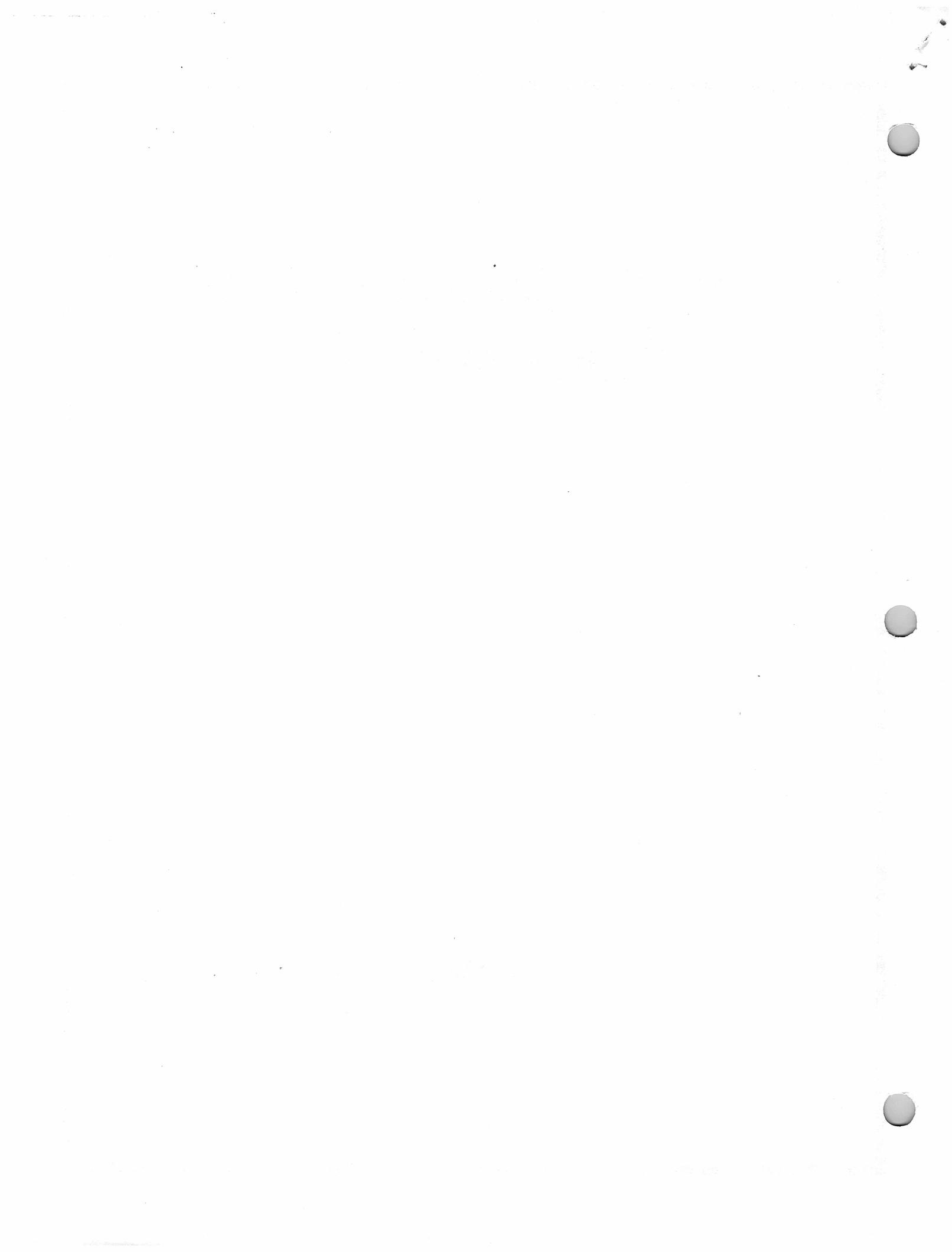


THE CHALLENGER C2-4P

USERS MANUAL

Sept. 1978

© Ohio Scientific Inc.



Before you begin...

Please follow the instructions on the following pages. They are organized as steps A., B., C., D.

a

Unpack and inspect the computer. Check supplied items against the enclosed check list.

b

Connect the computer to the console terminal (serial terminal or video display). Test the operation of the internal monitor program.

c

Connect mass storage device (i.e. audio cassette or disk) and run BASIC.

d

Connect any special features ordered.

e

In case of difficulty, check the troubleshooting section of this manual. To register your Challenger complete the Warranty card below and return it to:

Ohio Scientific, Inc.
1333 S. Chillicothe Rd.
Aurora, Ohio 44202

Warranty Card

NAME _____ PHONE(____)

STREET _____

CITY _____ STATE _____ ZIP _____

PURCHASED FROM _____

DATE OF PURCHASE _____

MODEL NUMBER (SEE NAMEPLATE ON BACK) _____

SERIAL NUMBER _____

For a Warranty description see the Limited Warranty statement in section E of this manual.

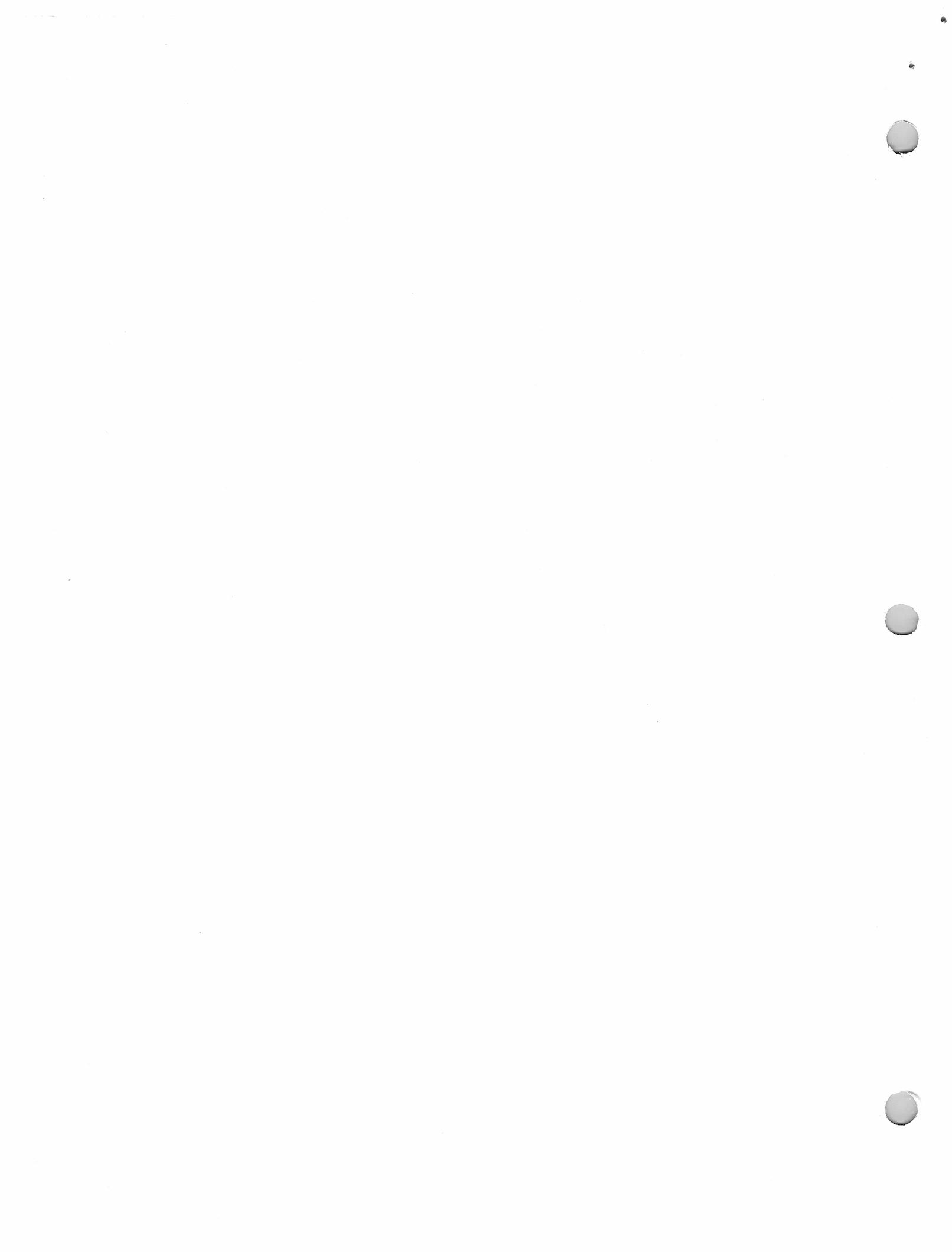


TABLE OF CONTENTS

Warranty Card	A-1
Unpacking Instructions	A-4
Setting Up The Computer	B-1
Video Display Connections	B-4
Keyboard Layout	B-8
Display Width Programming	B-9
Getting Up and Running	B-10
Cassette Storage and Hook Up	C-1
Using BASIC Program Cassettes	C-5
Introduction to Software	C-6
Short BASIC Programs	C-11
Lower Case (Keyboard)	C-18
Onwards To New Projects	C-20
Introduction To Hardware	D-1
Glossary of Terms	D-2
48 Line BUS	D-3
Installing OSI 48 BUS Boards	D-5
65V Machine Code PROM	D-6
Warranty	E-1
Troubleshooting Hints	E-1

Appendices

8K BASIC-in-ROM Reference Manual
502/540 Schematics and Memory Map
6502 Specification Sheets
The Challenger 2P Technical Report



UNPACKING AND ASSEMBLY INSTRUCTIONS

INTRODUCTION

This information outlines procedures to be followed during unpacking and assembly of Ohio Scientific microcomputer systems. Please follow these instructions carefully, preferably, read them completely BEFORE opening any cartons. You'll then be assured an up-and-running system with a minimum of problems. Please don't be guilty of the adage, "When all else fails, read the instructions"!

