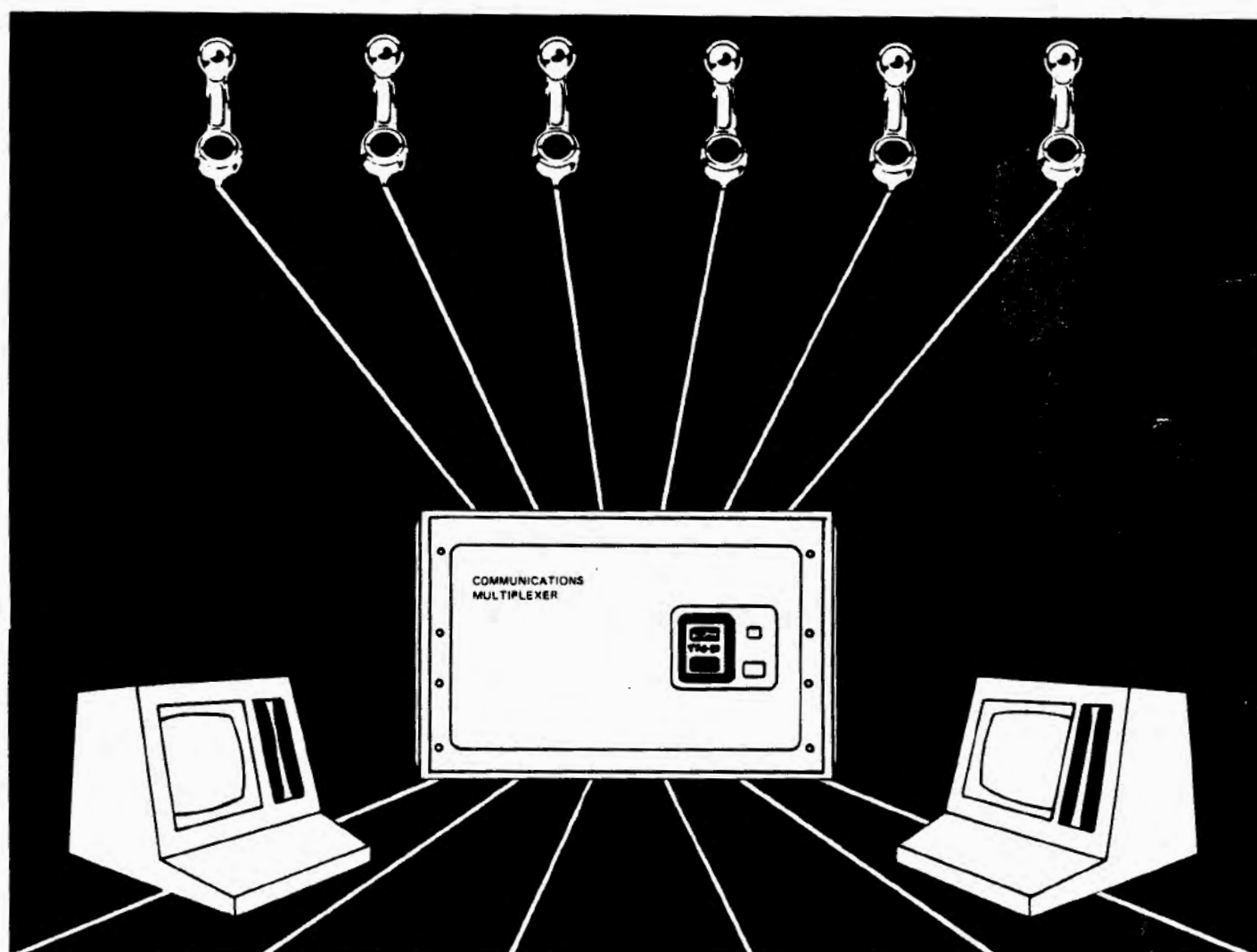


TRS-80[®] Communications Multiplexer


For use with TRS-80[®] Model II Computers

Catalog Number 71-3000/3001

Ax-8941



Radio Shack

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

Warning

This equipment generates and uses radio frequency energy. If not installed and used in accordance with the manufacturer's instructions, it may cause interference to radio and television reception.

As temporarily permitted by regulation, this equipment has not been tested for compliance with the limits for Class A computing devices in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation.

Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures are necessary to correct the problem.

*Communications Multiplexer Owner's Manual: © 1981
Tandy Corporation, Fort Worth, Texas 76012 U.S.A.
All Rights Reserved.*

Reproduction or use, without express written permission from Tandy Corporation, of any portion of this manual is prohibited. While reasonable efforts have been taken to assure its accuracy, Tandy Corporation assumes no liability resulting from any errors or omissions in this manual or from the use of the information obtained herein.

A BRIEF DESCRIPTION

The Communications Multiplexer controls up to 16 telephone lines, retrieves stored data from disk, and sends that data over the phone lines to callers using Videotex or other compatible terminals requesting that data. This data may also be updated concurrently with the requests.

