incoming bytes where, first, the status of the UART must be checked to see if a character has been received (If bit seven of the I/O address, EA in hex or 234 in decimal, is set (equals one), then a character has been received); and, second, a character must be read from the UART data register at I/O address EB in hex or 235 in decimal,

c. Transmit a byte where, first, the status of the UART is checked to see if the previous output character has been transmitted before the next is loaded (If bit six of I/O address EA in hex or 234 in decimal is set (equals one), then the new byte may be loaded.); and, second, the character is output to the UART data register at I/O address EB in hex or 235 in decimal.

7. In BASIC, the initialization, which is done one time at start-up, could be accomplished by the following:

```
10 OUT 232,0
20 OUT 233,85
30 OUT 234,164
```

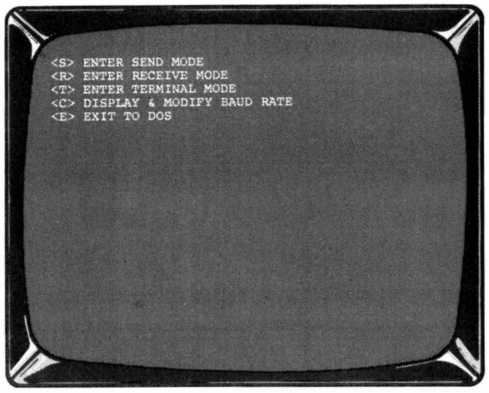
This resets the UART, sets the Baud rate at 3ØØ (85 in decimal or 55 in hex is the code for 3ØØ Baud.) and sets Parity even with one Stop Bit and DTR and RTS asserted on modem lines. (164 in decimal or A4 in hex is the code for this.)

8. To input ASCII from the serial port and display it on the screen:

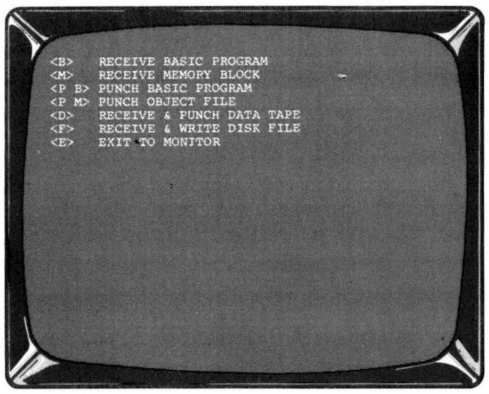
```
40 X=INP(235)
50 IF X=128 THEN GOTO 40
60 X=INP(235): PRINT CHR$(X);:GOTO 40
```

