THE CHALLENGER III
SET UP AND OPERATIONS MANUAL

Nov. 1978

Table of Contents

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

NOTE: All Challenger owners will receive a free one year subscription to Ohio Scientific's Small Systems Journal upon receipt of the Warranty card below.

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 11Ø VOLT WIRING WITHIN. ALSO, THE UNSHIELDED LOGIC CIRCUITRY MAY EMIT RADIO FREQUENCY INTERFERENCE.

Setting Up the Computer

The following instructions are designed for an Ohio Scientific representative or other technically qualified personnel.

All Ohio Scientific Challenger III systems come minimally equipped with:

*An RS-232C "Console Port" factory configured for 2400 baud operations.

*At least 32K RAM memory

*Dual 8" floppies for mass storage

*65DV3.0 Operating System Software

The manual will describe set up and operation in the following order:

1. Discussion of minimal safe operating requirements.
2. Connection of the floppy disk and terminal.
3. Initial Check out of the CPU.
4. Initial check out of the floppy disk.
5. Addition and check out of accessories including hard disk.
6. Computer operation.

Challenger III Minimal Physical Requirements

Temperature

Challenger III systems are to be operated in an air-conditioned environment where temperatures are not to exceed 60° to 75° F and humidity not to exceed 30% to 80%.

Air Circulation

The Challenger III systems require adequate air circulation in the vicinity of the computer equipment and peripherals. The C3-S1 computer may be stacked on top of the floppy disk drives, however, the floppy disk cannot be stacked on top of the computer. The C3-S1 and C3-OEM must have at least 6 inches clearance on both sides and the back of the computer for proper ventilation. The top of the computer system must not be blocked. Other heat producing devices such as terminals and printers must be at least one foot away from each other and the computer system. The C3-A and C3-B series computers must be one foot from the back wall to assure proper air flow through the rear vents of the computer. The computer systems are not to be exposed to direct sunlight since this can cause very high local temperature rises on the panels and heat sink surfaces of the computer.

Mechanical Vibrations

Computer systems and terminals should be isolated from printer mechanisms preferably by the use of a printer stand. Hard disk units are extremely susceptible to vibration damage when they are operating, therefore, any Ohio Scientific hard disk unit should be installed in a floor mounted equipment rack configuration such as utilized on the C3-A and C3-B computers. The hard disk units must be mechanically isolated from vibrations induced from line printer mechanisms for successful long term operation.

Electrical Connections

All Challenger III computer hardware and accessories require properly installed three wire grounded receptacles which provide 110 volts AC at 60.0 hertz. The safe operating range for Ohio Scientific equipment is 105 to 128 volts AC. Hard disk based computer systems require dedicated 30 amp three wire circuit. Typical Challenger installations require from three to six three wire receptacles. This can be provided by an UL approved plug strip connected to one 110 volt power circuit. All three connections of each receptacle should be tested by the technician with a voltmeter and an absolute ground reference before the computer system is connected. This is necessary because three wire building wiring and commercially available plug strips frequently have improper wiring and faulty grounds.

Interference Sources

Brush type DC motors, large AC motors (on starting) and faulty fluorescent lamp ballasts can provide disruptive sources of interference when connected to the same power line or operated in the immediate vicinity of the microcomputer.

Static Discharges

Static discharges produced typically by operators walking across carpeted areas and touching computer equipment can cause disruption in computer operation, loss of data and possibly permanent damage to the computer equipment. The best solution to this problem is to install the computer system in a non-carpeted area. If this is not feasible, the carpet in the vicinity of the computer system should be treated with an anti-static spray and the humidity of the room be kept above 50%.

Safe Operating Check List

1. Proper room air-conditioning.
2. Proper spacing between equipment.
3. Vibration isolation between hard disks and printer mechanisms.
4. Proper AC wiring.
5. Precautions against static discharges.

Note: In practice, the Challenger III system will work in far more severe environments than specified here. However, in professional applications involving several personnel and valuable data, these precautions must be followed to insure the safety of the equipment, personnel and the integrity of data.

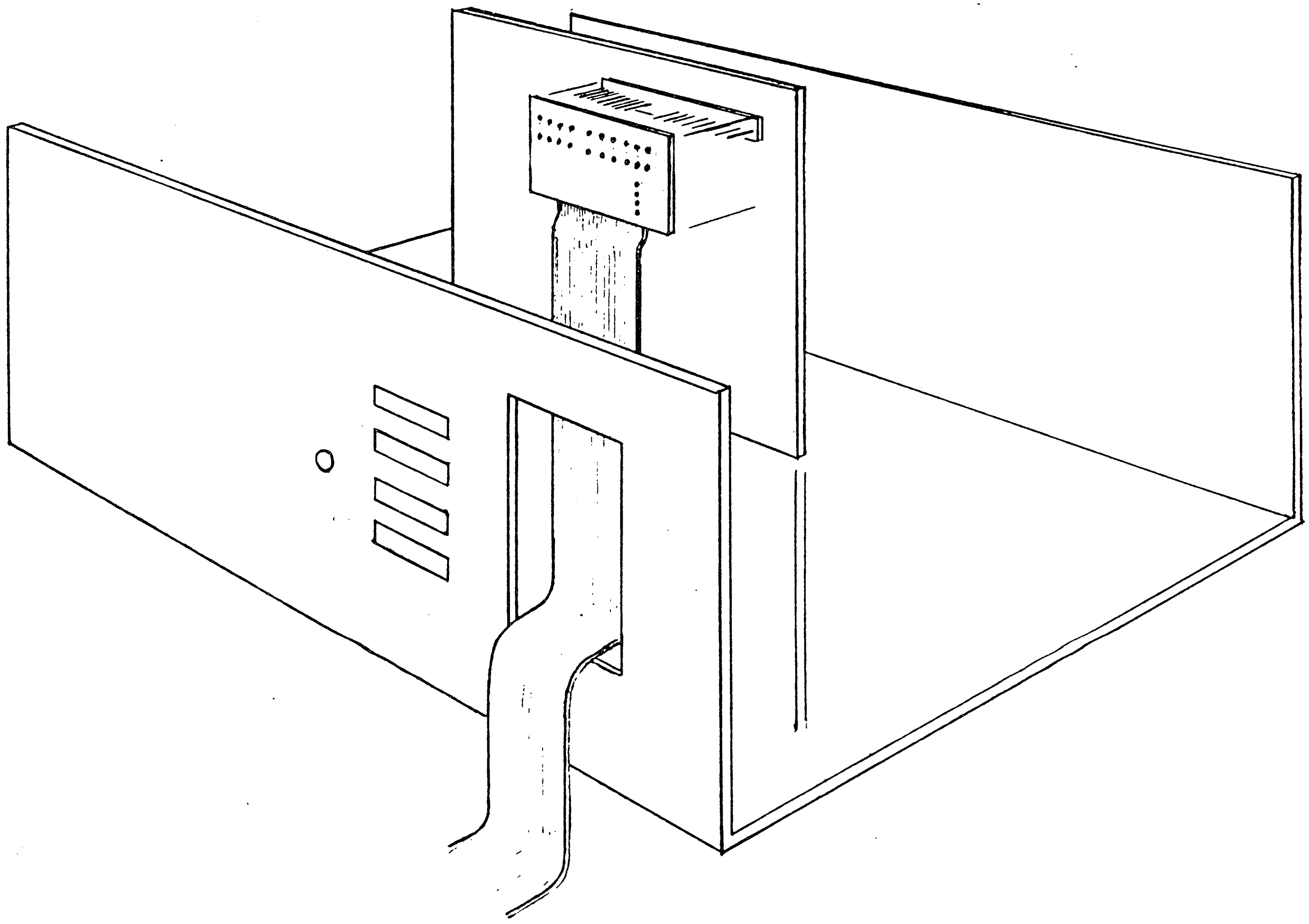


Diagram 1

Installing the floppy cable.

