

LYNC 2.0
Data Communications Program
User's Manual

LYNC 2.0
program
Copyright 1980,81
by
Midnight Software

LYNC 2.0 User's Manual
Copyright 1981
by
Pickles & Trout
P.O. Box 1206
Goleta, CA 93116
(805) 685-4641

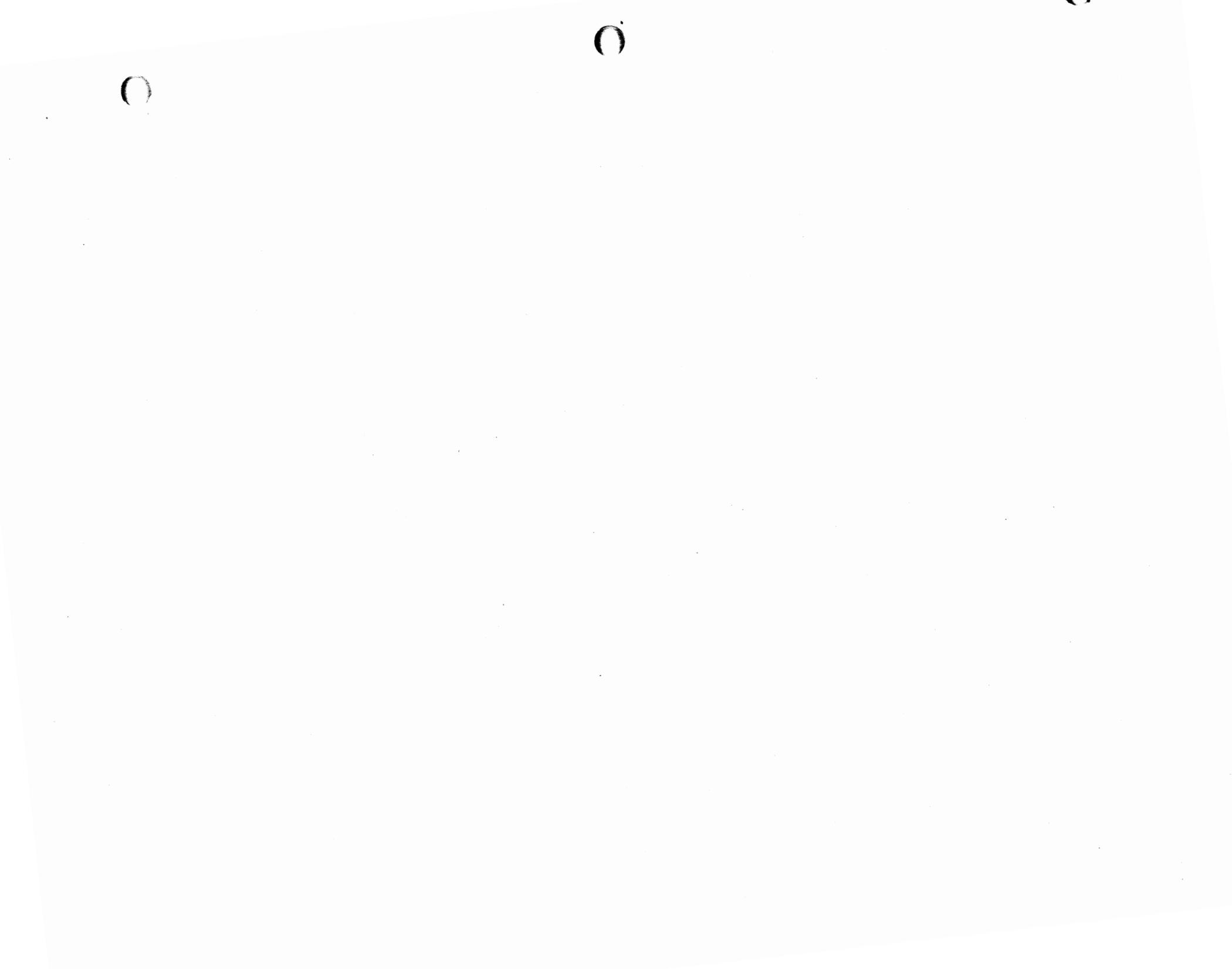
All rights reserved. No part of this manual may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, magnetic, optical, chemical, manual or otherwise, without the prior written permission of PICKLES & TROUT.

PICKLES & TROUT and MIDNIGHT SOFTWARE have made a reasonable effort to insure that the computer program described herein is correct and operates properly; however it is sold without warranties either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. PICKLES & TROUT and MIDNIGHT SOFTWARE will not be liable for consequential damages resulting from the use of this product either individually or in concert with other computer programs. Further, PICKLES & TROUT and MIDNIGHT SOFTWARE reserve the right to revise this manual and the program described in it and to make changes from time to time in the contents thereof without obligation of PICKLES & TROUT or MIDNIGHT SOFTWARE to notify any person of such revision or change.



TABLE OF CONTENTS

INTRODUCTION	1
NOTATION	2
GETTING STARTED	3
LYNC OPERATION	9
Modes of Operation	9
Control Key Functions	14
CUSTOM I/O ROUTINES	19
APPENDIX A	22
Files on the Distribution Disk	
APPENDIX B	23
ASCII Character Set	
APPENDIX C	24
Customization Parameters for Specific Hardware	



NOTATION

The following notation is used in this manual:

When a dialog between a user and the computer is being represented, all output from the computer is shown in normal type. Characters that are typed by the user are shown underlined.

When referring to a key on the keyboard that has a name printed on it the name is enclosed in angle brackets (eg. <carriage return>, <tab>). The symbol <tab> means to press and release the key which bears the legend TAB, not to type the 5 characters < t a b >. Some computer systems have an <enter> key which is equivalent to the <carriage return> key on other systems. In this manual, the <carriage return> or <enter> key is referred to by the symbol <cr>.

Control characters are referred to as follows: <ctl-A>. When a control character is to be typed, this symbol is shown underlined as follows: <ctl-A>. A control character is typed by holding down the control key (like the shift key) while typing a key on the keyboard.

In many figures line numbers are shown at the beginning of each line of the figure. These line numbers consist of one or more digits followed by a colon (eg. 14:). The line numbers are provided so that the written explanation can easily refer to a line number within a figure. All figures are numbered in their captions. For figures without line numbers, and when figures with line numbers are being referred to as a whole, the notation used is as follows: figure 12. Reference to a specific line within a figure is made by prefixing the line number with the figure number and a dash (eg. line 12-14 refers to line 14 of figure 12).

When LYNC is in use there are almost always two computers involved. When this manual refers to the local machine it means the machine with which you are working directly, ie. the machine on whose keyboard you can type. When this manual refers to the remote machine it means the machine connected to the other end of the serial communications link. Typically LYNC will be running at each end, however the remote machine might be a time-sharing system or some other computer which does not have LYNC available.

In the text of this manual, a hexadecimal (base 16) number will be denoted by following it with the letter 'h'. When a figure is illustrating a dialog between the user and the computer and a hexadecimal number is to be entered, the trailing h will not be shown since it is not typed when entering the number. Numbers shown in the text without the trailing h are decimal (base 10).

INTRODUCTION

LYNC is a data communications program which is optimized for file transfer between two micro-computers. LYNC requires a single serial port (either half or full duplex) on each computer. Data transfers take place with a full error checking protocol which permits reliable data transfer even in the presence of noise or other interference. LYNC's error checking and correction techniques are so reliable that the physical connection between two computers can be broken in the middle of a transfer and then re-connected without loss of any data. The link between the two computers can be by direct wire or modem - LYNC will work reliably up to 9600 baud.

LYNC is the ideal communications program for connecting micro-computers with incompatible disk formats, making it easy to transfer both text and program files between the machines. It also makes it possible for a software supplier to support his customers via the telephone. Changes to programs can be sent over the phone lines without the necessity of driving across town or waiting for the mail.

Before using LYNC, it is necessary to customize it for your computer system. The GETTING STARTED section of this manual describes the customization process in detail. In order to perform its communications functions, LYNC must be able to determine when an incoming character from the remote machine is available and when a character can be sent to the remote machine. Of course LYNC must also be able to send and receive characters.