B/Chart Of Options for COMPROG

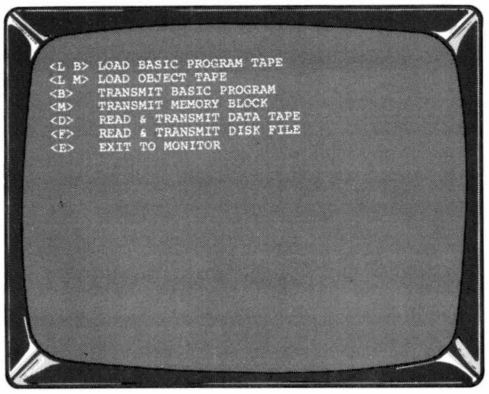
Monitor Mode



Receive Mode

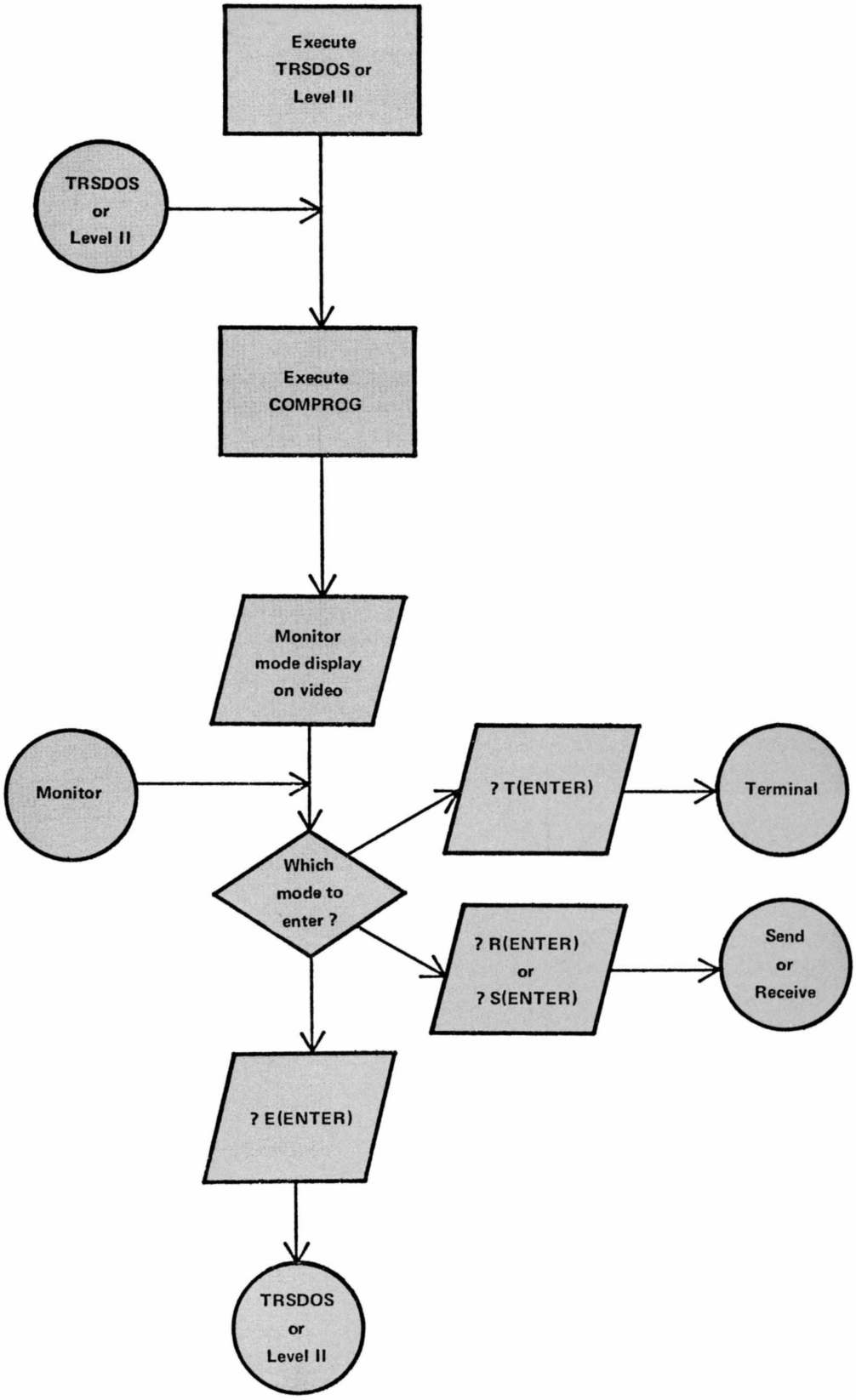


Send Mode

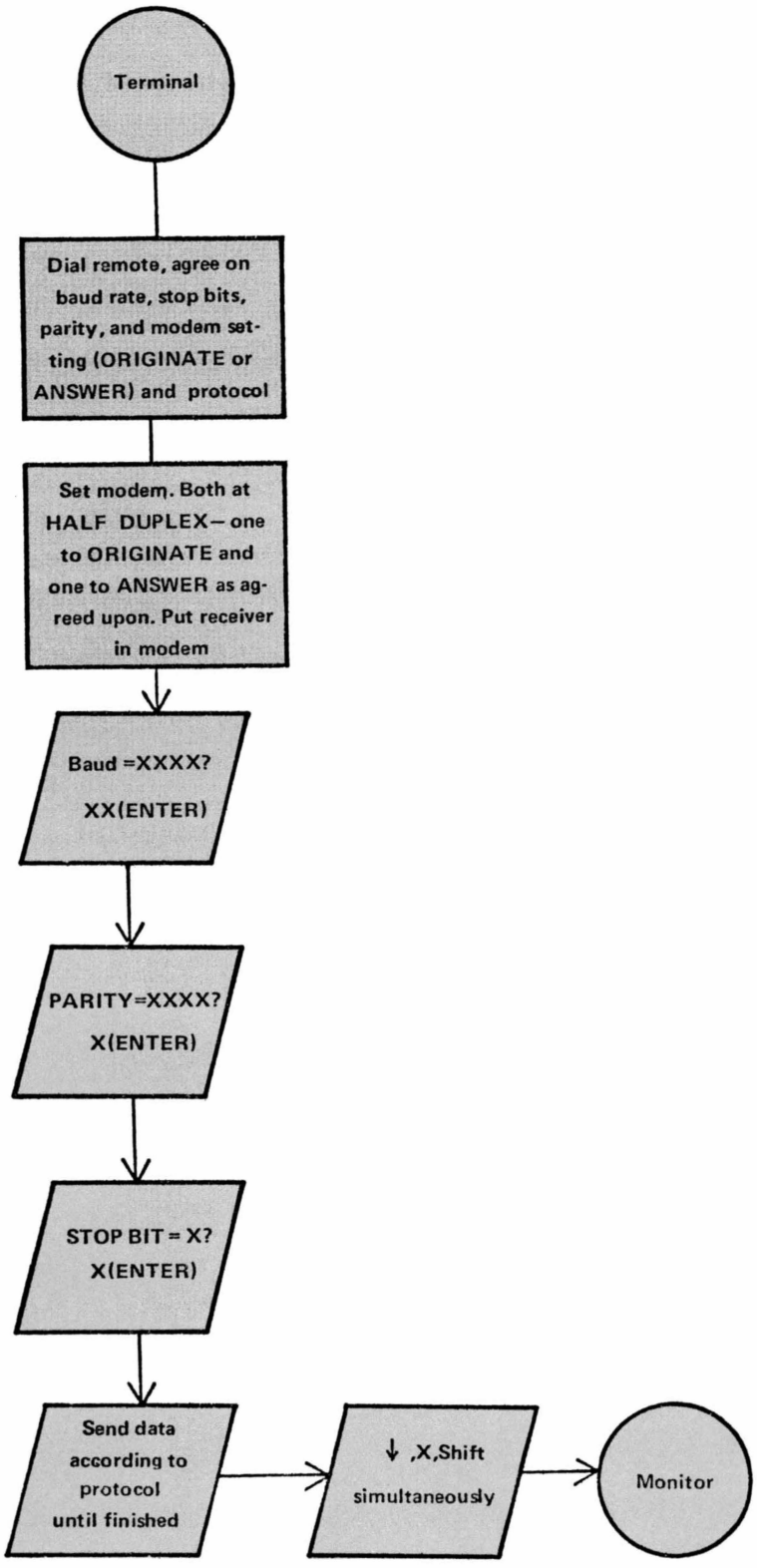