SYSTEM ARRIVAL

A CHALLENGER system may be delivered in from one to six boxes. Ohio Scientific normally ships via United Parcel Service (UPS). However, large or bulky items may require shipment by air or around freight carriers. Further, due to weight restrictions, the entire shipment may arrive over several days. Each package of the total shipment will be marked "(number) OF (total boxes)", i.e. 1 OF 3, 2 OF 3, etc. Equipment in each box will be accompanied by manuals and other materials pertinent to that equipment.

SHIPMENT EXTERNAL CHECK

Via a carton count, determine that you have the entire shipment. Inspect the boxes for signs of rough handling such as punctures, crushed sides, etc. If such damages are detected, check the contents of the box, preferably without removing the equipment. If the contents have sustained damage, this will make the determination of liability easier. In such cases, notify the carrier immediately.

UNPACKING

Carefully remove the system components from their boxes and SAVE ALL PACKING MATERIALS! These may be needed later to transport or ship components of the system. Most components require no further unpacking. However, a few components contain internal packing materials which must also be removed. These are listed below:

CHALLENGER C2-8P, CHALLENGER III - remove the 4 screws retaining the top cover (lower side edges), and lift the cover off. Compress and remove the foam from between the circuit boards. Cassettes, cables or other small components may be shipped within this cabinet. Remove these also. Be careful not to disturb any wiring, components, or PC boards. Replace the cover and screws.

FLOPPY DISK DRIVES - remove and save the cardboard dummy diskette which protects the disk head from vibration. The dummy should be re-installed when the disk drive is transported.

Other accessories such as CRT terminals, cassette machines, and line printers are shipped separately. Their boxes will contain the manufacturer's operating instructions, warranty cards, etc.

Check all unpacked components of the shipment to make sure the shipment is correct and complete.

===== WARNING! =====

NEVER OPEN ANY CABINET WHEN THE UNIT IS PLUGGED IN. NEVER ENERGIZE OR OPERATE ANY UNIT WITH THE TOP OR PROTECTIVE COVER REMOVED. YOU MAY CREATE A POTENTIAL SHOCK HAZARD BECAUSE THE COMPUTER AND PERIPHERAL EQUIPMENT HAVE EXPOSED 110 VOLT WIRING WITHIN. ALSO, THE UNSHIELDED LOGIC CIRCUITRY MAY EMIT RADIO FREQUENCY INTERFERENCE.

POLLED KEYBOARD NOTE

Several Ohio Scientific microcomputers employ a polled (software scanned) keyboard. These keyboards can be identified

by a latching "SHIFT LOCK" key immediately to the right of the "+;" key. The SHIFT LOCK must be latched in the "down" position before the machine can be reset and BASIC can be entered.

SETTING UP THE COMPUTER

Welcome to the world of personal computing! You now have a computer that was technically impossible just a few years ago! We at Ohio Scientific think you'll find your computer an interesting and entertaining device for years to come.

So that you can get the computer operating as quickly as possible, we have provided detailed instructions to assist you. Although a computer is a relatively rugged solid-state device, it may still be damaged if you fail to observe power supply, accessory, or safe-operating requirements. Therefore, follow the instructions carefully. Better yet, read all the instructions BEFORE you turn to the computer. Once you are familiar with these procedures, you can explore other areas of personal computing at your own pace.

The information we'll cover concerns the following:

1. Supplying a source of power to the computer. The Challenger 2P requires a 3-wire grounded 110V outlet.
2. Connecting a display device such as a closed-circuit TV monitor or ordinary television.
3. Use of a cassette recorder to play pre-recorded programs into the computer.
4. Use of a cassette recorder as a program storage device.
5. Computer activities you can turn to after you have mastered these basic skills.

POWER SUPPLY CONNECTIONS

CHALLENGER 2P

1. The Challenger 2P must be plugged into a grounded 3-wire 110V receptacle. This assures that the computer's cabinet is thoroughly grounded, which protect both the computer and you from possible damage or shock.
2. Optionally, you can run a wire from the computer's cabinet to a good ground such as a cold water pipe, and ONLY THEN use a two wire adapter on the computer's power cable.
3. THESE ARE THE ONLY ACCEPTABLE POWER CONNECTIONS FOR THE CHALLENGER 2P.

WARNING

ANY OTHER POWER CONNECTIONS FOR THE CHALLENGER 2P OTHER THAN OUTLINED ABOVE MAY ULTIMATELY PRESENT A SHOCK HAZARD OR LEAD TO COMPUTER DAMAGE VIA STATIC DISCHARGES. SUCH DAMAGES ARE SPECIFICALLY NOT COVERED UNDER THE WARRANTY.

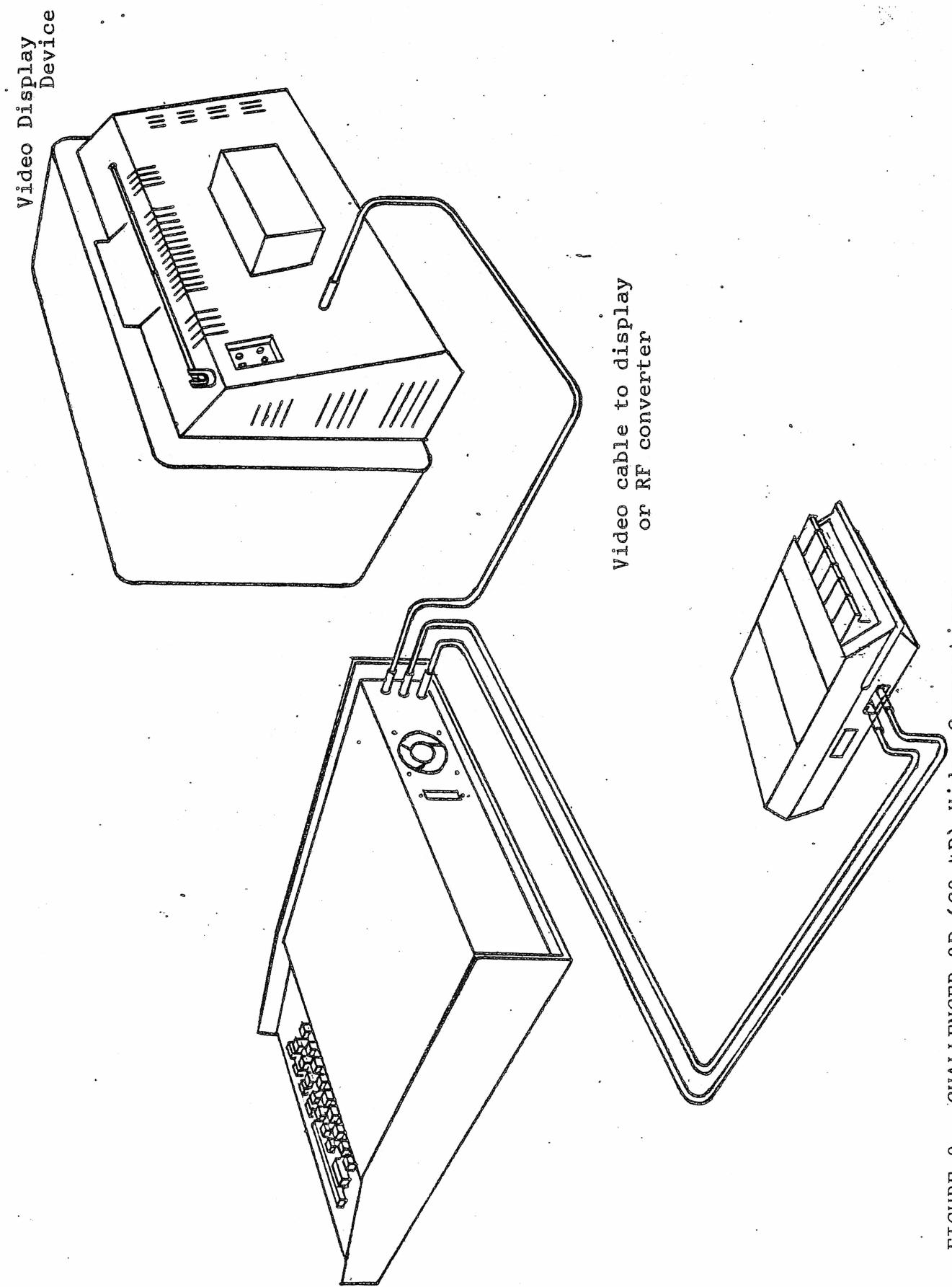


FIGURE 2. CHALLENGER 2 P (C2-4P) Video Connections.

VIDEO DISPLAY CONNECTION

There are three different methods of attaching a video display to the Challenger 2P computers. These are outlined as follows:

1. Preferred method - connect the supplied computer video cable to the high impedance (Hi-Z) input of a closed-circuit TV video monitor. Ohio Scientific offers the Model AC-3 12" monitor which is ideal for this application. The unit doubles as a television when the video cable is disconnected.
2. Connect the supplied computer video cable to an "RF modulator" which is, in turn, connected to a standard television's antenna terminals. RF Modulators are inexpensive and allow you to use almost any television with your computer.
3. Have a standard AC transformer-operated television modified to accept direct video entry. This requires special safety precautions which will be explained later.