Connecting the Terminal and Floppy Disks

Once the appropriate installation site is chosen and the power distribution has been checked out, the computer system can be unpacked. The cover, if present on the computer, should be removed. The inside of the computer should be inspected for packing material and small accessories which should be removed if present. Sometimes hard disk connector cables and other optional connector cables are present inside the computer in a coiled up fashion. These connectors should be uncoiled and routed to the large rectangular opening in the back of the computer. Exposed contacts of connecting cables should be temporarily wrapped in insulating material to preclude shorting conditions during the initial power up testing. Floppy disk drives should be unpacked; cardboard protection diskettes should be removed from the floppy disk drives. The cable from the floppy disk drives should be mated with the 470 or 505 floppy disk controller board which is generally the rear most board in the computer as shown in Diagram 1. The floppy disk cable should be routed into the computer via the rectangular opening in the rear of the system. The computer and floppy disk top should be put in place or the unit(s) should be slid into their rack with the disk in an upmost position and the floppy disk and the computer in the mid-range position (in rack configurations). In table top configurations, the units can be side by side or computer on top of floppy disk (not vice versa).

Connection of a Terminal

To test out the computer, you will minimally require an ASCII RS-232 compatible terminal and EIA standard 25 pin cable of appropriate length. The cable should be mated to the upper most EIA connector at the rear of the computer CPU and to the EIA input jack at the rear of the terminal. Carefully make these connections because there may be several other EIA jacks on the rear of the computer and on the terminal for other optional configurations. The terminal must be set as follows:

1. 2400 baud
2. No parity
3. Two stop bits
4. Full duplex
5. Manual carriage return

These selection options are usually accessible at the rear of the terminal or underneath a name plate or access plate on the front of the terminal. Consult the terminal's manual for further instructions. If the terminal has lower case capability, be sure that the shift lock key is in the down or locked position on the keyboard of the terminal. With these connections made, first turn on the terminal and allow a few seconds for it to warm up. A small line, dot or square should appear in the left hand corner of the screen. Then turn on the computer's CPU but not the floppy. The reset switch in the front of the computer should glow brightly. If it does not glow at all or glows dimly, turn the computer off immediately. Check the power connections, the fuse at the rear of the computer and the seating of the pilot lamp which can be accessed by removing the lens cap. If the problem persists, refer to the maintenance manual. On C3-A, C3-B and C3-OEM computers, two switches are present with a white and yellow lens cap. These two pilot lamps are connected to two separate power supplies so that if one lamp is out or dim, it indicates a problem with its corresponding power supply. Assuming that the pilot light on the front of the computer lights appropriately, proceed as follows:

If the computer system has a high speed option which is indicated by the existence of a red pilot lamp, be sure that this pilot lamp is off, indicating that the high speed option is not selected during the initial test. This is accomplished either by a toggle switch at the rear of the computer or by the alternate action yellow lens cap switch at the front of the computer. Now, perform a master reset of the computer by momentarily depressing the white reset button on the front of the computer and releasing it. The message H/D/M should appear on the screen. If it does not appear, repeat the process one or more times. Each time the reset button is depressed and released, the message H/D/M should appear on the screen. Once the message appears on the screen, depress the M key on the terminal which will select the computer's internal machine code monitor ROM. This will cause a carriage return and a line feed to occur on the terminal. The computer is now ready to accept the commands of the 65A serial monitor ROM. For initial testing purposes, type PØØØØ. Upon the completion of typing the fourth <u>zero</u>, the computer should immediately start printing out the hexadecimal information contained in memory location zero on in rows of eight locations across the screen each containing two hexadecimal characters. The print out should be of arbitrary characters within the range 0 through 9, A,B,C,D,E, and F in neat rows and columns. If there are additional spaces that occur or letters other than 0 through 9 and A,B,C,D,E, F, the baud rate of the terminal or the parity or the number of stop bits is incorrectly set or adjusted. To terminate the memory dump, type any key on the keyboard. This completes the initial test of the CPU indicating that the central processor is functioning and at least some of the memory of the computer is operational as well as the terminal interface and the terminal. Now proceed to the floppy disk operation.

Floppy Disk Care

The floppy diskettes and disk drives are delicate pieces of hardware, and should be treated as such. The following rules are strongly recommended to maintain their good condition.

A. Handling Floppy Diskettes

1. Do not touch the surface of the diskette or allow any dirt or dust to come into contact with the surfaces.

2. Be very careful in labeling diskettes, so as not to damage them.

3. Do not bend or fold the diskette.

4. Store the diskette only at temperatures from 10° to 125° F. (-18° to 51° C.) and only use a diskette in a drive if both are at the same temperature.

5. Do not allow magnets to come near the diskette.

6. Always place the diskette in its jacket and store it upright in its box when not in use.

7. If you must lay a diskette on a table, place it with the label side down, to avoid damaging the recording side.

8. When inserting a diskette in a drive, insert it carefully with both hands and an even pressure, until you hear a click. Make sure that it has not come back out slightly before you close the drive.

9. Do not try to clean the surface of the diskette.

10. Turn on the power to your computer before you insert the diskette, and turn power off only after you remove the diskette. Never turn the power on or off while the diskette is in the drive.

11. Insert the diskette in the disk drive with the label side up.

12. Use only 100% certified single index hole diskettes, such as the ones which OSI offers.

B. Handling Disk Drives

1. The disk drive should only be turned on or off when the computer has been turned on.

2. Diskettes should be inserted in the drive after the drive has been turned on, and removed before it is turned off.

3. Do not obstruct the air flow to the disk drive in the rear.

4. Disk drives and diskettes will not operate in very high or very low humidity environments. Air conditionins is generally not required unless the unit is operated in a basement, or other area where condensed moisture is likely to occur. Rugs and carpeting in the vicinity of the computer should be treated for anti-static.

5. The disk drive, being a mechanical rotational device, is susceptible to line voltage and line frequency variations. The unit must be operated at 60 .0Hz for write operations to work.

6. The floppy disk system is mechanical, and thus subject to wear on pulleys, belts, bearings, etc. It is a good practice to remove diskettes from disk drives when disk operations are not anticipated during the next hour or so. Also, turn off disk drives when not in use for prolonged periods of time.

Notes on OSI Dual Sided Floppy Operation

Only disks certified for operation on both sides should be used.

Disk drives will run all existing Ohio Scientific software with no modifications but will perform as normal single sided floppies with device "A" being the top drive and device "B" being the bottom drive.

OS-65D V3.0 and OS-65U will support the CD2+2 option.

