

THE OZROM 1E

A REPLACEMENT ROM FOR THE OSBORNE 1 COMPUTER

Copyright (C) 1984 Micro Management, Inc.

GUARANTEE: If you are not completely satisfied with the OZROM 1E, return it and the accompanying documentation in reasonable condition within thirty (30) days of purchase, and we will refund the purchase price of this product to you. Proof of purchase required for refund.

WARRANTY: Should the product fail to function due to electrical or mechanical failure during the first 90 days after purchase, Micro Management, Inc., will replace it free (and pay the return shipping charges, if necessary), provided the defective product is returned with proof of purchase within 100 days of the purchase date. Should the product fail to function due to electrical or mechanical failure after the warranty period, return it to Micro Management, Inc., with \$25.00 and we will replace it. Changes in the OZROM 1E or documentation may be made without notice.

LIMITATION OF LIABILITY: This product is intended for use in accordance with the attached documentation only. This product is not intended for any other use. It is the purchaser's responsibility to determine the suitability for a particular application.

Micro Management, Inc., shall not be liable for loss or damage caused, or alleged to be caused, directly or indirectly by the OZROM 1E or any other computer equipment or programs sold by Micro Management, Inc., including, but not limited to, any interruption of service, loss of business, or anticipatory profits of such computer equipment or computer programs.

The preceding statements of warranty and limitation of liability are in lieu of all other warranties, guarantees, or theories of liability, expressed or implied, including warranties of merchantability or fitness for a particular purpose.

COPYRIGHT: This manual and accompanying programs, including the OZROM 1E, are copyrighted. Purchaser receives a non-transferable, non-exclusive, personal license to use the copyrighted materials for their intended purposes. The materials may not be copied or sublicensed without prior written consent from Micro Management, Inc., with the exception that the purchaser may make a copy of the manual and software for use on his own system.

Trademarks and Tradenames: Osborne 1: Osborne Computer Corporation. WordStar: MicroPro International Corporation. DBase II: Ashton-Tate. SuperCalc: Sorcim. CP/M: Digital Research, Inc. Apple: Apple Computer, Inc. SMART-KEY: Heritage Software, Inc. Manual Rev. A 84/09/01

P.O. Box 165383

Irving, TX 75016

214/256-1960

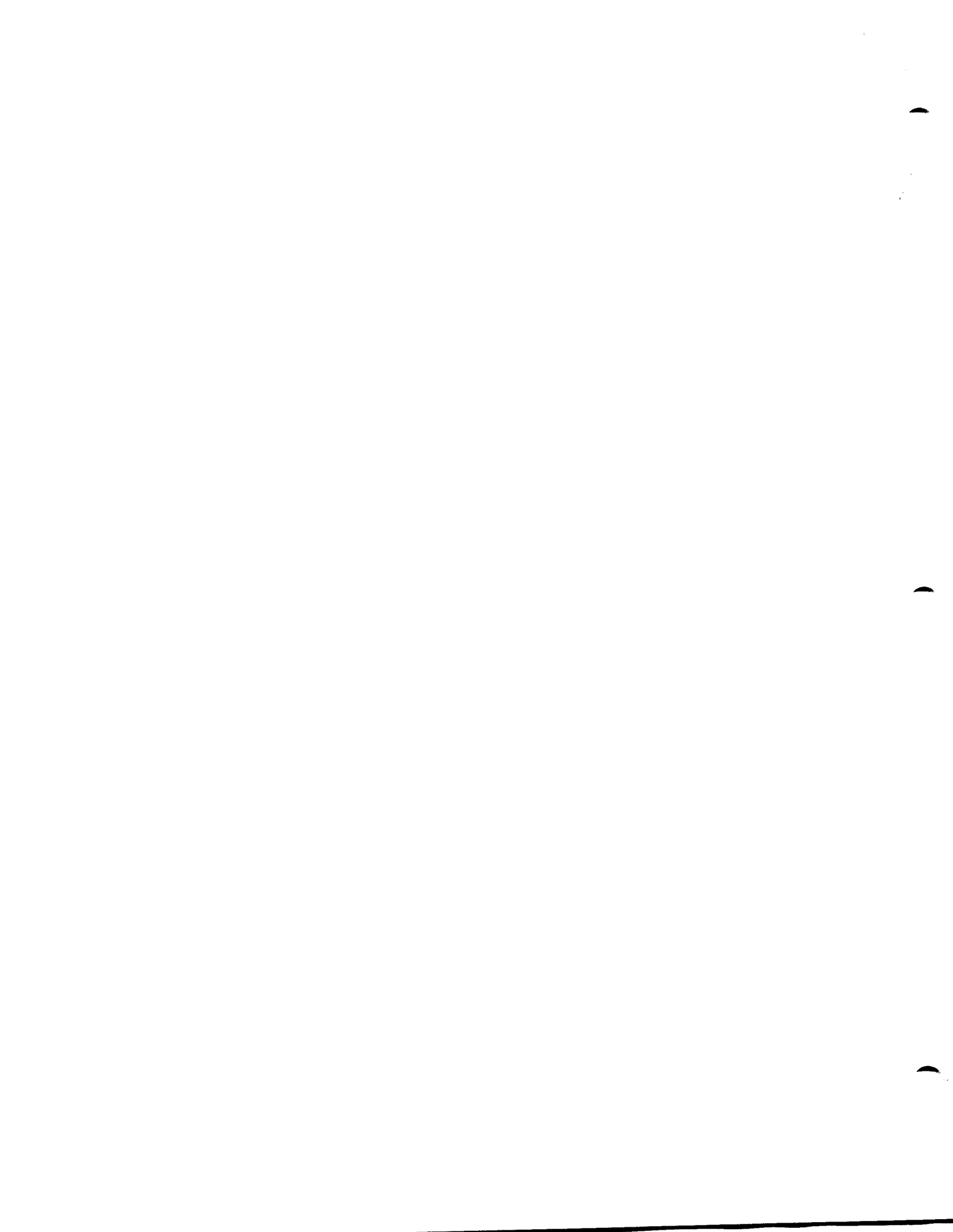
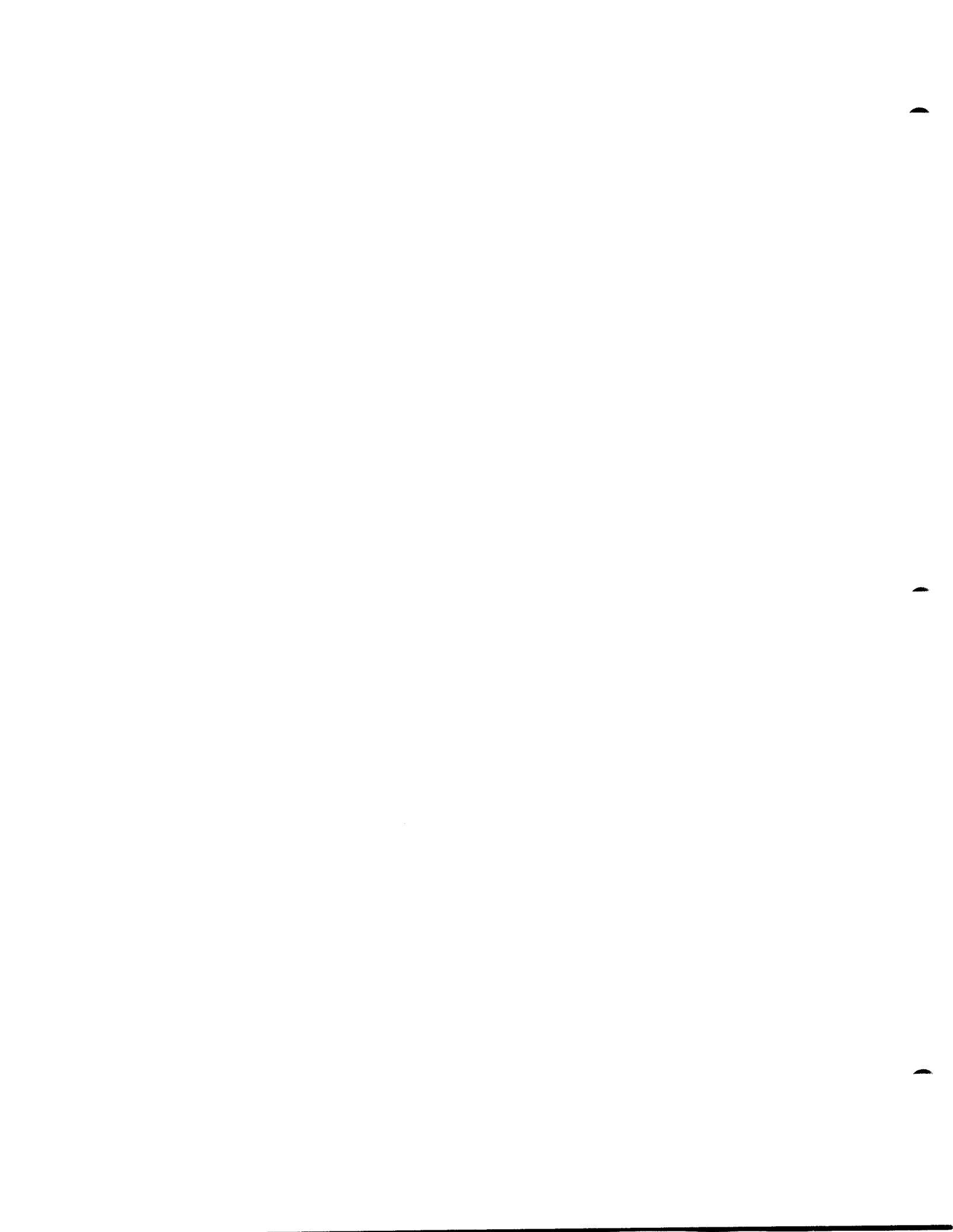


