

TC-8 cassette operating system

users manual





TC-8

high speed cassette system

JPC PRODUCTS COMPANY agrees to repair or replace, at our option, any TC-8 Cassette Interface which the purchaser has determined to be defective. No other warranty is expressed or implied. **JPC PRODUCTS COMPANY** shall have no liability to any entity with respect to loss or damage alleged to be caused by equipment or computer programs sold by us or by others, including but not limited to lack of merchantability or suitability for any purpose, or for interruption of service, or for loss of business or profits. Computer programs are sold by **JPC PRODUCTS COMPANY** on a strictly AS IS basis.

© COPYRIGHT 1980 PRINTED IN U.S.A.

Instruction Manual Release 4

TABLE OF CONTENTS

INTRODUCTION.....	1
LOADING UTIL.....	2
LOADING BOOT.....	5
DEFINITION OF TERMS.....	8
COMMANDS.....	11
SAVE.....	12
LOAD.....	13
LOAD?.....	14
LOADN.....	15
KILL.....	16
RSET.....	17
RUN.....	18
PUT.....	19
GET.....	20
GET?.....	21
GETN.....	22
STATEMENTS.....	23
OPEN.....	24
CLOSE.....	24
PRINT #.....	25
INPUT #.....	26
USEFUL INFORMATION.....	29
TC-8 CONNECTION.....	30
THE INDICATORS.....	31
THE TINY MONITOR.....	32
UTIL BACKUP.....	35
RELOCATING UTIL.....	37
PROGRAM CONVERSION.....	39
KEYBOARD DEBOUNCE.....	42
USE WITH PRINTER.....	42
GOOD PRACTICES.....	43
CTR-41 MODIFICATION.....	47

Introduction

Any computer, regardless of size, is only as good as its mass storage memory. That is where programs and data are stored when they are larger than the Random Access Memory (RAM) can hold. And since the computer uses many different programs and more than one type of data, these are stored in the mass memory and only one program or portion of the data is brought into the working memory at a time. There are many different types of mass storage: hard disk, floppy disk, bubble memory, paper tape, cassette tape, etc. Of these, magnetic storage on cassette tape offers an excellent compromise between performance and cost. A single cassette can store millions of bits of data at very reasonable cost. But to be a useful mass storage memory, the cassette interface must be fast, reliable, and must include a capable file manager. The standard TRS-80 cassette system does not adequately meet these requirements.

The TC-8 and the programs on the cassette that was supplied with it will make a significant improvement in the usefulness of the TRS-80.

The TC-8 provides an interface between the TRS-80 and the existing cassette recorder which increases the data rate by more than 5 times. The reliability is also substantially improved; the error rate for the modified Manchester pulse code used by the TC-8 has a demonstrated error rate of less than one error in a million bits.

UTIL is the file manager. It performs the routine tasks associated with finding, naming, saving, loading, and verifying programs. It also allocates space for the data files and packs the data efficiently into that space on the cassette. A system monitor, TINY, is provided. It is a tool, allowing you to look into memory or system tapes and identify important addresses, or to make changes to UTIL to customize it to your TRS-80 or to your special requirements.

LOADING UTIL

1. The cassette recorder should be plugged into the keyboard in the normal manner. The volume on the recorder should be set wherever you normally load SYSTEM tapes. The recorder should NOT be plugged into the TC-8 yet. The recorder motor may run continuously if plugged into the TC-8 prior to loading and executing UTIL.
2. Turn off power to the TRS-80. Refer to the "TRS-80 CONNECTION" and connect the TC-8 to the keyboard using the ribbon cable supplied. Be sure to get the red stripe in the correct orientation. Plug the power adapter cord into the TC-8. Plug the power adapter into a 115 V. AC outlet. Power up the TRS-80 computer. The "MEMORY SIZE" prompt should appear on the screen.

MEMORY SIZE? _

3. Type "31398".

MEMORY SIZE? 31398

WARNING: It is very important that memory size be set correctly or the stack may overwrite UTIL after it is loaded.

4. The LEVEL II banner should appear on the screen.

RADIO SHACK LEVEL II BASIC
READY
>

5. Place the UTIL cassette in the recorder and depress the "play" button.

6. Type "SYSTEM"

> SYSTEM

7. The "*?" prompt should appear on the screen. When it does, type "UTIL".

```
*? UTIL
```

8. The cassette should load, printing a message on the video screen which describes UTIL addresses. When the cassette has completely finished loading, a prompt will appear at the bottom of the video screen.

```
HIT ENTER TO CONTINUE....
```

9. Press the ENTER key. This will execute TINY, a small system monitor program which will allow you to relocate UTIL and perform some other tasks. TINY monitor commands are described in the useful information section. There is also a section which deals with relocation and backup of UTIL. The TINY prompt should now appear on the screen.

```
TINY MONITOR
 1 IDENTIFY TRS-80 SYSTEM FILE
 2 IDENTIFY TC-8 BINARY FILE
 3 EXAMINE MEMORY LOCATIONS
 4 RELOCATE UTIL PROGRAM.
 5 WRITE BACKUP COPY OF UTIL
 6 JUMP TO HEX ADDRESS
 7 RETURN TO BASIC
ENTER COMMAND > _
```

10. Now type "7".

```
ENTER COMMAND > 7
READY
>
```

11. You will return to BASIC. Type "SYSTEM". Then at the system prompt, type "/31400".

```
*? /31400
```

12. UTIL will report that it is ready and will return to BASIC.

UTIL READY

>

13. Rewind the UTIL cassette and store it in a safe place. Don't forget to make a backup copy soon.
14. Now unplug the cassette recorder from the back of the keyboard and plug it into the right hand jack on the TC-8 front panel. You are ready to load and save programs in high speed TC-8 format. Please read the manual for descriptions of the various UTIL commands. For initial checkout, type in a short BASIC program, "SAVE" it to cassette, and then "LOAD" it back in.
15. Now would be a good time to sit back and read the entire manual.

LOADING BOOT

To allow the greatest flexibility in using the File Manager, two entry points have been provided. UTIL is the larger program and supports all of the commands described in this manual. BOOT is a short version of UTIL that omits most of the commands to reduce memory requirements. You may prefer to use this version, since memory size is set to 32148. For very large BASIC programs, you may have to use the KILL command. Refer to the UTIL command description for further details on use of the KILL command.