C/Operational Flowcharts for COMPROG

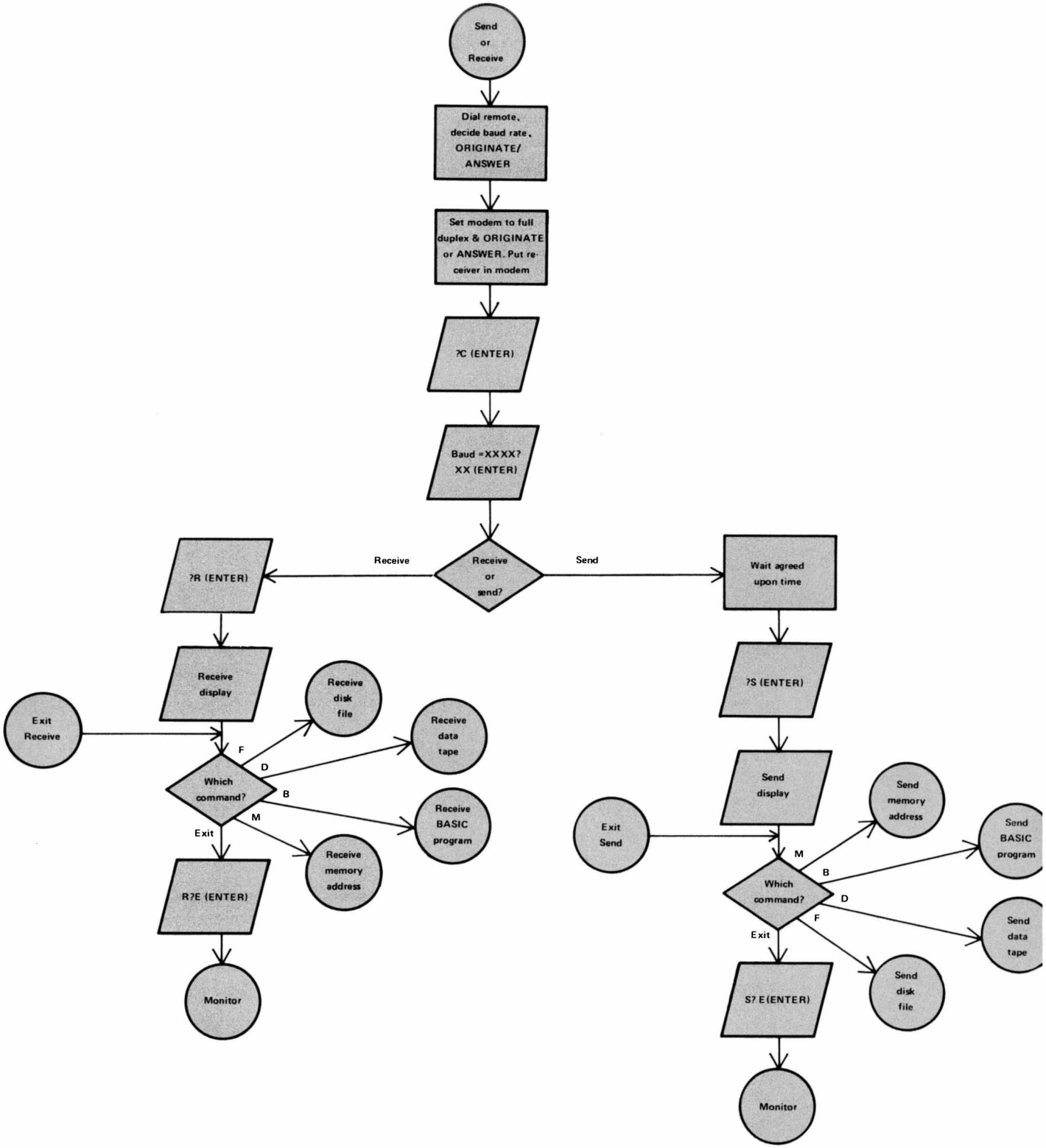
On the following pages are charts you can refer to for the operation of COMPROG. They include all the steps to follow to perform the functions in COMPROG. Some of you might find them useful while reading the detailed operating explanation or later on for reference; but it's not necessary to use them in order to run the program.