CLOSED-CIRCUIT VIDEO MONITOR CONNECTION

1. Refer to Figure 2. Attach the supplied video cable to the computer as shown. With the Superboard II this cable is part of a computer video/cassette recorder cable assembly.
2. Connect the other end of the cable to the high impedance input of the video monitor. The AC-3 monitor has a Hi-Z RCA-type phono jack input. On other monitors, a high impedance - low impedance selector switch is sometimes present, or there may be two or more inputs. Consult the manufacturer's instructions.
3. Observe the manufacturer's power recommendations. If the monitor has a 3-wire grounded plug, connect it to a properly grounded 3-wire AC outlet.
4. Turn on the computer and monitor.
5. Allow the monitor to warm-up. You should see the screen filled with random graphics characters, alphabet, etc.
6. If necessary, adjust the VERTICAL and HORIZONTAL controls to obtain a stable picture.

RF MODULATOR/STANDARD TV CONNECTION

1. Refer to Figure 2. Review the manufacturer's instructions included with the RF modulator.
2. Connect the computer video cable to the computer as shown.
3. Connect the video cable to the RF Modulator.
4. Connect the modulator to the television's antenna terminals (consult modulator instructions).
5. Plug in the television and computer.
6. Turn on the computer, television, and modulator (consult modulator instructions).
7. At this point you will have to select the proper TV channel and possibly adjust the television's fine tuning slightly (consult modulator instructions).
8. When the television warms up you should observe a screen filled with random graphics characters. If the picture is not stable, adjust the television's VERTICAL or HORIZONTAL controls as needed.

MODIFICATION OF A TELEVISION FOR DIRECT VIDEO ENTRY

1. A standard television may be modified to act as a video monitor. However, this conversion requires detailed knowledge of television circuitry, and will likely require a schematic of the television to be converted. Consult a qualified service person.

=====

WARNING

=====

ANY TELEVISION CONVERSIONS MUST BE PERFORMED ONLY BY A QUALIFIED PERSON, SUCH AS A TV SERVICEMAN. LETHAL VOLTAGES ARE PRESENT WITHIN THE TELEVISION. INCORRECT CONNECTIONS MAY PRESENT SHOCK HAZARDS OR DAMAGE THE COMPUTER. SUCH DAMAGE IS NOT COVERED BY THE WARRANTY.

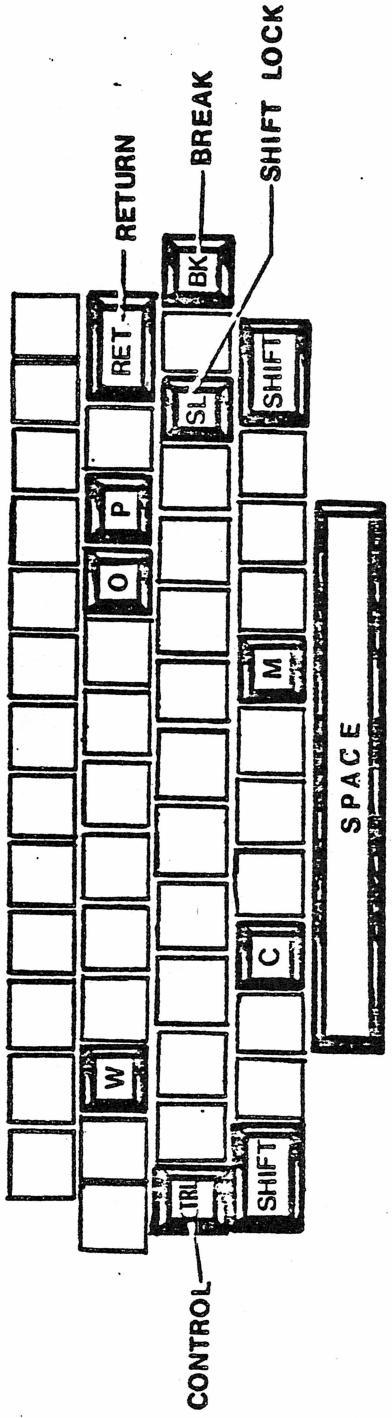
2. The television to be modified must be an AC-transformer operated television. Several solid-state TV sets are now available which can be operated from 110V AC, or from a 12 volt source such as a car cigarette lighter. These televisions can usually be converted easily. Some older "AC-DC" tube-type televisions are "hot chassis" types; that is, one side of the power line is connected to the chassis. These televisions do not have transformers and MUST NOT be used for conversions. Shock hazards are

present, and the computer may be damaged. These televisions are suitable for RF modulator operation. Refer to that section in this manual.

3. More characters per line can be displayed on the screen if the picture is "shrunken" slightly. On most 110V AC/12V DC televisions, this can be accomplished by adjusting the television's power supply regulator to give a lower voltage. Brightness will also be diminished, but this can be restored via the TV BRIGHTNESS CONTROL. Refer this adjustment to the service person at the time of conversion.
4. When the power supply voltage is adjusted, the picture may require re-centering. Equal borders can be restored to the screen by adjusting the picture tube centering coils. Refer this adjustment to the service person at the time of conversion.
5. When the television has been modified, it may then be treated as a video monitor and connected to the computer. Refer to that section of this manual.

Keyboard Operation

Almost all computer operations are performed via the keyboard. Some of the keyboard's functions are obvious from the labeling of the keys but many other functions are not. Please carefully study Figure 3 before proceeding to the next section.



1. <> - Brackets - Instruct user to press key whose label is contained between the brackets. DO NOT type in word between brackets.
2. SHIFT LOCK - (latching key) - must be in the locked (depressed) position before BASIC may be entered; or capital letters, numerals, etc., may be entered.
3. <BREAK> - Places computer in the "RESET" state any time after system is powered up.
4. C - May be pressed after <BREAK>. Initializes computer and clears system RAM.
5. W - May be pressed after <BREAK> except when computer is first powered up (C must be used). Initializes computer, DOES NOT clear system RAM. Any programs in RAM are preserved.
6. M - May be pressed after <BREAK>. Initializes computer, clears system RAM. Computer enters machine language monitor.
7. <SPACE> - provides a space when pressed.
8. <RETURN> - must be entered after a line is typed. Typed material is then stored in program memory space.
9. <SHIFT O> - Press <SHIFT> first, add O - erases last character typed.
10. <SHIFT P> - Press <SHIFT> first, add P - erases current line being typed. Provides a "@" carriage return and line feed.
11. <CONTROL C> - Press <CONTROL> first, add C. Program listing or execution is interrupted, "BREAK IN LINE XXX" is printed.

FIGURE 3. OPERATION NOTES - OSI POLLED KEYBOARD.

Display Width Programming

Your Challenger 2P uses OSI's 540 Video Board, which is capable of displaying up to 32 lines of 64 characters or 32 lines of 32 characters.

When the Challenger 2P is powered up normally, it should be displaying 32X64 characters. If it is turned off and on very quickly, it will come up with 32X32 characters. This is not the recommended way of getting the 540 to switch from 64 to 32 characters wide, however. The recommended approach is with the BASIC statement: POKE 56900,0 <return>. This programs an internal flag in the 540 board to drop to 32 characters wide. Conversely, it can be switched back to 64 by the statement: POKE 56900,1 <return>, which reprograms the board for a 64-character display. The user may want to use a 32-character-wide display in cases where his television monitor does not have adequate resolution for a 64-character width, and/or in video games where it is desirable to have a square character cell size instead of a rectangular cell size for the characters.

GETTING YOUR COMPUTER "UP AND RUNNING"

INTRODUCTION

These instructions will help you get your computer running in the computer language "BASIC". This language is permanently stored in the computer's memory, and can be quickly brought up when the computer is turned on.

Several of these instructions for bringing up BASIC contain words or letters which are bound by brackets "<" and ">", such as <BREAK> and <C>. The brackets indicate that a keyboard key labeled with the word or letter must be pressed. Do not type in a word contained between the brackets letter-by-letter. See Figure 3.

BEFORE YOU POWER-UP...

1. Check that all power supply connections are correct.
2. Make certain that your video monitor or television is connected to the computer properly.

GETTING INTO BASIC

These instructions should be followed very closely. If, at any time, the computer or television/monitor does not respond as indicated, turn off the power to both and review all hook-up procedures, wiring, etc.

1. Turn on the computer.
2. Turn on the television or monitor. After a short warm-up you will observe the screen filled with random characters.
3. Press <BREAK>. The prompt C/W/M? or D/C/W/M? will appear in the lower left corner of the screen.
4. Press <C>. The screen will scroll up one line and ask "MEMORY SIZE?"

5. Press <RETURN>. The screen will scroll up another line and ask "TERMINAL WIDTH?"

6. Press <RETURN>. The computer will reply:

XXXX BYTES FREE

OSI 6502 BASIC VERSION 1 REV. 3.2

COPYRIGHT 1977 BY MICROSOFT CO.

OK

Pay particular attention to the first line, "XXXX BYTES FREE".

"XXXX" represents a memory test which is performed each time the computer is turned on. Typically, "XXXX" is "3327" for a computer with 4K RAM, and "7423" for a computer with 8K RAM. If you get a number other than the correct one for your system, do not use the computer. Refer to your authorized Ohio Scientific dealer for help.