1. Set memory size to 32148.
2. Load "UTIL" in accordance with steps 5 through 10 in "LOADING UTIL".
3. You will return to BASIC. then type "SYSTEM". At the SYSTEM prompt, type "/32150". This will execute only the BOOT portion.

```
*? /32150
```

4. The BOOT prompt should now appear on the screen.

```
BOOT READY  
>
```

5. The following commands will be supported:

```
LOAD  
LOADN  
LOAD?  
KILL
```

CAUTION ! -- If it becomes necessary to change the size of the File Manager from BOOT to UTIL or from UTIL to BOOT, start over with a fresh load. Otherwise you may leave incomplete vectors or corrupt code which can cause unpredictable results. Start at step 1.

File Manager

user's guide

DEFINITION OF TERMS

ARGUMENT	Additional term which modifies a statement or command.
BEGIN	Four hexadecimal characters used as a beginning memory address for binary files.
BINARY	An absolute memory image, usually of a Z-80 machine language program. Also a number system with a base of 2.
BYTE	Two hexadecimal characters representing the contents of a memory location (8 bits).
COMMAND	A BASIC instruction which is entered in the immediate mode in response to the > prompt.
DEFAULT	The condition which occurs in the case of no operator selection.
DRIVE	Parameter which selects one of two cassette recorders as the active device. Only drive 2 is specified in UTIL since drive 1 is the default drive.
END	Four hexadecimal characters used as an ending memory address for binary files.
FILE	A group of bytes written on the cassette as one block.

FILENAME Any 8 or less alphanumeric (ASCII) characters used as a file name. The name must not contain any of the file type characters nor any imbedded spaces.

FILETYPE A symbolic character denoting the type of file:
\$ = basic program files
% = binary files
& = source files

HEXADECIMAL Representation of a base-16 digit by a numeral 0-9 or a letter A-F.

SOURCE A non-executable text file which is the input for an editor, assembler, compiler, or word processor.

STATEMENT A BASIC instruction which is used in the execution mode as a program line.

SYNTAX The required format or grammar of the input command.

TRANSFER Four hexadecimal characters used as the absolute execution address for a memory resident binary program or its cassette file image.

WORD Four hexadecimal characters representing the contents of two consecutive memory locations (16 bits).

Commands

SAVE

Syntax:

```
SAVE "<FILENAME>"  
SAVE2 "<FILENAME>"
```

Description:

The SAVE command creates a BASIC file (file type \$) on the cassette and writes out the current contents of the BASIC program buffer. The program name <filename> must be specified or a syntax error ("SN") will be displayed and the SAVE aborted. The <filename> may be 8 characters or less. The <filename> entered may not include any imbedded spaces. The program will be saved on the right hand drive (drive 1) unless the "SAVE2" form is used, in which case the program will be saved on the left hand drive (drive 2). The "record" button on the drive must be depressed in order for the save to actually occur. The SAVE routine will automatically write a leader between files on the cassette. When the save is finished, it will return to BASIC. The BASIC program in memory is not affected by the save. After a save of an important program, it should be verified by the "LOAD?" command.

Restrictions:

The SAVE command may only be used for BASIC programs (file type \$). This command is not available in the short form "BOOT" version.

Examples:

```
> SAVE  
?SN ERROR  
READY  
>  
  
> SAVE "TEST"  
$TEST READY  
>  
  
> SAVE2 "LONGNAME"  
$LONGNAME READY  
>
```

LOAD

Syntax:

```
LOAD
LOAD "<FILENAME>"
LOAD2
LOAD2 "<FILENAME>"
```

Description:

The LOAD command reads a BASIC program (file type \$) from the cassette. If the optional argument <filename> is not specified, the next BASIC program encountered on the cassette will be loaded. If <filename> is specified, UTIL will search for the requested program, writing the names of all files encountered to the video screen. When the requested file is found, it will be loaded into memory and UTIL will return to BASIC. The default drive is the cassette recorder plugged into the right hand jack of the TC-8 (drive 1). If the form "LOAD2" is entered, the active drive will be the recorder plugged into the left hand jack on the TC-8 (drive 2), and the program will be loaded from that recorder. Any BASIC programs previously in memory will be lost.

Restrictions:

The LOAD command may only be used for BASIC programs (file type \$); other file types will be ignored.

Examples:

```
> LOAD
$BLACKJAK READY
>
```

```
> LOAD "STARTREK"
$BLACKJAK $STARTREK READY
>
```

```
> LOAD2 "TAXES"
$CHECKBK $BUDGET $TAXES READY
>
```

LOAD?

Syntax:

```
LOAD?  
LOAD? "<FILENAME>"  
LOAD?2  
LOAD?2 "<FILENAME>"
```

Description:

This form is the verify command. A BASIC file is read from the cassette but is not loaded into memory. The contents of the file are compared to the contents of memory. If the file is not exactly the same as memory, a memory ("M") error will result. This allows the operator to determine that a program that has been saved is good. If the optional argument <filename> is not specified, the next BASIC program encountered on the cassette will be read and verified. If <filename> is specified, UTIL will search for the requested program. When found, it will be read and verified and UTIL will return to BASIC. If the "BAD" message is printed upon return, the program on cassette is not error free. The verify command does not alter any programs in memory.

Restrictions:

The LOAD? command may only be used for BASIC programs (file type \$); other file types will be ignored.

Examples:

```
> LOAD?  
$BLACKJAK READY  
>  
  
> LOAD?2 "BUDGET"  
$CHECKBK $BUDGET READY  
>  
  
> LOAD?2 "CHECKBK"  
$CHECKBK BAD  
>
```

LOADN

Syntax:

```
LOADN
LOADN "<FILENAME>"
LOADN2
LOADN2 "<FILENAME>"
```

Description:

This form is the directory command. It is, in effect, a "load names" command that prints a list of all the programs on a cassette. If the optional argument <filename> is not specified, the command will continue to load names and print the directory until it is interrupted by the operator pressing the "BREAK" key. If <filename> is specified, the tape will stop accurately at the end of the BASIC program \$<filename>. This command is useful in the named mode because it can be used to position the tape prior to writing a new program on a cassette which already contains other files. The "load names" command does not alter any programs in memory.

Restrictions:

The named mode can only be used to position the tape at the end of BASIC programs (type \$). The unnamed mode will print a directory of all files encountered, regardless of file type.

Examples:

```
> LOADN
$BLACKJAK $STARTREK $NIM %TBUG %HIBUG

> LOADN "NIM"
$BLACKJAK $STARTREK $NIM >
```

KILL

Syntax:

KILL

Description:

The KILL command removes the file manager UTIL from memory to release that memory to very large BASIC programs which need the full memory space. Memory size is reset by this command which will allow BASIC to utilize high memory, except for the keyboard debounce routine which is left in place (The top 45 bytes of memory).