The major components of this system are the Multiplexer (MUX), the Store and Forward (SF) and the Store and Forward Updater (SFU). The SF and SFU are standard TRS-80 Model II computers. The SFU uses TRSDOS 2.0a as its operating system. The SF and MUX operating systems are provided on the SF disk.

In addition to reading this manual, we also recommend that you read the following manuals:

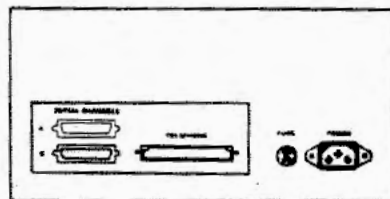
- Model II TRSDOS Manual (Cat. No. 26-4920)
- Model II Scripsit Manual (Cat. No. 26-4530)
- Tandy Text Editor (TED) Manual (Cat. No. 26-4710)

HOOK - UP

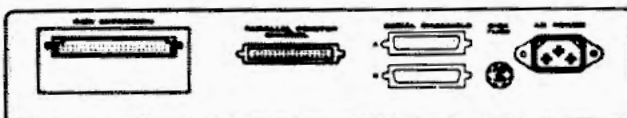
Before connections are made, you must have the local telephone company install a telephone rotary system. You must also notify the phone company of the:

Product: Communications Multiplexer.
Model No. 71-3000/3001
Manufacturer: Radio Shack
FCC Number: AA099R - 68793 - DM - R
Ringer Equivalence Number (REN): 0.7B

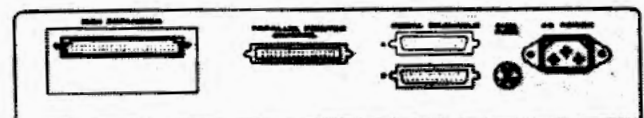
1. Connect the telephone rotary outlet to the large connector (labeled TELEPHONE) on the MUX using the provided telephone cable.
2. Use one of the provided RS-232 cables to connect the Channel A connector on the MUX to Serial Channel A on the rear of the computer to be used as the SF.
3. Use the other RS-232 cable to connect Serial Channel B on the SF computer to Serial Channel A on the computer to be used as the SFU.
4. Connect all power cords (MUX, SF, SFU) to 120 volt, 60 Hz, AC power outlets.



MUX



SF Computer



SFU Computer

COMMUNICATIONS SYSTEM CONNECTIONS

HOW THE COMMUNICATIONS MULTIPLEXER WORKS

Although at this point the communications system is not quite functional, it is important to understand its basic operation. Refer to **Figure 1** while reading the following description.

When power is applied, the SF will send the MTS/MUX operating system to the MUX. All other user tasks are inactive during this transmission. After the MUX receives its operating system, it begins answering calls and the system begins operation.

The telephone company rotary finds the first non-busy channel on the MUX when a phone call is received. The MUX "talks" to the caller's terminal and extracts the user identification code, sending it to the SF. The SF decodes the information request and translates it to menu request numbers. The SF locates pages from the menu and sends the data to the MUX which sends the data to the caller's terminal. The phone call is then terminated and the MUX waits for the next call. The SFU can send update information to the SF during this process. The existing data will be used for each request until the end of the update, at which time the updated text will then become available.

The software used by the MUX and the SF is MTS, an operating system which allows concurrent execution of tasks (referred to as "multi-tasking"). There are tasks for handling each channel, called user tasks, and tasks for management of resources, known as system tasks.