Unfortunately there are no standard system calls under CP/M which provide serial I/O with full status checking. LYNC, however, is capable of generating its own software routines to access a serial port. This includes the generation of status check and initialization routines in addition to the input and output routines. The user has only to supply the proper port addresses and status bits. A self check of the port operation is then made, and you are ready to communicate with other computers.

Although you may use any combination of stop bits and parity (you must use all 8 data bits!), we recommend the following for general compatibility:

8 data bits
1 stop bit
no parity

Note that parity checking is not necessary since built-in checksums are used in data transfer. When LYNC is used in its full protocol mode it insures accuracy of the data by transmitting the data in 128 byte blocks with a record number and checksum. The receiving end verifies the checksum and checks that the record number is the next in sequence. If the receiving end determines that an error has occurred in transmission (ie. checksum error or record number not in sequence) it requests a re-transmission. It should be noted that a checksum alone is insufficient - it would not show if a record was missing entirely or was sent twice.

GETTING STARTED

Be sure to make a working copy of the files on the distribution diskette before using LYNC. You should never make modifications to the distribution diskette since it is your ultimate backup copy.

When using LYNC on a system that has a serial terminal for the console, it is usually necessary for the baud rate to the terminal to be greater than (or at least equal to) the baud rate used for sending and receiving data with LYNC. If this is not the case some characters will most likely be lost when messages are displayed on the console. This is particularly noticeable in the [TERM] mode and when a remote directory is displayed in the [LYNC] mode.

Next, connect your serial output DIRECTLY BACK TO THE SERIAL INPUT (this is usually pins 2 and 3 on your RS-232 connector). This is necessary for the self test feature of the customization procedure. Also, be sure your serial port is active - the baud rate and status lines (like CTS) should already be set up.

After you have transferred the LYNC files to a working disk, set up your serial port, and connected the output of the serial port to its input, log onto the working disk and execute the LYNC customization program by typing the following command while in the CP/M command mode:

```
1: A>GENLYNC<cr>
```

Figure 1

LYNC will sign on with the following display:

```
1: LYNC Data Communications Program
2: (C) 1980,81 Midnight Software
3: version: 2.2 serial # 1-0000-0001
4: -----
5: : :
6: : dealer name appears here :
7: : :
8: -----
9: This program is for use exclusively by:
10:
11: user name appears here
12:
13: Data port = 00:
```

Figure 2

Lines 2-1 to 2-3 identify LYNC and give the program revision number and serial number. Lines 2-4 to 2-8 identify the dealer through whom LYNC was purchased and may contain a message from the dealer. Your name should appear on line 2-11. If your name does not appear on line 2-11, you may not have a legitimate copy of LYNC - PLEASE CONTACT PICKLES & TROUT AT ONCE.

Line 2-13 begins the customization dialog. It is necessary to customize LYNC to match your computer hardware. For most computer systems, the self-generating I/O feature of LYNC will be sufficient. The self-generating I/O routines are designed to operate with a system

where data input and output is done at one I/O port and serial port status is available at another I/O port. A single bit of the byte read from the status port should indicate when a character can be sent and a different single bit should indicate when a received character is ready to be read.

If the serial I/O hardware in your computer system does not conform to these specifications, you must install custom I/O drivers in order to use LYNC. If you have a hardware system which requires the direction of the data transmission lines to be switched you will need to install at least one custom routine to perform the switching. You should read both this section and the CUSTOM I/O section of the manual thoroughly before continuing with the customization process.

This customization need be done only once since a copy of the customized LYNC can be saved on diskette for future use. The customization dialog is shown in figure 3.

```
1: Data port      = 00:f4<cr>
2: Stat/cont port = 00:f6<cr>
3: Recv ready bit = 00:1<cr>
4: Xmit ready bit = 00:4<cr>
5: Number of drives = 02:3<cr>
6: Video backspace = 08:<cr>
7: Power up mode  = 00:<cr>
8: Term mode exit = 01:<cr>
9: View cont char = 00:<cr>
10: System clock  = 02:4<cr>
11: Init mode     = 00:<cr>
12: Usart init bytes =
13: :<cr>
14:
15: Alter I/O (Y/N) ? Y<cr> - PORT OK
16: Save on disk (Y/N) ? Y<cr>
17: Filename.typ = LYNC.COM<cr>
18: Linking
19: ..... OK
20: [LYNC]
21:
```

Figure 3

All the customization information is entered as hexadecimal numbers. Either upper or lower case is acceptable unless otherwise stated in the following description. A leading 0 is not required when a value from 0 to F is entered. Note that the previous value of the customization information is part of the prompts. When you run GENLYNC the previous values should be as shown in figure 3. This dialog can be initiated while running a configured LYNC so that the parameters can be modified if necessary. In this case, the existing parameters will be shown in the prompts. Any parameter can be left unchanged by typing only <cr> without any preceding characters.

Line 3-1: This line requests the data port which will be used for sending and receiving characters. Refer to your system documentation for this information. This port is often referred to as the DATA PORT in serial interface literature.

Line 3-2: This line requests the serial port status port associated with the data port specified on line 3-1. This port is often referred to as the STATUS PORT in serial interface literature.

Lines 3-3 and 3-4: Typically there is one bit within the byte of data read from the status port which will indicate that a data byte is ready to be input and another bit which indicates when a byte of data can be output. Line 3-3 specifies which bit indicates that a

data byte is ready to be read and line 3-4 specified which bit indicates that a data byte can be sent out. The bits are specified in the form of a mask byte. Figure 4 shows the value of the mask byte for each bit within the status byte. Refer to your system documentation to determine which bits are used for each of these functions.

1:	Bits of a byte are numbered as follows:	
2:	most signif ->	7 6 5 4 3 2 1 0 <- least signif
3:		
4:	<u>Bit Number</u>	<u>Mask Byte (hex)</u>
5:		
6:	0	01
7:	1	02
8:	2	04
9:	3	08
10:	4	10
11:	5	20
12:	6	40
13:	7	80

Figure 4

Line 3-5: The number of disk drives on the system is requested on this line. LYNC will allow a user to log onto any valid drive on the computer system for file transfer. This parameter specifies the number of valid drives on the system; LYNC assumes that all drives from A up to the number specified are available. For example, if you specify 5 drives, LYNC will allow a user to sign onto drives A, B, C, D, and E. If your computer system has non-contiguous drive assignments, you must specify the number of the last logical drive on the system. For example if your last logical drive is F, you must specify 6 drives if you are to have access to drive F. LYNC will not prohibit an attempt to access a non-existent drive on a system that has non-contiguous drive assignments if that non-existent drive comes before the last logical drive. By specifying 0 on line 3-5, LYNC will be able to access only the drive that was the current default drive when LYNC was executed.

Line 3-6: This line requests the character that causes your video terminal to backspace one character. The backspace may be either destructive or non-destructive. Under most conditions you may leave this value as-is by just hitting <cr>. If your terminal does not use the standard backspace character (08) then you will need to enter the value of the character it requires (5F or 7F is sometimes used for backspace).

Line 3-7: You have the option of having LYNC begin execution in one of three modes, [LYNC], [TERM], or [TERM] with half duplex. These various modes will be discussed in the LYNC OPERATION section of this manual. The first time you customize LYNC the [LYNC] mode is a good choice. The initial mode of operation is specified by entering a number on line 3-7. The numbers to enter for each of the modes are shown in figure 5.

1:	to begin execution	enter this
2:	<u>in this mode</u>	<u>number</u>
3:		
4:	[LYNC]	0
5:	[TERM]	1
6:	[TERM] half duplex	2

Figure 5

Line 3-8: When in the [TERM] mode, a single control key is used to exit the [TERM] mode and display a menu of available options. Since this control key is used to direct the operation of LYNC, it cannot be sent out to the remote machine in the [TERM] mode. Line 3-8 allows you to specify which control character you want to use for exiting from the [TERM] mode. You should choose a key that you will not need when running in terminal mode (this choice will be dictated by the remote machine you will be working with). LYNC defaults to <ctl-A> for the terminal mode exit character but you may change it if you need to send a

<ctl-A> to the remote machine. Since the terminal mode exit character must be entered in hexadecimal form, an ASCII to hex conversion chart is provided in appendix B.