CAUTION !!

After the KILL command has been executed, DO NOT attempt to use any of the UTIL commands until UTIL itself has been reloaded. Totally unpredictable results will occur. Once UTIL has been "killed" it may not be used until reloaded. KILL does not affect either the BASIC program in memory or the files on cassette.

Example:

```
> LOAD "STARTREK"  
$BLACKJAK $STARTREK READY  
> RUN  
?OM ERROR IN 20  
READY  
> KILL  
READY  
> RUN
```

RSET

Syntax:

RSET
RSET2

Description:

This is the "recorder setup" command which allows the operator to rewind and position the tape on older recorder models which have these functions tied to the motor control jack. The use of this command eliminates the need to remove the motor control plug everytime tape setup is required. With the newer CTR-80 recorders, these functions are independent and so the "recorder setup" command will be of little value.

When the RSET command is used, the recorder motor will remain on until the "BREAK" key is depressed by the operator. Return is to BASIC.

Restrictions:

This command is not available in the short form "BOOT" version.

Example:

```
> RSET  
                                (PRESS BREAK KEY)  
READY  
>  
  
> RSET2  
                                (PRESS BREAK KEY)  
READY  
>
```

RUN

Syntax:

RUN "<FILENAME>"

Description:

The RUN "<FILENAME>" command is a convenient combination of the UTIL load function and BASIC's RUN function, in effect, load-and-run-immediately. UTIL will search the cassette for the BASIC program <filename> and when it is encountered, will load it into the BASIC program buffer. UTIL will then jump to the RUN entry point in BASIC so that the program immediately begins execution. Programs to be run are loaded from cassette drive 1. Any BASIC program previously in memory is lost.

Restrictions:

The RUN "<FILENAME>" command may only be used for BASIC programs (file type \$). RUN without a <filename> is not allowed since that verb is reserved by BASIC. Likewise, RUN2 "<FILENAME>" is not allowed because that form is used by BASIC to begin execution at a line number. File types other than \$ will be ignored by this command. This command is not available in the short form "BOOT" version.

Example:

```
> RUN "BLACKJAK"  
1 OR 2 PLAYERS _
```

PUT

Syntax:

```
PUT "<FILENAME>" <BEGIN>,<END>,<TRANSFER>  
PUT2 "<FILENAME>" <BEGIN>,<END>,<TRANSFER>
```

Description:

The PUT command is the binary (system) equivalent of the SAVE command. It allows you to put binary programs on cassette for future use. Binary files are programs such as KBFIX, EDTASM, OR TBUG. The <filename> must be specified or a syntax error will be displayed and the PUT aborted. The <filename> may be 8 characters or less and may not include any imbedded spaces. All memory locations from <begin> through <end> will be put on the tape. The address <transfer> is the entry point into the program and must be specified. If no transfer addresses is desired, then 0000 must be entered for <transfer>. When the PUT is finished, it will return to BASIC. The contents of memory are not altered. After a put of an important file, it should be verified by the "GET?" command.

Restrictions:

The PUT command may only be used for binary files (file type %). This command is not available in the short form "BOOT" version.

Examples:

```
> PUT "HIBUG" 7380,7830,73A0  
%HIBUG READY  
>
```

```
> PUT "WRONG" 7380,7830  
?SN ERROR  
READY  
>
```

GET

Syntax:

```
GET
GET "<FILENAME>"
GET2
GET2 "<FILENAME>"
```

Description:

The GET command is the binary equivalent of the LOAD command. It gets the binary file from the cassette back into memory. If the optional argument <filename> is not specified, the next binary file encountered on the cassette will be read. If <filename> is specified, UTIL will search for the requested file, writing the names of all files encountered. When the requested binary file is found, it will be loaded into memory, and UTIL will return to SYSTEM. It can then be executed using the slash (/) command. The default drive is the recorder plugged into the right hand jack of the TC-8 (drive 1). If the form "GET2" is entered, the program will be loaded from the recorder plugged into the left hand jack (drive 2). The previous contents of memory between <begin> and <end> will be lost.

Restrictions:

The GET command may only be used for binary files (file type %); other file types will be ignored. This command is not available in the short form "BOOT" version.

Examples:

```
> GET
$STARTREK $NIM %TBUG READY

*? _

> GET "HIBUG"
$STARTREK $NIM %TBUG %HIBUG READY

*? _
```

GET?

Syntax:

```
GET?  
GET? "<FILENAME>"  
GET?2  
GET?2 "<FILENAME>"
```

Description:

This form is the binary file verify command. A file is read from the cassette but is not loaded into memory. The contents of the file are compared to the contents of memory. If the two are not exactly the same, a memory ("M") error will be displayed. This command allows the operator to determine that a binary file that has been put on tape is good. If the optional argument <filename> is not specified, the next binary file (file type %) encountered will be read and verified. If <filename> is specified, UTIL will search for the requested file. When found, it will be read and verified and UTIL will return to SYSTEM. If the "BAD" message is printed upon return, the file on cassette is not error free. The verify command does not alter the contents of memory.

Restrictions:

The GET? command may only be used for binary files (file type %); other file types will be ignored. This command is not available in the short form "BOOT" version.

Examples:

```
> GET?  
$STARTREK $NIM %TBUG READY  
  
*? _  
  
> GET? "HIBUG"  
$STARTREK $NIM %TBUG %HIBUG BAD  
>
```

* M

GETN

Syntax:

```
GETN
GETN "<FILENAME>"
GETN2
GETN2 "<FILENAME>"
```

Description:

This form is the directory command. It is, in effect, a "get names" command that prints on the video screen a list of all the files on a cassette. In terms of the simple directory function, it is identical to the LOADN command. However, if the optional argument <filename> is specified, the tape will stop accurately at the end of the named binary file %<filename>. This command is useful for positioning at the end of existing binary files prior to writing new files.

Restrictions:

GETN can only be used to position binary files; LOADN is used to position BASIC program files. Either may be used to list the directory. This command is not available in the short form "BOOT" version.

Examples:

```
> GETN
$STARTREK $NIM %TBUG %HIBUG %EDTASM

> GETN "TBUG"
$STARTREK $NIM %TBUG >
```

Statements

OPEN

Syntax:

OPEN

Description:

The OPEN statement is required in BASIC programs before cassette input or output of data may be performed. This prepares UTIL to handle a data file. If an attempt is made to open a file which is already open, a BASIC "FD" error message will be printed, indicating the line which contains the illegal OPEN. CLOSE the file first.