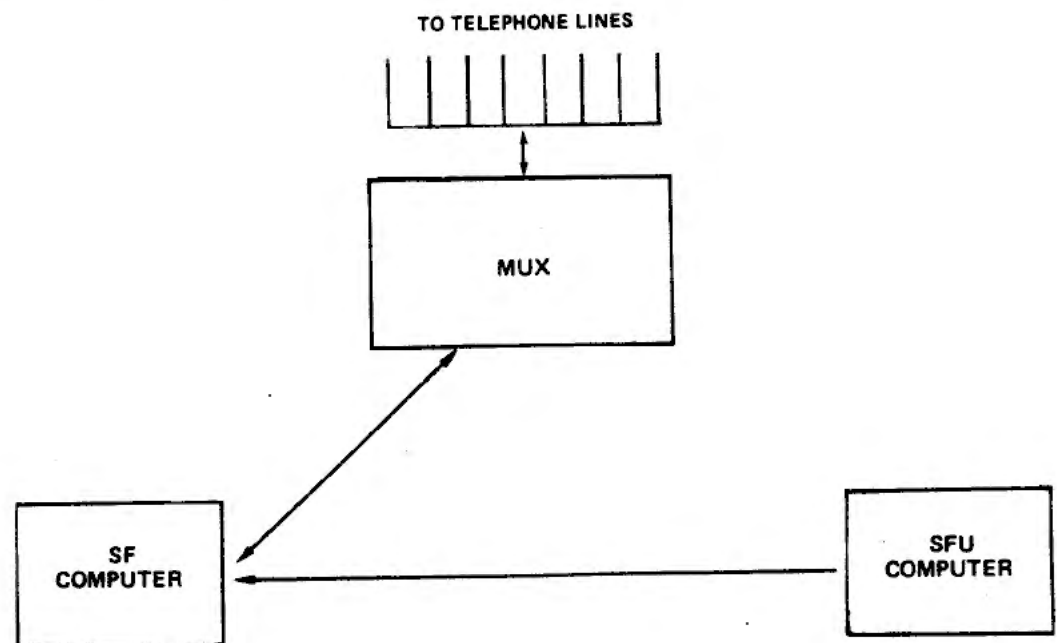


FIGURE 1. Communications System Block Diagram

GETTING STARTED

Before start - up of the system, the following files must be "on - line":

- | | |
|--------------|---|
| MTS/SYS | - operating system for the SF; provided on the SF diskette. |
| MTS/MUX | - operating system for the Multiplexer; provided on the SF diskette. |
| MXLOAD | - sends MTS/MUX to the Multiplexer; provided on the SF diskette. |
| DATABASE/DAT | - the information to be sent over the telephone lines to the requesting terminals; not provided. Refer to <i>Updating a Database File</i> for information on maintaining this file. |
| DATABASE/NDX | - index to the DATABASE/DAT file; initially empty. This file is updated along with DATABASE/DAT. |
| USERHOOK | - checks user identification, decodes information requests, sends data to the MUX, etc.; not provided. Refer to the Appendix for information on creating this file. |
| INIT/CMD | - executes system operation: provided on the SF diskette. This file consists of commands to: |

set date	(DATE MM/DD/YY)
set time	(TIME HH,MM,SS)
download MUX	(EXEC MXLOAD)
start channels	(START MUXCHN00)

(START MUXCHN15)

Editing INIT/CMD is accomplished by using Tandy Text Editor (TED). It is desirable to edit INIT/CMD before starting the system so that the date and time are correct.

Before operating the system, it is necessary to transfer the SFU software (file "SFU") from the SF disk to the TRSDOS disk to be used in the SFU computer. It is also wise to make a "working copy" of the SF diskette and save the original for making additional copies as needed. Use TRSDOS 2.0a and the BACKUP command to make duplicates of the diskette.

START - UP PROCEDURES

1. Power up or reset the MUX, SF, and SFU computers.
2. Insert the SF diskette into Drive 0 when prompted. The boot - up diagnostics will start (64K MEMORY displayed on screen) and then **INITIALIZING** will appear on the screen.
3. The operating system will be loaded and executed by the INIT/CMD file. After the operating system is loaded, the message **INDEX FILE SUCCESSFULLY LOADED** will appear in the upper left part of the screen. The message **ENTER SYSTEM COMMAND** will appear in the lower left of the screen.
4. No other commands are needed for the system to function, however, for debugging the system or for manual control purposes, CI commands may be used directly on the SF.

CI COMMANDS

The following notations are used in defining the commands:

()	denotes an optional repeatable parameter
ADDR	-- a one to four digit hex memory address (leading zeros are assumed if less than four digits are entered).
BP#	single digit breakpoint number (1 to 8)
BYTE	-- two digit hex value
FILESPEC	-- standard TRSDOS file specification
LINE#	one or two digit decimal number denoting a line on the video screen
RPN	register pair name (AF, BC, DE, HL, IX, IY, AF', BC', DE', HL')
TASK#	number of task (two digit decimal value)
TASKNAME	name of task (same as FILESPEC, but without extension, password, or drive number)
WORD	-- a four digit hex value

The following are the actual CI commands:

- *** — This symbol must be followed by **LINE#** and text. This command prints text on the SF screen at **LINE#** given.
- CLS** — Clears the video screen.
- DATE** — This command prints the current system date. If a new date is entered in **MM/DD/YY** format, the date is set to the value given.
- DEFPRI** — Must be followed by **BYTE**. This command sets default priority given to any task subsequently loaded to the value given by **BYTE**.
- DEFTIM** — Must be followed by a one to five digit decimal value. **DEFTIM** sets the default time slice given to any task which gets loaded after this command to the decimal value given.
- EXEC** — Must be followed by **FILESPEC**. This command loads the **FILESPEC** (the **FILESPEC** becomes **TASKNAME**) and executes the **TASKNAME**.
- LOAD** — Must be followed by **FILESPEC**. This command simply loads **FILESPEC** without starting the task. **FILESPEC** becomes **TASKNAME**.
- MINPRI** — Must be followed by **BYTE**. **MINPRI** sets the minimum priority a task must have in order to get execution time. Any task with a lower priority will not be executed.
- NAME** — Must be followed by **TASK#**. This command shows the corresponding **TASKNAME** on the screen.
- PRI** — Must be followed by **BYTE** and **TASK#**, respectively. This command assigns a priority (**BYTE** value) to a task (designated by **TASK#**).
- SHOW** — This command prints different information on the screen depending upon what command or letter follows the **SHOW** command. The options are listed below:
 - S** shows status of system tasks
 - U** shows status of user tasksRefer to *Appendix B* for definitions of the state codes used with this command.