The nomenclature for device selection is:

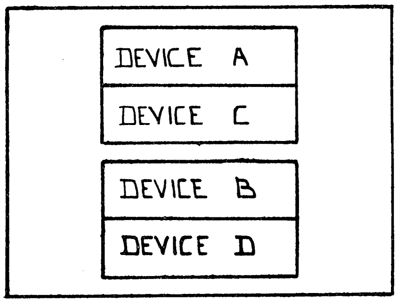
DEV "A is top surface of top disk drive

DEV "B is top surface of bottom disk drive

DEV "C is bottom surface of top disk drive

DEV "D is bottom surface of bottom disk drive

This is illustrated below:



Checking Out a Floppy Disk Drive

With the terminal and computer turned on, turn on the floppy disk drive. Place a 65D V3.0 diskette, label side up, in the upper or "A" drive of the disk, heeding all the warnings and guide lines on the preceding pages about disk usage. With the diskette firmly in place, reset the computer. Depress the "D" key. You will hear a series of several clicks, then the message OS-65V3.0 development disk and some other messages and finally FUNCTION? on the screen. To the question FUNCTION?, type UNLOCK return. The computer will respond "SYSTEM NOW OPEN FOR MODIFICATION". The computer system is now ready to accept commands of the standard programming language BASIC. The computer system operations to this point indicates that the floppy disk controller and the major portion of the memory CPU "A" drive of the floppy disk system are operational. If this is the user's or installer's first encounter with Ohio Scientific computers, the 65D V3.0 disk operating system manual should be followed from this point to gain a familiarity with 65D V3.0 operating system.

If this is an installation for an end user who will receive specific training on the computer, one can omit this step for the moment.

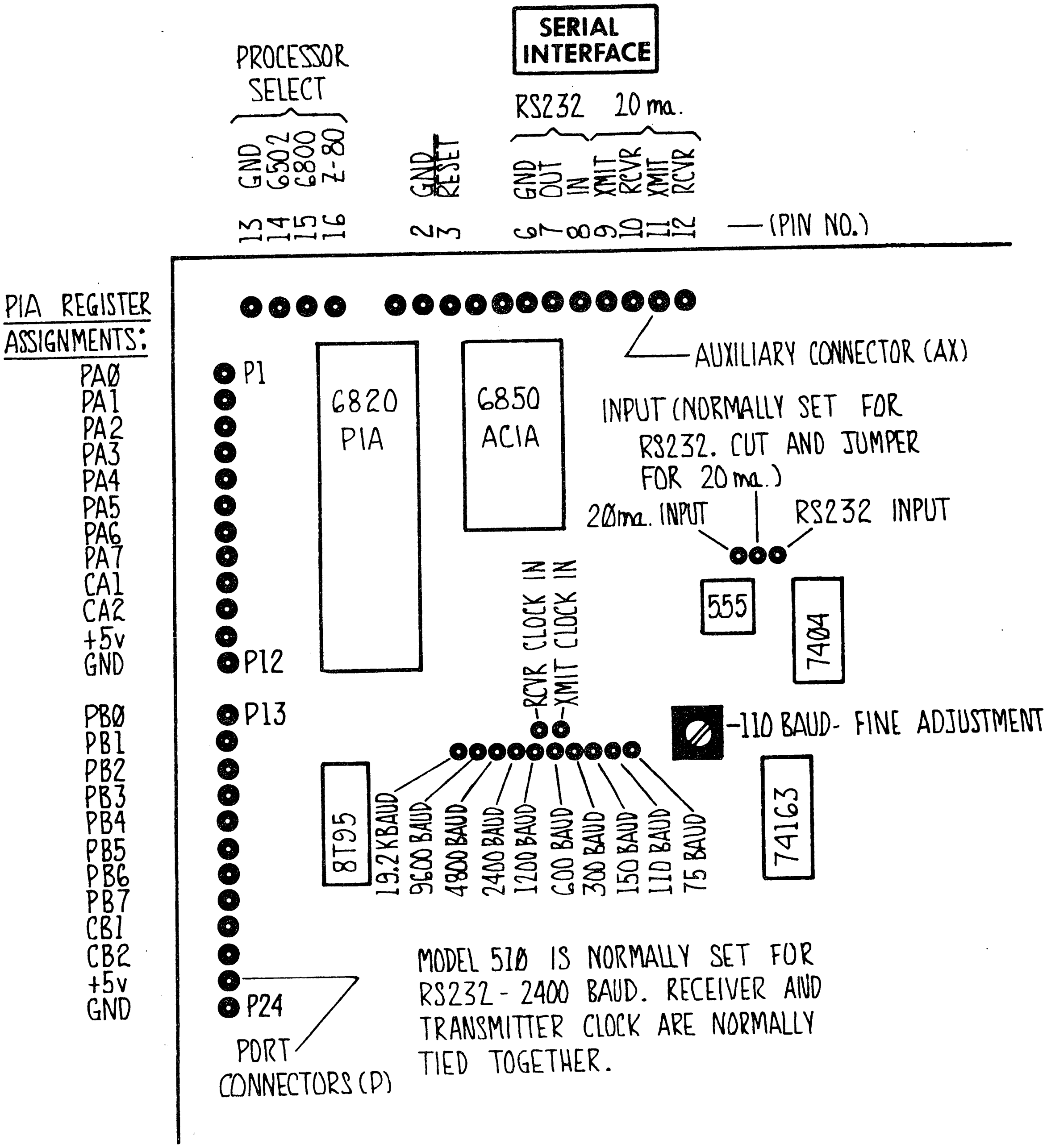
Diagnostic Check Out of the Computer

Two performance tests can be optionally carried out as outlined in the maintenance manual for Challenger III systems. One test is the memory test which can be performed by simply placing the memory test diskette in the computer, depressing the reset switch and typing "D". During the initial installation phase of the computer, this test should be run for several operations, and periodically every few hours for the first several days of the computer's use. This is because memory failures due to infant mortality are most likely to occur during the first few days of operation. Likewise, the floppy disk drives can be exercised by the selection of the dual drive test diskettes or the dual drive double sided test diskettes. The procedures are outlined for these diskettes in the maintenance manual.

Once the memory and the floppy disk drives of the computer have been tested, one can proceed to installation of additional accessories.

Installation of Additional Accessories

Once the computer system has been thoroughly checked out with a 2400 baud terminal, it can be reconfigured for use at a different baud rate or for use with a 20 mill current loop device such as a Teletype$^{(R)}$. The model 510 CPU board in the Challenger III provides the user with two classes of options. One is the selection between RS-232 and 20 mill current loop, and the other is baud rate selection.



MODEL 51Ø IS NORMALLY SET FOR RS232 - 2400 BAUD. RECEIVER AND TRANSMITTER CLOCK ARE NORMALLY TIED TOGETHER.

(UPPER LEFT CORNER OF BOARD. VIEW FROM FRONT.)

MODEL 510 I/O CONNECTIONS

Diagram 2

Baud Rate Selection

Refer to Diagram 2 for the connections on the 510 board for selecting baud rates other than 2400 baud. With the terminal and the computer turned off and disconnected from the 110 volt source, the CPU board of the computer can be carefully removed, carefully noting the cabling connections to the CPU board with magic marker so that these connections can be properly reinstalled. Locate the jumper area on the board and cut off the old jumper. Using a piece of 22 ga. or thinner tinted wire and 60/40 rosin core solder, rejumper the board for the desired baud rate. Likewise, make the baud rate selection at the terminal.

Current Loop Operation

For current loop operation, it may be necessary to change the baud rate from 2400 baud and cut the RS-232 jumpers and replace them with the 20 mill current loop jumpers as specified. It will also be necessary for the installer to fabricate his own wiring harness directly from the CPU to his terminal as necessary. Diagram 3 shows the pin outs for 20 mill current loop and RS-232 at the CPU's connector. Diagram 4 indicates the corresponding serial current loop connections for a standard ASR33 Teletype. Particular caution must be maintained in connecting cables to an ASR33 Teletype since 110 volts AC and possible high voltage DC are both present on the terminal strip of the Teletype. Even a momentary connection to these terminals will cause immediate destruction to a major portion of your computer. Damage due to faulty hook ups is specifically not covered under the warranty.