PROGRAM EXAMPLE

The following program example demonstrates some of the more fundamental concepts of BASIC. This program may be entered once the computer replies "OK". Enter the program exactly as it appears, including all punctuation, etc.

```
10 PRINT "HELLO! I'M YOUR NEW COMPUTER!" <RETURN>
20 PRINT <RETURN>
30 END <RETURN>
```

Now, check the program to be sure you have entered it correctly. Type in the word LIST and <RETURN>. This instructs the computer to print out the program as stored within the computer's memory.

LIST <RETURN>

To have the computer execute ("RUN") the program, type in:

RUN <RETURN>

The computer should then print:

HELLO! I'M YOUR NEW COMPUTER!

The BASIC language makes it easy to modify ("EDIT") a program. Errors within a line may be corrected by retyping the line. Additional statements may be incorporated into a program by sequencing the new line numbers within the existing program. The following additions to the example program demonstrate these editing concepts.

5 FOR X=0 TO 30 <RETURN>

25 NEXT X <RETURN>

To examine the program as amended, type LIST <RETURN>.

To execute the new program, type RUN <RETURN>.

Refer to one of the many BASIC programming texts now available for an in-depth study of BASIC.

CASSETTE STORAGE TECHNIQUES

CONNECTION

A standard cassette tape recorder, such as the Model AC-2 offered by Ohio Scientific, may be used for program storage and playback. The recorder must have a microphone input jack and an audio output jack (usually labeled "EARPHONE" or "SPEAKER"). The cassette recorder should be connected to the C2P as shown in Figure 4.

The necessary cables are supplied with the computer. Note that the placement of the microphone and audio output jacks may vary with different cassette recorders. If you plan to use a recorder other than the AC-2 be certain your connections are correct.

PLAYING BACK A PROGRAM

The following steps show how to load the computer with a program stored on cassette.

1. Check for correct cable connections between the recorder and computer.
2. Rewind the cassette so that the tape "leader" is visible on the take-up spool.
3. Turn on the computer and get into BASIC as indicated by the letters "OK" in the lower left corner of the screen.
4. Type in NEW <RETURN>. This erases anything now stored in the computer.
5. Type LOAD. Do not press <RETURN> .
6. Turn on the recorder to PLAY the tape.
7. As soon as the tape (dark brown) begins to wind onto the spool, press <RETURN> .

8. Shortly, the program will begin listing on the screen. When program loading is complete, the following will appear in the lower left corner of the screen:

OK

S- ERROR

OK

9. Press <SPACE> .
10. Press <RETURN> .
11. To inspect the program, type in LIST <RETURN> .
12. To execute the program, type RUN <RETURN> .

RECORDING A PROGRAM

These instructions show how to record a program contained in the computer's memory.

1. Check for correct cable connections between the recorder and computer.
2. Use a new or thoroughly erased cassette. This will minimize "noise" and other problems associated with old cassettes.
3. Have your program ready in the computer.
4. Rewind the cassette so that the tape "leader" is visible on the take-up spool.
5. Type SAVE <RETURN> .
6. Type LIST. Do not press <RETURN> .
7. Turn on the cassette recorder in the RECORD mode. When the tape (dark brown) begins winding onto the take-up spool, wait 5 seconds and press <RETURN> .
8. Observe the program listing on the screen. When the last line of the program is listed, wait a few seconds and turn off the recorder.
9. Type in LOAD <RETURN> .
10. Press <SPACE> <RETURN> .
11. Label the cassette. If you wish to protect the contents from accidental erasure, break out the appropriate "record protect" tab located on the cassette's rear edge.

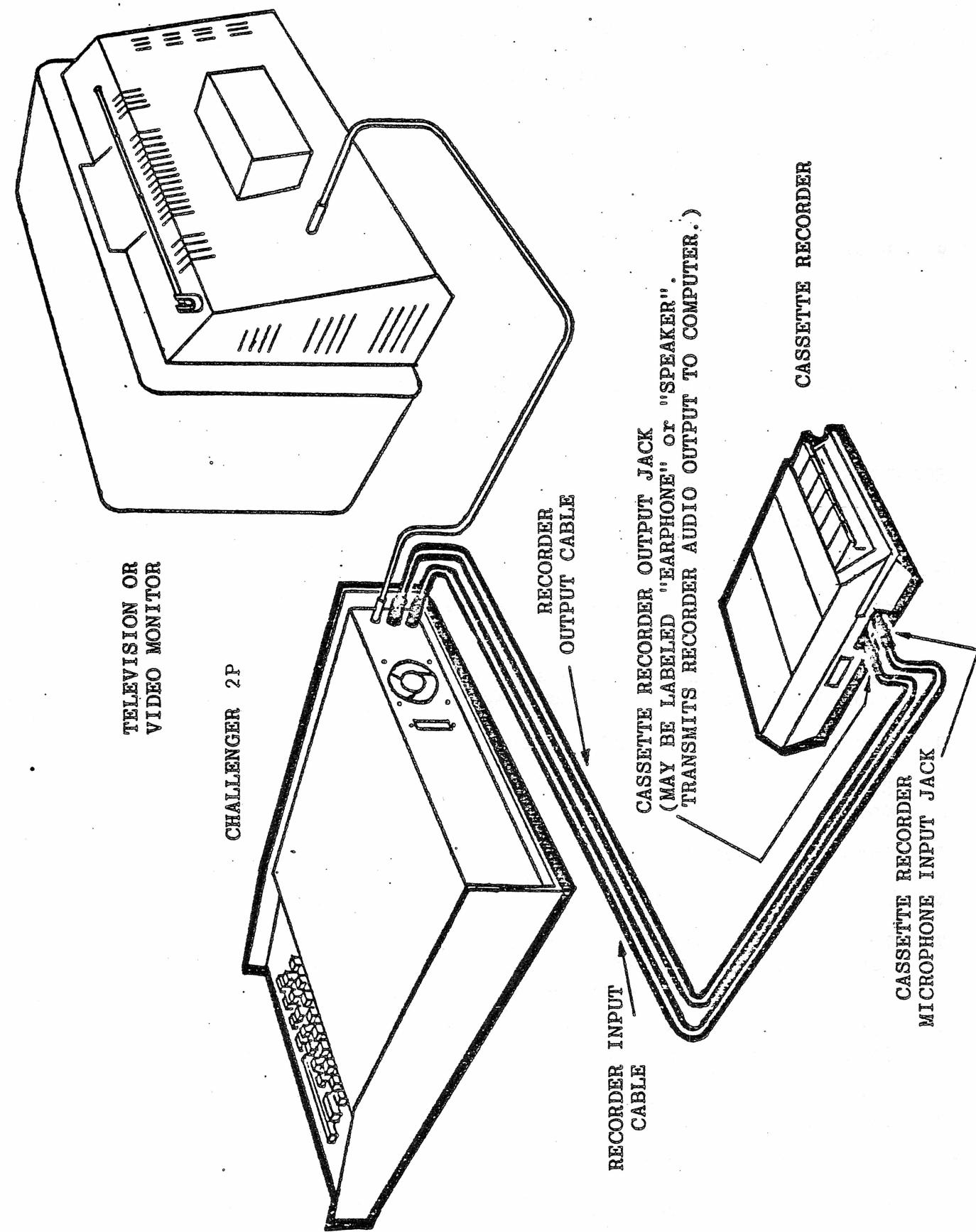


FIGURE 4. CHALLENGER 2P CASSETTE RECORDER CONNECTIONS

such excellent manuals as Schmidt's outline series Programming with BASIC (McGraw-Hill), it can at least give you some insight into the method for writing your own programs in BASIC.

Refer to the instructions provided with your individual unit to bring up BASIC in the OSI Challenger. Establish the memory size and terminal width for your particular program. When you see an OK appear on your video monitor or terminal, the computer is ready to start accepting BASIC commands from the keyboard.

Every statement in your program must begin with a statement number. These need not be typed in numerical order, since the computer will automatically rearrange them according to statement number when you have finished typing the program. But they must be numbered in the same order in which they are to be run. In OSI's 8K BASIC for the 6502, a variable can consist of one or two characters. If longer variables are to be used, BASIC will recognize only the first two characters. The first character in a variable must be alphabetic. The second character, if present, may be either alphabetic or numeric. Functions, commands, etc., already used by BASIC must not be employed as variables. In order to set a variable equal to a desired value, e.g., Z equal to 10, you use the LET statement, as follows:

(line number →) 20 LET Z=10

Since LET is optional in OSI's 8K BASIC, you may also type:

20 Z=10

You may wish the value of the variable to change each time you run the program, without having to rewrite the whole program every time. To take advantage of the option to alter the values of variables, you make use of the INPUT statement, for example, as follows:

```
10 INPUT A,B,C  
20 LET X=A  
30 LET Y=B+C
```

In this way you can cause X and Y to take different values each time the program is run. Later, when you do run the program, you will see a ? on the terminal. You then type the values for A,B, and C which are relevant to the particular program. If there are no other INPUT statements in the program, it will begin to run immediately with the values you have entered, unless some built-in control prevents this. If the program contains additional INPUT statements, BASIC will keep asking you (by means of a ?) to input whatever data it needs to run the program, until each INPUT statement has been answered.

It can be that a variable has a value which is to change at a regular rate during the course of a single program run. This will require you to set up a loop which makes calculations using these increasing or decreasing values each time a new value is employed. For this you need to use a FOR-NEXT loop. This loop begins with a FOR statement and ends with a NEXT statement. The FOR statement identifies the initial and final values of the variable in question, and includes the constant amount of increase or decrease:

```
100 FOR Z=10 to 20 STEP 3  
110 LET A=Z+(2*4)  
120 NEXT Z  
130 (resumption of program)
```

Step 3 means an increment of 3 upon each pass through the loop. Therefore, the above FOR-NEXT loop will be run four times, namely, when Z=10,13,16, and 19. When the value of Z exceeds 20, BASIC resumes the program by going to the first statement following the FOR-NEXT loop. In addition to signifying the end of the loop, the NEXT statement also contains the variable identifying which loop it terminates. As you may later discover, this is most useful in nesting one loop inside another.