TABLE OF CONTENTS

What you gain with the OZROM 1E	1
What you lose with the OZROM 1E	1
BEFORE YOU BEGIN	2
Things to remember	2
FEATURES	3
A blinking cursor	3
A more complete ASCII keyboard	3
MyKey -- A redefinable keyboard	4
MyKey II -- New function keys	7
Immediate control over system parameters	8
No more parallel port paralysis	8
Print your screen	8
Control your printer	9
Time-of-day clock	9
True clear-screen	9
System crash damage protection	9
A system memory tester	10
1200-baud modem operation considerations	11
Additional display control codes	11
"Booting" with the left/right arrow keys	12
SUPPORT PROGRAMS	13
MEMORIZE	13
Autostarting with MEMORIZE	14
TIME	14
ALARM and TIMER	15
QUIKTIME , CRASH , and SHOW	16
The OZPATCH system patcher	17
The FK function key programmer	19
The SET utility	27
 APPENDICES	
Program patches	A-1
A smoother, faster-scrolling WordStar	A-1
A fix for dBase II function key operation ...	A-3
Installing your OZROM 1E	B-1
Technical information on the OZROM 1E	C-1
Differences between the OZROM 1E and others .	C-1
Truer Televideo 912/920 operation	C-2
An interrupt hook for programmers	C-2
1793/1797 floppy disk controller operation ..	C-3
Control-shift-key notes	C-3
Damage protection, NZCPR, and CRC checking ..	C-4
The OZROM 1E memory map	C-5
The ROM jump table	C-6
The OZROM 1E keytable	C-7
ROM error codes	D-1
ASCII table	E-1
A quickie control-shift-key reference table	E-2
 INDEX	 The Last Page



WHAT YOU GAIN WITH THE OZROM 1E:

User-selectable blinking cursor, keyclick and display of system clock.

A software-redefinable keyboard -- individual keys can be redefined almost anytime, even right in the middle of most programs -- in any program that requires user input.

Eight locally-redefinable functions, similar to the regular function keys, which can be assigned to any regular key, or shift- or control-key combination. Each can be up to 63 characters long.

Local, immediate control over system parameters through the use of control-shift-keys: you can toggle scrolling, clock display, cursor blink, and keyclick; you can select between 52-, 80-, or 104-column display (with the Osborne Screen-Pac); you can free your computer from the "Centronics printer not connected" lockout (double-density operating systems only); you can toggle easily between the left 52 columns and the right 52 columns of an 80-column display if you don't have an 80-column upgrade.

Memory tester, to verify the integrity of your computer's random-access memory. If you have a bad main memory chip, this routine will even show you which chip on the board to replace.

Optional drive step-rate speed-up booting.

True, local clear-screen, including attributes (half-intensity, underline, and graphics).

No more phantom lines in WordStar due to a bug in previous ROMs. Truer Televideo 912/920 emulation.

More keycodes available, including 1Dh and 1Eh (^] and ^^, respectively), from the keyboard.

A shorter, saner beep length (about 1/2 as long as the original).

A 50 per cent faster key repeat.

An operating system patcher program, which adds the following features: a resident print-screen function which does not affect your screen or program (you choose from the keyboard whether you want the "screen dump" to be 80 columns wide or 128); and the ability to send any non-zero ASCII character to your printer -- again, in the middle of your program! ("Resident" means that once patched, always available.)

A handful of feature programs, like one to save your new Dvorak (or whatever) keyboard definition, including any special new functions you may have defined "locally"; or one to set the time-of-day clock you now have, as well as other system parameters, like baud rate and system printer.

WHAT YOU LOSE WITH THE OZROM 1E:

Although Centronics parallel printer operation is not affected (which means printers that use the IEEE-488 port as a parallel port will work fine), you can no longer use the port as an IEEE-488 port without additional software.

BEFORE YOU BEGIN

If you are installing the OZROM 1E:

For the sake of your safety, BE SURE that you have disconnected all AC power from your computer before you install the integrated circuit that we have provided. (Pulling the power cord plug out of its socket is sufficient.) Refer to APPENDIX B for installation instructions.

If your OZROM 1E is already installed:

For the sake of your sanity, we recommend reading the rest of this documentation prior to using your Osborne 1 with the new ROM, and refer to the appropriate sections frequently as you get familiar with each new feature you try. Also, you should make a backup copy of the supplied diskette before you do anything else.

THINGS TO REMEMBER

Control-Shift Keys

In the following sections, we will be referring to "ctrl-shift" keys. To enter the corresponding "ctrl-shift-key", hold down the control and shift keys, and press the key requested. For example, to enter a ctrl-shift-A, hit the A key while holding down both the control and shift keys. Then release the keys.

Toggles

A toggle is a switch function, like a push-button switch. Every time you use it, you reverse the condition associated with the switch. An example of a toggle is the control-V (Insert ON/OFF) toggle of WordStar.

No More Vertical Scrolling

Vertical scrolling with the control-up arrow and control-down arrow keys has been eliminated. These control-arrow keys now have different functions: control-up-arrow repositions the window so that the display starts at column 29, and control-down-arrow positions the window to start at column 76.

This allows those of you who do not have an 80-column option to quickly "toggle" the screen to see the right-hand two-thirds of an 80-column screen (the right hand column is column 80) with the control-up-arrow key combination. Since this is right next to the left-square-bracket key, you can hold down the control key with one hand, and with the other quickly switch back and forth between the first 52 columns and the last 52 columns of an 80-column screen. (The left-square-bracket key in combination with the control key "homes" the screen window.)

The control-down-arrow combination allows you to view the remaining portion of the screen.

FEATURES

Although you can use your new OZROM 1E in almost the same way as previous OI ROMs, there are quite a few new features now available to you.

A BLINKING CURSOR --

Once installed, the OZROM 1E provides you with a very nice aid in the use of your computer -- a blinking cursor. When you turn on your computer, the cursor will not be blinking. You can either toggle the blinking cursor feature on or off as described below, or, if you like the feature enough to desire it as a default condition, you can say so when you run the OZPATCH utility, which serves as a SETUP program for your new features. You can even change the blink rate for a faster or slower blink if you wish.

This is not a "hardware" cursor; it will not blink all the time. It will blink any time a program asks for a character, which, you will soon find, is a good indication when your program is ready to take more input. (Some programs will not ask for a character until one is ready. In this instance, the cursor will not blink. Examples include DBASE II version 2.41, a couple of the supplied OI utilities, like SETUP and COPY, and those programs which supply their own cursor, like SuperCalc and VDO.)

To turn the blink feature on or off, just type ctrl-shift-B. You will hear a "bip" to tell you that something has happened, and you should be able to see on the screen the effect of the toggle.

Note: Some programs, like WordStar, have public-domain patches available to blink the cursor. This blinking is done in the program itself. We recommend that you try out a copy of your original (without the patch), and use this version if the cursor will blink with the OZROM 1E. (Now, the blinking is being done where it should be -- in the console input routine in the ROM -- so you will not lose your cursor on the screen due to some of those patches.) If you desire that "smoother, faster-scrolling" version of WordStar, our version is given in one of the appendices.

Remember, ctrl-shift-B is a toggle, reversing the function of the blink every time you use it. Should you encounter an application where a blinking cursor is not desired, you can easily turn it off this way.