Line 3-9: In [TERM] mode LYNC provides an option to display all control characters (except carriage return and line feed) coming in from the remote machine as a caret followed by the corresponding printing character. For example, <ctl-Z> would be displayed as ^Z when the view control character option is in effect. When the option is not in effect, all control characters are sent to the system console as control characters. You may have LYNC begin execution with the control characters viewable by entering 1 on line 3-9 or you may have LYNC begin execution with the control characters sent directly to the system console by entering a 0 on line 3-9.

Line 3-10: On this line you are requested to specify the system clock rate. LYNC uses this information for generating internal delays when necessary. The number you enter here should be the clock frequency of your system in Mhz rounded to the nearest integer. This number is not critical; rounding it to an integer will not affect the performance of LYNC.

Line 3-11: This line requests you to specify what type of initialization (if any) LYNC should perform on the serial I/O port before sending and receiving data. Four initialization modes are available as follows: Initialization mode 0 will cause no initialization to be done. Mode 1 sends a string of bytes out to the status and control port specified in line 3-2. Mode 2 is for systems that require a short initialization program to be run to set up the serial ports. Mode 3 is for use in systems that require initialization bytes to be sent out to a port other than the status/control port or to more than one port.

Consult your system documentation to determine if your serial port needs any initialization before it is used. If not, or if the operating system initializes the port for you, enter a 0 on line 3-11. If initialization is required, enter 1, 2, or 3 depending on the type needed.

Line 3-12: After specifying the type of initialization on line 3-11, LYNC will show you the current initialization bytes on line 3-12. For an uninitialized copy of LYNC there are no default initialization bytes, hence none are shown.

Line 3-13: On this line LYNC allows you to specify the initialization bytes to be used for serial port initialization. If you specified initialization mode 0, simply press <cr> at line 3-13 to continue.

If you specified initialization mode 1 you will need to provide the string of bytes to be sent out to the serial port status/control port (specified in line 3-2). Consult your system's operating manuals to find out what string of bytes is needed to accomplish initialization. To specify the initialization sequence, enter the bytes to be output to the status/control port separated by a space.

If you have specified initialization mode 2 you will need to provide the object code of a subroutine to be executed to initialize the serial port. Since you have no control over where LYNC will store the program in memory, it must be written in a position independent form. This means that you should avoid absolute jumps and calls within the subroutine (note: absolute jumps or calls to system entry points and Z-80 relative jumps are OK). The subroutine you write should terminate with a return (RET) instruction. Consult your system documentation to determine the subroutine that is needed to initialize the serial port and write out the program; then hand assemble the program to obtain the object code. Enter the machine code one byte at a time with each byte separated from the next by a space.

If you have specified initialization mode 3 you will need to provide a set of port number/data byte pairs. The first byte of each pair determines the output port to which the second byte will be sent. Consult your system documentation to determine the initialization bytes that need to be sent out and the ports to which they must be sent. Organize this

information in the order in which the bytes are to be sent with each byte preceded by the port number to which it is to be sent. Enter this list of bytes on line 3-13 separating one byte from the next by a space. Figure 6 gives an example of how this information is entered.

```

1: to send these bytes to the indicated ports:
2:
3:      port           byte
4:      D7             34
5:      D9             7F
6:      E5             1C
7:
8: enter the following for the Usart init bytes:
9:
10:      D7 34 D9 7F E5 1C<cr>

```

Figure 6

Line 3-15: After you have completed the specification of the initialization bytes, LYNC will ask if you want to alter its I/O drivers. Check over the information you have entered at this time and if it is all correct, respond with Y<cr>. If you have made a mistake or were merely checking over the parameters that were entered, enter N<cr>. Note that the response given on line 3-15 must be upper case. Entering a lower case response or one that cannot be understood by LYNC will cause LYNC to print a question mark on the console and wait for further input. If you give an affirmative answer on line 3-15, LYNC will generate and install the necessary I/O routines and will test the serial port to make sure the routines work properly.

If everything checks out to be working properly, LYNC will respond with "**- PORT OK**" as shown on line 3-15. If the testing procedure does not produce satisfactory results, you will receive one of several error messages.

TRANSMIT BAD is the message given if the Xmit ready bit does not seem to respond. This usually occurs if you have specified the Xmit ready bit mask improperly or have specified the wrong port as the status/control port.

NO RECEIVE DATA indicates that when LYNC sent out a test character to the serial port it did not receive anything back. Usually this means you forgot to connect the serial data output back to the serial data input for the setup and test. This message can also result from incorrectly specifying the data ready bit mask or from specifying the wrong port for the status/control and/or data port.

RECEIVE DATA BAD indicates that the test character received from the serial port was not the same as the character which was sent out. When this message occurs, it might indicate a bad connection from the serial output to the serial input or an incorrect setup of the serial port such as having the transmit and receive baud rate set to different values.

If you get an error be sure that you have the serial output connected back to the serial input. If they are connected together, consult the hardware documentation on your serial port to be sure you have specified the correct parameters. If your hardware does not talk to the serial port with a simple status port and data port, you may need to install your own I/O routines (see the section on CUSTOM I/O).

Line 3-16: If the I/O test is successfully completed, LYNC will ask you if you want to save a customized version of LYNC on disk. If you are merely making a temporary change to the configuration parameters, you may want to give a negative response at this point. A negative response will cause LYNC immediately to begin execution in the power up mode you specified in line 3-7. Usually, however, you will want to save the customized LYNC on disk; enter Y<cr> on line 3-16. Note that the response given on line 3-16 must be upper case. Entering a lower case response or one that cannot be understood by LYNC will cause LYNC to

print a question mark on the console and wait for further input. Once the customized version is saved you will not need to go through the customization process again.

Line 3-17: If you indicate that you want to save the customized version of LYNC on disk you will be prompted for a file name as shown on this line. Enter the complete file name (including the file type extension). LYNC will store the customized file on the drive that was currently logged on when the program was executed. Note that only files with the extension .COM are considered to be executable by CP/M; you will probably want to use a .COM extension.

After you have specified the file name LYNC will save the customized version of the program and then will begin execution in the power up mode specified on line 3-7. In figure 3 the LYNC mode was specified for power up and lines 3-18 to 3-21 show the prompts for the LYNC mode.

At this point LYNC has been customized to your hardware and is ready for use.

LYNC OPERATION

The operation of LYNC is directed by control key functions. When certain control keys are typed LYNC will switch from one mode to another or perform various other functions. All of the control key functions are described later in this section of the manual.

Modes of Operation

LYNC has 7 modes of operation as shown in figure 7.

1:	[TERM]	perform as an intelligent terminal
2:	[LYNC]	console-to-console communication
3:	[SEND]	send files to remote machine with protocol
4:	[FETCH]	retrieve files from remote machine with protocol
5:	[DUMP]	send file with no protocol
6:	[LOAD]	receive file with no protocol
7:	[HEXDUMP]	convert file to HEX format and send with no protocol

Figure 7

Control keys are used to enter and exit different modes of operation and to perform other housekeeping functions. In addition two control characters are defined for execution of user supplied routines. These routines may be used to do anything you like. In order to implement the user defined routines you must patch the routines into the custom I/O area of LYNC (see CUSTOM I/O). The various control key functions are described later in this section.

[TERM] mode:

This mode is for communication with some type of remote access system such as a time sharing system or a bulletin board system. When in [TERM] mode LYNC can operate in either FULL DUPLEX or HALF DUPLEX. In FULL DUPLEX, LYNC expects characters to be echoed by the remote computer. LYNC will send out all characters typed from the keyboard to the remote machine but will only display characters that come back from the remote machine on the console.