- SLICE** – Must be followed by a one to five digit decimal value and a TASK#. This command assigns a maximum amount of time (the decimal value) which is sliced for use by a specific task (identified by TASK#).

- STOP** – Must be followed by TASKNAME. This command puts the task in a halted condition.

- TIME** – This command prints the current system time. If a new time entered in HH,MM,SS format, the system time is set to the value specified.

- UNLOAD** – Must be followed by TASKNAME. This command removes the designated task from memory (i.e., the inverse of LOAD).

- DEBUG** – Must be followed by the TASKNAME to be debugged. Once in the Debug mode, the previous commands are inactive and only the following commands can be used.
 - BRK** – must be followed by BP# and ADDR. This command sets a breakpoint at designated ADDR. If ADDR = 0, the breakpoint is removed.

 - CHG** – if followed by ADDR, BYTE, (BYTE), this command sets memory at ADDR to first BYTE value given; if a second BYTE value is given, the memory is set at ADDR + 2 to the second BYTE value; if a third value is given, the memory is set at ADDR + 2 to the third BYTE value, etc. If this command is followed by RPN, WORD, the value given is placed into register pair.
 NOTE: A comma must follow the RPN to distinguish it from an ADDR (e.g.: CHG BC,12 will change the BC register pair, whereas CHG BC 12 will change memory location 00BC).

 - EBT** – empties the breakpoint table (there are eight breakpoints).

 - GO** – when used alone, this command activates the present task in DEBUG with present PC. If followed by DDR, it sets the PC to ADDR and activates the task under DEBUG.
 NOTE: In both cases the BREAK key is used to return control to Debug. Control is also returned if a breakpoint is encountered.

 - IN** – must be followed by PORT#. This command reads the input port (given by PORT#) into the A register.

- MOV must be followed by startADDR, stopADDR, and destinationADDR. This command moves memory data in range given by startADDR and stopADDR to an equal sized block of memory starting at destinationADDR.
- OUT - must be followed by PORT# and BYTE. This command outputs BYTE to output port given by PORT#.
- RAM - if followed by ADDR, this command will display 128 byte block of memory (in hex and ASCII) starting at ADDR. If followed by N, it will display the next 128 byte block of memory. If followed by P, the previous 128 byte block of memory is displayed.
- SYS - this command quits DEBUG and returns to system mode. The system commands now become active and the task under DEBUG is left in a stopped condition.

UPDATING A DATABASE FILE

The SFU is used to update the database file (DATABASE/DAT) which has a fixed record length of 256. Text files are created using SCRIPSIT then are merged onto a non-Scriptit disk for transmission. Please read this entire section before starting on the file.

The first character typed into a SCRIPSIT file must be a menu mark. This mark is obtained by pressing **< CONTROL > < 6 >** on the TRS-80 Model II keyboard**, and appears as "~" on the video screen. The menu mark is followed by a MENU# and **< ENTER >**.

The pages for that menu will be typed next. Each page must begin with a page mark obtained by pressing **< CONTROL > < 9 >** and displayed as "\". The page mark should be immediately followed by **< ENTER >**. All characters after the page mark and **< ENTER >** will be considered as page data up to the next page mark, menu mark, or the end of the file. (It should be noted that any lower case letters in the file will appear as upper case letters on the Videotex screen.)

The length of data must be between 0 and 16 pages. If 0 pages are transmitted to the SF during an update, the entire designated MENU# will be deleted from the file.

When the file is complete, it should be merged onto a non-Scriptit disk using the **SCRIPSIT MERGE** utility.

Database Rules

1. MENU#'s must be between 0000 and 9999.
2. MENU# 9999 must be present. This menu should contain a message similar to "The information requested cannot be found. Please check and retype your request". MENU# 9999 will be sent if a MENU# requested cannot be found.
3. As many as 600 MENU#'s may be used if they are one page each. Otherwise, there will be less than 600 MENU#'s. The number of pages is the limiting factor.
4. There must be a maximum of 16 pages per MENU#. Each page may contain up to 16 data lines (not counting page mark) with no more than 32 characters per line.
5. Do not use tabs or the Hyphenate function in SCRIPSIT as these control codes are not screened out during the MERGE process and will cause alterations in the appearance of data or could cause the SFU software to abort transmission altogether.