CLOSE

Syntax:

CLOSE

Description:

The CLOSE statement is required in BASIC to end a cassette data file. A new data file may not be opened until any open file is closed. Also a CLOSE statement should be used between TRS-80 cassette data (PRINT #-1) statements and UTIL data statements. If a file is open when the program exits through a STOP, BREAK, END, OR ERROR, the file will be closed automatically.

Example:

```
100 OPEN
110 PRINT #1,A;"",B
120 PRINT #1,"THIS IS A STRING"
130 CLOSE
```

PRINT

Syntax:

```
PRINT #1,<VARIABLE>;",";<VARIABLE>;",";<VARIABLE>  
PRINT #2,<VARIABLE>;",";<VARIABLE>;",";<VARIABLE>  
PRINT #1,"LITERAL STRING"  
PRINT #2,"LITERAL STRING"
```

Description:

The PRINT # statement allows numerical and string data to be written into a cassette file. No file names are used by UTIL for data files. The user may manage the file name within BASIC by writing a string as the first item. Multiple variables may be output in a single line but they must be separated by the explicit delimiter string:

```
;" , " ;
```

With the efficient record packing of UTIL, there is little advantage to multiple variables in a PRINT # statement.

The cassette drive must be specified in the PRINT # statement. Drive must be the numeral 1 or 2. A variable may NOT be used as drive specification. Cassette drive specification must not be changed while a file is open. Close the file and reopen a new one to effect a drive change. The PRINT # command may not be used unless a file has been opened ("FC" error will result). Following the last PRINT # to a file, the file must be closed. There is no limit to the amount of data that may be written to a single file. Strings may NOT contain commas. String length is limited to 127 characters in each string.

Example:

```
100 OPEN  
120 PRINT #1,"JULY"  
130 FOR J=1 TO 10  
140 PRINT #1,N$(J)  
160 NEXT J  
180 CLOSE
```

INPUT

Syntax:

```
INPUT #1,<VARIABLE>,<VARIABLE>,<VARIABLE>  
INPUT #2,<VARIABLE>,<VARIABLE>,<VARIABLE>
```

Description:

The INPUT # statement allows numerical and string data to be read from a cassette file. No file name is used. The data must be input in the same order and format that was used in the PRINT # statement which created the file. The cassette drive must be specified in the INPUT # statement. Cassette drive specification must not be changed while a file is open. The INPUT # statement may not be used unless a file has been opened ("FC" error will result). If an INPUT # is attempted after all data has been read from a file, an "OD" error will result.

Study the example below and the longer example program on the next page. These examples illustrate use of the TC-8 data handling statements.

Example:

```
100 OPEN : ON ERROR GOTO 300  
120 INPUT #1,F$  
130 IF F$ = "JULY" THEN 200  
140 INPUT #1, X$: GOTO 140  
200 J=1 : ON ERROR GOTO 400  
210 INPUT #1, N$(J)  
220 J=J+1  
230 GOTO 210  
300 IF ERR=6 THEN RESUME 120  
400 IF ERR=6 THEN RESUME 410  
410 CLOSE : ON ERROR GOTO 0  
420 END
```

Note that in the example above, the program searches the tape reading files until it finds the string "JULY" stored as a file name. Then it reads in data from the file until it encounters the end of file ("OD" error). Then it closes the file and ends. Notice that no for-next loop was required because of the OD ERROR technique.

```

1 / THE FOLLOWING, WHILE NOT A COMPLETE MAILING LIST PROGRAM,
2 / DOES ILLUSTRATE SOME OF THE TECHNIQUES FOR USING THE TC-8
3 / AND UTIL TO CREATE BASIC DATA FILES.
4 /
5 /
6 /
7 /
100 CLEAR 200
120 ON ERROR GOTO 9000
130 CLS
150 PRINT:PRINT:PRINT"SELECT OPTION"
160 PRINT:PRINT"    1 - CREATE ADDRESS TAPE"
170 PRINT:PRINT"    \ 2 - LIST ADDRESS TAPE ON PRINTER"
180 PRINT:PRINT"    3 - RETURN TO BASIC
190 PRINT:INPUT"WHICH OPTION";Q
200 IF Q < 1 OR Q > 3 THEN 150
210 ON Q GOTO 1000,2000,9090
990 /
992 / * ADDRESS INPUT *
995 /
1000 OPEN
1010 PRINT:PRINT:INPUT "READY CASSETTE FOR WRITE - HIT ENTER";Q$
1020 PRINT:INPUT"NAME > ";N$
1030 IF N$="STOP" THEN 1200
1040 INPUT"STREET >";S$
1050 INPUT"CITY & ZIP >";C$
1060 PRINT : INPUT"    ..... IS ADDRESS OK TO SAVE (Y/N)";Q$
1070 IF Q$ = "N" THEN 1020
1080 IF Q$ <> "Y" THEN 1060
1090 PRINT #1,N$ : PRINT #1,S$ : PRINT #1,C$
1100 GOTO 1020
1200 CLOSE : GOTO 130
1990 /
1992 / * ADDRESS PRINT *
1995 /
2000 OPEN
2010 PRINT:INPUT"READY CASSETTE FOR READ - HIT RETURN";Q$
2020 PRINT:INPUT"READY PRINTER - HIT RETURN";Q$
2030 INPUT #1,N$ : INPUT #1,S$ : INPUT #1,C$
2040 LPRINT N$
2050 LPRINT S$
2060 LPRINT C$
2070 LPRINT : LPRINT : LPRINT
2080 GOTO 2020
9000 /
9001 / * ERROR HANDLER *
9002 /
9010 IF ERR=6 THEN CLOSE : RESUME 130
9020 IF ERR=42 THEN PRINT "CASSETTE READ ERROR" : RESUME NEXT
9090 ON ERROR GOTO 0 : END