In HALF DUPLEX, LYNC echos each character to the console after sending it out to the remote machine. If the remote machine echos the characters back to LYNC they will appear twice on the console.

In [TERM] mode all control characters typed at the console keyboard are sent out to the remote device and will not activate the other modes. The exception to this is the TERMINAL MODE EXIT character (default is <ctl-A>). The TERMINAL MODE EXIT character will force an exit from the terminal mode and display a table of the various options available to you as shown in figure 8.

```
1: [TERM]
2: <ctl-A>
3: OPTIONS:
4: H - Half/full duplex
5: V - View control char
6: O - Open input file
7: W - Write data to file
8: L - Load program
9: D - Dump disk file
10: X - Hex dump file
11: P - System parameters
12: U - User 1
13: Y - User 2
14: C - Go to LINK
15: E - Exit to CP/M
```

Figure 8

After typing the terminal mode exit character you can use any of the control characters specified in figure 8. For example you could use <ctl-O> to set up a memory buffer and open a disk file. Once the file is set up each byte that comes in will be stored in memory in addition to appearing on the console. The exit character can again be used to activate the options followed by a <ctl-W> to write the data onto disk. This procedure can be repeated until you run out of disk space. Should the memory buffer overflow, an error message will appear but you may still write the data received before that point out to the disk file. Receiving a file from a remote terminal device is covered in more detail when the control key functions are described later in this section.

[LYNC] mode:

The [LYNC] mode is used for communications between two micro-computers where both are running LYNC. Whenever you enter the [LYNC] mode LYNC will print the lines shown in figure 9 on the console.

```
1: Linking
2: ..... OK
3: [LYNC]
4:
```

Figure 9

On line 9-1 LYNC indicates that it is trying to establish a link with the remote machine. The dots on line 9-2 appear as special test characters are sent between the two machines to test the communications line. If these dots do not appear, the link between the machines has probably been disrupted. This can occur when a machine is shut off, is disconnected, or the phone line in use mysteriously hangs up on you. Line 9-3 indicates that you are in the [LYNC] mode.

While in the [LYNC] mode you may type messages at your console and they will be displayed at the remote machine's console. This provides a technique for communication between two users without the need for a separate phone line or constantly switching the line between voice and data operation.

You have access to all other modes of operation from the [LYNC] mode by typing the appropriate control character.

[SEND] mode:

The [SEND] mode is used for sending one or more files to the remote computer. This causes the file to be sent with the full error checking protocol of LYNC. When you enter the [SEND] mode you will be prompted for the name of the file to send as shown in figure 10.

```

1: [SEND]
2: Filename.typ = PAYROLL.COM<cr>
3: PAYROLL.COM - COMPLETE
4: Linking
5: ..... OK
6: [LYNC]
7:

```

Figure 10

There are a variety of ways in which the file or files to be transferred can be specified. You may respond to the prompt on line 10-2 with any of the following:

1. A single file name (i.e. PAYROLL.DAT).
2. A list of file names separated by spaces or commas (i.e. PROG.COM,MAIL.LST PAYROLL.DAT ...).
3. A wildcard name (i.e. PR*.DAT or PAY??ROL.DAT). The wildcard specification is the same as for CP/M: An asterisk will match any number (including 0) of characters at its position. The asterisk can only appear at the end of the primary file name or at the end of the extension. A question mark will match any single character at its position. Question marks may be embedded between other characters in either the primary file name or its extension.
4. An indirect file. An indirect file is specified by a commercial each at sign before its name (ie. @FILELIST.1). The indirect file is a sequential file containing a list of file names to be transferred separated by carriage return line feed pairs. An indirect file may contain wildcard file names but must not contain an indirect file reference.
5. Any combination of 1-4, for example: *.ASM,PAY.HEX,@GROUP1,XYZ*.LST

The files transferred in the [SEND] mode will be stored on the remote machine with the same file name they have on the local machine. If a file with a given name already exists at the remote machine it will be overwritten by the new file as it is transferred.

As each file is being transferred its name will be displayed on the console (line 10-3) and the console cursor will be positioned just following the file name on the display. The cursor will move forward or backward one space as each block of the file is transmitted. If an error is detected during transmission, a single character error indicator will be printed. These indicators will be overwritten as subsequent blocks are transmitted or subsequent errors occur. It is completely normal for some errors to occur during file transfer since disk accessing and other computer operations may cause a timeout error. The error checking protocol of LYNC is such that these errors will not affect the integrity of the file that is transferred. After the file has been transferred, the message "- COMPLETE" is displayed on the console (line 10-3) and LYNC proceeds on to the next file to be transferred (if any).

The following error indicators may be displayed at the receiving machine:

- T - Indicates a timeout error or incomplete block of data. If more than about two seconds elapse between successive blocks of incoming data, a timeout error has occurred. If a pause of greater than about two seconds occurs when receiving a

block of data, an incomplete block error has occurred. This error can be caused by delays in reading data from disk on the sending computer and may occur even when two computers are connected by a hard wire.

- C - Indicates a checksum error in a received block. This is typically caused by noise in the transmission medium. If a large number of checksum errors occur you should check for problems with your transmission equipment (modem, telephone handset microphone, etc.).
- D - Indicates a duplicate block has been received. The sending machine has duplicated the transmission of a block of data (perhaps because it did not receive an acknowledgment of the block) which has previously been received without error.
- M - Indicates that a received block is not the next one expected in the sequence of blocks transmitted; one or more blocks are missing.

The following error indicator may be displayed at the transmitting machine:

- R - Indicates that a block is being re-transmitted. This will occur when the receiving end requests a re-transmission due to an error or when an acknowledgment of a block is not received within the allotted time.

[FETCH] mode:

This mode allows you to fetch files from the remote machine without operator interaction at the remote machine. [FETCH] works just like the [SEND] mode except that files will be transferred from the remote machine to the local machine.

[DUMP] mode:

This mode is used to send out disk files to some system that does not have LYNC. It is usually used from the [TERM] mode since [LYNC] mode is only useful if the remote machine is running LYNC. It requests the name of a file (only one file may be sent at a time), and sends it out continuously at your current baud rate. No handshaking of any sort is performed so the remote machine must be able to accept the continuous stream of characters at the current baud rate. When all of the file has been sent, it will say " - COMPLETE" and wait for you to hit RETURN twice before returning to [TERM]. This is to allow for manual closing of the file at the other end if necessary.

[HEXDUMP] mode:

This works just like [DUMP] except that the file to be sent is first converted to a standard HEX file. This mode is very useful for sending a copy of LYNC.COM to another computer. The other computer can use PIP or some other data communications program to receive the file and save it on disk. The CP/M LOAD program can then be used to produce an executable COM file. The hex file will be terminated with a <ct-Z> so that file transfer programs like PIP will correctly recognize the end of file.

[LOAD] mode:

The [LOAD] mode provides a means of receiving files from computer systems that are not running LYNC with a rudimentary form of error checking. Use of the [LOAD] mode is illustrated in figure 11.

```

1: [TERM]
2: <ctl-A>
3: OPTIONS:
4: H - Half/full duplex
5: V - View control char
6: O - Open input file
7: W - Write data to file
8: L - Load program
9: D - Dump disk file
10: X - Hex dump file
11: P - System parameters
12: U - User 1
13: Y - User 2
14: C - Go to LINK
15: E - Exit to CP/M <ctl-L>
16: [LOAD]
17: Filename.typ = PAYROLL.COM<cr>
18: PAYROLL.COM
19: Ready to receive file !
20: DATA NOT VERIFIED
21: [TERM]
22: <ctl-A>
23: OPTIONS:
24: H - Half/full duplex
25: V - View control char
26: O - Open input file
27: W - Write data to file
28: L - Load program
29: D - Dump disk file
30: X - Hex dump file
31: P - System parameters
32: U - User 1
33: Y - User 2
34: C - Go to LINK
35: E - Exit to CP/M <ctl-L>
36: [LOAD]
37: Filename.typ = DUMMY.1<cr>
38: DUMMY 1
39: Ready to receive file !
40: DATA VERIFIED
41: [TERM]

```

Figure 11

On line 11-17 you are asked for a file name just as for the [SEND] or [FETCH] modes. Enter the name followed by <cr>. As the file is created on the disk, its name is shown on the console as illustrated in line 11-18. After the file is created on the disk, LYNC informs you that it is ready to receive data as shown in line 11-19. At this time data transmission from the remote machine should be started.