A MORE COMPLETE ASCII KEYBOARD --

The OZROM 1E provides more ASCII keycodes than previous ROMs. You can now generate 1Dh (^_) and 1Eh (^ ^) directly from the keyboard. The first code is used to select compressed print in some dot-matrix printers, and the second code provides the Home Cursor function of a Televideo 920 terminal, which the Osborne 1 emulates.

To get a ^_, just type ctrl-shift-]. To get ^ ^, type (you got it) ctrl-shift-^. (The "^^" is on top of the 6.)

MyKey -- A REDEFINABLE KEYBOARD

With the OZROM 1E, the characters you get when you type on your keyboard are no longer cast in concrete. Each key can be redefined to give you any character you wish. (There is also the capability of assigning a string of characters to a key; this is discussed in the next section.)

Four Keyboards

Think of your keyboard as actually being four keyboards: one "normal" keyboard, one "shift" keyboard, one "control" keyboard and one "ctrl-shift" keyboard.

The "normal" keyboard returns the character printed on the key, or the bottom character if there are two. You'll note that on this "normal" keyboard, the control and shift keys do not produce any results when pressed by themselves.

The "shift" keyboard gives you upper-case letters, or the top characters from the two-character keys, when you press the shift key in combination with the corresponding key.

The "control" keyboard returns the "control character" that corresponds to the key, or a few remaining ASCII characters that are not displayed on your keyboard, like {, }, ' and ~, when you press the control ("ctrl") key in combination with another key. You probably have used the "control" keyboard extensively in WordStar.

Until now, the "control-shift" keyboard on your computer was limited to one key -- the ctrl-shift-/ key, which gave you ASCII 7Fh, the DELETE character. If you knew about this key, you are a rare person, and if you used it, you are a very rare person indeed. (It can be used while editing in WordStar to delete the last character -- but you probably used ^- instead.)

Well, you now have two additional ctrl-shift keys, ^^ and ^| as discussed earlier. There are many more ctrl-shift keys that are used for functions, like the ctrl-shift-B described earlier to toggle the blinking cursor on and off.

Key Redefinition

To redefine a key, type ctrl-shift-X. You will hear a short "bip". If your cursor was blinking, it will stop. If you did not have the keyclick enabled, you will now hear your next two keyboard entries "click". The first should be the key you wish to redefine. The second should be the key definition that you want. (The ORIGINAL key definition will always apply here.)

For example, you've always wanted a double-quote on a key that you did not have to shift while you were using SuperCalc. You probably have not used the backslash key at all in Supercalc. Here, you can make the backslash key a double-quote key as follows: Just type ctrl-shift-X, followed by the backslash (\) key, followed by the double-quote (") key (the shift of the single-quote (') key). Your

cursor will start blinking again if you had that option selected. Now, hit the backslash key. You'll see that it gives you double-quotes!

If you need the backslash key, you can define the original double-quotes to be a backslash the same way. If you wish to redefine a key back to its original definition, just type ctrl-shift-X, followed by the key TWICE.

Three Keyboards are Definable

You can define the entire normal and shift-keyboards, and almost all the control-keyboard, in this fashion. You can assign your control-number function keys to keys on the "normal" keyboard to get true single-key function keys. You can even assign the control-shift functions to regular keys (like making the backslash key into a ctrl-shift-X key). AND, after you have come up with your favorite keyboard, you can use the MEMORIZE program that we've provided to save the keyboard definition in a file. MEMORIZE creates a program file on diskette that you can execute by itself to restore your keyboard definition.

A word of caution in redefining letter keys: The second key entered, if it is a letter key, is affected by the caps-lock key. If you define a Dvorak keyboard, for example, all your letter keys will be in upper case if you leave the caps-lock key on (i.e., down). If you find that a key you defined is in upper case when you wanted lower, just repeat the ctrl-shift-X sequence after making sure the caps lock key is up.

Incidentally, the caps-lock function is separate from the shift function. This means that when the caps-lock key is down, the capital letters you get when you press the shift key are different than the capital letters you get from the "normal" keyboard (which are due to the caps-lock key being down). This gives you extra keys if you wish, or you can even have a keyboard that gives upper-case on the "normal" keyboard and lower-case letters when you use the shift key!

MyKey does not have the problem that SMARTKEY or the public-domain QWIKKEY cannot solve -- namely, the fact that the keyboard is tied to the ROM. For example, the Dvorak keyboard requires the letter S to be located where the QWERTY keyboard has the semi-colon. Then, when the caps lock is down, you would still get a lower-case "s". With MyKey, however, the caps-lock function will work properly on any non-control/non-shift key.

The only keys you cannot redefine from the keyboard are the control-shift keys, and the following control keys: all ctrl-arrow keys, and ctrl-I, which are used for screen positioning. (Actually, the screen-positioning keys are programmable, but not from the keyboard.)

This means you can also assign keys (and functions; see the MyKey II section following) to the following keys: ctrl-esc, ctrl-tab, ctrl-return, ctrl-space, shift-esc, shift-tab, shift-return, shift-space, and shift-any-arrow-key, without affecting the normal key definition. Ctrl-

semicolon and ctrl-apostrophe can also be reprogrammed in this manner.

You can use these extra keys for the MyKey II functions (they're coming, they're coming!) or for those keys that get "displaced". For example, your apostrophe key might be programmed to be double-quotes (") for SuperCalc. Out of deference to your habit of occasionally using the original double-quotes key, you do not wish to redefine it to be the apostrophe key, but you use the apostrophe enough to warrant putting it someplace. You can assign it to ctrl-tab, for instance, and always have it available without having to restore the original definition. (The default keyboard already has ctrl-apostrophe and ctrl-semicolon assigned to apostrophe and semicolon for just this purpose.)

Although your key definitions will survive warm boots (that is, if you type a ctrl-C at the CP/M prompt, you won't lose your redefined keyboard), pressing RESET will always restore the original key definitions. Remember, use MEMORIZE to save those favorite definitions in a file.

Some Weird Programs...

One more note -- If this or any other special function should not work or should cease working, one of two things has happened:

1. You are using a program that does not use the normal BIOS method of console input. In this instance, the program will return strange characters instead of executing the desired functions. If you have to use this type of program, just don't use ctrl-shift-X or ctrl-shift-R (discussed in the next section). You can, however, redefine your keyboard OUTSIDE of this type of program and save the key redefinitions with MEMORIZE. Then, when you load them back in later, the new definitions will be available to you even though you can't change them in the middle of the program. This only applies to ctrl-shift-X redefinitions.

2. You are using a program that has "walked" on the special reserved key-definition space, up in video RAM. The ROM has a special routine to detect this change and restore control to the keyboard definitions in ROM. In this instance, you can go through the motions of redefining a key, but when you attempt to use the redefined key, you will find that the key has its original definition. (The OZROM 1E special features will work with ALL standard CP/M programs, and with 98 per cent of programs that are specially written to work with the Osborne 1. The remaining 2 per cent don't use the standard video functions of the Osborne 1; they wrote their own routines which may walk on the reserved memory space of the OZROM 1E. Please let us know if you come across one of these programs. In most cases the program can be easily modified to stay within the required limits.)

MyKey II -- NEW FUNCTION KEYS

In addition to redefining a key to give you a different character, you can redefine a key to give you a string of up to 63 characters, and you can have up to 8 such "function" keys at a time (in addition to the regular function keys).

Ctrl-shift-R starts the MyKey II function key definition. When you press **ctrl-shift-R**, you will see the bottom line of your Osborne 1 display disappear, to be replaced by the word **KEY:** followed by a block cursor. This is a prompt asking you for which key you wish to define. When you type a key, the prompt will disappear, and the block cursor will move to the beginning of the line.

Now you can enter your function definition, up to 63 characters in length. The left-arrow key is used in this mode to backspace and delete the previously-entered character. To exit this mode (which will always be indicated by the block cursor), just type another **ctrl-shift-R**. The bottom line of your OI display will be replaced, and your key has now been redefined to be a function key, just like your regular **ctrl-number** function keys.

If you do not enter anything, or if you have backspaced to the beginning of the line, typing **ctrl-shift-R** will exit without redefining the key.

Here's an example: In generating this documentation, the writer found it convenient to redefine a key to produce the phrase "**ctrl-shift-**". Since the underline key was not often used, it was selected as the "function" key. The key sequence was:

ctrl-shift-R, shift-underline, c,t,r,l,-,s,h,i,f,t,-, ctrl-shift-R.