Diagram 3

INTERFACE CONNECTIONS

500, 510 Auxiliary Connector	Description	EIA Standard Pinout
6	RS-232 Ground	7
7	RS-232 output from CPU (input to terminal)	3
8	RS-232 input to CPU (output from terminal)	2
9	20ma loop CPU output (+) connect to pin 7 on TTY	13
10	20ma loop CPU input (+) connect to pin 4 on TTY	12
11	20ma loop CPU output (-) connect to pin 6 on TTY	25
12	20ma loop CPU input (-) connect to pin 3 on TTY	24

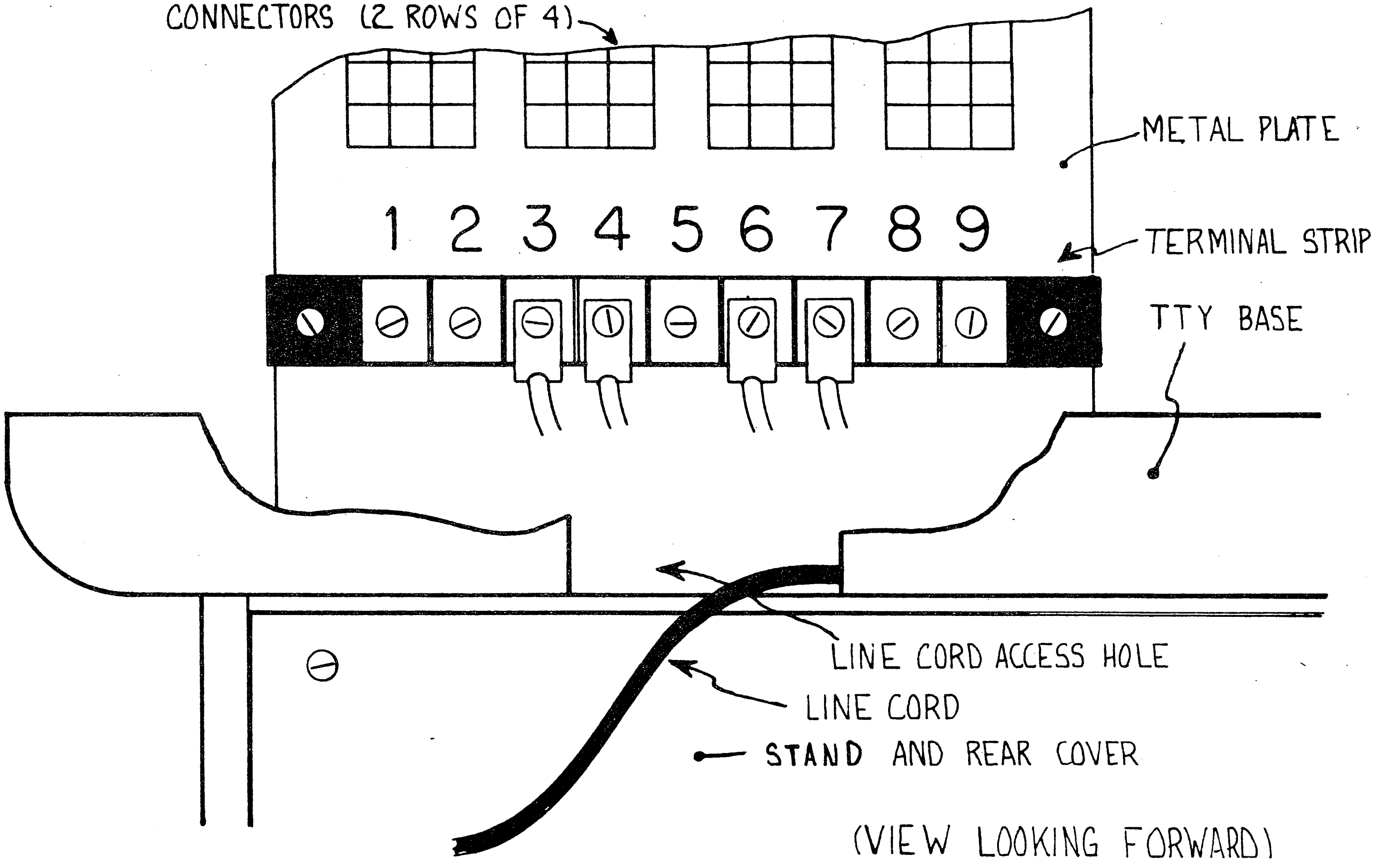


Diagram 4. Connecting to a Teletype

Adding Memory

In order to run OS-DMS, OS-AMCAP and other 65U applications programs, 48K of RAM must be present. If the system is configured minimally for 32K, an additional CM-3 RAM memory board must be added. This CM-3 memory board must be addressed for the third 16K memory space as specified in the maintenance manual. To operate OS-CP/M and OS-65U Level 2 or 3, the computer system must minimally have 56K of RAM memory. This is accomplished by adding either a CM-10 memory board to standard systems which have 48K or a CM-2 memory board to computer systems configured with a hard disk. The CM-10 memory board is automatically configured for use in this application and can be plugged into any open slot. The CM-2 memory board must be configured as specified in the maintenance manual for use at DØØØ hex.

Once the desired amount of memory has been added to the computer, the memory test diagnostic and floppy disk test diagnostic should both be performed. If difficulty is encountered, refer to the maintenance manual.

Hard Disk Care

1. Disk drives should always be turned on after the computer is turned on and turned off before the computer is turned off.

   A. Never turn the hard disk on while the computer is off.

   B. Never switch the computer and disk on and off together.

2. Turn the computer on, reset it. Turn the hard disk on, wait two minutes minimum and then reset the computer again.

3. You must wait at least 15 minutes before turning on the disk after it has been turned off.

4. Never operate the hard disk during brown outs or any other period of unstable power such as lightning storms.

5. Never operate the system outside the recommended temperature range.

6. The hard disk is very susceptible to vibration damage while running. Never move the drive (even a few inches when running. Always install the shipping clamps when moving the hard disk any distance.

The following optional procedures are recommended for the ultimate data integrity on hard disks.

A. Perform an operational test of the computer with a floppy disk (such as the memory test) each morning before powering up the hard disk.

B. Wait five minutes after hard disk power up before writing to the disk.

C. Refresh the data on the hard disk every three months by copying files to other areas of the disk and then back to their original positions. (Mechanical alignments within the disk drift slightly with aging, the refresh operation keeps the data "in tune" with the mechanics.)

Installing a Hard Disk

All Ohio Scientific hard disk options utilize Model 590 and 525 controller combinations in the computer which are connected together both at the computer backplane and via the special memory channel. Connections to the hard disk drive are clearly labeled and should be made with all computer system and accessories disconnected from the AC line. On most hard disk options, innerconnections include a 12 ga. ground, two 16 pin ribbon cables and one twisted pair ribbon cable set. Once these connections are made; replace all protective covers on the computer's CPU and disk drives; reconnect all computer devices; power up the computer, terminal, floppy disk as usual and verify normal floppy disk operation.

Turning on the Hard Disk

With the computer system up and running, turn on the hard disk drive. If there is any noticeable flicker in the pilot lights on the computer when the hard disk is turned on, turn the hard disk off immediately. This indicates that the power wiring to the computer system is inadequate to support the starting current of the hard disk. A heavier duty circuit will have to be used with the computer system utilizing heavier gauge wire from the primary feed of the building. After the hard disk has been turned on, wait approximately two minutes for the unit to come up to speed. Then verify that the computer system is still working satisfactory as a floppy disk only computer. Providing that it is, reset the computer and select "H" for hard disk bootstrap. After a few seconds, an OS-65U operating system message should appear on the screen. If this is the installer's first encounter with OS-65U, the operating system should be thoroughly investigated on floppy disk before proceeding with hard disk operation. The 65U operating system provides extensive hardware error checking and will constantly check the integrity of the hard disk unit during normal operation. The hard disk error codes are listed in the OS-65U manual. The installers and users should pay particular attention to the hard disk usage warnings listed immediately preceding this section during initial set up and normal use.