Sometimes you will want the program statements to be run in a different order, if a certain condition is met. In order to change the order of execution, you may use the IF . . . GOTO statement, for example,

```
100 IF X=10 GOTO 150
110 (another program line)
140 (another program line)
150 LET Y=X+5
```

Here, the IF . . . GOTO diverts execution to a non-consecutive statement, line 150, omitting lines 110 and 140, provided only that the value of X is equal to 10. If X is not equal to 10, the program would resume with line 110. A simple GOTO command may also be employed without an accompanying IF, if no condition must first be met.

An IF . . . THEN statement is used to jump to a statement other than the one directly following. It can also be used to issue any other statement allowed in BASIC. For example,

```
100 IF X>10 THEN PRINT "X IS GREATER THAN 10."
```

This will cause the terminal to display X IS GREATER THAN 10 only if $X > 10$. The program will then proceed as normal, with the next consecutive line. If $X \leq 10$, the program will, of course, proceed as normal, ignoring the PRINT command.

A PRINT statement will cause the terminal to display whatever follows. If you type:

```
100 PRINT A
```

the value of some previously defined variable A will be printed. If you type:

```
100 PRINT "A"
```

the simple letter A will be printed.

The END statement terminates the program and allows you to run the program, change it, or start to write a new program. As in the case of the LET statement, the END statement is optional in OSI's 8K BASIC.

If you want to erase a program and start a new one, simply type NEW and enter your next program.

BASIC is provided with a large number of mathematical functions, such as sine [SIN(X)], square root [SQR(X)], and absolute value [ABS(X)]. These functions automatically cause the computer to calculate the pertinent value without figuring by the user. For example, in the following statements:

```
100 X=121
110 PRINT SQR(X)
```

the value of the square root of 121 will appear on the terminal when the program is run.

At any time while entering your program, or after you have finished entering it, you can list all the statements up to that point by typing LIST. You can thus list the whole program, or by typing a specific line number after LIST, such as LIST 140, you can display just the one line. If you desire to see a certain block of program lines only, then you can specify the desired range, such as:

LIST 100 TO 200

If you want to correct a line previously typed, simply type the correction, using the same line number. It is recommended that you number the program statements by jumps of 10 rather than consecutively, so that you can later easily insert additional lines if you wish. To do this, type a line number which falls between the interval where you want the new statement to appear, and add the missing line. If you want to delete a line, simply type the line number, then <return>. By using the LIST command, you can easily verify any changes you have made. This will cause every program line to scroll up the terminal, with each line number in consecutive order. If you want to stop the scrolling, type Control-C, examine the listing to your satisfaction, then type CONT (=continue), after you see BREAK IN LINE XX on the terminal.

The following example gives an illustration of editing procedures: Suppose you want to modify the following statements:

```
90 INPUT A  
100 LET X=2*A  
110 PRINT "THIS IS A PROGRAM."  
120 PRINT "EXAMPLE"
```

If you want to insert a line Y=A between lines 90 and 100, you could, at this point, type:

91 LET Y=A

If you want to delete line 110, simply type 110, then <return>. If you want line 100 to read LET X=3*A, simply type the correction, using the same line number. At any time you could confirm the alteration by typing LIST.

Following these corrections, if you are ready to run the program, type RUN <return>. If your program contains any INPUT statements, you will now see a ? on the terminal. Type in the data desired, as explained above, and the program will run. Following program execution, you can start over again by typing RUN, or enter a new program by typing NEW.

The following sample program demonstrates the INPUT, LET, PRINT, GOTO, and END statements, the FOR-NEXT loop, and the IF . . . GOTO command, as well as the SQR function.

Problem: Print the square root of a number; increase the number by five six times, and each time print the square root. If the largest square root is less than twice the first square root, indicate this. Otherwise, indicate only the fact that the program prints square roots.

After the programmer has typed the above program in BASIC, he will see an OK on the screen, signifying that the computer is ready for the next command from the user. If he wants to run the program, he types RUN. The computer will show a ? on the terminal. The user types on the keyboard that number with which he wants to begin the program. The six values (with constant increments of 5, see line 40) will scroll up the screen, each accompanied by its square root. If by chance you have made an error in typing (not including improper spacing), you

will probably see an error message on the screen. If you do, simply edit the line containing the error, as explained above. You can always run the program again by typing RUN. Here are the lines of the program:

```
10 INPUT A
20 LET Y=SQR(A)
30 LET Z=A+30
40 FOR X=A TO Z STEP 5
50 PRINT "THE SQUARE ROOT OF";X;"IS";SQR(X)
60 PRINT
70 NEXT X
80 IF SQR(Z)<2*Y GOTO 110
90 PRINT "THIS PROGRAM PRINTS SQUARE ROOTS."
100 GOTO 120
110 PRINT "THE LAST ROOT IS LESS THAN TWICE THE FIRST ROOT."
120 END
```

All of the special features of Ohio Scientific's 6502 8K BASIC are described in the OSI 8K BASIC Users Manual. For a more fundamental introduction into BASIC, refer to any of the following books:

Gottfried, B.S.: Programming with BASIC, Schaum's Outline Series, McGraw-Hill, New York, 1975.

Gottfried, B.S.: BASIC Programmer's Reference Guide, Quantum Publishers, New York, 1973.

Greunberger, F.: Computing with the BASIC Language, Canfield Press, San Francisco, 1969.

Kemeny, J.G. and T.E. Kurtz: BASIC Programming, 2nd ed., Wiley, New York, 1971.

Short BASIC Programs

The following short BASIC programs are provided here to allow you to gain some experience with your computer through fully debugged programs which are known to be working. These programs in no way depict the total capability of your computer. They are simple programs which are very short to facilitate manual entry. Each of the programs can be entered in your computer as listed. Remember to type NEW before entering each program. This clears out the computer's workspace. You can substitute a ? for the word PRINT. Ohio Scientific's 8K BASIC allows you this particular shorthand notation wherever the word PRINT occurs. Before you try to write lengthy programs of your own in BASIC, try modifying or customizing any of these programs to get a good feel for how BASIC works.

PROGRAM 1: Number Guess

In this program the computer generates random numbers, and you try to guess what the number is. When you guess the correct number, the computer tells you how many attempts you took to arrive at the correct number.

```
10 PRINT "I WILL THINK OF A"  
15 PRINT "NUMBER BETWEEN 1 AND 100"  
20 PRINT "TRY TO GUESS WHAT IT IS"  
25 N=0  
30 X=INT(RND(56)*99+1)  
35 PRINT  
40 PRINT "WHATS YOUR GUESS ";  
50 INPUT G  
52 N=N+1  
55 PRINT  
60 IF G=X THEN GOTO 110  
70 IF G>X THEN GOTO 90  
80 PRINT "TOO SMALL, TRY AGAIN ";  
85 GOTO 50  
90 PRINT "TOO LARGE, TRY AGAIN ";  
100 GOTO 50  
110 PRINT "YOU GOT IT IN ";N;" TRIES"  
113 IF N>6 THEN GOTO 120  
117 PRINT "VERY GOOD"  
120 PRINT  
130 PRINT  
140 GOTO 10  
150 END
```

PROGRAM 2: Heads-Tails Flipping

This program exercises the RND function of the computer by producing heads and tails. The long-term average out of many runs of this program should be approximately fifty percent heads, fifty percent tails.

```
5 REM HEADS/TAILS FLIPPING  
10 Y=1  
20 C=0  
30 X=1  
40 F=INT(RND(45)*2)  
50 IF F=1 GOTO 80
```

```

60 PRINT "T",
70 GOTO 100
80 C=C+1
90 PRINT "H",
100 X=X+1
110 IF X<51 GOTO 40
120 PRINT
130 PRINT C; " HEADS OUT OF 50 FLIPS"
132 PRINT
133 PRINT
135 Y=Y+1
140 IF Y<11 GOTO 20
150 END

```

PROGRAM 3: ESP Test

This is another number-guess program where you are simply guessing heads or tails as the computer flips a coin. The computer keeps constant tabs on how many right and wrong answers you have given.

```

10 REMESP TESTER
15 REMTYPE E TO END
20 H=1
25 W=0
30 T=0
35 C=0
37 E=10
40 F=INT(RND(12)*2)
42 IF F=0 THEN A$="H"
43 IF F=1 THEN A$="T"
50 PRINT "H OR T ";
60 INPUT X$
70 PRINT
80 IF X$=A$ THEN GOTO 100
83 IF X$="E" THEN GOTO 150
85 W=W+1
87 PRINT "WRONG"
90 GOTO 120
100 C=C+1
110 PRINT "RIGHT"
120 PRINT "W="; W; " R="; C
130 PRINT
140 GOTO 40
150 PRINT "BYE"
160 END

```

PROGRAM 4: Power Generation

This program generates powers of two up to the mathematical limit of the computer. It demonstrates the fact that BASIC automatically reverts back to scientific notation (E-format) when numbers are more than about six digits long up to a maximum of 10 to the 32d power. BASIC can also handle fractions as small as 10 to the -32d power.

```
5 PRINT
7 PRINT
10 PRINT "POWERS OF TWO"
20 PRINT
30 PRINT "POWER      VALUE"
40 X=0
50 Y=1
60 PRINT X, Y
70 Y=Y*2
75 X=X+1
80 IF X=126 THEN GOTO 100
90 GOTO 60
100 END
```