```

Useful Information

cassette operation

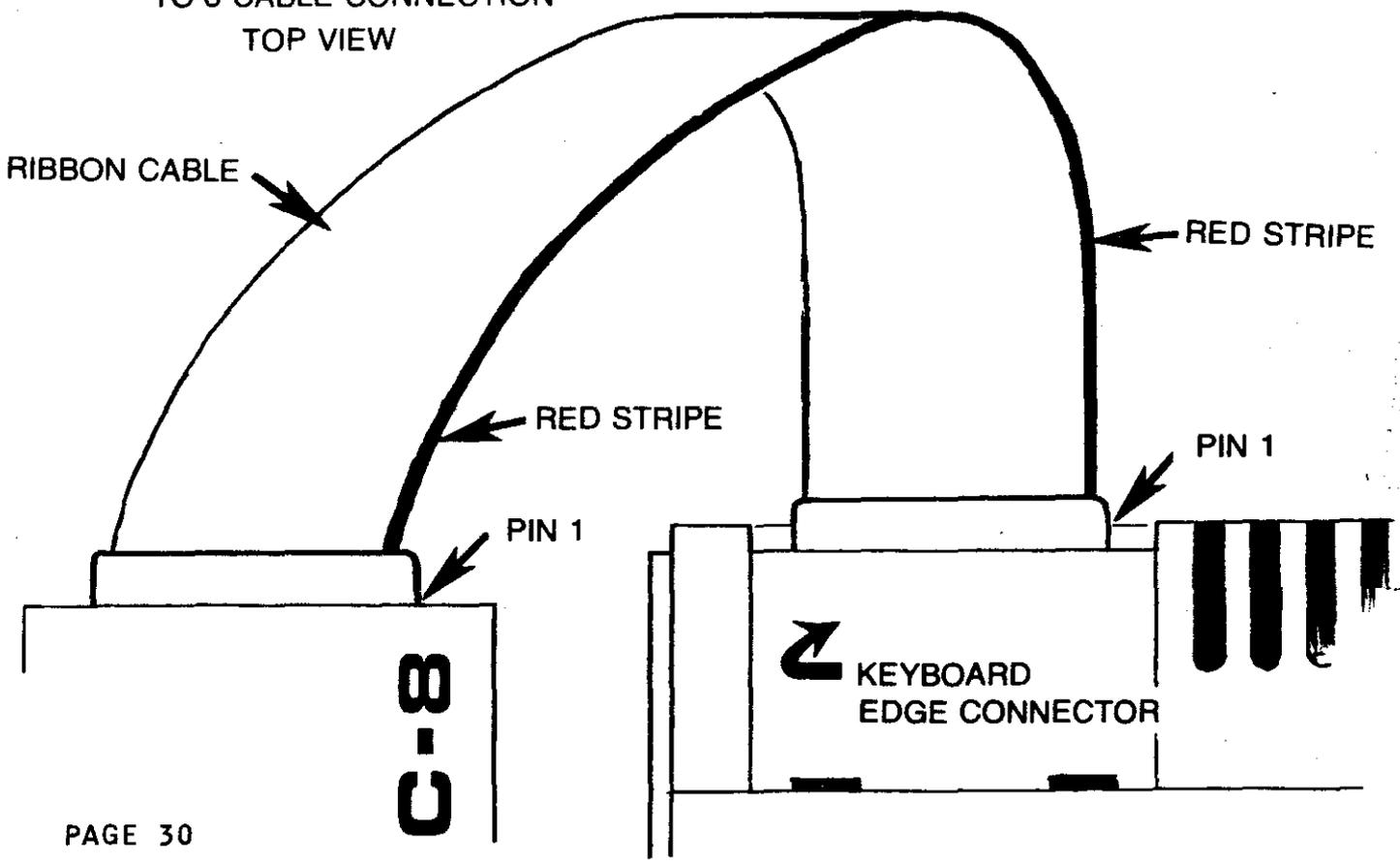
TC-8 CONNECTION

Turn off electrical power to the TRS-80 and connect one end of the ribbon cable to the keyboard expansion-port edge connector, or to the bus card-edge connection on the left side of the Expansion Interface. Connect the other end to the TC-8. Refer to the drawing below and be sure that the orientation is correct (Note the red stripe).

WARNING: SOME TRS-80 KEYBOARD AND EXPANSION INTERFACE CARD-EDGE CONNECTORS ARE OVERSIZE AND ARE A VERY TIGHT FIT IN THE RIBBON CONNECTOR. IN MOST CASES THE CONNECTOR WILL FIT ACCEPTABLY WELL. IF NOT, DO NOT USE EXCESSIVE FORCE AS YOU MAY DAMAGE THE CIRCUIT BOARD. IF YOU CANNOT GET THE CONNECTOR TO FIT, TAKE THE TRS-80 TO YOUR RADIO SHACK STORE AND ASK TO HAVE THE EDGE CONNECTOR FILED DOWN TO PROPER DIMENSIONS.

Plug the power cord into the power connector on the TC-8. Then plug the power adapter into a 115 VAC outlet. Now turn on power to the TRS-80. Note that the cassette motor may run continuously when plugged into the TC-8 if UTIL has not been loaded and executed.

TC-8 CABLE CONNECTION
TOP VIEW



THE INDICATORS

UTIL - In the upper right hand corner of the video screen, two indicators are displayed to let you know what UTIL is doing. Two bars indicates that you have called UTIL to do something.

— —
* *

READING - Two stars, the right one blinking, indicate that UTIL has found a file on the cassette and is reading it.

* ↑

GOOD FILE - When an up-arrow appears to the right of the star, it indicates that UTIL has finished loading the file you requested and that no errors were detected.

* C

CHECK SUM ERROR - If a "C" appears to the right of the star, it means that UTIL has read the file, but the check sum that it calculates does not match the one on the cassette. That usually means that the program you have just loaded into memory is incorrect. Try again.

* M

MEMORY ERROR - The contents of a memory location were not as expected. If this error occurs during a LOAD or GET, then it may mean that your TRS-80 has memory problems. If this error occurs when doing a verify, it simply means that the cassette file is not exactly like what is in memory.

THE TINY MONITOR

The TINY MONITOR included on your UTIL tape will allow you to perform certain system functions. Whenever TINY is in memory, it can be executed from SYSTEM by typing:

*? /25000

All of the functions in TINY are general purpose routines with the exception of the relocater, which can ONLY be used to relocate UTIL from its original address space. When TINY is executed it will print a menu:

```
TINY MONITOR
  1 IDENTIFY TRS-80 SYSTEM FILE
  2 IDENTIFY TC-8 BINARY FILE
  3 EXAMINE MEMORY LOCATIONS
  4 RELOCATE UTIL PROGRAM
  5 WRITE BACKUP COPY OF UTIL
  6 JUMP TO HEX ADDRESS
  7 RETURN TO BASIC
ENTER COMMAND > _
```

1 IDENTIFY TRS-80 SYSTEM FILE

This function reads a TRS-80 format system tape and identifies the name and addresses of the program. Additional details on use of this command are given in the section "PROGRAM CONVERSION".