When the first byte of data comes in LYNC will display a "I" on the console to indicate that the transfer has started. All of the incoming data will be stored in memory until a pause of approximately 15 seconds is detected in the incoming data stream. When such a pause is detected the data is written out to the specified disk file and the file is closed. Note that you are limited to transferring files that will fit within the available memory. LYNC takes up about 6 Kbytes of memory leaving the remainder of the transient program area (TPA) available for buffering incoming data.

In order to provide a rudimentary test that the data has been received correctly, LYNC compares the data received to what is already in memory. When the transfer is complete LYNC will display the message "DATA VERIFIED" on the console if the data that was read was identical to what was already in memory. If a difference is found between the incoming data and the data already in memory, the message "DATA NOT VERIFIED" will be displayed on the console.

Data verification is accomplished, as is shown in figure 11, by transmitting the same file twice. After the first transmission the message DATA NOT VERIFIED (line 11-20) is given

since there was no data in the memory buffer before the transfer, hence the incoming data did not match it. At this point the memory buffer still contains the file that was received and saved on disk. LYNC returns to the [TERM] mode after it has finished the [LOAD] operation.

Lines 11-22 to 11-40 show the file being re-transmitted but this time it is stored in a dummy file on the disk. After the file transmission is complete the "DATA VERIFIED" message is given (line 11-40) indicating that the second transmission matched the first. Since two successive transmissions yielded the same result the chances are pretty good that the file was received without error.

Control Key Functions

Figure 12 summarizes the various control key functions and indicates in which mode or modes the functions may be used. Attempting to use a control key function in a mode for which it is not defined will have unpredictable results. Note that control key functions that are accessed from the [TERM] mode must be prefaced by the TERMINAL MODE EXIT character specified when LYNC was configured (the default is <ctl-A>). In the following descriptions the terminal mode exit character is represented by <tme> for those functions that are used from the terminal mode.

1: 2: 3:	control key	function	available in mode(s)
4:	<ctl-A>	Display local directory	[LYNC]
5:	<ctl-Z>	Display remote directory	[LYNC]
6:	<ctl-R>	Set current drives	[LYNC]
7:	<ctl-C>	Return to [LYNC] mode	all
8:	<ctl-S>	Enter [SEND] mode	[LYNC]
9:	<ctl-F>	Enter [FETCH] mode	[LYNC]
10:	<ctl-D>	Enter [DUMP] mode	[TERM]
11:	<ctl-X>	Enter [HEXDUMP] mode	[TERM]
12:	<ctl-L>	Enter [LOAD] mode	[TERM]
13:	<ctl-T>	Enter [TERM] mode	[LYNC]
14:	<ctl-O>	open file and reset buffer	[TERM]
15:	<ctl-W>	write buffer to disk	[TERM]
16:	<ctl-P>	Reset system parameters	[LYNC],[TERM]
17:	<ctl-E>	exit to CP/M	[LYNC],[TERM]
18:	<ctl-V>	toggle control char display	[TERM]
19:	<ctl-H>	toggle full / half duplex	[TERM]
20:	<ctl-U>	execute user routine #1	[LYNC],[TERM]
21:	<ctl-Y>	execute user routine #2	[LYNC],[TERM]

Figure 12

<ctl-A> - Display Local Directory ([LYNC] mode)

This control key function will display the directory on the current disk at the local machine. Note that the current disk drive may be changed using the <ctl-R> control key function. This function should be used only while in the [LYNC] mode. The form of the directory display is illustrated in figure 13.

1:	Local files -
2:	C: GENLYNC COM : LYNCTESTCOM : LINKDOC BAK : LINKDOC
3:	C: LYNC COM :
4:	Linking
5: OK
6:	[LYNC]
7:	

Figure 13

<ctl-Z> - Display Remote Directory ([LYNC] mode)

This control key function will display the directory on the current disk at the remote machine. Note that the current disk drive may be changed using the <ctl-R> control key function. The form of the remote directory display is the same as that shown in figure 13 for the local directory display.

<ctl-R> - Set Current Local and Remote Drives ([LYNC] mode)

This control key function allows you to change the current drive assignment at the local and remote machine assuming LYNC was configured to permit the drive assignment to be changed. When LYNC is executed the current drive is set to be the system default drive. All disk I/O is performed on the current drive. To access a different disk drive you must use the <ctl-R> control key function to change the current drive to the desired drive.

When the <ctl-R> control key function is used a disk system reset is performed; this allows you to change diskettes on a given drive and be able to write to the new diskette. Figure 14 shows the form of the prompts given when the <ctl-R> control key function is used. If you do not want to change the current drive assignment for either of the local or remote machines, merely type the <cr> key in response to the prompt for a drive. If all you want to do is change diskettes, make the change, type <ctl-R>, and do not specify new drives. This will retain the same current drives but will cause a disk system reset so that you can write to the new diskette.

```

1: Local drive = <cr>
2: Remote drive = <cr>
3: Linking
4: ..... OK
5: [LYNC]
6:

```

Figure 14

<ctl-C> - Return to [LYNC] mode (any mode)

This control key function operates as a sort of general reset to the LYNC program. At nearly any time, you can press <ctl-C> and LYNC will abort whatever it is doing and return to [LYNC] mode. This is useful for prematurely halting a transfer if a mistake has been made. The only time <ctl-C> will not return to the [LYNC] mode is when in the [TERM] mode. In [TERM] mode, the only character recognized as a control key function is the terminal mode exit character specified when LYNC was configured. In order to return to [LYNC] mode from the [TERM] mode you must type the terminal mode exit character followed by <ctl-C>.

<ctl-S> - Enter [SEND] Mode ([LYNC] mode)

This control key function will enter the [SEND] mode from the [LYNC] mode. See the description of the [SEND] mode earlier in this section.

<ctl-F> - Enter [FETCH] Mode ([LYNC] mode)

This control key function will enter the [FETCH] mode from the [LYNC] mode. See the description of the [FETCH] mode earlier in this section.

<tme><ctl-D> - Enter [DUMP] Mode ([TERM] mode)

This control key function will enter the [DUMP] mode from the [TERM] mode. See the description of the [DUMP] mode earlier in this section.

<tme><ctl-X> - Enter [HEXDUMP] Mode ([TERM] mode)

This control key function will enter the [HEXDUMP] mode from the [TERM] mode. See the description of the [HEXDUMP] mode earlier in this section.

<tme><ctl-L> – Enter [LOAD] Mode ([TERM] mode)

This control key function will enter the [LOAD] mode from the [TERM] mode. See the description of the [LOAD] mode earlier in this section.

<ctl-T> – Enter the [TERM] Mode ([LYNC] mode)

This control key function will enter the [TERM] mode from the [LYNC] mode. See the description of the [TERM] mode earlier in this section.

<tme><ctl-O> – Open Disk File and Reset Buffer ([TERM] mode)

This control key function will cause a disk file to be opened and a memory buffer made ready to receive any incoming data. After this is done LYNC will return to the [TERM] mode. As characters arrive from the remote machine they will be displayed on the screen and stored in the memory buffer. This action will continue until either the memory buffer fills up or you use the <ctl-W> function to write the contents of the memory buffer out to disk. If the buffer fills up an error message will be displayed and no more characters will be saved in the buffer. The contents of the memory buffer can be saved with the <ctl-W> function even after the buffer has filled up.

Note that it is your responsibility to ensure that the remote machine stops sending data before the buffer fills up. During the reception of data in the [TERM] mode you can send out characters from the console keyboard so you periodically can send whatever command the remote machine needs to stop sending data so that you can empty the buffer with the <ctl-W> command. The buffer size can be estimated by subtracting 6 Kbytes from the size of the transient program area of your computer.