Now, when the underline key is struck, it produces the phrase "**ctrl-shift-**".

Eight's the Limit At Any One Time

You can have up to eight function definitions at one time. If you redefine a ninth key as a function (as described above), then the first key that you had defined in this way will now have the same definition as the ninth key. To correct this, or to assign a function key back to its original character value, use **ctrl-shift-X**. See the technical notes appendix for an explanation of how these function keys are generated and interpreted.

These functions are also saved by the **MEMORIZE** program when you save your keyboard definitions. You may want to assign **ctrl-shift-R** to a single key, like the backslash, to make it easy to program a quickie function in the middle of your application.

It is recommended that you do not redefine the left-arrow key (the backspace key), since only the original left-arrow definition will be accepted for backspace editing while using MyKey II.

IMMEDIATE CONTROL OVER SYSTEM PARAMETERS

With the OZROM 1E, you also have immediate control over some of your system parameters, like automatic horizontal scrolling, cursor blink, time-of-day clock display, key-click, and 52-, 80-, or 104-column mode (with Screen-Pac).

As mentioned earlier, **ctrl-shift-B** toggles cursor blink on and off. In like manner:

Ctrl-shift-S toggles the automatic horizontal scrolling feature of the Osborne 1.

Ctrl-shift-T toggles the display of the new time-of-day clock in the OZROM 1E. (Turning off the display does not erase what was previously put on the screen.)

Ctrl-shift-K toggles the keyclick feature of the OZROM 1E on and off.

If you have the Osborne Screen-Pac 52-80-104 column option, then the following is available to you:

Ctrl-shift-5 selects the 52-column mode.

Ctrl-shift-8 selects the 80-column mode.

Ctrl-shift-1 or **ctrl-shift-0** select the 104-column mode.

NO MORE PARALLEL PORT PARALYSIS

For those of you with printers that use the parallel port of your O1: Ever lock up your computer because your printer was not plugged in and you tried to print something? Well, if you should ever lock up your system again for this reason, remember the following -- **ctrl-shift-escape** will assign your parallel printer to a "null" device (that is, any characters you try to print will be thrown away and not printed). This will allow you to recover control of your computer. If you are using WordStar, for example, after you type **ctrl-shift-escape**, you can then type a P to stop the printing. **Note:** If you have assigned your printer to a "null" device with **ctrl-shift-escape**, you can restore your parallel printer to normal operation by typing **ctrl-shift-7**.

PRINT YOUR SCREEN (with OZPATCH only)

If you have patched your system with the supplied OZPATCH program, you also have the ability to print your screen and send any character to your printer, right in the middle of your program. **Ctrl-shift-P** will send a non-graphics image of your screen's first 80 columns to your printer. **Ctrl-shift-W** will send the entire 24 lines by 128 columns of your screen to your printer. If your cursor was blinking when you selected either of these functions, it will stop, until the computer is ready to accept more input. Graphics characters will be converted into asterisks (*).

Output will be to the device you have selected to be the printer in the SETUP program. (You can tell your computer to print to a different port -- serial, parallel, or even your Osborne screen can be the "printer" -- by changing the IOBYTE with DDT or the SET program provided.)

CONTROL YOUR PRINTER FROM YOUR KEYBOARD (with OZPATCH only)

Ctrl-shift-O tells your computer that the next character you type is to be sent to your printer. For example, if you have a printer that requires a control-O for compressed print and you want compressed print for your printout, you can type ctrl-shift-O, control-O to put your printer in the compressed mode. If you want to send a formfeed to your printer, just type ctrl-shift-O followed by control-L, which is the formfeed character for most printers. See the OZPATCH documentation for more information.

TIME-OF-DAY CLOCK

You now have a time-of-day clock available to you. It can be displayed anywhere on the first two lines of your display (see the Technical Notes in the appendix). You can set the time with the TIME program provided. The clock is very accurate as long as no disk access takes place. Some "fudging" has been added to make the clock reasonably accurate even with floppy disk access.

The Technical Notes section of the appendix gives the info you would need if you wished to make use of the clock in your programs.

TRUE CLEAR-SCREEN

You also have the ability to clear your screen any time your program is expecting keyboard input. Type **ctrl-shift-Z** to zap your screen completely. Use caution with this one, though; if you do it in the middle of your WordStar or SuperCalc program, it will be a while before your screen "returns to normal." Ctrl-shift-Z will prove most useful for clearing the screen of garbage from listing WordStar files, etc., or preparing the screen for a screen-print. Another use is recovering from the graphics mode, or half-intensity or underline modes, should you accidentally get stuck in one of these modes.

Now, any time your program clears your screen by sending a control-Z to your console, the half-intensity and underline attributes will be reset, also. Should you need to clear the graphics attribute as well, send an escape followed by a control-Z; otherwise, the control-Z will be interpreted as a graphics character.

SYSTEM CRASH DAMAGE PROTECTION

There is a program called CRASH on the diskette we've provided which demonstrates what will happen almost every time any of your programs should do something wrong. With the previous ROMs, your same program under the same conditions would have turned on both drives and the bell, which ran a very good chance of ruining any diskette in the drive. The OZROM 1E reduces the chance of diskette damage from about one in two to about one in one thousand.

A SYSTEM MEMORY TESTER --

The OZROM 1E has a memory tester built in to test your system RAM. You can execute the memory test program at power-up or after you push the RESET button. Just press the TAB key to start the tester going. The display will show:

RAM TEST

XXXX

where XXXX will be a rapidly changing address display, indicating in hexadecimal which RAM location is being tested. The test is cyclic, starting each time at memory location 0 and testing (0, 0FFh, 0AAh, and 55h) thru 0EFFFh. Video memory is not tested; a problem with video memory will not only be visible on the display, it will be generally trapped as an error starting around 0C000h (video RAM is only 1/4 of one of the 16K banks of RAM).

Should a failure be detected, the address display will freeze and two sets of binary numbers will be displayed:

XXXX bbbbbbbb
 bbbbbbbb

This information can be used to determine which IC on the main board (Rev. E or later) has failed. Here's how to use the information:

XXXX bbbbbbbb <--- what the memory tester wrote
 bbbbbbbb <--- what the memory tester read

 01234567 <--- the Y value, used below

First, determine which 16K bank of memory (A, B, C, or D) has the bad chip:

if the value of XXXX is:	then the bank is:
0000-3FFF	A
4000-7FFF	B
8000-BFFF	C
C000-EFFF	D

Then look at the binary display. There will be a difference between what the tester program wrote and what it read. That difference will generally be in one column only. Here the columns are numbered from left to right, zero thru seven. If the difference is in the fourth column, then the corresponding number is 3.

The bad chip will then be at X2Y, where X is the bank letter and Y is the number obtained from the binary display. For example, say the display has stopped changing and shows:

402F 00000000
 00000100

The 402F indicates bank B and the 1 in the sixth column says that the Y value is 5. Then we should replace the chip at location B25 on the main circuit board.

Note: The memory test is continuous and will not stop unless you either press RESET or turn the computer off. Some memory chip failures only occur after the chip gets warm, so it may be a good idea to test your memory AFTER the machine has been on a while. Run the memory tester any time your computer does not seem to be acting properly. If your computer passes the test, at least you have eliminated the system RAM as part of the problem. (Memory failure can actually result in Bdos errors, which might lead you to believe the disk drive had a problem.)

1200-BAUD MODEM OPERATION CONSIDERATIONS

The new video software routines on your OZROM 1E do not make use of the entire screen of video RAM in order to give you the some of the features previously mentioned, like MyKey and MyKey II. As a consequence, the video screen cannot be moved down like a window placed over the video RAM, as discussed in your O1 manual. Instead, when the cursor is at the bottom of the screen and your program sends a linefeed character "to the screen", the video subroutines move the bottom 23 lines of your display up one line, first the regular video stuff, then the half-intensity stuff. This is slower than the original ROMs, but most of your software is memory-mapped or uses cursor addressing, so you will generally not notice.

However, although 300-baud operation is not affected, 1200-baud operation WHERE THE VIDEO SCREEN IS CONCERNED may be unreliable. (Object file transfer with XMODEM or programs like MOVE-IT will not be affected.)

Viewing text on the screen at 1200 baud will result in characters near the beginning of the line being dropped due to this delay, UNLESS YOUR HOST COMPUTER CAN SUPPLY NULLS appropriately at the end of each line. When your host computer says "How many nulls?", reply with 4. That is, tell the host computer you need four nulls to operate correctly.

Again, 300-baud operation is not affected.