2 IDENTIFY TC-8 BINARY FILE

This function is similar to 1 above, except that it works on TC-8 files. It is useful when you are making a copy of a TC-8 file and you have forgotten the addresses. The recorder must be plugged into the TC-8 for this function.

3 EXAMINE MEMORY LOCATIONS

This function allows you to examine the contents of memory. You may also modify memory contents. After you select this function by typing 3, TINY will prompt for a hexadecimal address. Type in 4 hex digits. Example: 61A8

To view the contents of successive addresses, press the ENTER key. Memory contents will not be changed.

```
# 61A8 C3
# 61A9 BB
# 61AA 65
```

To stop and get out, press the break key. To modify memory, type the new hex contents following the old hex contents. Do not press the ENTER key.

```
# 5000 00 21
# 5001 00 22
# 5002 00 23
```

Again, press BREAK to stop. If you accidentally type a non-hexadecimal digit for any of the addresses, the command will abort and return to the TINY prompt.

4 RELOCATE UTIL PROGRAM

This function allows you to move UTIL in memory to a location which meets your special requirements. It is described in detail in the section "RELOCATING UTIL".

5 WRITE BACKUP COPY OF UTIL

This function writes a TRS-80 format tape similar to that produced by the "P" command in TBUG. It is described in more detail in the section "UTIL BACKUP".

6 JUMP TO HEX ADDRESS

This function allows you to execute memory-resident binary programs by jumping to the transfer address. For example, if you have just relocated UTIL to BAA8 as in a previous example, then you can execute it by typing 6 and then BAA8

```
ENTER COMMAND > 6
# BAA8
UTIL READY
>
```

7 RETURN TO BASIC

Does just what it says.

BREAK

The BREAK key will allow you to escape from most of the TINY functions if you change your mind. You may have to wait for cassette I/O to finish before the BREAK will work.

UTIL BACKUP

It is a good idea to backup UTIL as soon as possible in case the original tape becomes lost or damaged. The backup should be in the TRS-80 format of course, since otherwise you will have no way to get it into the system from power up. UTIL will probably be the only program that you will maintain in the TRS-80 format.

1. The recorder should be plugged into the keyboard in the normal manner.
2. Set memory size to 31398 before beginning the backup operation.
3. Load the original UTIL tape according to the instructions in the front of the manual in the section "LOADING UTIL".
4. After loading is complete, press enter to start TINY. The TINY menu will be displayed.

TINY MONITOR

- 1 IDENTIFY TRS-80 SYSTEM FILE
- 2 IDENTIFY TC-8 BINARY FILE
- 3 EXAMINE MEMORY LOCATIONS
- 4 RELOCATE UTIL PROGRAM
- 5 WRITE BACKUP COPY OF UTIL
- 6 JUMP TO HEX ADDRESS
- 7 RETURN TO BASIC

ENTER COMMAND > _

5. Type 5 to select the backup option.

ENTER COMMAND > 5

NAME	BEGIN	END	TRANS
------	-------	-----	-------

6. Rewind and store the original tape. Mount a blank tape, preferably degaussed, in the recorder. Be sure the tape has been run forward well past the end of the clear leader. Press both the record and play buttons.

7. Now type in EXACTLY SIX LETTERS for a file name. For example, if you name the file UTIL, type 2 spaces at the end to make six characters. Type the beginning and ending addresses and the entry point (transfer address) exactly as shown below. Do NOT press the enter key at any time.

NAME	BEGIN	END	TRANS
UTIL	7AA8	7FFF	7AA8

8. The recorder will start as soon as you have entered the last number. It will write the copy of UTIL and then return to TINY. You may write several copies of UTIL on the tape at this time by simply repeating steps 5 through 7. Then return to BASIC by selecting command 7.
9. The backup copy can be loaded from SYSTEM by typing "UTIL".

*? UTIL

10. To run the program from SYSTEM, you will only need to type "/ENTER"

*? /<ENTER>

RELOCATING UTIL

1. The UTIL cassette that we sent you with the TC-8 kit was assembled for use at the top of 16K systems. To use it in that location, all you need do is execute it as is. But if you have a 32K or 48K system, or if you need to move it to leave room for some other system program like RENUMBER, then you can use the RELOCATOR in TINY to accomplish it.

NOTE: THE RELOCATOR IS NOT A UNIVERSAL RELOCATOR. IT IS ONLY DESIGNED TO RELOCATE UTIL FROM ITS ORIGINAL ADDRESSING AT 7AA8 HEX. IT CANNOT BE USED TO RELOCATE OTHER PROGRAMS.

2. Be sure that memory size is set to 31398.
3. Load the original UTIL tape according to the instructions in the front of the manual in the section "LOADING UTIL".
4. After loading is complete, press ENTER to start TINY. The TINY menu will be displayed.

TINY MONITOR

```
1 IDENTIFY TRS-80 SYSTEM FILE
2 IDENTIFY TC-8 BINARY FILE
3 EXAMINE MEMORY LOCATIONS
4 RELOCATE UTIL PROGRAM
5 WRITE BACKUP COPY OF UTIL
6 JUMP TO HEX ADDRESS
7 RETURN TO BASIC
ENTER COMMAND > _
```

5. Type 4 to select the relocation option. TINY will prompt you for the new start address. You may relocate to any address you choose, but be sure to leave enough room for UTIL which is 550 hex bytes long. The following examples are for typical relocation within 32K or 48K systems.

For 32K systems:

```
ENTER NEW START ADDRESS (HEX)
# BAA8
```

For 48K systems:

ENTER NEW START ADDRESS (HEX)
FAA8

6. TINY will perform the relocation and report the new addresses. Pay special attention to the END address reported, to be sure that it fits below the end of memory, or below the program above it.

For 32K systems, the following would be reported:

NAME	BEGIN	END	TRANS
UTIL	BAA8	BFF8	BAA8

For 48K systems, the following would be reported:

NAME	BEGIN	END	TRANS
UTIL	FAA8	FFF8	FAA8

7. To put the relocated program on cassette in TRS-80 format, select TINY backup option 5. Set up the cassette recorder for recording on a blank tape. Then enter the following information:

For 32K systems:

NAME	BEGIN	END	TRANS
UTIL32	BAA8	BFF8	BAA8

For 48K systems:

NAME	BEGIN	END	TRANS
UTIL48	FAA8	FFF8	FAA8

8. The decimal addresses for memory size and execution are:

For 32K systems:

MEMORY SIZE 47782 *? /47784

For 48K systems:

MEMORY SIZE 64166 *? /64168

PROGRAM CONVERSION

One of the things you will probably want to do is to convert all of your existing programs from the TRS-80 format into the faster TC-8 format. This section will give you some guidelines for making the conversions.