When the <ctl-O> function is used LYNC will prompt you for a file name as shown in figure 15. If the file already exists on the current disk it will be erased and a new file created.

```
1: Filename.typ = TEST.1<cr>
2: TEST 1
3: (Buffer Cleared - ready)
4: [TERM]
5:
```

Figure 15

<tme><ctl-W> – Write Contents of Buffer to Disk ([TERM] mode)

This control key function causes the information which has been saved in memory as a result of the <ctl-O> function to be written out to the disk file specified when the <ctl-O> function was used. After the information is written to the disk file the file is closed and the memory buffer de-activated. After <ctl-W> has been used incoming data will no longer be buffered in memory for subsequent writing to disk and LYNC will return to the [TERM] mode. Figure 16 shows the console interaction as the file is written to disk and closed. Note that line 16-1 corresponds to the last line of the terminal mode exit menu (figure 8) since <ctl-W> is used from the [TERM] mode. If you need to transfer more data than can fit into the buffer you can use the following technique (note this will only work with textual material):

1. Open a file and enable the memory buffer with a <ctl-O>.
2. Type the commands on the console keyboard to initiate transfer for data from the remote machine.
3. When the buffer is close to full (you will have to estimate this from the serial port baud rate and the length of time the transfer has been running) issue commands from the keyboard to halt the output of the remote machine.
4. Use <ctl-W> to write the received data to the disk.

5. Use <ctI-O> to open a new data file.
6. Issue the necessary command from the keyboard to start data transmission from the remote machine again.
7. Repeat steps 3, 4, 5, and 6 until all the data has been transferred.
8. Exit from LYNC and use the PIP utility to concatenate all of the data files together into one large file.

```

1:  E - Exit to CP/M <ctI-W> - COMPLETE
2:  (Buffer Off)
3:  [TERM]
4:

```

Figure 16

<ctI-P> - Re-specify Configuration Parameters ([LYNC] mode)

This control key function causes LYNC to repeat the configuration process described in the GETTING STARTED section. This function is useful when various parameters such as the terminal mode exit character must be changed or when re-configuring LYNC for different hardware.

<ctI-E> - Exit to CP/M ([LYNC] or [TERM] mode)

This control key function causes LYNC to return to the operating system. Note that when in the [TERM] mode you must type the terminal mode exit character before you can use the <ctI-E> control key function.

<tme><ctI-V> - Toggle Control Character View ([Term] mode)

This control character function provides a means for you to change between viewing and not viewing control characters without re-running the LYNC configuration procedure. When the <ctI-V> function is used LYNC will change from the control character view option currently in effect to the other. When the control character view option is enabled all control characters (except carriage return and line feed) will be displayed on the console as a caret (or up arrow) followed by the printing character corresponding to the control character. For example, <ctI-D> will be displayed as D. When the control character view option is disabled all control characters will be sent directly to the console. Note that when the control characters are sent directly to the console they may be interpreted by the console device as commands for special functions (ie. reverse video, erase screen, etc.).

<tme><ctI-H> - Toggle Full/Half Duplex ([TERM] mode)

This control character function switches the operation of LYNC in the [TERM] mode between full and half duplex. When you use this control key function LYNC will change to the type of operation that is not currently active. For example, if LYNC is in half duplex, using this control character function will switch it to full duplex. If in full duplex this control character function will switch LYNC to half duplex. In full duplex LYNC will display on the console only those characters coming in from the remote machine, ie. the remote machine must echo the characters LYNC sends to it. In half duplex, LYNC will echo all characters typed on the keyboard to the console as they are sent to the remote machine, therefore the remote machine should not echo the characters sent to it.

<ctl-U> - Execute User Defined Routine #1 ([LYNC] or [TERM] mode)

This control character function causes the first of two possible special user supplied functions to be executed. Instructions for installing user supplied functions are given in the CUSTOM I/O section of the manual. These special functions can be used to perform special functions such as changing baud rates, hanging up a phone line, dialing a phone number, etc. They are usually highly hardware dependent and their implementation requires assembly language programming abilities.

<ctl-Y> - Execute User Defined Routine #2 ([LYNC] or [TERM] mode)

This control character function causes the second of two possible special user supplied functions to be executed. See the notes for <ctl-U> above.

CUSTOM I/O ROUTINES

It will be necessary to write and install routines to provide the I/O functions LYNC needs if your computer system has a serial I/O channel that cannot be accommodated by the self-generating I/O routines in LYNC. Specialized routines are also needed if you have a half duplex modem that requires line switching or if you want to install the user defined functions. In order to do this you will need some ability to program in assembly language and to use the CP/M debugger, DDT.

LYNC always calls the direction switch routine before transmitting any characters and then calls it again after the transmission is completed. This means that LYNC operates with the transmission line in the receiving direction until it needs to send a character. The transmission line is switched to the send direction only as long as it needs to make a transmission. The direction switch routine in the standard self-generated I/O routines contains only a return instruction, hence it takes no action.

If you are going to install custom I/O routines, it is still necessary to run GENLYNC as described in GETTING STARTED. If you will be supplying a complete custom set of I/O routines for LYNC, you may just ignore the first 4 questions LYNC asks in its customization procedure (lines 3-1 to 3-4 of figure 3). Simply type a <cr> in response to these 4 questions. You should enter the required information on lines 3-5 to 3-10. You can ignore lines 3-11 to 3-13, entering a <cr> when information is requested. On line 3-15 you should enter N<cr> to indicate that the self-generated I/O routines are not to be installed. Respond with Y<cr> on line 3-16 and a file name on line 3-17. This will save a copy of LYNC on disk so that you can patch in your custom I/O routines. At this point LYNC will attempt to begin execution and will probably cause your system to lock up since the serial I/O drivers have not yet been installed. This lockup can be avoided by having LYNC start in the [TERM] mode. You may need to re-boot your system in order to continue.

If you are going to install only the serial line direction control routine and/or the user defined functions, supply all of the information requested by the customization procedure exactly as described in GETTING STARTED. This will generate a copy of LYNC on your disk with the self-generated I/O routines installed ready for you to install the additional routines.

All of the I/O functions used by LYNC are accessed by a jump table starting at location 10Ch. This jump table is shown in figure 17. If you are installing a complete set of I/O drivers you will need to provide routines for each of these functions. If you are merely installing the half duplex line direction switching routine, only the fifth jump is of interest. The sixth and seventh jumps are used for the user defined functions.

Each of the I/O routines need not preserve the contents of any registers. LYNC ignores the contents of all registers (except the register, if any, that returns information) returned by the I/O functions.

	<u>location</u>	<u>jump</u>	<u>comments</u>
1:			
2:			
3:	010C	JMP XISP	;serial port input status check
4:	010F	JMP XID	;serial input routine
5:	0112	JMP XODR	;serial output routine
6:	0115	JMP XINIT	;port initialization routine
7:	0118	JMP SWITCH	;line direction switch routine
8:	011B	JMP USR1	;first user function <ctl-U>
9:	011E	JMP USR2	;second user function <ctl-Y>

Figure 17

Line 17-3 shows a jump to a routine that checks the serial port status to find out if an incoming character is ready to be input. This routine should return with the zero flag cleared if a character is available and the zero flag set if no character is ready.

Line 17-4 shows a jump to a routine to input a character from the serial port. This routine should return with the input character in register A. It is not necessary to check the serial port status in this routine since LYNC will not call it unless the input status check routine (line 17-3) shows that a character is ready.

Line 17-5 shows a jump to the serial port output routines. LYNC will call this routine with the character to be output in register A. This routine should check the serial port status and wait until the serial port can accept an output character before sending the character. It should not return to the calling routine until the character has been sent.

Line 17-6 shows a jump to the serial port initialization routine. LYNC calls this routine at the beginning of execution to set up the serial port. This routine should perform the operations (if any) that are necessary to make the serial port ready for use by LYNC.