Installation of Other Options

There is a broad range of other options available for the C3 series computers including line printer interface and two line printer options and additional RS-232 ports, parallel interfaces, etc. In most cases, these options are factory supplied properly addressed and configured such that they can be directly plugged into the computer system. In each case, the installer should confine his work to one operation at a time, performing memory and disk I/O diagnostic tests after the installation of each accessory so that problems can be immediately spotted if they occur. Refer to the maintenance manual for more information on installing additional options to the computer.

This section concludes the installation portion of the manual. The following section concerns introductory operation of the computer system for the installer or end user who does not have the benefit of formal training on the computer system.

Using Applications Diskettes

At this point, you have set up your computer system and should be able to run and utilize commercially available diskettes for the Challenger III. Several programs on diskette are provided with the unit and a large library of applications software is available from Ohio Scientific and your local Ohio Scientific dealer. Your Ohio Scientific computer can provide you with years of benefit by utilizing commercially available applications software alone. However, you may wish to learn how to program yourself and may even wish to explore and expand its hardware capabilities. The following sections of this manual are concerned with expanding your horizons in both programming and the hardware capabilities of the computer. The next section specifically deals with an introduction to the programming language BASIC. This section should be used in conjunction with the accompanying Ohio Scientific BASIC Reference Manual and the accompanying book, "BASIC And The Personal Computer".

The 65D V3.0 User's Manual provides an in-depth discussion on the use of DOS (Disk Operating System) in conjunction with the BASIC and Machine Code Sections in this manual.

INTRODUCTION TO SMALL COMPUTER SOFTWARE

In order for a computer to perform even simple operations, it obviously needs a means by which the user can communicate instructions to it. Any such means which consists of a set of rules to convey information, is called a computer language. The numerous languages in use today offer a wide range of specific applications and varying degrees of understandability for the user. They can provide direct communication with the computer at the complex level of machine language, or enable the programmer to use an indirect communication by means of a higher level language which corresponds more closely to human speech.

Machine level languages are really the most practical device from the computer's point of view, because when you use them, you are really speaking the computer's own jargon, and are thus making more efficient use of memory space. On the other hand, when you use an upper level language, every instruction you give, in what resembles "plain English", has to be converted into one or more separate instructions in a machine level language. Therefore it is obviously less wasteful in terms of time and effort to write in machine language, and skip translations from other languages altogether. The major drawback in machine languages, however, is that they are difficult for the average person to learn.

Machine languages consist of binary codes used in all of the commands which are entered by the programmer. These codes are 8-bit groups of on-or-off switches, which in various combinations, serve as instructions for the computer. Although the majority of users will have no need or desire to learn these combinations, there are some who, for one reason or another, will want to program their computer directly. Since it is quite troublesome to commit several dozen combinations of numbers to memory, a system of abbreviations (mnemonics) has been devised which exactly correspond to machine language instructions. The program which converts these mnemonics into machine language is called an Assembler. By following the instructions provided with your computer, you can make use of the Assembler and write a program in mnemonic code. This is directly translated into the binary object code which the machine understands. After you gain proficiency, you can even begin to use the actual object code to do your programming, examine and change memory locations, etc., and thus be in ever greater control of your machine.

Upper level languages, in contrast to machine level languages, are much easier for humans to master. Nevertheless, every upper level language has to originate at the machine language stage, and usually represents a long, tedious effort on the part of the author or authors of that language. Probably the most common upper level language is BASIC (Beginner's All-Purpose Symbolic Instruction Code). An 8K version of BASIC written by Microsoft, Inc. (i.e., it occupies 8×2^{10} locations in memory) is used in all of OSI's 6502 computers. Because of BASIC's popularity, simplicity, and versatility, OSI has made it a standard feature in its product line, either by placing it in a computer's permanent memory (also known as Read Only Memory [ROM], which does not "forget" once the power is turned off), or by reserving special tracks for it on floppy or hard disks. In addition, in OSI products, BASIC is always immediately available to use, because it comes up automatically the instant the computer is reset. Therefore, the programmer is free of the burden of manually bringing in BASIC, which would demand that he be thoroughly versed in the computer's internal thinking processes and machine language. The fact that BASIC comes up automatically is very convenient for computer programmers, most of whom probably have programs they would like to run or write in BASIC.

There are a large number of publications available which describe in detail the commands and functions of BASIC. While this introduction can in no way duplicate 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≤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 desi--e 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 progams 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 "T00 SMALL,TRY AGAIN ";
85 GOTO 50
90 PRINT "T00 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 FND
```

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 digits long up to a maximum of 10 to the 32d power. BASIC can also ndle 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-Deucy

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 30
32 IF Y=1 THEN Y=2
40 A=X
50 GOSUB 500
60 A=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 A=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
```

```
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 A=(9*X)/5+32
100 PRINT "   =";A;"F"
110 PRINT
120 GOTO 70
200 A=(5*(X-32))/9
210 PRINT "   =";A;"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
```

Software Horizons

This section will direct you to software available from Ohio Scientific as of November, 1978.

BASIC

There are three forms of BASIC available for use on the Challenger III. They are as follows:

OS-65D V3.0 9-Digit BASIC - Supplied standard with the system. 65D is used as the basis of most personal applications diskettes. It is useful for personal, educational and industrial applications.

OS-65U 9-Digit BASIC - Ohio Scientific's advanced virtual data memory system for business applications. 65U has extensions for multiple users, time share and distributed processing applications.

OS-CP/M BASIC (variable precision) - This Extended BASIC has several programmer enhancements such as "print using", renumbering and character editing. However, its disk file I/O is much less sophisticated than OS-65U. This BASIC should be used mainly for compatibility with 8080 (S-100) software.

BASIC Applications Software

Personal applications disks in the areas of Games, Education, Personal Finances and Small Business under 65D V3.0.

OS-AMCAP, a 65U based Small Business Accounting System including A/R, A/P, C/R, C/D, Inventory, General Ledger, Payroll, etc.

OS-DMS, 65U based Data Base Management System designed specifically for small business use. DMS based applications packages are available for A/R, A/P, General Ledger, Inventory, Payroll and Personnel.

Non-BASIC Oriented Software

WP-2 word processing system for general office use.

FORTRAN, COBOL are available as part of OS-CP/M. Other computer languages are planned.

Development Software

Assemblers

A 6502 assembler is available under 65D V3.0.

WP-1B contains a 6502 assembler, 6800 assembler, compatible editor and 6800 run time package.

OS-CP/M contains two 8080 assemblers, an editor and an extensive on line debugger.

Machine Code. The following section discusses machine code operation.

Multiple User Operation

Challenger III systems can be configured for a broad range time share, distributed processing and network configurations. As of November, 1978, all multiple user software options are extensions of OS-65U.

Every configuration requires at least one CA-10X (550 board) and most configurations require multiple memory partitions, and the real time clock option on the 470 floppy disk controller board.

Consult the multiple user manuals and the CA-10X manual for further details.

Machine Code

The 6502 can be directly accessed in machine code by typing "M" after reset. This executes the 65A machine code monitor ROM which is documented on the following pages.