PROGRAM 5: Decimal-Binary Conversion

It is important that the user become familiar with binary in hexadecimal notation if he is to master machine language or assembly language.

```
50 PRINT
60 PRINT
70 PRINT "DECIMAL TO BINARY"
80 PRINT "    CONVERTER"
90 PRINT
93 PRINT
95 PRINT
100 INPUT X
101 IF X<0 THEN GOTO 330
102 IF X>32767 THEN GOTO 330
104 PRINT
105 PRINT "X=";
110 Y=16384
120 A=INT(X/Y)
130 IF A=0 THEN GOTO 200
140 PRINT "1";
150 X=X-Y
160 GOTO 300
200 PRINT "0";
300 Y=Y/2
310 IF INT(Y)=0 THEN GOTO 320
315 GOTO 120
320 GOTO 90
330 END
```

PROGRAM 6: Prime Number Generation

Try to figure out how this program works.

```
10 PRINT "PRIME NUMBER"
11 PRINT "GENERATOR"
13 Y=2
```

```

15 A=1
17 GOTO 80
18 X=1
20 X=X+1
50 Z=INT(Y/X)
60 IF INT(Z*X)=Y GOTO 85
70 IF X*X>Y GOTO 80
75 GOTO 20
80 PRINT A,Y
82 A=A+1
85 Y=Y+1
90 GOTO 18
100 END

```

PROGRAM 7: Acey-Ducey

This is a longer program that should be fun to play. Once you get this program in and running, it would be wise to store it on audio cassette or disk for future use.

```

10 PRINT "ACEY-DUCEY"
12 PRINT "YOU WILL GET 25 HANDS"
13 H=1
15 PRINT
17 T=100
19 PRINT "YOU HAVE $";T
20 X=INT(7*RND(67)+6)
21 IF X>12 THEN GOTO 20
30 Y=INT(X*RND(23)+1)
31 IF Y>X THEN GOTO 38
32 IF Y=1 THEN Y=2
40 R=X
50 GOSUB 500
60 R=Y
70 GOSUB 500
80 PRINT
100 PRINT "YOUR BET";
110 INPUT B
111 IF B<=T THEN GOTO 120
112 PRINT "YOU DONT HAVE THAT MUCH"
113 GOTO 100
120 Z=INT(13*RND(99)+2)
121 IF Z>14 THEN GOTO 120
130 R=Z
140 GOSUB500
150 PRINT
160 IF Z<=Y GOTO 200
170 IF Z>=X GOTO 200
180 PRINT "YOU WIN"
181 PRINT
182 PRINT
190 T=B+T
195 GOTO 300
200 PRINT "YOU LOSE"
201 PRINT
202 PRINT

```

```

210 T=T-B
220 IF T<=0 GOTO 380
300 H=H+1
310 IF H>25 GOTO 400
320 GOTO 19
380 PRINT "YOUR OUT!"
390 STOP
400 PRINT "THATS 25 HANDS"
410 STOP
500 IF A<11 THEN GOTO 505
501 IF A>14 THEN PRINT "ERROR": STOP
502 ON A-10 GOTO 522, 524, 526, 528
505 PRINT A;
510 RETURN
522 PRINT "JACK ";
523 RETURN
524 PRINT "QUEEN ";
525 RETURN
526 PRINT "KING ";
527 RETURN
528 PRINT "ACE ";
529 RETURN

```

PROGRAM 8: Multiplication Quiz

This demonstrates the use of the computer as a teaching aid.

```

10 PRINT "MULTIPLICATION QUIZ"
13 N=0
15 C=0
16 I=0
20 X=INT(RND(56)*13)
30 Y=INT(RND(54)*13)
40 Z=X*Y
50 PRINT
60 PRINT X; "*"; Y; "=";
70 INPUT W
75 PRINT
80 IF W=Z GOTO 120
90 PRINT "STUPID!"
91 PRINT "THE ANSWER IS"; Z
100 I=I+1
110 GOTO 140
120 PRINT "YOU ARE RIGHT!"
130 C=C+1
140 PRINT C; " ARE RIGHT"
150 PRINT I; " ARE WRONG"
160 N=N+1
170 IF N<=9 GOTO 20
180 IF C>=6 GOTO 190
183 PRINT "YOU FLUNKED!"
184 PRINT "PRACTICE!"
185 GOTO 13
190 IF C>=9 GOTO 200
195 PRINT "YOU DID OK"
198 GOTO 210

```

```
200 PRINT "NICE JOB!"  
210 PRINT "TRY AGAIN?"  
220 INPUT T$  
230 IF T$="Y" GOTO 13  
240 END
```

PROGRAM 9: Fahrenheit-Celsius and Celsius-Fahrenheit Conversions

```
10 PRINT "THIS PROGRAM CONVERTS"  
20 PRINT "FAHRENHEIT TO CENTIGRADE"  
30 PRINT "AND VICE-VERSA"  
40 PRINT  
41 PRINT "TYPE THE TEMPERATURE TO CONVERT"  
42 PRINT "A COMMA AND A"  
43 PRINT " 0 TO GET FAHRENHEIT  
44 PRINT " 1 TO GET CENTIGRADE  
50 C=0  
60 F=1  
70 INPUT X, Y  
75 IF Y>1 GOTO 250  
80 IF Y=1 GOTO 200  
90 R=(9*X)/5+32  
100 PRINT " ="; R; "F"  
110 PRINT  
120 GOTO 70  
200 R=(5*(X-32))/9  
210 PRINT " ="; R; "C"  
220 PRINT  
230 GOTO 70  
250 END
```

PROGRAM 10: Subroutines

A simple demonstration of the use of subroutines.

```
10 REMOSI ADVERTISING PROGRAMS  
20 GOSUB 500  
22 GOSUB 500  
25 GOTO 100  
30 PRINT" XXX XXX X"  
35 RETURN  
50 PRINT" X X X X"  
55 RETURN  
70 PRINT" X X X X"  
75 RETURN  
80 PRINT" X X XXX X"  
90 RETURN  
100 GOSUB 30  
110 GOSUB 50  
115 GOSUB 50  
120 GOSUB 50
```

```
130 GOSUB 80
140 GOSUB 70
150 GOSUB 70
160 GOSUB 70
170 GOSUB 30
180 GOSUB 500
190 GOSUB 690
200 GOTO 20
500 X=1
510 PRINT
520 X=X+1
530 IF X>8 GOTO 550
540 GOTO 510
550 RETURN
690 Z=1
700 Z=Z+1
710 IF Z>30 GOTO 720
715 GOTO 700
720 RETURN
```

Lower Case

Ohio Scientific's computer systems are capable of generating lower case characters as well as numerous graphics characters. Under normal operation, the shift lock key is in the depressed or released condition. It must be in this state for normal systems level software to operate. With the shift lock key down, depressing any alphabetic, numeric or punctuation key on the keyboard will cause the keyboard to generate upper case alphabetics and numerics. By depressing the left or right shift key in conjunction with another key, punctuation and special control codes will be generated. For example, depressing the shift key and the 5 key together generates a per cent (%) sign. Depressing the shift key and the P key together generates a commercial at (@) which is recognized by BASIC as being the line delete code.

Shift Lock Key Up

The shift lock key can be released for certain special applications. Specifically, to generate lower case characters as part of literal strings in BASIC and for use in conjunction with word processing software. With the shift lock key in the up position, the keyboard will act considerably different than with it in a locked position. With the shift lock key up, only standard alphabetic characters will generate expected results. Specifically, depressing any alphabetic key will cause the generation of a lower case alphabetic character. In this mode of operation, the left shift key has a different function than the right shift key. Depressing the left shift key in conjunction with alphabetic or numeric keys generates upper case alphabetics and numerics. The right shift key in conjunction with other keys generates upper case punctuation. For example, depressing the 5 key without either shift key generates "garbage". Depressing the 5 key in conjunction with the left shift key generates numeral 5. Depressing the 5 key in conjunction with the right shift key generates the per cent (%) sign. As stated in numerous other places, the shift lock key should be kept in a depressed or locked mode except when lower case characters are explicitly desired.

Advanced Features

Auto Repeat

The Challenger keyboard has a built-in auto repeat feature. By depressing any key and holding it down, first that character will be generated once and then after approximately one-half second, the character will be repeated at a rapid rate.

Programming and Graphics

The Challenger keyboard is directly read out via microcomputer and can be programmed for special functions. The computer system is also capable of a wide range of graphics and gaming displays in

addition to standard upper and lower case characters. Refer to the Challenger Character Graphics Reference Manual for a complete discussion of graphics, and programmable keyboard operation of the computer.

Onwards Towards New Horizons

This is the end of the formal portion of the manual. The remaining sections are specific appendices and deal with the hardware and advanced software topics of the computer system. The following table can act as a guide line towards expanding your computer horizons in areas of more complex software and hardware investigations.

A. More on BASIC

Your Ohio Scientific microcomputer system is capable of storing files on cassette. It is also capable of having program key functions and elaborate graphics. Discussion of the cassette data file capabilities is in the BASIC reference manual. The Ohio Scientific character graphics manual covers the programming of the keyboard and graphics capabilities in detail.