ADDITIONAL DISPLAY CONTROL CODES

You might have purchased a program that you had to "install" on your O1 through a lengthy procedure where you told the program that your computer used ESC) to turn on half-intensity, etc. If you ever have to do it again, the following additional display control codes may prove useful:

ESC j turns on BOTH underlining AND half-intensity
 ESC k turns both underlining and half-intensity off
 ESC Y clears from the cursor position to end-of-screen

"BOOTING" USING THE LEFT/RIGHT ARROW KEYS

When you boot your computer with the <RETURN> key (or the <"> key for the right-hand drive), the OZROM 1E acts similar to the previous O1 ROMs. If the drive has trouble reading the diskette, you will see the message "OH, NO!" appear on your screen (instead of "BOOT ERROR") and the disk drive head will be moved to track 5 and then back again for another attempt. However, there is another way to boot your diskettes, using the horizontal arrow keys.

When you first boot using the left or right arrow, your disk drives are assigned a step rate delay of 12 ms. (The original step rate delay is 20 ms.) This causes your disk drives to step from track to track a little faster than before. (You may have used a FAST program to "speed up" your drives in this manner.) Some Osbornes, however, cannot stand the speed-up. The OZROM 1E attempts to read the system tracks of a diskette with this speed-up, but if the first attempt fails, then the step rate is set to 20 ms, the disk drive head is moved out to track 5, and then "homed" again to try to read the first track again. This is the case if you frequently see "OH, NO!" appear once before your system boots.

There are two possible types of problems that can occur with the drive speed-up. One is a home-sensor-adjustment problem, and will occur (if at all) only at system boot or warm boot (i.e., when you type a ctrl-C at the CP/M prompt). The reason is that the speed-up causes the head to "home" so fast it actually passes track 0 before the computer can recognize that it's there. The sensor in this instance is generating the signal too late for the computer to react to it. Unfortunately, it goes on telling the computer that the head is at track 0, even though the head is actually past it -- consequently, the drive head cannot read any valid data and the computer returns the boot error. THE FIX: Either get the home sensor adjusted to accommodate the speed-up (the "stops" may have to be adjusted also), or don't boot using the horizontal arrow keys (use the regular <RETURN> or <"> keys to boot instead).

In the second possibility, the stepper motor cannot stand the delay reduction. You may hear a laboring sound from your drive (making it sound like an old-time Apple disk drive), but there will be an increase in the Bdos errors you get. Sometimes, they will show up after the stepper motor heats up a bit from use, which means you will probably be able to boot with the arrow keys, but after a while you will have problems. If this is the case, then refrain from using the arrow keys to boot your diskettes.

OZROM 1E SUPPORT PROGRAMS

Several programs are included on the diskette supplied with your OZROM 1E package. These programs are intended to make your use of the OZROM 1E more convenient.

MEMORIZE

The main one is the MEMORIZE program. This one allows you to save the keyboard definitions that you have created with ctrl-shift-X and ctrl-shift-R. The MEMORIZE program creates a .COM file on diskette that can be run to restore your keyboard definition. For example, if you type **MEMORIZE<cr>** at the CP/M prompt, your current key definitions and any MyKey II function key definitions will be saved on a diskette in the default disk drive in a file called **KEYBOARD.COM**. At any time thereafter, you can type **KEYBOARD<cr>** from the CP/M prompt, which will restore the definitions you had when the MEMORIZE program was initially run.

You are not limited to the filename **KEYBOARD.COM**. Just enter the filename you want after the word **MEMORIZE** on the CP/M command line. For example, you have defined a **DVORAK** keyboard that you want to save under the name **DVORAK.COM**. Just type:

```
MEMORIZE DVORAK<cr>
```

at the CP/M prompt.

Nor are you limited to saving the file on the default drive. Just put an A: or a B: in front of the filename to save the file on the corresponding drive. Example: You want to save your WordStar special definitions on the B: drive under the name **WSKEYS**. Then type:

```
MEMORIZE B:WSKEYS<cr>
```

at the CP/M prompt.

If a file by the same name already exists on the destination diskette, MEMORIZE will attempt to delete it. If the file is read-only, MEMORIZE will tell you so and ask if you still want to erase the old file to save a new file of key definitions. MEMORIZE will also give you an error message if the attempted delete fails. (MD hard disk drive owners, take note: If you have a file in user 0 and you try to save a file by the same name in a different user area, you will encounter this error. You'll need to rename one of the files, or erase the file in user 0.)

Now, when you want to restore the WordStar keyboard definition that you created in the example above, just type:

```
WSKEYS<cr>.
```

Your keys will be changed and the CP/M prompt will reappear.

Autostarting with MEMORIZE

MEMORIZE can also be used to create a program that not only loads in a new keyboard definition, but also autostarts any desired program. You can create this autostarting program in one step:

```
MEMORIZE AUTOST WS<cr>
```

for example, creates a program called AUTOST.COM on the default diskette which, when the diskette is booted, will load WordStar after loading the new keyboard definition.

You can create a file by any name, not just AUTOST, and you can put any CP/M command following it. When you run the keyboard definition program that you have created, a new set of key definitions will be loaded, then your program will be executed as if you typed it on the CP/M command line by itself. DO NOT, however, attempt to run such a program from WordStar (with the R command).

Please note that some programs change the CP/M operating environment in such a way to keep this option from executing properly. One such example is the QWIKKEY program in the public domain. Some hard disk operating systems may also prevent the option from running correctly. If this is the case, do not attempt to run programs in this manner.

TIME

The TIME program can be used to set the system clock. To set the time, just type

```
TIME HHMMSS<cr>
```

without any colons or separators. H stands for hours, M for minutes, S for seconds. Seconds are optional, and you can set the time with only three digits -- for example, it's 8:30 in the morning:

```
TIME 830<cr>
```

You can also specify P.M. on the command line:

```
TIME 830 P<cr>
```

will set the time to 20:30:00 (military time -- it's a 24-hour clock). The space is required. For a quick reminder of the syntax, just type TIME<cr> at the CP/M prompt.

(Remember, ctrl-shift-T will toggle the time display on the screen. You may need to clear the screen to remove any remaining display of the time after you have toggled the time display off, as this action does not erase "what has gone before.")

INTERRUPT HOOK PROGRAMS

Several programs are included with your OZROM 1E to make use of the Interrupt Hook feature (described in the Technical Notes Appendix). These programs are ALARM, TIMER, and QUIKTIME. Only one of these "Interrupt Hook" programs can be active at any one time; if you run QUIKTIME after setting the alarm with the ALARM program, the alarm will not sound.

ALARM

The ALARM program actually allows your computer to do two things at once. You can be engaged in word processing and be reminded by your computer that a certain time has arrived.

To use the ALARM program, you need to set the time first, using the TIME program. Then run the ALARM program the same way, with the same syntax, selecting the time when you want the alarm to go off. Typing `ALARM<cr>` will give you a reminder of the correct syntax. Note: Only hours and minutes can be programmed with the ALARM program.

Now you can proceed with whatever you wish to do, like spreadsheet analysis or word processing. As long as you do not press RESET, the alarm is active and will go off at the selected time, giving you ten short double beeps, one per second.

Note: The less disk accessing you do, the more accurate the system clock. Although fudge factors are built in to accommodate the disk drive requirements (accessing a drive results in the system clock being momentarily shut down), variation from system to system on diskette reads means that the clock has no guarantee. If you do not turn on the drives, the accuracy should be better than three minutes a day.

TIMER

The TIMER program is similar in function to the ALARM program, except that the alarm will sound after every interval of time that you have selected. For example, if you had typed:

```
TIMER 20 5<cr>
```

then the alarm would sound every twenty minutes (starting twenty minutes from the time you executed TIMER), and you would hear five be-beeps each time. You can program any interval up to one hour, and specify the number of be-beeps per alarm from 1 to 9. One use of this program would be to remind you to save your data periodically. Another use would be to quickly set an alarm for a time less than an

hour away (you do not have to figure out the required alarm setting).

Again, type *TIMER*<cr> for a reminder of the syntax.

QUIKTIME

Normally, when the system clock is displayed on the screen, the display is only updated once every second. When you run the QUIKTIME program, the display will be updated 60 times each second.

Please remember that only the last "Interrupt Hook" program executed will be active.

CRASH

CRASH is a simple demonstration program to illustrate a safeguard feature built into your OZROM 1E. Run the program by typing

CRASH<cr>

at the CP/M prompt. You will see two lines at the top of your screen behave very strangely, and the only way to stop this display without turning off the machine is to press RESET. With any other ROM, your computer's drives and bell would turn on, and you might damage the data on the diskettes in the drives.