Machine code programming and reference material for all three microprocessors of the Challenger III is covered in the book "How to Program Microcomputers", by William Borden which comes with the Challenger III system.

Machine Code on the 6800 and Z80

The multiple processor operations manual covers operation of the software switch and user programmable system vectors. Utility software is present in the form of a 6800 machine code monitor ROM (68A2) which is documented here and utilities disk.

The Challenger III utilities disk contains processor switching routines and monitor programs for the 6800 and Z-80. The utilities diskette is documented in the utilities manual, however, the diskette is based on 65D V2.0 and assumes some knowledge of this system. Since 2.0 is no longer supplied standard, it may be desirable to purchase a copy of this system and manual if extensive use of the utilities diskette is planned.

A quick guide to machine code operation of the 6800 and Z-80.

1. Place the utilities diskette in the A drive.
2. Reset the computer.
3. When the A* prompter appears, type "VU".

4. The "A*" prompter will then appear.

   A. To enter the 68A2 monitor under the 6800, type "R6".

   B. To enter the Z-80 monitor under the Z-80, type "RZ". (The Z-80 monitor is documented in the Challenger III utilities manual.

5. Simply reset the machine to get back to the 6502.

6. Refer to the manual for additional operating capabilities.

65A PROM MONITOR
INSTRUCTIONS

The 65A PROM Monitor is used with 6502 serial systems by the programmer who wishes to write at the machine language level. When the reset button is pressed, the letters D/M? or C/W/M? may appear on the screen. To get into the monitor, type an M on the keyboard (D is used only in conjunction with the diskette, which contains BASIC). While using the Monitor program, you can directly manipulate the computer's memory, and write programs using the computer's own language.

First of all, to examine memory locations before changing them, type a P, then the initial location in the block of addresses you wish to inspect. When you do this, the contents of that block will scroll up the screen. You may halt this scrolling by typing any key on the keyboard.

To change memory contents, type an R to return to the Command Mode. Then type an L, together with the location whose contents you wish to change, then an optional space for clarity, followed by the "new" contents which you select. If you are altering the contents of consecutive addresses, simply type the new contents one after the other. You may type spaces, carriage returns, and line feeds between these contents if you wish to make it more legible, but this is not necessary. In any case, the next successive address in memory is opened with each set of contents you type. If the next location you wish is not immediately consecutive, type R to get back into the Command Mode, then type L and the new address, plus the contents you wish to place there. Continue typing new contents if you are changing those of consecutive addresses, otherwise type R, then L, and so on.

To verify any changes you have made, use the P command to examine memory blocks as explained above.

While you are using the L command, the Monitor ignores all non-hexadecimal characters except R. When you use the P command, the monitor inserts spaces, carriage returns, line feeds and nulls.

The fourth command available when using the 65A Monitor is the G command which is used to run programs. This will be illustrated in the sample program below. Some of the following subroutines are used in the course of the program.

Subroutines

FE00	INCH (input character and echo)
FE0B	OUTCH (output character)
FE35	CONTROL (Note: FE40 will bypass ACIA initialization)
FE77	LOAD
FE8D	PRINT
FEC7	BUILD ADDRESS (constructs an address from input at 00FC [low] and 00FD [high])

Go and Breakpoint Locations

0129	Index Register Y
012A	Index Register X
012B	Accumulator
012C	Status Register
012D	Stack Pointer
012E	Program Counter High
012F	Program Counter Low

Vectors:

NMI	0130
RESET	FE35
IRQ	01C0

Sample program to illustrate OSI 65A Monitor

This program prints in double any character you type on the keyboard. Beginning at location 0200, the program would look as follows in user source code:

```
10*=$200
20 JSR INCH
30 JSR OUTCH
40 JMP $200
```

The assembled version of this short program would look as follows:

```
10 0200         *=$200
20 0200 2000FE  JSR INCH
30 0203 200BFE  JSR OUTCH
40 0206 4C0002  JMP $200
```

These lines are interpreted as follows:

Line 10: initialization of program counter

Line 20: actual program begins at given initialization point (0200); 20 is the ASCII code representation for JSR; 00 is the low address byte of INCH; FE is its high address byte.

Line 30: since three bytes have been used since program initialization, we are now at location 0203; 20 is ASCII for JSR; 0B is low address byte for OUTCH; FE is its high address byte.

Line 40: as this is the sixth byte since program initialization, we are at location 0206; 4C is the ASCII code for JMP; 00 is the low address byte for location 0200; and 02 is its high byte.

The bytes in this program are all to occupy consecutive memory locations. Therefore, only one L command will be necessary while we are in the Monitor, until we are ready to run the program. To enter it, engage in the following dialogue with the computer: press reset (your responses are underlined).

```
D/M? M
L02002000FE200BFE4C0002R
```

To verify that these contents truly are loaded into memory, type: P0200 The contents of all the addresses beginning with location 0200 will immediately scroll up the screen. To stop the scrolling, type any key and examine the contents displayed on the screen. Then type R to get back into Command Mode.

To run the program, you need to set the stack pointer (located at address 012D) to 28, and the program counter high (012E) at 02 and low (012F) at 00, because the starting address is 0200. Since these locations are consecutive, you need only type : L012D280200R

To execute the program, type G

Then any character you type will appear in duplicate on the screen.

68A2 MONITOR

The monitor has four commands:

(L) Load memory from keyboard or paper tape. The first four characters are the starting address and are given in hexadecimal. Each pair of succeeding hexadecimal characters is loaded into successive memory locations. The monitor ignores all non-hexadecimal characters except (R).

(P) Print or punch memory in hexadecimal starting with the location specified by the four hexadecimal numbers entered by keyboard. 8 bytes are listed per line and the monitor injects spaces, carriage returns, line feed, and nulls.

(R) Return to Command Mode from either (L) or (P).

(G) Go to the user program starting at the location specified by two locations, on page 1F.

Operation:

The L Command

The user can enter programs via the keyboard or teletype based paper tape reader with the L command. To manually load memory, simply type an "L", the starting address, and enter data.

For example, load location 0146 with 4E 6F 01.

Type:

```
L 0146 4E 6F 01 R
```

The "R" is used to return to the command mode. The load command will directly accept paper tapes generated by the P command. Simply type an "L", place the tape in the reader, and turn the unit on.

The P Command

The Punch/Print command allows the user to examine memory and punch tapes of programs. The user simply types a "P", then a four digit starting address. Then, the monitor will start listing memory in rows of 8 bytes. Typing any key on the keyboard will terminate this operation at the end of the existing line.

The G Command

68A, like the Motorola MIK Bug monitor, has a register stack starting at 1F29 as is given below:

1F29 - Condition Codes
1F2A - B Accumulator
1F2B - A Accumulator
1F2C - Index Register High
1F2D - Index Register Low
1F2E - Program Counter High
1F2F - Program Counter Low
1F30 - Stack Pointer High
1F31 - Stack Pointer Low

The contents of these locations are installed in the corresponding processor registers when the GO command is executed. When the processor encounters a software interrupt (3F), the processor registers are placed in these locations and the monitor is re-entered. 3F is a "breakpoint" command which allows the user to examine all processor conditions at any point of a program.

To execute a program, the user must load the starting location of the program into 1F2E (High) and 1F2F (Low). It may also be necessary to set other registers at this time, particularly the stack pointer. On 6800 systems, the stack pointer may be anywhere in memory so that if it is arbitrarily set, it may destroy important code. OSI recommends setting the stack to 1F80 for small programs. For example, execute a program at 0200 which does not initialize itself.