BASIC PROGRAMS

Conversion of BASIC programs is very straightforward. Plug the cassette recorder into the keyboard connector and CLOAD the BASIC program in the normal manner. Then plug the cassette recorder into the TC-8 and SAVE the program onto a different cassette. From then on you will be able to load the program with the LOAD command, and it will load about 5 times faster than before. Read the descriptions of the LOAD and SAVE commands. When you make these programs, you will probably want to verify them using the LOAD? command to be sure that you have a good SAVE.

SYSTEM PROGRAMS

Converting system programs to the TC-8 format can be somewhat more complicated. The major difficulty is that often you will not know the begin and end addresses, nor the transfer (execution) address for these programs.

We have provided functions within the TINY monitor to assist you in performing these conversions. The following steps will show you how to convert a system program. As an example, we will convert the Radio Shack TBUG program from SYSTEM format into TC-8 binary format. A similar procedure should be used for most other SYSTEM tapes.

1. First you must determine the begin, end, and transfer addresses of the SYSTEM program; use TINY to do that. Plug the recorder into the keyboard and load the original UTIL cassette (or simply load TINY if you have made a copy). Also be sure UTIL is loaded and has been executed (UTIL READY). Execute TINY. You will see the menu.

```
TINY MONITOR
1 IDENTIFY TRS-80 SYSTEM FILE
. . .
```

2. Be sure that the recorder is plugged into the keyboard in the normal manner. Place the SYSTEM tape that you want to convert into the recorder. Press play. Select TINY option 1.

ENTER COMMAND > 1

NAME	BEGIN	END	TRANS
------	-------	-----	-------

3. The recorder will begin to read the SYSTEM tape. The program is not loaded into memory, but TINY will read all of the address information from the tape. The file name will display first. When the tape has finished, the addresses will be displayed. Write them all down.

NAME	BEGIN	END	TRANS
TBUG	4380	4824	43A0

4. Now that you know the addresses, type 7 to return to BASIC. Load the program into memory from SYSTEM.

*? TBUG

5. When the program is successfully loaded, press ENTER to get back to BASIC. Plug the recorder into the TC-8. Place a blank (preferably degaussed) tape into the recorder. Press the play and record buttons. Now PUT the program onto tape.

```
> PUT "FASTBUG" 4380,4824,43A0
%FASTBUG READY
>
```

6. You will now be able to load the new TBUG into memory through the TC-8 about 5 times faster by typing:

```
> GET "FASTBUG"
```

NOTE: Some system programs are structured such that TINY can not properly identify the addresses. Some of these programs load several isolated blocks of memory with a single load. The UTIL tape that came with the TC-8 is of that type. If you want to make a copy of TINY in TC-8 format, use these addresses.

> PUT "TINY" 61A8,67FF,61A8

The execution (transfer) address of TINY is 61A8 hex (/25000 decimal).

Other programs that will be difficult to convert are those commercial programs which the seller has intentionally made difficult to copy to prevent theft. SARGON II is one of these. However, most Radio Shack and Instant Software programs should be easy to convert. For the "protected" ones, you will have to use your own ingenuity, since each one may use slightly different methods. Keep in mind that ANY system program CAN be converted.

KEYBOARD DEBOUNCE

A keyboard debounce routine is included in UTIL which will prevent multiple characters being entered by dirty key contacts. This routine accomplishes the same effect as the Radio Shack KBFIX program. Therefore, you should NOT use the KBFIX program when using UTIL.

If you have an especially noisy keyboard, you may want to increase the delay in the debounce program. Use TINY to make the following change. In this example, the delay constant is changed from 05DC hex to 07FF hex.

```
# 7FEC DC FF  
# 7FED 05 07
```

If you want to disable the debounce routine for any reason, set the delay constant to 0001.

```
# 7FEC DC 01  
# 7FED 05 00
```

USE WITH PRINTER

If you use a printer which plugs into the expansion port on the keyboard, you had to unplug it to use the TC-8 and probably wondered how you can use both at the same time. We offer a bus expander cable in 2-FOR-1 and 3-FOR-1 versions. Similar expanders are also available from other suppliers. With the appropriate expander, you can plug several devices into the keyboard connector.

GOOD PRACTICES

There are several things that you can do to improve the reliability of the programs and data that you are storing on cassette tape. These "Good Practices" are habits that you should acquire.

CLEAN THE HEADS REGULARLY

This is one of the most important maintenance items in your entire system. The oxide coating on the tape wears as it passes over the heads and deposits a coating on the critical contact surfaces. At first, the coating is microscopic, but it builds up fairly quickly and pushes the tape away from the head. This greatly reduces the ability of the head to read or write data on the tape. In addition, dust and other particles can accumulate on the heads, also reducing tape contact. CLEAN THE HEADS EVERY DAY. It only takes a few seconds...less time than it will take to reload a file with load errors. Use a Q-TIP swab moistened with a pure grade of alcohol or a good head cleaner solvent. Or use a tape head cleaner cassette (Radio Shack CAT 44-1160 for example). By all means, clean the heads before a session in which you expect to write important cassettes. It is worth the effort.

USE GOOD QUALITY TAPE

The TC-8 writes data at high speed and the resulting waveform contains relatively high frequencies. Cheap tape just cannot provide the required frequency response. We have done extensive testing of various tapes for use with the TC-8 and have selected two as being superior for data use. The best is DuPont Crolyn, a chrome tape with a good combination of high frequency response, high output, and low dropout incidence. We offer this tape in a high quality 5-screw shell in C-10 length.

We also offer AGFA PE-611 ferric tape which is far superior to most tapes for data use and is less expensive than the Crolyn. PE-611 is highly calendared and has a negligible dropout incidence. Like the Crolyn, it comes in a quality 5-screw shell in C-10 length.

We have had good success with the Radio Shack Certified Data tapes, but we feel that the cost is probably too high for most applications.

REDUCE ELECTRICAL NOISE

If the air conditioner comes on right in the middle of a SAVE and the video display shrinks, you can be fairly sure that the file you have saved is worthless. Computer systems are very sensitive to electrical disturbances and the TC-8 is no exception. Connect your TRS-80 to an electrical circuit that isn't shared with a refrigerator, or a washing machine, or an air conditioner, or any other "noise producing" appliance. As an additional safeguard, you might consider purchasing a power line filter. A good one is available from Radio Shack especially for the TRS-80.