** If a terminal other than a Model II is used, the keys for menu mark and page mark will vary.

Sample input stream:

~ 301

\

THIS IS THE FIRST PAGE OF AN
ITEM. THE PAGE COULD CONTAIN AS
MANY AS 512 CHARACTERS,
INCLUDING SPACES, BUT NOT
INCLUDING THE PAGE MARK. NOTE
THAT ANY CARRIAGE RETURN WILL
GET STORED AS TWO CHARACTERS ONE
FOR A 0DH AND ANOTHER FOR A 0AH.

----- imaginary SCRIPSIT page boundary -----

\

THIS WOULD BE THE SECOND PAGE
ON THE MENU. IT IS SEPARATED
FROM THE PREVIOUS PAGE BY THE
PAGE MARK (\).

----- imaginary SCRIPSIT page boundary -----

~ 302

\

THIS WOULD BE THE NEXT MENU. IT
IS SEPARATED FROM THE PREVIOUS
MENU BY THE MENU MARK, MENU#,
AND PAGE MARK.

USING THE SFU PROGRAM

The SFU program is used to transfer the SCRIPSIT-created text files for inclusion in the DATABASE file.

To start execution of the SFU program, type:

SFU (filename)

on the TRSDOS command line. The filename is the name of the file to be transmitted to the SF.

The SFU program will automatically initialize the serial port drivers, prompt for a filespec (if one was not entered on the command line), and open the source file.

Menus are read one at a time and screened for format errors before transmitting. When the entire source file has been transmitted, a prompt for a new filespec will be given. If it is desired to transmit another file during this session, enter the new filename. To terminate the session press **< ENTER >**.

The Break Key

Pressing **< BREAK >** will terminate the update session when the SFU program is editing or transmitting. This termination should only be done in emergencies as the session will be cut-off after transmission of the current page, which could result in a partially updated menu. **< BREAK >** may not be used during prompting for a new filespec.

Input Errors

If a syntax error is detected, the current menu being edited will not be transmitted, the current filespec will be closed, and a new filespec will be requested. All previous menus will have been transmitted to the SF and therefore updated. The last MENU# transmitted will be displayed on the CRT screen.

Other Errors

Some errors, such as drive door open, etc., can be corrected by the operator. This type of error will cause **CONTINUE?** to be displayed on the screen after the error message is displayed.

If **< Y >** **< ENTER >** is typed in response to the prompt, the program will repeat the operation it was performing when the error occurred. If **< N >** is entered, the program will terminate the session, close the file, and enter the "halt" state. Some errors, however, will abort the transmission of the current file and the program will enter the "halt" state automatically. If this occurs, the program can be restarted by pressing any key or you can restart TRSDOS by pressing **< BREAK >**. Refer to *Appendix C* in the back of this manual for technical information on how the SFU program works.

APPENDIX A

This appendix provides the information necessary for an experienced programmer to write a Userhook routine. To develop a Userhook program, you must be able to use TRSDOS, the Editor/Assembler (TED), and Z-80 Assembly Language.

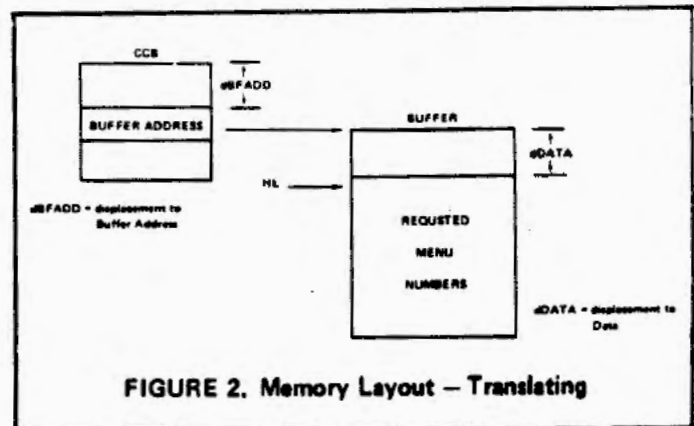
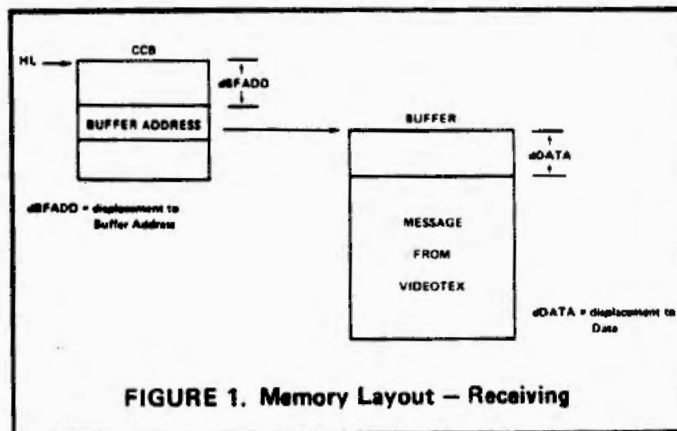
An example program and its description is supplied. This program may be used, with the exception of a few sections which will need to be customized to fit your specific installation and use.