Line 17-7 shows a jump to the serial line direction switching routine. This routine is called immediately before the transmission of one or more characters and is called once again immediately following the completion of the transmission. The transmission line should be initialized to start in the incoming direction (to receive data). This routine should switch the serial transmission line to the transmit direction the first time it is called and back to the receive direction the second time it is called (ie. it alternates the direction of the line).

Lines 17-8 and 17-9 show jumps to the two user defined functions. These routines are executed whenever the user types their associated control keys (<ctl-U> and <ctl-Y>). These functions can be used to perform special operations that are unique to your system (like send a certain string of characters). If your computer system has hardware that is capable of generating a break on the serial output line you may want to install a routine which causes a break for one of the user functions.

The standard LYNC self-generating I/O routines use the memory space from the end of the jump table (121h) up to location 1BFh. If you are providing a complete set of I/O routines, you may use all of the space from 121h to 27Fh. If you are only adding the direction switching or user defined functions, you may use only the memory from 1C0h to 27Fh.

As an example, figure 18 shows a program that will generate a user function to send out a certain string to the remote device whenever the first user function control key (<ctl-U>) is typed. This might be used for logging onto a time sharing computer system or for sending a string that often is needed.

```

1:      ;      Example of program for a user installed function
2:
3:      jdout  equ    112h  ;address of jump to serial port output
4:                        ; routine
5:
6:      org    11bh  ;install a jump to supplied function
7:      jmp    user1
8:
9:
10:     org    1c0h  ;put routine in available patch area
11:     ; The following routine will output a string of characters to
12:     ; the serial output port. The string is stored in memory starting
13:     ; at location STR1 and is terminated by a null character.
14:
15:     user1: lxi    h, str1 ;get the beginning address of the string
16:     uloop: mov    a,m     ;get a character
17:            ora    a      ;check for zero
18:            rz      ;return on first null (zero)
19:            push   h     ;save pointer (may not be necessary)
20:            call  jdout  ;output the character
21:            pop    h     ;retrieve pointer
22:            inc    h     ;increment pointer
23:            jmp    uloop ;continue with another character
24:
25:     str1: db    'This is the message to send',0
26:
27:     end

```

Figure 18

The program shown in figure 18 must be assembled and installed in LYNC. An example of how this might be done is given in figure 19. This dialog assumes that the file containing the source code shown in figure 18 is named EXAM16.ASM and resides on drive A. It also assumes that a configured copy of LYNC called LYNC.COM as well as the standard CP/M utilities ASM.COM and DDT.COM are stored on drive A. The modified copy of LYNC with the user function installed is saved on the disk as LYNCU1.COM.

```

1:  A>ASM EXAM16<cr>
2:  CP/M ASSEMBLER - VER 2.0
3:  01EB
4:  000H USE FACTOR
5:  END OF ASSEMBLY
6:
7:  A>DDT LYNC.COM<cr>
8:  DDT VERS 2.2
9:  NEXT PC
10: 1780 0100
11: -IEXAM16.HEX<cr>
12: -R<cr>
13: NEXT PC
14: 1780 0000
15: -<ctrl-C>
16: A>SAVE 23 LYNCU1.COM<cr>
17: A>

```

Figure 19

The file JUMP.ASM on the distribution LYNC diskette contains the jump table as shown in figure 17. The example of figure 18 is also on the distribution diskette in a file called EXAM16.ASM.

APPENDIX A
Files on Distribution Diskette

<u>file name</u>	<u>description</u>
GENLYNC.COM	un-configured copy of LYNC
JUMP.ASM	source code for the jump table at the beginning of the I/O routine area of LYNC - see details in CUSTOM I/O section of this manual
EXAM16.ASM	source code for the example of a user control key function - see details in CUSTOM I/O section of this manual

APPENDIX B
ASCII Character Set

<u>dec</u>	<u>hex</u>	<u>character</u>	<u>dec</u>	<u>hex</u>	<u>character</u>	<u>dec</u>	<u>hex</u>	<u>character</u>
0	00	NUL	43	2B	+ (plus)	86	56	V
1	01	SOH <ctl-A>	44	2C	, (comma)	87	57	W
2	02	STX <ctl-B>	45	2D	- (minus)	88	58	X
3	03	ETX <ctl-C>	46	2E	. (period)	89	59	Y
4	04	EOT <ctl-D>	47	2F	/ (slash)	90	5A	Z
5	05	ENQ <ctl-E>	48	30	0	91	5B	[(lf bracket)
6	06	ACK <ctl-F>	49	31	1	92	5C	\ (back slash)
7	07	BEL <ctl-G>	50	32	2	93	5D] (rt bracket)
8	08	BS <ctl-H>	51	33	3	94	5E	^ (caret)
9	09	HT <ctl-I>	52	34	4	95	5F	_ (underscore)
10	0A	LF <ctl-J>	53	35	5	96	60	` (accent grave)
11	0B	VT <ctl-K>	54	36	6	97	61	a
12	0C	FF <ctl-L>	55	37	7	98	62	b
13	0D	CR <ctl-M>	56	38	8	99	63	c
14	0E	SO <ctl-N>	57	39	9	100	64	d
15	0F	SI <ctl-O>	58	3A	: (colon)	101	65	e
16	10	DLE <ctl-P>	59	3B	; (semi-colon)	102	66	f
17	11	DC1 <ctl-Q>	60	3C	< (less)	103	67	g
18	12	DC2 <ctl-R>	61	3D	= (equals)	104	68	h
19	13	DC3 <ctl-S>	62	3E	> (greater)	105	69	i
20	14	DC4 <ctl-T>	63	3F	? (question)	106	6A	j
21	15	NAK <ctl-U>	64	40	@ (each at)	107	6B	k
22	16	SYN <ctl-V>	65	41	A	108	6C	l
23	17	ETB <ctl-W>	66	42	B	109	6D	m
24	18	CAN <ctl-X>	67	43	C	110	6E	n
25	19	EM <ctl-Y>	68	44	D	111	6F	o
26	1A	SUB <ctl-Z>	69	45	E	112	70	p
27	1B	ESC	70	46	F	113	71	q
28	1C	FS	71	47	G	114	72	r
29	1D	GS	72	48	H	115	73	s
30	1E	RS	73	49	I	116	74	t
31	1F	US	74	4A	J	117	75	u
32	20	space	75	4B	K	118	76	v
33	21	! (exclamation)	76	4C	L	119	77	w
34	22	" (quote)	77	4D	M	120	78	x
35	23	# (pound)	78	4E	N	121	79	y
36	24	\$ (dollar)	79	4F	O	122	7A	z
37	25	% (percent)	80	50	P	123	7B	{ (lf brace)
38	26	& (ampersand)	81	51	Q	124	7C	(logical or)
39	27	' (apostrophe)	82	52	R	125	7D	} (rt brace)
40	28	((left paren)	83	53	S	126	7E	~ (tilde)
41	29) (right paren)	84	54	T	127	7F	DEL
42	2A	* (asterisk)	85	55	U			

APPENDIX C Customization Parameters for Specific Hardware

The information in this section is provided to help you configure LYNC to work with various different types of hardware. The information presented has been gathered from users, manufacturer's specifications, and direct experience. Every attempt has been made to insure that this information is accurate and correct but Pickles & Trout cannot guarantee its accuracy or correctness. If the parameters given do not seem to work in your system, please read the documentation for your serial interface hardware CAREFULLY before contacting Pickles & Trout.

We at Pickles & Trout have every desire to be helpful but please realize that it is the user's responsibility to locate the information required to customize LYNC. If you have trouble getting LYNC customized for your machine, we will try to help but support for hardware to which we do not have access is limited.

The information in this appendix is gathered largely from the experience of our users. If you customize LYNC for hardware that is not listed here we would like to add it to our documentation. Please SEND us complete information about your hardware and all of the parameters you specified when customizing LYNC. If we do not already have the information you send in we will add it to our list and send you \$10 as a thank you for your help in improving the LYNC documentation. As new customization information is collected we will periodically add it to our manual. If you want a copy of the latest version of this appendix, send us a stamped, self-addressed business sized envelope with postage for two ounces and we will send you one.