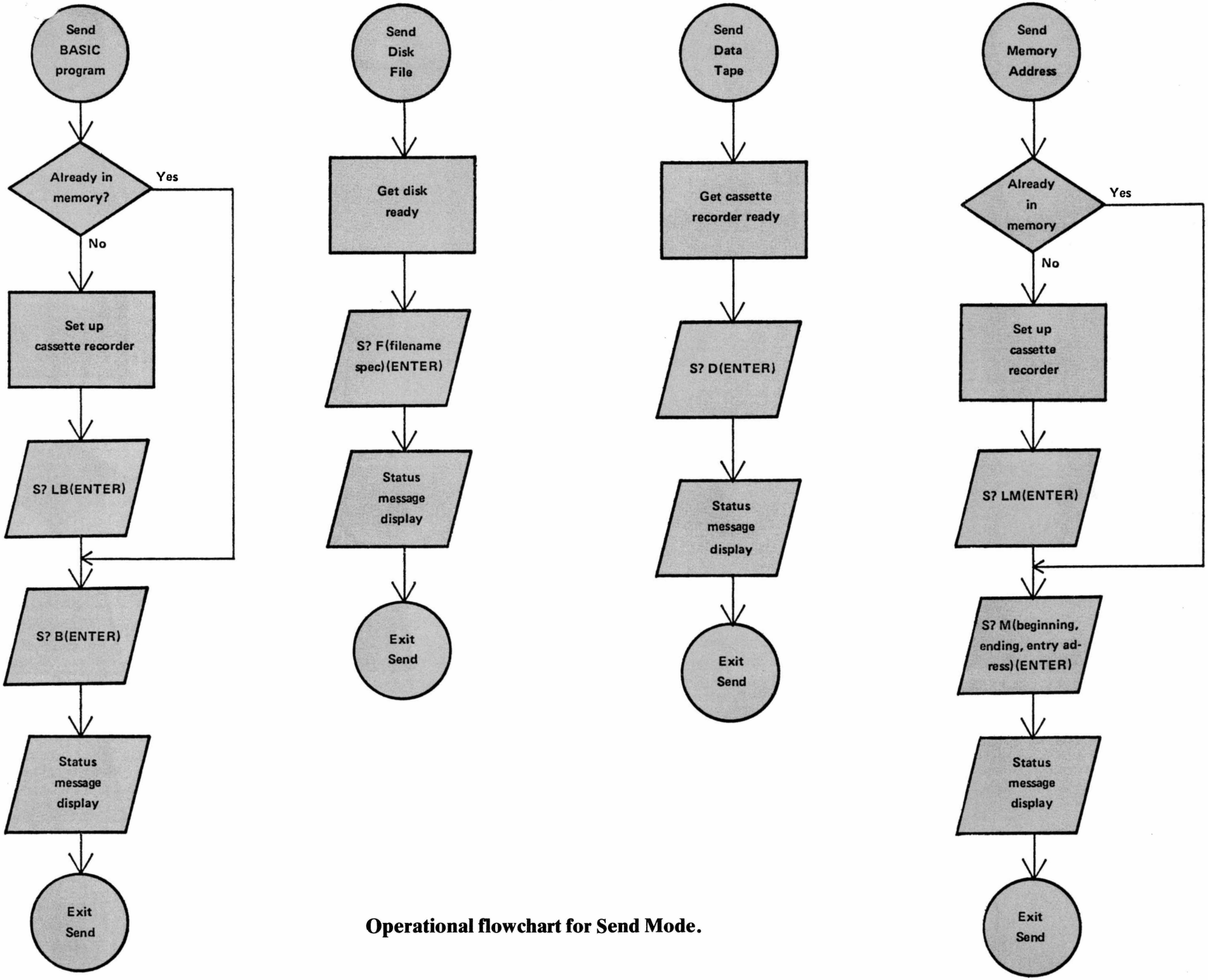
Operational flowchart for Monitor Mode.