The first part of this program, as marked, should not be altered. This section defines the offset of the Channel Control Block (CCB), Buffer Address, Data Address, etc. The rest of the program has been divided into sections and sub-sections for easy explanation.

Section 1 of the program receives the request for data from the operating system. The HL register pair contains the CCB location. The CCB is a small block in memory which contains a Buffer Address (HL + dBFADD). This address designates the first character of the request data (Buffer Address + dDATA). See Figure 1.

Section 2 of the program reads the request and translates it into the appropriate Menu numbers. See Figure 2. This section can also be used to keep a record of how many calls were received, for requiring a password, etc. These procedures are performed by a subroutine called RWDET, which is explained below.

Section 3 returns the CCB to the operating system.



RWDET Subroutine Description

Subroutine RWDET performs the actual translation of the request message into a string of Menu numbers. This subroutine, and its associated data tables, is the only part of the sample program which would vary depending upon the installation and use of the system. To help you understand this subroutine, it has been divided into different parts with an explanation of each part given below. For most systems, a simple re-coding of the data tables (part F) is all that is needed.

Part A of the subroutine has the job of filtering out leading spaces.

Example:

```

  00INFO (Before)
  ↑
  HL
  00INFO (After)
  ↑
  HL
```

Part A then puts the Address Data into register HL.

In **Part B**, the length is first loaded into register BC. Register pair HL is then incremented by one. The Operating System Service, SVCCMP, is used to compare the request message with each reserved word. This comparison continues until a match is found or a length of zero (0) is reached indicating the end of the table or no match. If a match is found the subroutine goes to Part C. If a match is not found the subroutine goes to Part D.

Part C replaces the request message with the list of Menu numbers found at the Menu number address following the matched reserved word.

Part D returns the original message unchanged. If the original message is not a valid list of Menu numbers, MENU# 9999 will be used.

Part E consists of the Reserved Word Table (RW table) which is structured like this:

length	reserved word
address of menu number list	
length	reserved word
address of menu number list	
⋮	
0	end of table

Part F contains the Menu number list which corresponds to entries in the RW table. Each list of Menu numbers consists of a list of numbers separated by commas and is terminated by two periods or two commas.