B. Machine Code

This manual includes documentation on the 65V monitor PROM which is built into your Ohio Scientific computer. This allows you to examine memory, load and run machine code programs. Ohio Scientific offers an excellent book HOW TO PROGRAM MICROCOMPUTERS which covers the machine language programming techniques and procedures for the 6502 which is used in this computer and the 6800 and Z80 which is utilized in the Challenger III triple processor system. Ohio Scientific also offers an extended monitor for debugging machine programs, an interactive assembler/editor which will operate on your computer provided it has 8K of RAM or more.

C. Hardware Expansion

The small computer hardware appendix of this manual and the Challenger Technical Report provide an in-depth discussion of the hardware configuration of your computer and its expansion capabilities. By adding more memory and a mini-floppy system, you will be able to broadly expand your computer's capability. By adding a mini-floppy, you will gain the ability to instantly load and save programs and to have random access data files. Your system can be further expanded via the OSI 48 line BUS giving you a wide range of real world interface capability.

D. Other Advanced Topics

Your purchase of an Ohio Scientific small computer entitles you to a one year subscription to the Ohio Scientific Small Systems Journal. The journal covers a broad range of topics for personal and small business users of Ohio Scientific computers. It typically contains software ideas and new products which will be of interest to you as an owner of an Ohio Scientific small computer system.

INTRODUCTION TO SMALL COMPUTER HARDWARE

Small computers are made up of several modules, or blocks. The first of these, the microprocessor, is an integrated circuit much like those used in modern watches and calculators. It performs the function of a large computer, which a few years ago would have been prohibitively expensive. This integrated circuit makes the whole field of personal computing possible and affordable.

Next, one must have some memory, which can be in the form of ROM, PROM, EPROM, or RAM. The first three devices provide permanent storage of programs and data, that is, they do not "forget" when the power is turned off. RAM provides modifiable storage, that is, programs and data can be written in and read out repeatedly. However, almost all types of RAM "forget" whenever the power is turned off. Therefore, RAM is used for temporary storage, and ROM, PROM, and EPROM are used for permanent storage of programs which will not change. Generally, a small computer will have a large amount of RAM for general purposes, and very little PROM or ROM. It does need some of the latter, to give it some intelligence when it is first turned on, and this is usually in the form of a monitor program which allows the user to load additional programs from some external device such as a tape recorder into RAM. Today, the most advanced computers put BASIC, the most commonly used programming language in ROM. This has only recently become possible because BASIC requires approximately 8,000 bytes of ROM, which had been a costly feature until now.

Along with memory, the microprocessor requires some form of I/O device (Input/Output), that is, some way of "talking" to the outside world. The computer communicates through interfaces such as the ACIA-Based Serial Interface and the PIA-Based Parallel Interface to external devices called peripherals, such as CRT terminals, Teletypes, paper tape readers, paper tape punches, line printers, and audio cassettes. Other types of interfaces include D/A converters and A/D converters.

The microprocessor communicates with its interfaces and memory with a series of wires, or lines, called buses. There are generally three buses in any microcomputer: an address bus, a data bus, and a control bus. These three buses are combined in what is called a system bus.

The address bus is generally made up of 16 lines. The microprocessor always is the signal generator for this bus. The 16 lines carry specific addresses, that is, 16-bit binary words which select a memory or I/O location. This location can be thought of as a post office box, and the address word can be thought of as the box number. The microprocessor can, therefore, through its address bus, specify memory or I/O locations.

It can place data in these locations, or read memory from them via an 8-bit wide data bus. Its 8-bit width indicates that the microcomputer can read or write one byte at a time. With ROMs, PROMs, and EPROMs, the microcomputer only reads what is already in those locations, and acts accordingly. In the case of RAM and some I/O locations, the microprocessor can also place data in these locations. Unless the computer has a large amount of ROM, it will generally be very "stupid" when first turned on. That is, its main memory, or RAM, has nothing of value in it. The user must enter a program which the microcomputer can then execute in its RAM memory. It does this by use of a PROM Monitor Program, that is, a short program which the computer runs, allowing it to take data from some interface, and ultimately from some peripheral, and place it into its operating memory, or RAM. It can then later perform functions and write or store additional programs based on this stored program. The typical peripherals used for this are a video display interface, and a keyboard, or a Teletype or CRT terminal. Additional mass storage devices, such as paper tape readers, audio cassettes, and floppy disks, are utilized for storage of programs.

The OSI system utilizes a 48-line system bus made up of an address bus of from 16 to 20 lines (depending on the CPU board used in the system), an 8-line data bus, a 7-line control bus, power connections, and spare lines for user connections. The system utilizes 8" x 10" PC boards plugged into an 8-slot backplane, which spaces the system boards one inch apart. For very small computers, the Model 500 can be used without a backplane board as a stand-alone computer, that is, it can be populated with the microprocessor, PROM and ROM memory, RAM memory, and a serial interface, so that it can function as a complete computer by itself. For larger systems, however, a backplane board and additional system boards are used.

It is necessary for anyone servicing or building an OSI system to be somewhat familiar with the 48-line bus utilized by the computer. This bus is outlined on page

Glossary of Small Computer Terms

ACIA- (Asynchronous Communications Interface Adapter) An IC used for serial data transfer between a device such as a small computer and a serial terminal.

A/D- (Analog/Digital) refers to changing an analog signal to a digital signal which the computer can use.

Backplane Board- (Sometimes called mother board) allows simple interconnection between small computer boards using the same bus.

Bit- The smallest amount of data possible; a bit is expressed as a high or low (on or off) state (normally 1 or 0).

Bus- Refers to the set of foils or wires needed to interconnect between system boards provided that the pattern of how each of the connections is used is consistent for all system boards.

Byte- 8 bits of data. The most fundamental microprocessor commands are organized into sets of 8 bits (i.e. bytes).

CPU- (Central Processing Unit) the portion of a microprocessor which does the actual arithmetic calculations and decision making.

D/A- (Digital/Analog) Refers to changing digital signals (from the computer) into analog signals.

EPROM- (Erasable Programmable Read Only Memory) information stored in an EPROM IC can only be removed by special light sources or specific voltages (depending on the type of EPROM). Through the use of a special programming device, the user can store a set of information in the EPROM after it has been erased.

Hardware- that part of a computer consisting of actual electronic circuitry, printed circuit boards, case, and power supply as opposed to software which is the set of commands the hardware is executing.

I/O- (Input/Output) refers to bringing information into the machine in a form it recognizes and allowing the machine to transmit information. In other words, communicating with the outside world.

Memory- a general term referring to parts of the computer where information is stored.

Microprocessor- a large IC (electronic part) which functions as the CPU of the microcomputer. The 6502 on Ohio Scientific's 500 board is a microprocessor.

PC Board- (Printed Circuit Board) a card with foils (electronically conductive pathways) connecting electronic components which are mounted on the board.

PIA- (Peripheral Interface Adapter) IC used for parallel data transfer.

PROM- (Programmable Read Only Memory) Memory which can have information stored on it once, but, is not normally changeable.

RAM- (Random Access Memory) the data stored in this type of memory is easily changed by the user while the machine is in use (unlike ROM, PROM, EPROM). However, it is erased whenever electrical power is turned off.

ROM- (Read Only Memory) preprogrammed, unchangeable memory.

Software- programs or instructions that the machine will execute,

48 LINE SYSTEM BUS OUTLINE

- B1 - low true WAIT When pulled low by a system board, causes processor clock to slow down to speed of approximately 500KHz on most processor boards. This is used to service slow memory and I/O devices.
 - B2 - NMI (non-maskable interrupt) When brought low, a non-blockable interrupt occurs, causing the processor to stop its operation and service this interrupt, that is, go to a specific memory location and start executing an interrupt service routine.
 - B3 - IRQ (interrupt request) An interrupt which can be masked by the processor, that is, the processor can choose to ignore this interrupt under program control. If the interrupt is not masked, it will cause the processor to stop executing the program it is in, and jump to a different location.
 - B4 - DD (data direction) When pulled low by system board, it changes the direction of the 8T26 buffers on the CPU board, and thus switches the processor from outputting data to the bus to listening to the bus.
 - B5 - D0
 - B6 - D1
 - B7 - D2
 - B8 - D3
 - B9 - D4
 - B10 - D5
 - B11 - D6
 - B12 - D7
 - B13
 - B14
 - B15
 - B16
 - B17
 - B18
 - B19
 - B20
 - B21
 - B22
 - B23
 - B24
 - B25
 - B26
 - B27
 - B28
 - B29 - A6
 - B30 - A7
 - B31 - A5
 - B32 - A8
 - B33 - A9
 - B34 - A1
 - B35 - A2
 - B36 - A3
 - B37 - A4
 - B38 - A0
 - B39 - Ø2
 - B40 - R/W (read/write)
- } Bi-directional eight-bit wide data bus for communication of data between the processor and system boards.
- } Upper data bits on some systems
- } Optional reset line used to clear all PIAs and similar I/O circuitry in the system.
spare line
- } Memory management address lines: Lines 21 and 22 are used on systems with a 500 CPU Board; all 4 are used with the 510.
- +12 Power connection
-9 Power connection
+5 Power connection
- } Ground Connection
- } Ten low-order address lines
- Used to clock external circuits or external I/O interfaces, such as the A/D converter.
Originates at the microprocessor and specifies read or write operations on the data bus.

B41 - VMA (Valid memory address) Only used in conjunction with the 6800 microcomputers. The 6502's always have this line high.

B42 - Ø2-VMA Master timing signal for enabling memory and I/O in the system.