Type:

```
L1F2E 0200 1F 80 R G
```

Vectors:

Reset - FFA8
Software Interrupt - FFA8
IRQ - 1FD0
NMI - 1F30
INCH - FF00
OUTCH - FF88

Notes on MIK Bug Compatibility

MIK Bug compatible paper tapes are in a special block oriented checksum format. Each block is usually 16 bytes long. The block also contains a starting identifier (S1), a block length indicator, and a check sum. Each block is separated by a carriage return, line feed, and some nulls. The tape is terminated by a (S9). Thus, the format allows for alphabetic labeling of tapes before the first (S1) error checking via checksums and an indication of where one is in the load or dump process via absolute addresses.

The OSI MIK Bug compatible loader is part of the listing here from 1D00 to 1D62. The loader must be at these locations in conjunction with a 68A 8K mod or similar monitor. To use the loader, load 1F2E, 2F (program counter in 68A 8K) with 1D00 and 1F30, 31 with 1F80 (stack) then return and type a "G". Load a MIK Bug format tape into the reader and turn it on. If a check sum error occurs, the terminal will stop echoing the tape and will type a "?". If this occurs, stop the reader immediately and back up at least one complete data block which is distinguishable by a group of nulls between blocks and restart the program as above. Since each block of data has its own address, the tape will automatically load at the right place. The end of the listing is the OSI MIK Bug compatible tape puncher. This is located starting at 1D70 to 1DE3. The user must load the starting address of the dump into 1E02 and 1E03 and the last address of the dump into 1E04 and 1E05 (8K system). The program is started by loading 1F2E and 1F2F with 1D70 and 1F30 and 1F31 with 1F80 (or other stack pointer), returning to the monitor, turning the paper tape punch on, and typing a "G". The program will dump a full MIK Bug compatible tape with all the features of that format and will automatically stop when done.

Many small programs written on MIK Bug based systems rely heavily on subroutines in MIK Bug to save memory space. Fortunately, there are functionally equivalent subroutines for most of these in the OSI 68A monitor. One often used subroutine which 68A does not have is PDATA1. This is included in the MIK Bug loader/dumper from 1D63 to 1D6D.

The following table lists the commonly used MIK Bug subroutines and the OSI 68A locations of equivalent routines.

Table 1

Subroutine	MIK Bug Location	OSI 68A 8K Location
OUTCH or OUTEEE	E1D1	FF88
INCH or INEEE	E1AC	FF00
OUTHL	E067	FF7A
OUTHR	E06B	FF7E
OUTS	E0CC	FFA1
PDATA1	E07E	1D67*
CONTROL	E0E3	FFA8
INHEX	E0AA	FF12
BADDR	E047	FF37

*must be loaded in

Any use of the 128 word buffer memory at A0XX should be relocated to 1EXX.

To summarize the conversion process:

1. The user must change his 68A monitor PROM to a 68A 8K format or similar.

2. The user must locate RAM memory at 1EXX and 1FXX (for 8K systems).

3. The user must convert the subroutine calls of MIK Bug programs as per Table 1.

4. The user must convert program stack and temporary memory usage from A0XX to 1EXX.

5. The user must have resident in memory PDATA1 and the MIK Bug Loader.

An example of programs available in MIK Bug format are the excellent program offerings from Technical Systems Consultants (TSC), Box 2574, West Lafayette, Indiana 47906. Their programs are fully documented including assembler listings. TSC programs can be modified to run on OSI systems in a few minutes using the guide lines above.

```
ØSI CHECKSUM LØADER DUMP AND PDATA
FØR 8K 6800


P 1D00
BD FE 00 81 53 26 F9 BD
FE 00 81 39 27 29 81 31
26 EE 7F 1E 0A BD 1D 4D
80 02 B7 1E 0B BD 1D 3C
BD 1D 4D 7A 1E 0B 27 05
A7 00 08 20 F3 7C 1E 0A
27 CE 86 3F BD FE 88 7E
FE B2 02 02 BD 1D 4D B7
1E 0C BD 1D 4D B7 1E 0D
FE 1E 0C 39 02 BD FE 12
48 48 48 48 16 BD FE 12
84 0F 1B 16 FB 1E 0A F7
1E 0A 39 BD FE 88 08 A6
00 81 04 26 F6 39 71 FB
0D 0A 00 00 00 00 53 31
04 86 12 BD FE 88 FE 1E
02 FF 1E 0F B6 1E 05 B0
1E 10 F6 1E 04 F2 1E 0F
26 04 81 10 25 02 86 0F
8B 04 B7 1E 11 80 03 B7
1E 0E CE 1D 70 BD 1D 67
5F CE 1E 11 8D 25 CE 1E
0F 8D 20 8D 1E FE 1E 0F
8D 19 7A 1E 0E 26 F9 FF
1E 0F 53 37 30 8D 0C 33
FE 1E 0F 09 BC 1E 04 26
B3 20 0E EB 00 7E FE 95
13 0D 0A 14 00 00 00 2A
04 7E FE B2 21 64 40 41
00 51 C1 21 4C 8C 00 28
```

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. Fo.· 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 <u>outputing</u> data to the bus to <u>listening</u> to the bus.

B5 - D0
B6 - D1
B7 - D2
B8 - D3
B9 - D4
B10 - D5
B11 - D6
B12 - D7

Bi-directional eight-bit wide data bus for communication of data between the processor and system boards.

B13
B14
B15
B16

Upper data bits on some systems

B17 Optional reset line used to clear all PIAs and similar I/O circuitry in the system.

B18 spare line

B19
B20
B21
B22

Memory management address lines: Lines 21 and 22 are used on systems with a 500 CPU Board; all 4 are used with the 510.

B23 +12 Power connection

B24 -9 Power connection

B25
B26

+5 Power connection

B27
B28

Ground Connection

B29 - A6
B30 - A7
B31 - A5
B32 - A8
B33 - A9
B34 - A1
B35 - A2
B36 - A3
B37 - A4
B38 - A0

Ten low-order address lines

B39 - Ø2 Used to clock external circuits or external I/O interfaces, such as the A/D converter.

B40 - R/W (read/write) 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 - B15

The Challenger III system will physically accommodate any OSI 48 line BUS boards with the following restrictions:

CM-3 is the only recommended memory expansion option because CM-4 and other dynamics do not work with the Z-80. Also, C2-4P memory boards (CM-7, 8, 9) can overload the Challenger III power supply because of their high current draw on the +5 supply.

OS-CP/M and 65U Level 2 and 3 require memory at $D000 and $E000 so the Challenger III is not compatible with the 540 video board.

WARRANTY AND TROUBLESHOOTING HINTS

Ohio Scientific fully-assembled products are covered by a limited warranty. Challengers are covered for a period of sixty days against defects in materials and workmanship to the extent that any malfunction not caused by abuse, misuse, or mishandling will be repaired or corrected without charge to the owner provided that the unit is returned postpaid to Ohio Scientific within sixty days of day of receipt by the user.

Beyond this sixty-day period, up to one year from day of receipt by the user, the system is further warranted against defects in materials to the extent that Ohio Scientific will repair or replace them, charging only for labor on the portion of the electronic component that is manufactured by Ohio Scientific, without charge for the part(s). This warranty includes power supplies and floppy disk drives. It specifically excludes hard disks, terminals, video monitors, audio cassettes and some keyboards. Ohio Scientific's only obligation under these terms, in either case, is to repair the unit and return it once it has been delivered postpaid to Ohio Scientific. Typical turn-around time under this warranty is two to three weeks plus shipping time from the factory. Ohio Scientific cannot be held responsible for delays beyond its control such as those caused by shipping or long delivery of replacement components, e.g., floppy disk drives, etc.