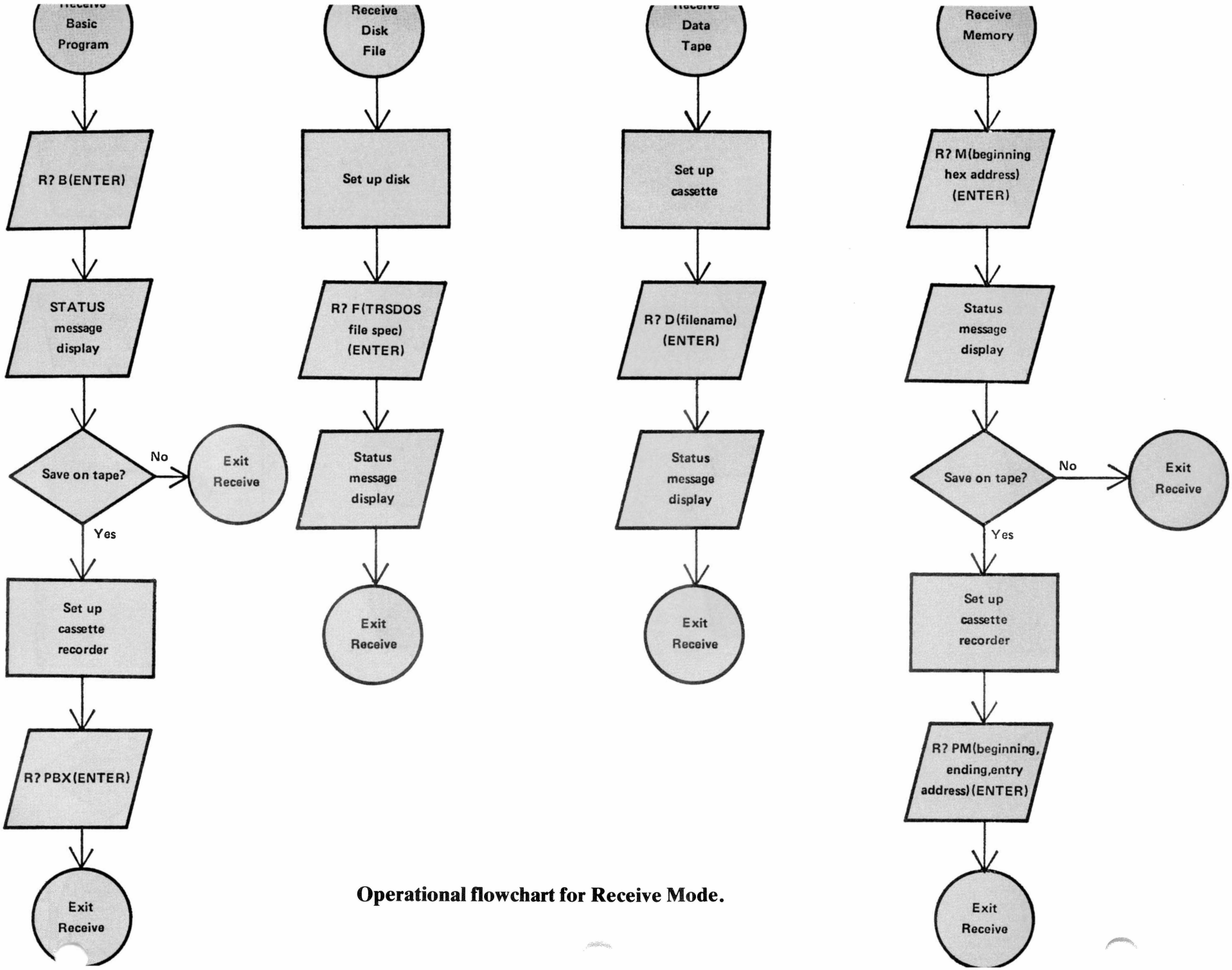
Operational flowchart for Terminal Mode.



Operational flowchart to enter Send or Receive Modes.



Operational flowchart for Send Mode.



Operational flowchart for Receive Mode.

IMPORTANT NOTICE

ALL RADIO SHACK COMPUTER PROGRAMS ARE DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTY

Radio Shack shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by computer equipment or programs sold by Radio Shack, including but not limited to any interruption of service, loss of buisness or anticipatory profits or consequential damages resulting from the use or operation of such computer or computer programs.

NOTE: Good data processing procedure dictates that the user test the program, run and test sample sets of data, and run the system in parallel with the system previously in use for a period of time adequate to insure that results of operation of the computer or program are satisfactory.

RADIO SHACK A DIVISION OF TANDY CORPORATION

U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

TANDY CORPORATION

AUSTRALIA	BELGIUM	U. K.
280-316 VICTORIA ROAD RYDALMERE, N.S.W. 2116	PARC INDUSTRIEL DE NANINNE 5140 NANINNE	BILSTON ROAD WEDNESBURY WEST MIDLANDS WS10 7JN

PRINTED IN U.S.A.

479