B43 - A10

B44 - A11

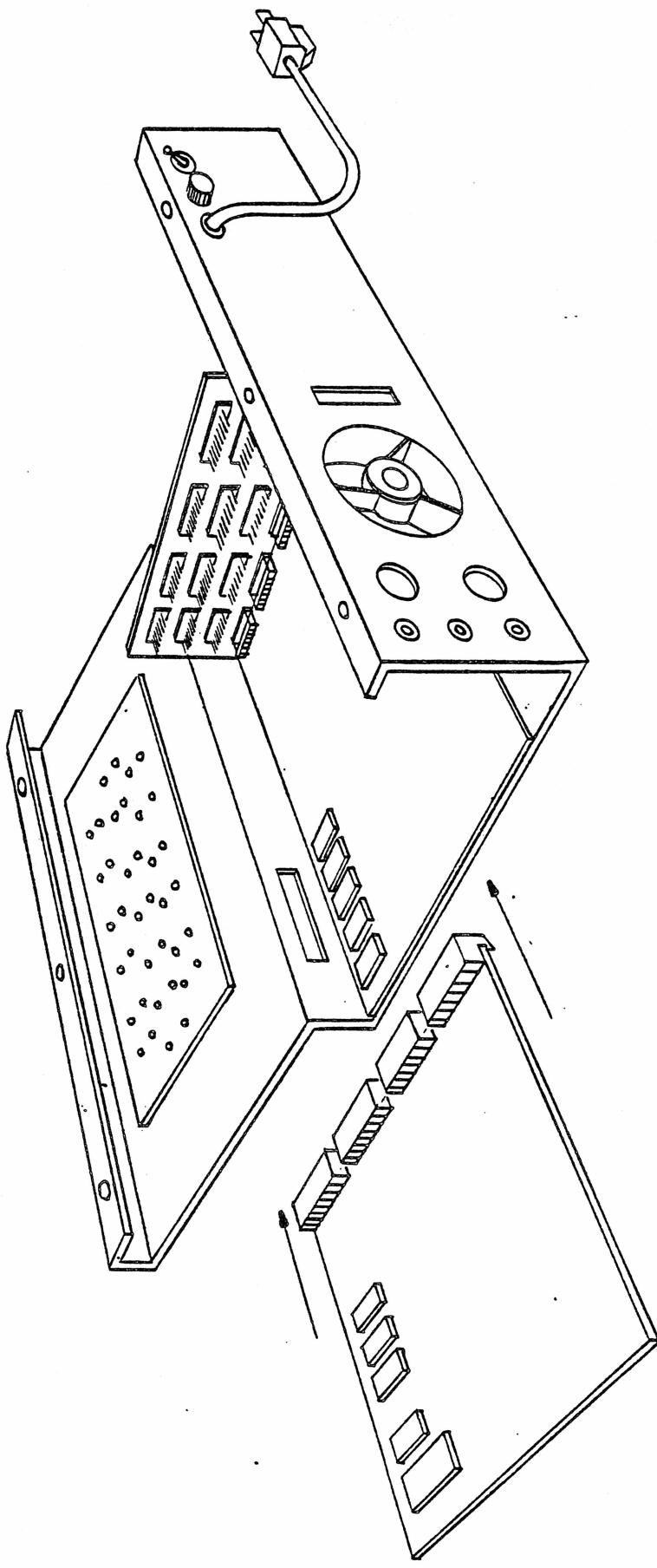
B45 - A12

B46 - A13 Six high-order address lines

B47 - A14

B48 - A15

The C2-4P utilizes a 4-slot backplane. A 4K or 8K unit typically has a Model 502 CPU/memory board and a 540 video display board, leaving two slots open for expansion. Popular options include mini-floppies which use one slot and 8, 16 or 24K 527 RAM board. This yields a maximum configuration of 32K RAM and dual mini-floppies. The CA-12 96 line parallel board is also a popular option on C2-4P's. Note that the C2-4P has only +5 volts so not all OSI 48 line boards can be used in it without modification.



INSTALLING OS1 48 BUS BOARDS IN THE C2-4P

65V PROM MONITOR

INSTRUCTIONS

Anyone wishing to become proficient in programming at the machine language level should be well acquainted with the PROM Monitor which is necessary to examine and change the contents of memory locations. The Monitor used in conjunction with the 6502 video system is the OSI 65V PROM Monitor.

As explained above, OSI's 8K BASIC comes up when the computer is reset. When this is done, the letters "C/W/M?" or "D/M?" appear on the video screen or terminal. If you wish to enter a program by means of the Monitor instead of BASIC, type an "M" on the keyboard. This brings up the Monitor and starts it in the Address Mode, that is, the mode in which you can specify memory addresses or locations to simply examine their contents. Appearing on the screen are four digits in hexadecimal notation followed by two spaces and, finally, another two digit number, also in hex. The four digit number is a location, and the two digits are the contents of that location. To examine another address, type the address on the keyboard. The same address will appear on the terminal, as will the corresponding contents of that address.

If you wish to change those contents and thus enter programs using the Monitor, you must exit the Address Mode and get into the Data Mode. To do this, type a slash (/). Then type any two hex characters, and they will be inserted into that location as its new contents. The normal procedure for entering programs via the Monitor is to use consecutive memory locations. While still in the Data Mode, you can open the next address by typing the return key. You can do this continually, each time altering the memory contents according to the needs of the program you are writing. If you want to jump to a non-consecutive location, you need to get back into the Addressing Mode by typing a period (.). Then type the new address you want. Type another slash to get back into the Data Mode and continue as before until you want to open a non-consecutive address. Extreme caution should be used whenever the Monitor is in the Data Mode as you are directly manipulating the computer's memory.

If you wish to enter a program from an audio cassette instead of manually from the keyboard, first get into the Address Mode, then turn on the cassette. Let the tape advance to the point where the program of interest begins, and type L. This transfers control to the audio cassette, such that all ASCII commands are supplied by the cassette instead of by the keyboard. The L command also puts the Monitor into the Data Mode. If the contents of 00FB are 00, the Monitor will accept commands from the keyboard.

If the cassette does not load 00FB (hex) with 00, to transfer control back to the keyboard, press reset. Otherwise, commands are accepted from the Audio Cassette UART.

To run any program which you have entered via the Monitor, get into the Address Mode, i.e., type a period, type the starting address of the program, and type a "G".

<u>Label</u>	<u>Program Entry Points</u>
VM	FE00 - Restart Location FEOC - Bypasses UART and Stack Pointer initialization and the clearing of decimal mode but does clear the screen.
IN	FE43 - Entry into address mode, bypass initialization
INNER	FE77 - Entry into data mode, bypass initialization
<u>Label</u>	<u>Subroutines</u>
OTHER	FE80 - Input an ASCII character from Audio Cassette UART
LEGAL	FE93 - Returns stripped ASCII number if 0-9 or A-F. Otherwise returns a FF.
INPUT	FEED - Input an ASCII character from keyboard

Required Hardware

The 65V Monitor requires as a minimum the following hardware: an OSI Model 400 board with a 6502 microprocessor, 1,024 words of RAM memory located from 0000 to 03FF, and the 65V monitor itself. It also requires an OSI Model 440 Board populated for alphabetic display and keyboard input. The 440 Video Board must be located at DXXX which will automatically locate the keyboard input at DFXX.

The keyboard must be a seven-bit high true ASCII keyboard with a positive or negative going strobe pulse at least 100 microseconds long.

The 65V Monitor will additionally support input from a generalized serial communications subsystem of an OSI 430 board located at FBXX. Specifically, the monitor contains a load program for a 430 board-based audio cassette interface. The same program can be used with a 430 board configured for digital cassette or ASCII teletype input.

Commands

Address Mode Commands:

- / - Change to Data Mode
- G - Go -- Jump to location seen on screen and execute program found there.
- L - Transfer control to audio cassette.

Data Mode Commands:

- . - Change to Address Mode
- RETURN - Open next address. In other words, increment location pointer by 1.

If the 65V is in address mode, typing 0 - 9 or A - F will cause that number to be rotated into the LSD of the location pointer. Typing a 4 causes 0123 XX to become 1234 XX.

If it is in Data Mode, the number is rotated into the data contents and memory is thus modified. This permits the easy correction of errors. If, for example, the user typed 0478 when intending to look at location 047B, he would simply type 047B.

All of the non-command keys and non-hexadecimal characters are ignored by the monitor.

65V Demonstration Program

The following is a program which may be entered using the 65V Monitor from the keyboard or audio cassette. An "*" indicates a return key depression.

.0002 Loads the ASCII Message Starting at Location 0002
/4F * 53 * 49 * 20 * 36 * 35 * 56 * 2E * 5F

.0200 Loads the Main Program at 0200
/ A9 * 02 * A2 * 00 * 20 * 00 * 03 * A2 * 00 * 20 * ED * FE * 9D * 24 *
D2 * E8 * 4C * 09 * 02

.0300 Loads the Subroutine at 0300 to Output an ASCII Character String.
/85 * 00 * A9 * 00 * 85 * 01 * A0 * 00 * B1 * 00 * C9 * 5F * F0 * OA * 9D *
E4 * D1 * E8 * E6 * 00 * D0 * F2 * E6 * 01 * 60

.0200G Loads the Starting Address of the Program and Execute it.

You should see the message "OSI 65V." on the screen. Now, you may type any keys and they will be echoed just below the message. Press reset to re-enter the 65V Monitor.

If this were entered off of the audio cassette, it would be self-loading and auto starting. Since the cassette is in complete control, it can load the starting address and execute the program without user interruption.