You will also see this effect any time you run a program that "does a no-no" with respect to your computer. It could be a program meant for a different operating system (like a program designed for a single-density operating system, and you have a double-density one), or a program that attempts to make use of the IEEE-488 functions, which the OZROM 1E does not directly support. System programmers may also see this effect during program development and debugging.

SHOW

SHOW is a public-domain program first written by Alan Miller to display text files a screenful at a time. This version has been modified to display WordStar document files without any garbage, and clears the screen before displaying each page (for a faster display on the O1 screen). To display a file, type:

SHOW FILENAME.TYP<cr>

where FILENAME.TYP is the name of the text file you wish to display. The first 22 lines of text will be displayed. Hit the space bar to view the next 22 lines, or any other key to quit back to CP/M.

*SET-UP For
OZBORNE***THE OZPATCH SYSTEM PATCHER PROGRAM**

This program will patch your Osborne 1 operating system, version 1.3 or later, to give you the following additional features:

Print-screen, 80 columns by 24 lines, with ctrl-shift-P, and 128 by 24, with ctrl-shift-W;

Send the next character you type, with ctrl-shift-O;

Query your system to check if ^P printer echo is active, with ctrl-shift-Q (your system will beep if echo is on -- does not apply to hard disk);

And allow you to set up a blinking cursor, keyclick, and/or time display as default conditions when you boot a OZPATCHed system diskette.

To use the program, just type

OZPATCH<cr>

at the CP/M prompt. The program will ask you a series of questions, starting with whether you desire a blinking cursor. (The defaults you select here through OZPATCH do not affect your ability to toggle blink, keyclick, etc.)

If you reply with a Y to the question concerning blink, you will then be asked to choose a blink rate from 1 to 9. Five is nominal, and the OZROM 1E default. You can speed up or slow down the rate here. If your choice turns out to be not quite what you wanted, you can run the OZPATCH program again, and again, as often as needed, until you get the set of conditions that you want.

After the blinking cursor comes the question as to whether you desire the keyclick feature enabled when you boot your diskette. (At any time before saving the new system on diskette, you can cancel the patch with ^C.)

Then the program will ask you if you want the system clock displayed on the display. If the answer is yes, you will be asked where on the first two lines that you wish to have it displayed.

Finally, the program will ask you which drive contains the diskette whose system you wish to patch. This will only patch floppy diskettes, even when run from a hard disk, as the program makes direct floppy BIOS calls.

The program will verify the patch and tell you that you can now use your new system. You will have to press RESET to boot the newly-patched system. Please note that OZPATCH will have to be run every time you "do a MOVCPM" on your operating system.

When you print your screen, the cursor will stop blinking (if it was blinking) until all the characters have been sent to your printer. The OZPATCHed system strips off all high-order (underline) bits on the characters, and converts graphics characters to asterisks (*). Also, the entire 80 x 24 screen of characters, including spaces, will be sent to your printer (128 x 24 if you used ctrl-shift-W).

Printer Control with an OZPATCHed System

To send a character to your printer, like a formfeed or a code for compressed print, just type ctrl-shift-0 and follow it with the character you wish to send. Example: To send a formfeed, type ctrl-shift-0, ctrl-L.

You will have to type ctrl-shift-0 for each character you wish to send to your printer. For example, you may wish to send the characters ESC, "W", 1 to your Epson or Panasonic dot-matrix printer to turn on the wide print mode. You will need to type:

ctrl-shift-0, ESC, ctrl-shift-0, W, ctrl-shift-0, ctrl-A

You'll note that we used control-A to send the value of 1 to the printer. You can send any ASCII character that can be generated by the keyboard, and some non-ASCII characters as well. The only character that cannot be directly generated from the keyboard is the ASCII null (0 hex), as this is used by the keyboard handler to indicate that no character is ready. Although you can use the FK program to program a normal function key to give you an ASCII null, the ctrl-shift-0 function will "parse" the function key codes (80h through 8Dh, including arrow keys) before they get "expanded" into normal function keys, which means you'll be sending the codes 80h through 8Dh when you use ctrl-shift-0 followed by a ctrl-number or arrow key.

(Since many printers allow you to send the required control code in the form 80h + code, to send a null you might use ctrl-shift-0, ctrl-0. The ctrl-0 in this case will generate 80h, which will be trapped and sent to the printer by the OZPATCHed system before it gets expanded into a normal function definition.)

With MEMORIZE, you can really control your printer from your computer. Just program (using MyKey II) a few not-normally used keys (like ctrl-esc, ctrl-tab, shift-esc, shift-tab, etc.) with your desired printer control codes, after the manner of the example for wide print above. Test them by pressing a key you've defined, then printing a test paragraph. Save them with MEMORIZE. Any time thereafter that you execute the resulting .COM file, you'll redefine those keys to immediately send the codes to select different print styles, etc., when the appropriate key is pressed.

(Remember that you can program up to 8 functions with MyKey II. If you have programmed four keys to send control codes to your printer, you have four "keys" remaining to reprogram inside your application program before you will walk on the printer-control-code definitions.)

Single and Double Density OIs Only

OZPATCH is designed to patch standard OI single- and double-density systems only, for use with the OZROM 1E. An error message will appear if your system cannot be patched.

FK, A FUNCTION KEY PROGRAMMER FOR THE OSBORNE 1
(c) 1984 M. Joel Guerra

As an OZROM 1E owner, you now have two sets of function keys available to you: the standard control-number function keys of your O1, plus the new MyKey II function keys. Here is a utility to greatly extend the flexibility and usefulness of the standard function keys. (You can load in different sets of MyKey II function keys with programs created with MEMORIZE, as discussed earlier.)

The FK function key programmer utility gives a user the following capabilities:

The ability to examine/modify the current function key definitions.

The ability to load a new set of function key definitions (from one of eight internal sets) without re-booting.

The ability to examine/modify the eight internal sets of function key definitions.

The ability to automatically execute a CP/M command after loading a new function key set.

The ability to redefine the O1's arrow keys, individually, to anything desired -- space permitting.

PROGRAM REQUIREMENTS:

The FK program will work with all standard versions of the O1 that support function keys (ROM versions 1.3, 1.4, 1.43, 1.44 -- and it should work with version 1.2, although this hasn't been tried) and with at least one hard disk subsystem (the MD series by Media Distributing). It will work with MOVCPM'ed systems and it might work with others when patched; see PATCHING FK, below. It works very well with the OZROM 1E.

The program takes up 4K bytes of disk space. This includes the eight sets of function keys that are stored within the FK.COM program itself. IF YOU WISH TO MAKE CHANGES TO THESE INTERNAL SETS, BE SURE THE FK PROGRAM IS NOT WRITE-PROTECTED. You can write-protect the program when you do not wish to make any further changes. This is due to the fact that FK writes itself back to disk (over the original FK program) when you answer "Y" to the question, "Do you wish to save the changes?" For this reason also, you should have backups of the FK program, just in case of "BDOS Err on X: Bad Sector" messages, in which case FK probably has been lost.

GENERAL DESCRIPTION

FK has four basic functions: examining or modifying the function key definitions currently in memory; examining or changing one or more of the eight internal sets of function key definitions; loading one of these internal sets into memory to be the current set; and executing a CP/M command after loading one of the internal function key sets.

A brief description of how to execute each of these functions can be obtained by typing `FK<cr>` at the CP/M prompt (if `FK.COM` is on the currently-logged drive). This is the same message you will see if FK does not understand your command, or if you attempt to load in a function key set that was programmed on a different machine.

For the following walk-through, log onto the drive containing the FK program. `<ESC>` represents the escape key and `<CR>` denotes the RETURN key or the ENTER key.

EXAMINING/CHANGING THE CURRENT SET DEFINITIONS:

At the CP/M prompt, type `FK M<cr>`. You will see a listing of your function keys currently in memory (from whence the "M") and a request to select a key to change -- 0 through 9 or U, R, D, and L. These last four letters represent the up-arrow key, the right-arrow key, the down-arrow key, and the left-arrow or backspace key. Note that, unlike the `SETUP` program supplied with your O1, the "0" function key definition is below the "9" definition.

The arrow keys are given in a different fashion than you may be expecting. Rather than defining the whole set to be either WordStar (tm - MicroPro) or CP/M (tm - Digital Research), you are shown the definition of each individual key. This is because you can define each key individually, just like one of the numeric function keys.