The mention of specific hardware in this appendix in no way constitutes an endorsement by Pickles & Trout of that hardware for any particular use.

The information given for each type of hardware consists of those parameters which are specific to the transmission and reception of serial data. Items such as the number of disk drives on the system, console backspace character, and power up mode are not given because they are not determined by the serial interface hardware. The information for each type of hardware is given in a tabular form with columns labeled as follows:

port	indicates which hardware port (when there are more than one) the row of the table refers to
data	shows the data port number
stat	shows the status/control port number
recv	gives the receive data ready mask byte
xmit	gives the transmit ready mask byte
clock	shows the clock rate for the system
imode	indicates the initialization mode to use
ibytes	indicates the initialization bytes, if any, to specify

TRS-80® Model II

<u>port</u>	<u>data</u>	<u>stat</u>	<u>rcv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
A	F4	F6	1	4	4	1	18,4,44,3,C1,5,EA,0
B	F5	F7	1	4	4	1	18,4,44,3,C1,5,EA,0

Note that with the initialization specified, LYNC will re-program the serial I/O in the computer. This may cause problems with the operating system or other programs that use the serial ports. If you set up the serial port with standard system facilities, specify 0 for the init mode and do not enter any initialization bytes.

SUPERBRAIN™

<u>port</u>	<u>data</u>	<u>stat</u>	<u>rcv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
main	40	41	2	1	4	0	0
aux	58	59	2	1	4	0	0

Note that in order to send data out from the serial ports the CTS line on the RS-232 connector must be in the true state. To do this connect lines 5 and 20 together at the computer. The serial port data rate, word length, etc. should be set up using the standard system facilities supplied with the machine.

NORTHSTAR HORIZON

<u>port</u>	<u>data</u>	<u>stat</u>	<u>rcv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
0	2*	3*	2	1	4	0	0
1	4*	5*	2	1	4	0	0

* The data and status/control port numbers shown are the manufacturer's recommended default values. These port assignments may be changed by means of dip switches. If they are changed, the appropriate port numbers must be specified when LYNC is configured. See the manufacturer's documentation for details on setting the port numbers.

The serial port baud rate, word length, etc. are set by means of switches within the computer. Refer to the manufacturer's documentation for setting these parameters.

GODBOUT INTERFACER™ 1 & INTERFACER™ II

<u>port</u>	<u>data</u>	<u>stat</u>	<u>rcv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
A	0*	1*	2	1	†	0	0
B	2*	3*	2	1	†	0	0

* The data and status/command port numbers shown are ones that are commonly used. Since these port assignments are set by dip switches on the INTERFACER board you may set them to other ports. Please refer to the manufacturer's documentation for setting these switches. If the ports are re-assigned, the appropriate port numbers should be used in the LYNC customization.

† The clock rate is dependent on the processor in the system. Refer to your system documentation for this information.

The baud rate, word length, stop bits, etc. are set by jumpers on the INTERFACER board. Refer to the INTERFACER documentation for setting these jumpers.

GODBOUT SYSTEM SUPPORT 1

<u>data</u>	<u>stat</u>	<u>recv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
5C*	5D*	2	1	†	3	5E*,4E,5E*,75,5F*,5§

* These are the manufacturer's recommended port assignments for the serial interface. The port assignments may be changed by a dip switch on the circuit board. Please refer to the documentation for the board for setting these switches. If the ports are re-assigned, the appropriate port numbers should be used in the LYNC customization.

† The clock rate is dependent on the processor in the system. Refer to your system documentation for this information.

§ This value will set the serial port for 300 baud. Other baud rates may be programmed by changing this number. See the manufacturer's documentation for details.

The initialization bytes shown above will set the serial port for asynchronous operation with 8 bit words, one stop bit, no parity and 300 baud. It is necessary to install jumpers at J2 on the board before using it. If these jumpers are not installed, refer to the manufacturer's documentation for instructions on how to install them.

CALIFORNIA COMPUTER SYSTEMS MODEL 2810 Z-80 CPU

<u>data</u>	<u>stat</u>	<u>recv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
20*	25*	1	20	§	3	24*,F,23*,83,20*,80†,21*,1†, 23*,3,21*,0,25*,1

*These are the manufacturer's recommended port assignments for the serial interface. The port assignments may be changed by means of jumpers on the board. If the port assignments are changed, the ports specified in the LYNC customization must be changed to match. See the manufacturer's documentation for instructions on changing the port assignments.

§ The clock rate will be either 2 or 4 depending on whether the processor clock rate is set to 2 or 4 Mhz. See the manufacturer's documentation for instructions on setting the processor clock rate.

† These numbers set the port baud rate to 300 baud. See the manufacturer's documentation for numbers corresponding to other baud rates.

The initialization bytes specified set up the serial port for 8 bit words, 1 stop bit, no parity and 300 baud.

VECTOR GRAPHIC BITSTREAMER™ II

<u>port</u>	<u>data</u>	<u>stat</u>	<u>recv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
A	2*	3*	2	1	†	1	0,0,0,40,4E,27
B	4*	5*	2	1	†	1	0,0,0,40,4E,27
C	6*	7*	2	1	†	1	0,0,0,40,4E,27

*These are the manufacturer's recommended port assignments for the serial interface. The port assignments may be changed by means of jumpers on the board. If the port assignments are changed, the ports specified in the LYNC customization must be changed to match. See the manufacturer's documentation for instructions on changing the port assignments.

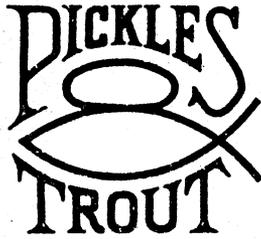
§ The clock rate is determined by the processor board in the system. For Vector Graphic systems, the clock rate is 4 Mhz.

VECTOR GRAPHIC ZCB

<u>data</u>	<u>stat</u>	<u>rcv</u>	<u>xmit</u>	<u>clock</u>	<u>imode</u>	<u>ibytes</u>
4*	5*	2	1	4	1	0,0,0,40,4E,27

*These are the manufacturer's recommended port assignments for the serial interface. The port assignments may be changed by means of jumpers on the board. If the port assignments are changed, the ports specified in the LYNC customization must be changed to match. See the manufacturer's documentation for instructions on changing the port assignments.

INTERFACER is a trademark of Godbout Electronics
TRS-80 is a trademark of Tandy Corporation
SUPERBRAIN is a trademark of Intertec Data Systems
BITSTREAMER is a trademark of Vector Graphic



PICKLES & TROUT

P. O. BOX 1206, GOLETA, CA 93116. (805) 685-4641

SMALL COMPUTER INTERFACES & SOFTWARE

LYNC 2.0 Additional Notes

After the LYNC 2.0 User's Manual had been printed and distributed, we discovered a few potential points of confusion:

1. re: customizing LYNC 2.0

Any time you run the GENLYNC program, you MUST respond with Y<cr> when asked if you want to alter the I/O (line 15 of figure 3). If you respond negatively, LYNC 2.0 will not properly generate the serial I/O routines necessary to communicate to your hardware and will not work. See pp 3-8 of the manual for complete instructions on getting started.

2. re: TRS-80 Model II parameters given in Appendix C

If you are using LYNC 2.0 with the TRS-80 Model II, it is not necessary to specify initialization mode 1 if you are using a system that will set up the serial port for LYNC 2.0. For example, if you are using P&T CP/M 2, you could use the SETUP utility to initialize the serial port, eliminating the need for LYNC 2.0 to set it up. You may still use initialization mode 1 and the initialization bytes shown but be aware that LYNC 2.0 will re-program the serial port which may cause some confusion for other programs that use the serial ports.

3. re: using LYNC 2.0 with P&T CP/M 2 for the TRS-80 Model II

When using LYNC 2.0 with P&T CP/M 2, you should always set the xmit on character to 0 on the serial port you are using. This will defeat all protocol within the serial port driver and will insure that there can be no conflict between the serial port driver and LYNC 2.0.

4. re: Superbrain parameters given in Appendix C

A user has reported to us that the port numbers given for the main and aux ports are reversed.