Also, DO NOT place the cassette recorder near the LEFT HAND side of your video display unit. There is a flyback transformer in the video unit that emits strong magnetic fields that will interfere with cassette operation.

DON'T BUMP THE RECORDER

If you bang on the recorder while it is doing a LOAD or SAVE, the result will undoubtedly be an error. The same effect will also be produced by bumping or kicking the table on which the recorder rests. The tape must be in critical contact with the heads. But the cassette is not held in position very securely in low-cost recorders and the mechanical bumps cause it to move slightly...and you lose a bit of data.

GET YOUR CTR-80 FIXED

Some CTR-80 recorders supplied with TRS-80 computers have a design flaw which allow them to "write" a spike on the tape when the motor stops. This can even occur in "play". Radio Shack has been providing a free retrofit service to fix this problem. Check with your local Radio Shack store.

A DEGAUSSER ??

Do you need a bulk tape eraser (degausser) ?? Well you don't absolutely HAVE to have one. The recorder has an erase head. And it works pretty well when you are recording music. After all that is what the recorder was designed for. But it doesn't completely erase old data from the tape, and that just increases the background noise. A bulk eraser works a lot better. And if you are using the chrome tapes with their higher coercivity, then we strongly recommend the use of a degausser. A good one is Radio Shack CAT 44-210.

INTERRUPTS

Interrupts must be turned off while the TC-8 is performing cassette I/O. UTIL turns interrupts off automatically, so for most TRS-80 systems, you need not be concerned about interrupts. However, if your system uses interrupts in some manner, then you will want to "ENABLE INTERRUPTS" after each UTIL operation.

FAST CLOCK

There are several modification kits available for the TRS-80 that increase the clock speed. When used with the TC-8, these modifications can cause a problem because they may increase the BAUD rate beyond the capacity of the CTR-80 cassette recorder. Try it. If the reliability is acceptable to you at the higher speed then operate that way. Use our chrome cassette tapes and keep the head immaculately clean.

MODIFY YOUR CTR-41

Motor control of the CTR-41 recorder will not function properly with the TC-8. A simple modification of the recorder will correct this problem and will also provide improved operation with the normal TRS-80 cassette functions. Modification instructions are included in this manual. The CTR-80 recorder does NOT require modification.

TWO RECORDERS

The TC-8 will support two recorders. There are two connectors on the front of the unit, one for each recorder, and the UTIL program allows drive specification. But not all cassette recorders can be connected in the dual-drive configuration. Best results can be achieved when two identical recorders are used. We would recommend two CRT-80 recorders. Many TC-8 owners are using the dual-drive system successfully, but some report having to experiment with volume control settings at first. Use one recorder for writing out new data and the other for reading in old data. Be sure to CLOSE and OPEN between input and output routines.

RECORDER VOLUME SETTING

The volume control setting on the cassette recorder is not at all critical with the TC-8. Anywhere above #2 will work acceptably well. But there is an optimum setting for each recorder which is the maximum output without significant distortion. For the CTR-80 recorder, full volume is best, giving the greatest tolerance to tape dropouts. For the CTR-41 recorder, full volume creates very bad distortion, and so a setting of #7 is best. The tone control on the CTR-41 MUST be set to HIGH.

DUST IS DIRTY STUFF

Keep the cassette recorder clean; keep the cover over the heads closed. Store the cassettes in a clean place preferably in a dust proof container. Dust between the head and the tape causes dropouts and loss of data.

CLEAN THE HEADS

We hate to sound redundant, but it is very important!

CTR-41 MODIFICATION

If your TRS-80 uses the CTR-41 recorder, it MUST be modified for use with the TC-8. This modification will not degrade its operation with CLOAD and SYSTEM cassette I/O. In fact it will reduce the incidence of cassette relay "micro-welding".

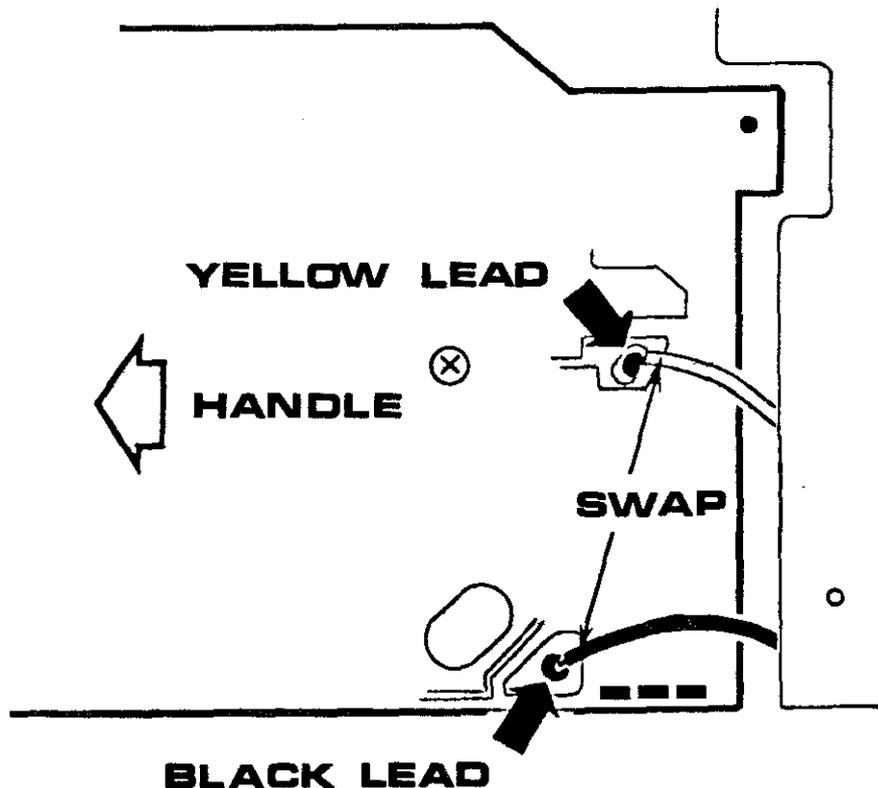
() Remove the 5 screws holding the bottom on the CTR-41 recorder (2 are in the battery compartment).

() Position the recorder in front of you so that the handle is to your left. On the right side of the circuit board, locate the YELLOW lead and the BLACK lead. This modification will swap these two leads.

() Unsolder the YELLOW lead.

() Unsolder the BLACK lead. Solder the BLACK lead on the pad where the YELLOW lead was previously unsoldered.

() Solder the YELLOW lead on the pad where the BLACK lead was previously unsoldered.



notes