Ohio Scientific reserves the ultimate authority to determine what constitutes in-warranty repair in circumstances where circuit modification, abuse, misuse, or shipping damage occur. The warranty is also subject to the use of proper packing material in any returns. This is the only warranty expressed or implied by Ohio Scientific and the only warranty which any Ohio Scientific agent is authorized by Ohio Scientific to give in conjunction with the product. Any maintenance or extended warranties that the end user may entertain with an Ohio Scientific representative or dealer are solely between that representative and the customer and are in no way authorized or supported beyond the extent of the above stated warranty by Ohio Scientific. The support of such warranty or maintenance contract is the sole responsibility of the agent offering the warranty.

Ohio Scientific software offers absolutely no warranty. The software is always thoroughly tested and thought to be reasonably bug-free when released. Ohio Scientific maintains a full staff of software experts and will endeavor to correct any serious bugs that may be discovered in the software after release in a reasonable amount of time. However, this is a statement of intent and not a guarantee in such matters.

TROUBLESHOOTING

If you encounter any difficulty in procedures in this manual, first refer to the following troubleshooting guides. If they do not provide sufficient help for you to solve your problems, proceed to the end of this section.

1) Order does not seem complete. First check to see that all packages specified have arrived. Carefully look over the packing lists, manuals, and this manual to determine what is supposed to be present in your system. If you have further doubts, check with the dealer or representative from whom you purchased your system, or the factory, <u>if and only if you ordered directly from the factory</u>.

2) Unit(s) mechanically damaged in shipment. Report damages or losses immediately to carrier. All units are shipped by Ohio Scientific fully insured. Under no circumstances should you ship the un't back in such condition as it would then be impossible to determine where the unit was damaged. This can cause a long drawn-out dispute with the carrier especially if the unit was transported by different carriers.

3) User has difficulty in following manual because of high level of technology involved. Suggestions: Obtain assistance from your local Ohio Scientific dealer or representative. If you ordered factory direct, or, are at a considerable distance from the dealer, contact your local hobby club and see if any members can assist you. Hobby club members are generally very willing to help out, which is a major reason they are in the club. Current club activities are listed in BYTE, Kilobaud, Interface-Age, and bi-weekly publications such as ON LINE. Any local computer store should be able to assist you in becoming a computer club member.

4) Reset light does not illuminate on power-up. Carefully check power connections. Check to see if unit is plugged in, that the power switch is on and that power is present at the power outlet. If so, turn the unit off and unplug it. Check the 2 amp fuse at the back of the unit and check the reset light itself by pulling the lens cap out and making sure that the lamp is properly seated in its socket.

5) Reset switch is dimly lit or not lit at all after you have checked with the above procedures. Carefully inspect the PC board portion of the computer for foreign matter such as a wire cutting or something leading out from the PC board. Also check to see that all PC boards are properly seated, and that any ribbon cables are properly seated in their sockets. If the unit light is only dimly lit, remove about half of the PC boards. If the light comes up to full brightness with these out, put those boards back in and pull the other ones out. If the same condition occurs, it means that there is a power supply malfunction and that the unit will have to be returned for repair. If the power supply folds back when some PC boards are out, and not with others, you should be able to isolate the board causing the foldback. That board most likely has foreign matter across it, causing the short on the board.

6) Power supplies look fine, but computer doesn't seem to reset at all or properly. Symptoms: nothing comes out on serial terminal or screen doesn't clear on video system. Solution: again, give the system a careful visual inspection. At this point, it would be invaluable to have access to another Ohio Scientific computer system by way of a dealer or another computerist. If neither is available, and you do not wish to or are not able to attack the actual circuitry of the system, it will most likely be necessary to return the unit for repair.

7) Computer appears to reset properly, but there is no response to the keyboard. On serial systems, carefully check your interface. Make sure the terminal is set up for full duplex and no parity. On video systems, carefully check your wiring. Make sure the keyboard strobe is properly connected. It is possible that the keyboard strobe pulse is too short for the computer's polling routine to see. The keyboard's strobe should be at least 5 ms long for proper operation.

8) Last key strobe on video systems is not acknowledged until next key is depressed. Solution: Keyboard strobe is set wrong. Change polarity via the jumper on the 440 Board.

9) Keyboard operation erratic; some keys are missed. Solution: Keyboard strobe is too short. Strobe pulse must be lengthened, usually be adding capacitance to the strobe circuit on the keyboard. Strobe pulse should be on the order of 5 to 10 ms for proper operation.

10) System works fine in machine code, but in BASIC you consistently receive SN error message (Syntax error). Carefully refer to the example given in the BASIC User's Manual.

11) Floppy disk does not appear to work properly. Consult OS-65D Version 2.0 Manual for any problems concerning the Floppy Disk Drive.

12) Audio Cassette Interface does not appear to work. Carefully look at your connections between audio cassette recorder and computer making sure that you are connecting the microphone jack to the cassette interface output and the speaker jack to the cassette interface input. Be sure that you are using shielded cables and that there are no shorts or opens in your cables. Volume controls should be mid-range, tone controls should be mid-range. Be sure you are using high-quality tape and that you are running the cassette on fresh batteries or 110V AC.

13) If you experience errors when playing back cassettes, try adjusting your own control upward or downward from midrange when recording and playing back cassettes. You may also need to adjust your tone controls. The audio cassette is generally extremely reliable with most medium-quality tape recorders. If your system is subject to a noticeable or an appreciable error rate, it is quite likely that your tape recorder is not suitable for use as a computer storage device. We highly recommend the Panasonic RQ309 tape recorder for such applications.

In Case of Difficulty

If you encounter a problem with your system, first carefully look over the trouble-shooting hints in your procedures. The great majority of problems encountered on new computers result simply from the user's unfamiliarity with the computer system. If you decide that you cannot resolve the problem yourself, contact the representative or dealer from whom you purchased the computer. If you purchased it directly from the factory, contact the factory at 216-562-3101. This number is for customer service. Your local OSI dealer representative should be able to help you by providing guidance on operating procedures, and in the case of an actual computer malfunction, should be able to substitute PC boards and subassemblies to isolate the problem. He should then also provide the service of getting the replacement or repair for the malfunctioning unit. If you ordered the computer factory-direct, then you must call Ohio Scientific to obtain a return authorization number before returning any unit for in-warranty repair. When you call, please immediately state that you currently own Ohio Scientific equipment and are experiencing difficulty with it. By stating this, you will quickly be routed to the service department. Please have your situation well organized so that you can present it to the service department personnel. They may then be able to help you immediately, or may need to consult with engineers and production staff to see if there is a simple solution to your problem, which will require another telephone call. If it is then decided that there is a hardware problem which must be repaired, you will then be offered a return authorization no., so that you can send the faulty equipment back for repair.

All equipment returned for repair must have a return authorization no., and must be packaged in the same containers or equivalent containers upon return. That is, all Challenger products are to be shipped in double containers with at least one inch of standard packing material lining all six sides of the inner container to preclude the possibility of puncture and damage. All packages must be stuffed with packing material so that the computer cannot vibrate or move around in the box; that is, it should pass a manual shake test, where there is no shift of load when you hold the box and shake it. Computers not shipped in double-wall containers or double boxes will not be accepted for in-warranty repair by Ohio Scientific, since it is our experience that such units are often damaged in transit. The preferred method of shipping is by UPS insured. Please enclosed a note clearly stating thc problems you have encountered with the unit when you return it for repair, and be sure to include the return shipping address on that note.