USERHOOK SAMPLE PROGRAM

USERHOOK FOR DEMOS

MAIN

Addr Obj Fl Ln # Source Line

			00130	*****
			00140	;
			00150	;
			00160	USERHOOK ROUTINE
			00170	;
			00180	THIS ROUTINE TRANSLATES A VIDEOTEX REQUEST INTO A SET OF VALID
			00190	MENU NUMBERS. IF THE REQUEST TEXT IS NOT RECOGNIZED AS A ONE OF
			00200	THE ALLOWABLE ITEMS, THEN THE TEXT IS NOT MODIFIED.
			00210	;
			00220	;
			00230	;
			00240	;
			00250	;
			00260	;
			00270	;
			00280	;
			00290	;
			00300	;
			00310	;
			00320	;
			00330	;
			00340	;
			00350	;
			00360	;
			00370	;
			00380	;
			00390	;
			00400	;
			00410	;
			00420	;
			00430	;
			00440	;
			00450	;
			00460	;
			00470	;
			00480	;
			00490	;
			00510	;
			00520	;
			00530	;
			00540	;
			00550	;
			00560	;
			00570	;
			00580	;
			00590	;
			00600	;
			00610	;
			00620	;
			00630	;
			00640	;
			00650	;
			00660	;
			00670	;
			00680	;
			00690	;
			00700	;
			00710	;
			00720	;
			00730	;
			00740	;
			00750	;
			00760	;
			00770	;
			00780	;
			00790	;
			00800	;
			00810	;
			00820	;
			00830	;
			00840	;
			00850	;
			00860	;
			00870	;
			00880	;
			00890	;
			00900	;
			00910	;
			00920	;
			00930	;
			00940	;
			00950	;
			00960	;
			00970	;
			00980	;
			00990	;
			01000	;
			01010	;
			01020	;
			01030	;
			01040	;
			01050	;
			01060	;
			01070	;
			01080	;
			01090	;
			01100	;
			01110	;
			01120	;
			01130	;
			01140	;
			01150	;
			01160	;
			01170	;
			01180	;
			01190	;
			01200	;
			01210	;
			01220	;
			01230	;
			01240	;
			01250	;
			01260	;
			01270	;
			01280	;
			01290	;
			01300	;
			01310	;
			01320	;
			01330	;
			01340	;
			01350	;
			01360	;
			01370	;
			01380	;
			01390	;
			01400	;
			01410	;
			01420	;
			01430	;
			01440	;
			01450	;
			01460	;
			01470	;
			01480	;
			01490	;
			01500	;
			01510	;
			01520	;
			01530	;
			01540	;
			01550	;
			01560	;
			01570	;
			01580	;
			01590	;
			01600	;
			01610	;
			01620	;
			01630	;
			01640	;
			01650	;
			01660	;
			01670	;
			01680	;
			01690	;
			01700	;
			01710	;
			01720	;
			01730	;
			01740	;
			01750	;
			01760	;
			01770	;
			01780	;
			01790	;
			01800	;
			01810	;
			01820	;
			01830	;
			01840	;
			01850	;
			01860	;
			01870	;
			01880	;
			01890	;
			01900	;
			01910	;
			01920	;
			01930	;
			01940	;
			01950	;
			01960	;
			01970	;
			01980	;
			01990	;
			02000	;
			02010	;
			02020	;
			02030	;
			02040	;
			02050	;
			02060	;
			02070	;
			02080	;
			02090	;
			02100	;
			02110	;
			02120	;
			02130	;
			02140	;
			02150	;
			02160	;
			02170	;
			02180	;
			02190	;
			02200	;
			02210	;
			02220	;
			02230	;
			02240	;
			02250	;
			02260	;
			02270	;
			02280	;
			02290	;
			02300	;
			02310	;
			02320	;
			02330	;
			02340	;
			02350	;
			02360	;
			02370	;
			02380	;
			02390	;
			02400	;
			02410	;
			02420	;
			02430	;
			02440	;
			02450	;
			02460	;
			02470	;
			02480	;
			02490	;
			02500	;
			02510	;
			02520	;
			02530	;
			02540	;
			02550	;
			02560	;
			02570	;
			02580	;
			02590	;
			02600	;
			02610	;
			02620	;
			02630	;
			02640	;
			02650	;
			02660	;
			02670	;
			02680	;
			02690	;
			02700	;
			02710	;
			02720	;
			02730	;
			02740	;
			02750	;
			02760	;
			02770	;
			02780	;
			02790	;
			02800	;
			02810	;
			02820	;
			02830	;
			02840	;
			02850	;
			02860	;
			02870	;
			02880	;
			02890	;
			02900	;
			02910	;
			02920	;
			02930	;
			02940	;
			02950	;
			02960	;
			02970	;
			02980	;
			02990	;
			03000	;
			03010	;
			03020	;
			03030	;
			03040	;
			03050	;
			03060	;
			03070	;
			03080	;
			03090	;
			03100	;
			03110	;
			03120	;
			03130	;
			03140	;
			03150	;
			03160	;
			03170	;
			03180	;
			03190	;
			03200	;
			03210	;
			03220	;
			03230	;
			03240	;
			03250	;
			03260	;
			03270	;
			03280	;
			03290	;
			03300	;
			03310	;
			03320	;
			03330	;
			03340	;
			03350	;
			03360	;
			03370	;
			03380	;
			03390	;
			03400	;
			03410	;
			03420	;
			03430	;
			03440	;
			03450	;
			03460	;
			03470	;
			03480	;
			03490	;
			03500	;
			03510	;
			03520	;
			03530	;
			03540	;
			03550	;
			03560	;
			03570	;
			03580	;
			03590	;
			03600	;
			03610	;
			03620	;
			03630	;
			03640	;
			03650	;
			03660	;
			03670	;
			03680	;
			03690	;
			03700	;
			03710	;
			03720	;
			03730	;
			03740	;
			03750	;
			03760	;
			03770	;
			03780	;
			03790	;
			03800	;

Source: [Link](#)

A

B

C

D

16

Source Line

17

APPENDIX B

MUX CHANNEL STATE CODES

The **SHOW U** command on the MUX terminal gives the state code for each of the 16 lines (task name MUXCHN00 through MUXCHN15). Following is a list of the codes and their meanings:

Code	Meaning
0	Channel is being initialized
1	Waiting for SF to OK handling a call
2	SF did not ok handling a call (error condition)
4	Waiting for a call
5	Waiting for a list of desired MENU#'s
8	Aborting call due to bad menu block
12	Sending menu block to SF
14	Sending "Receiving Page xx" message
16	Sending data block to Videotex
18	Sending request for next block to SF
32	Terminating call
48	Disabling channel hardware
50	Sending error message to SF

Codes other than those listed are not used by the software. Occurrence of any unused codes in the **SHOW U** display is an error.

APPENDIX C

USER SFU PROCEDURES

This section will furnish the experienced programmer with enough information to write a customized SFU program.

Control characters used in data transmission are:

<STX>	02 Hex	signifies start of update
<SOH>	01 Hex	signifies start of an item (file)
<FS>	1C Hex	field separator
<RS>	1E Hex	record separator, signifies end of current item
<ETX>	03 Hex	signifies end of current update
<ACK>	06 Hex	block received acknowledgement
<NAK>	15 Hex	signifies a negative acknowledge

Updates are made through the SF Serial Channel B at 9600 Baud, 8 bit word length, no parity, and 1 stop bit. An update sequence begins when the SF receives a <STX> character. It is terminated by an <ETX> character.