For example, if you have to fill out many forms, repeating the same information (like city, state, zip) over and over again, you can use your up- and/or right-arrow keys as single-key function keys. A warning here, though: If you use any other programs that change the arrow key definitions (for example, a WordStar patched to toggle the arrow keys coming and going), be sure to restore the arrow keys to a standard setting before using those programs.

At this point, pressing `<ESC>` will return you to the CP/M operating system. This can be a quick and convenient way to examine your current key definitions. (If you want a print-out, you are out of luck unless you have a print-screen feature, like the one available if you've run the `OZPATCH` program -- `OZROM 1E` only -- or the `PATCHER` program in the public domain.)

To Program a Key

If you wish to program a key, just hit the number or letter corresponding to that key. Instantly, a new help-screen will appear, telling you that you need to hit `<ESC>` twice to quit reprogramming your function key, and telling

you how to enter in those unusual characters that you can't normally use in the SETUP program. Since the escape key is used as a prefix to get these keys, you will need to type <ESC> followed by the open square bracket ([) key to put an escape in your function key definition.

You will also be told how many characters remain for your use in reprogramming that function key.

To Exit From Programming a Key

You must use <ESC><ESC> to quit programming the key. If you do this at the beginning of the line, the previous definition is unchanged. (The same applies even if you have entered characters -- if you backspace to the beginning of the line and exit, you won't affect the previous definition.)

Current Function Key Usage

You can use the present function keys in your definition. For example, if you just want to change the last part of function 1, type ^1 (control-1) to restore the definition at the beginning of the line, then edit the line and exit with <ESC><ESC> to redefine the key. **THE KEY IS NOT REDEFINED UNTIL YOU EXIT WITH <ESC><ESC>.** The newly defined key will be redisplayed along with all the rest, and you will be given another opportunity to change one of the keys or exit.

EXAMINING/CHANGING THE INTERNAL FUNCTION KEY SETS

There are eight internal sets of function keys stored inside FK.COM. To examine and/or to change one or more of these function key sets, type

FK C<cr>

at the CP/M prompt. If this is the first time you have done this, you may see a message saying that all sets have been initialized. The FK program always checks the compatibility of the internal sets with the current version of operating system that you are using. If they differ, FK will initialize the internal sets to correspond to the operating system you are using.

You will be asked which set to examine/change. You can respond with a number from 1 through 8, or press <ESC> to exit. All other characters will be ignored.

Picking a number from 1 through 8 will display the internal set corresponding to that number, just like the display of the current function key set earlier. You are prompted to select one of the keys to program or press <ESC> to exit.

You can program the set in exactly the same way you programmed the set in memory, with a couple of exceptions:

The set you are working on is NOT loaded at this time as the current function set, so changes made to the set do not affect the current definitions.

This also means that when you enter a function key to redefine a function key (e.g., ^1 when you are programming the definition for ^1) you will be inserting the current definition from memory, not from the set being changed.

When you exit from programming a set, you will not return to CP/M. Instead, you will be given the option of examining/changing one of the internal sets again, or exiting.

A Special Set of Arrow Keys

The arrow keys in each set are initially configured in a WordStar/CP/M combination, which works well with most application programs, like WordStar, dBASE II (tm Ashton-Tate), SuperCalc (tm Sorcim), while still giving CP/M a true backspace key. Since this setting (which I call Special) works so well with so many programs, I recommend leaving it as is unless you run into a program that requires the arrow keys to be one way or the other.

But, If You Have To Have 'Em

If you wish to have a true CP/M or WordStar setting, here's the way the keys must be programmed:

Key	CP/M	WordStar
U	^K	^E
R	^L	^D
D	^J	^X
L	^H <--	^S <-- entered with <ESC>

You'll note that to program in a ^H (control-H) or a ^S (control-S) you must hit the escape key and then the corresponding H or S key.

After you have finished programming one of the keys, entering <ESC><ESC> causes the new function key definition to be entered into the set, and gives you the option of reprogramming another key in that set. Pressing <ESC> at this point takes you back to the prompt to examine/change one of the eight sets.

If you press <ESC> to exit from the examine/change prompt, FK will return you to CP/M if the internal sets have not been changed. Otherwise, you will see the question "Do you wish to save the changes?" (Note: If the sets were initialized, you will be asked this question even if you do not change any set definitions.) Typing an "N" will result in an exit to CP/M, as will typing a ^C. Typing "Y" will make FK then ask which drive, A or B. You can change diskettes if you wish, as the program will reset the drive before saving a new FK.COM on disk. Then select A or B to save the changes. When the changes have been saved successfully, the program will say so and exit to CP/M.

Note to MEDIA DISTRIBUTING HARD DISK USERS: When you

first put this program on your hard disk, put it in user 0 ONLY. It will still be available to all user areas, and you will avoid the problem of having the file on user 0 and elsewhere, which does not work well. When FK saves the new set definitions, it will save itself on user 0 of the drive you select.

LOADING ONE OF THE INTERNAL SETS

The internal sets of function key definitions are numbered, as you've noticed, from 1 to 8. To load a particular set, just put the number after the FK command on the CP/M command line. For example,

```
FK 1<cr>
```

loads set #1 into memory. It will also display the set on the screen, giving you the opportunity to verify that the correct set has been loaded and to refresh your memory. You will see a comment that the set has been loaded, and the CP/M prompt will return.

(Note: If you have not programmed any of the internal sets, you may see the statement that the set must be programmed on your computer before it can be loaded, followed by the syntax message. See EXAMINING/CHANGING THE INTERNAL FUNCTION KEY SETS, above.)

EXECUTING A CP/M COMMAND AFTER LOADING A SET

Although it's fairly easy to load a set by typing FK n<cr> (where n is the set #) and then execute another CP/M command by entering it at the next CP/M prompt, there is a way to load a set and immediately execute the next command. Just put the command on the same line as the FK command. For example, if you wish to load WordStar after loading in a set of WordStar function keys (say, set #2), your command line would look like this:

```
FK 2 WS<cr>
```

This would load in set #2, display the set, and run WordStar.

The command you put after the FK command can be more than 100 characters long. If it will run from the CP/M command line, it will run in this manner, too, except...

N.B.: As discussed earlier, some programs change the CP/M environment in such a way to keep this option from working properly. For example, you can use the R command of WordStar to load a set or redefine a key, but DO NOT ATTEMPT to execute a program through FK when running under the control of WordStar. Also, this option [of running a program after loading in a set of function keys] will NOT work when the warm-boot vector at location 0 has been changed by another program, like QWIKKEY.

Although there's not much difference between the two

methods, the second method allows you to program your "superdisk" or your Media Distributing hard disk boot diskette with several of these combined commands, one for each application you wish to run. For example, your keys could look like this:

```
^1    FK 1 SC<cr>
^2    FK 2 WS<cr>
^3    FK 3 DBASE<cr>
^4    FK 4 MBASIC<cr>
^5    FK 3 DBASE ACCOUNTS<cr>
```

and so on. In this way, a single function key both redefines the function keys for a particular program and runs the program. Note that in the example, ^5 loads in function set #3, loads DBASE, and tells it to look for ACCOUNTS.COM to execute. We suggest that, if you use this approach, you should reserve a set of keys to restore this set -- function set #8 could be used to do this. After you finish an application, you would then load set #8 prior to selecting another application.

The nice thing about the approach given above is that the application program itself is not modified. The function key sets can be easily changed without affecting the application program.

A Function Key Aid

If you have a grey-case O1, you may want to get an old broom handle and cut it to size to fit just above the function keys on your keyboard. Wrap it with a sheet on which you have put your favorite function keys, one row on the paper for each set, spaced to correspond to the number keys. Then just put it in the groove above the keys on the keyboard, and turn it to correspond to the function keys that you load. Or you might find some clear plastic tubing that fits, and insert the function key definitions inside the tube.

JUST 8 SETS?

If you need more than eight sets, rename FK.COM to FKA.COM or some name convenient to you (as long as it ends in .COM), execute the new FKA (or whatever), change the sets you wish to change, and save the changes. This will create a new FK.COM program on your disk, which should be renamed also to protect it, should you decide to change the previous set. Each time you run the program with the "C" option ("C" stands for "change") and save the changes that you make, any existing FK.COM is deleted from the selected diskette, and a new FK.COM is saved. By renaming FK.COM to something else, you preserve the old set definitions.

This program was originally designed to run on O1s with Media Distributing MD hard disk drives. A design goal was ease-of-use coupled with small size -- mutually exclusive objectives from the author's point of view. The maximum

program size had to be 4K, and it had to have as many sets internally as possible. Although it's somewhat inconvenient not being able to refer to a set by a name (as of this version!), the program loads and executes very quickly because it does not have to search the disk for any function key files, with the added advantage of no extra room needed on the disk for these function key sets.