An update sequence consists of one or more items (files or menus). Each item must be transmitted in two or more sections:

- 1) a header block followed by
- 2) 0 to 16 pages of data; if 0 pages are transmitted then the entire MENU# will be deleted from the file.

Each header block has five parts:

- 1) <SOH>
- 2) MENU#
- 1) <SOH>
- 2) MENU# (one to four ASCII decimal digits)
- 3) <FS>
- 4) number of pages (one or two ASCII decimal digits)
- 5) <RS>

Note that the MENU# is the same as the FILE# which is used to reference an item after it has been stored in the SF.

Each page of data (up to 16 page per file) consists of 1 to 512 bytes of data followed by a <RS>. Data values should be restricted to values greater than 1F hex except for 09 hex (tab), 0A hex (line feed), and 0D hex (carriage return).

The SF will respond with an <ACK> or a <NAK> after it receives the <STX> and after each <RS>.

Example of a standard update sequence:

```
< STX > *  
< SOH > < '001' > < FS > < '02' > < RS > *  
< '----- ASCII Data -----' > < RS > *  
< '----- ASCII Data -----' > < RS > *  
< SOH > < '0002' > < FS > < '03' > < RS > *  
< '----- ASCII Data -----' > < RS > *  
< '----- ASCII Data -----' > < RS > *  
< '----- ASCII Data -----' > < RS > *  
< ETX >
```

NOTE: Each * signifies where the transmitter must wait for an < ACK > .

The above example updates MENU# 1 with two pages of data and MENU# 2 with three pages. Note that if an item header declares that there will be X number of pages, the SF must receive X pages for that menu before it will accept an < ETX > or a < SOH > .

SERVICE POLICY

Radio Shack's nationwide network of service facilities provides quick, convenient, and reliable repair services for all of its computer products, in most instances. Warranty service will be performed in accordance with Radio Shack's Limited Warranty. Non-warranty service will be provided at reasonable parts and labor costs.

Because of the sensitivity of computer equipment, and the problems which can result from improper servicing, the following limitations also apply to the services offered by Radio Shack:


1. If any of the warranty seals on any Radio Shack computer products are broken, Radio Shack reserves the right to refuse to service the equipment or to void any remaining warranty on the equipment.
2. If any Radio Shack computer equipment has been modified so that it is not within manufacturer's specifications, including, but not limited to, the installation of any non-Radio Shack parts, components, or replacement boards, then Radio Shack reserves the right to refuse to service the equipment, void any remaining warranty, remove and replace any non-Radio Shack part found in the equipment, and perform whatever modifications are necessary to return the equipment to original factory manufacturer's specifications.
3. The cost for the labor and parts required to return the Radio Shack computer equipment to original manufacturer's specifications will be charged to the customer in addition to the normal repair charge.

LIMITED WARRANTY

For a period of 90 days from the date of delivery, Radio Shack warrants to the original purchaser that the computer hardware unit shall be free from manufacturing defects. This warranty is only applicable to the original purchaser who purchased the unit from Radio Shack company-owned retail outlets or duly authorized Radio Shack franchisees and dealers. This warranty is voided if the unit is sold or transferred by purchaser to a third party. This warranty shall be void if this unit's case or cabinet is opened, if the unit has been subjected to improper or abnormal use, or if the unit is altered or modified. If a defect occurs during the warranty period, the unit must be returned to a Radio Shack store, franchisee, or dealer for repair, along with the sales ticket or lease agreement. Purchaser's sole and exclusive remedy in the event of defect is limited to the correction of the defect by adjustment, repair, replacement, or complete refund at Radio Shack's election and sole expense. Radio Shack shall have no obligation to replace or repair expendable items.

Any statements made by Radio Shack and its employees, including but not limited to, statements regarding capacity, suitability for use, or performance of the unit shall *not* be deemed a warranty or representation by Radio Shack for any purpose, nor give rise to any liability or obligation of Radio Shack.

EXCEPT AS SPECIFICALLY PROVIDED IN THIS WARRANTY OR IN THE RADIO SHACK COMPUTER SALES AGREEMENT, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL RADIO SHACK BE LIABLE FOR LOSS OF PROFITS OR BENEFITS, INDIRECT, SPECIAL, CONSEQUENTIAL OR OTHER SIMILAR DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR OTHERWISE.

RADIO SHACK  A DIVISION OF TANDY CORPORATION

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

TANDY CORPORATION

AUSTRALIA

280-316 VICTORIA ROAD
RYDALMERE, N.S.W. 2116

BELGIUM

PARC INDUSTRIEL DE NANINNE
5140 NANINNE

U. K.

BILSTON ROAD WEDNESBURY
WEST MIDLANDS WS10 7JN