HOW MANY CHARACTER SPACES DO I GET?

It depends on which version of the CP/M operating system you are using. If you have one of the double-density versions, you get as many character spaces per set as you would with the SETUP utility. If you have a single-density operating system like the one for ROM 1.3, you get one character space less than what you are supposedly getting when you run the SETUP utility for that operating system. However, if you programmed all your available spaces with that version of SETUP and saved the changes on diskette, your system will crash when you next use it (don't just take our word for it -- try it!). You still get about 110 total character spaces, including the arrow keys, for single density; there are 89 or 90 total keys for double-density systems, depending on whether or not you have the operating system that supports the 80-column card. Note that the number refers to TOTAL character spaces; if each key had only one character assigned to it, you would have 14 less character spaces remaining to program.

If you can patch your operating system so that the original set has more function key spaces, FK will adapt itself to your new configuration provided that the function key pointers start at BIOS BASE + 6Bh and the function key definitions are in the same page of memory, starting after BIOS BASE + 80h.

SOME SHORT CUTS

You may have noticed that the syntax listing you get when you type FK<cr> had more options than the four listed above. You can also go directly to programming a function key or function key set by entering the key or set number as the second parameter on the command line. FK M U<cr> will take you directly to programming your up-arrow key, while FK C 7<cr> will display set #7 and ask which key you wish to change.

PATCHING FK (version 2.0 or earlier)

If you have "done a MOVCPM" on your system, your function keys have moved with your system, and attempting to run an unmodified FK on your system will result in an error message: "Unusual BIOS. Program aborted." You can patch a single location in FK.COM to enable FK to run on your system, but it will then be incompatible with those unmodified systems. First determine the page which has the BIOS jump

The OZROM 1E

table AND the function key pointers and function keys. If all you have done is a MOVCPM operation, then load DDT with FK.COM as follows:

```
DDT FK.COM<cr>
```

At the hyphen prompt, type S2<cr>. You will see

```
0002 xx _
```

where xx is the byte you need to remember. Then type a period followed by a carriage return, then

```
Dxx80<cr>
```

substituting the byte you saw for the xx.

Look for the AUTOST followed by your function key definitions. If you don't see them, do not continue with this patch -- sorry, but your system is probably not compatible with FK at all.

If you do see them, then type

```
S42A<cr>  
xx<cr>          <-- again, substitute for xx  
^C
```

and when the CP/M prompt returns, type

```
SAVE 16 FK.COM<cr>
```

and your patched FK should be ready to try. If you do not see the same error message again, and the function key definitions display properly, congratulations.

PATCHING FK (version 2.1 or later)

If you have the OZROM 1E, FK should be able to find your system's standard function keys without any patching. If it cannot, you either have a radically different system, or there is something wrong with your hardware or software.

THE SET PROGRAM FOR THE OSBORNE I
(C) 1983 M. Joel Guerra

SET is a program to immediately and temporarily change almost all of the parameters (settings) of your computer (and some of the parameters of your printer as well). In a way, it functions like the SETUP program included with your computer, but it is quicker, easier to use, and does more.

With SET, you can:

- Set your arrow keys to WordStar, CP/M, or a special setting that may eliminate the need for either WordStar or CP/M arrow keys for you.
- Set the baud rate on your serial port instantly to any of three settings -- 300, 1200, or 19200.
- Change the logical width of your Osborne screen from 52 to 80 to 128 characters per line.
- Set the physical width of your Osborne screen or external monitor (on those Osbornes with the 80-column upgrade from Osborne) to 52, 80, or 104 columns displayed. (This feature is directly available through the OZROM 1E via ctrl-shift-1, -5, -8, or -0.)
- Set horizontal scrolling off for programs like SuperCalc, or set scrolling on for programs like WordStar. This can also be accomplished with control-shift-S when using the OZROM 1E.
- Set the LST: (printer) device to be the serial port, the parallel port, the IEEE-488 port, or even your Osborne screen (very useful if you don't have a printer, or don't want to waste paper).
- Set the protocol of your serial printer to be ETX/ACK, or XON/XOFF, or no protocol.
- Set your printer to top-of-form by sending it a form-feed command. (Note: Owners of the OZROM 1E can send any character to the printer at any time, even in the middle of a program, if they have patched their operating system with OZPATCH.)
- Set your Epson or Okidata dot-matrix printer to compressed, expanded, emphasized, or normal printing (without MBASIC!).
- Set any memory location in your computer's memory to a desired value, like CP/M's IOBYTE for special applications, or those locations for "speeding up" your drives, locking and unlocking your keyboard, etc. Consult your manual for specific locations.

This program is designed to be both useful and easy to use. It may easily become a favorite utility for your computer, and its 2K size may make it a resident of all your program diskettes.

USING SET

First follow the instructions for backing up the Utilities diskette that we've sent you. Then copy SET.COM onto the disk(s) of your choice with PIP or other file-copying program. "Boot" your computer so that you see the CP/M prompt on the screen, either an "A>" or a "B>". Put the disk with SET.COM into this drive if it is not already there.

Type the letters "SET" followed by a carriage return key to see a list of options you can choose from. Should you ever enter a command that the program does not understand, this is the menu that you will see.

Try the following command. This is the long form of the command; the short form will be shown later.

`SET ARROWS WORDSTAR<cr>` where <cr> means press the carriage return key.

Your drive will turn on for a moment and a message will appear:

WordStar arrow keys loaded.

Test this out by hitting the left-arrow key; you should see a "^S" appear every time you hit the key. Hit the down-arrow key (^X) to make the "^S" characters disappear.

Now type in the following command:

`SET ARROWS CPM<cr>`

You will see the message:

CP/M arrow keys loaded.

You can prove this by typing some characters and then deleting them with the left-arrow key, which is now CP/M's backspace key (^H).

The short form of the command is simply the initials of the arguments, in most cases. For example:

`SET A C<cr>`

will set the arrows to CP/M, and

`SET A W<cr>`

will set the arrows to WordStar's configuration. (There is a special combination of WordStar and CP/M arrow keys that I call Special -- the left arrow key is CP/M's backspace, while the rest of the arrow keys are configured for WordStar. This works well with SuperCalc, dBASE II, and most other programs, as well as CP/M and WordStar.)

The one exception to single-letter arguments is the **SET SCROLL ON** or **SET SCROLL OFF** command. You need at least two letters for the second argument because they both start with the letter "O".

To set the different printer protocols, you need both a slash ("/") and a letter for the second argument. To set the printer to the X-on/X-off protocol, you would need the command

SET P /X<cr> or **SET PRINTER /XONXOFF<cr>**

which would return a message that the printer was set to XON/XOFF. To return the printer protocol to normal, use this command:

SET P /N<cr>

instead of **SET P N**, because this last command is to reset the Epson printer to normal after the printer had been set to compressed, emphasized, or expanded.

Most of the options requiring a number argument need only the first digit to work correctly:

SET B 3<cr> will set the baud rate to 300.
SET L 5<cr> will set the logical screen width to 52 columns
SET W 1<cr> will set the physical screen width to 104 columns when the SCREEN-PAC is installed. (Note: If your OPERATING SYSTEM supports the SCREEN-PAC even if you don't have one, typing **SET<cr>** will display the option anyway. Executing the command won't affect operation, however. Also, OZROM 1E owners will find it more convenient to use ctrl-shift-1, -5, -8, and/or -0 to set the physical width.)

An exception to the single-digit short form is the **SET BAUD** option. **SET BAUD 1<cr>** will set the baud rate to 1200, but **SET BAUD 19<cr>** will set it to 19,200, unsynchronized.

Please don't let this confuse you. If you are not sure about an option, just type **SET<cr>** to see the list of options. Then try out a command. The program will tell you what it has done after it does it!

The form for changing memory with the **SET** program is:

SET Hxxxx xx<cr> or **SET Hxxxx xxxx<cr>**

The first form is for changing a byte. The first four x's are the address; the remaining two are the new byte, in hexadecimal. For example, **SET HEF13 01<cr>** will change memory location EF13 hex to 01 hex. (You could leave off the leading zero: **SET H3 0<cr>** changes your IOBYTE so that

all your printer output also goes to the Osborne screen.)

The second form is for changing a word (two bytes) at a time. In this instance, SET assumes an address value for the second set of x's, so it stores the word high-order-byte-last. SET H0001 E103<cr> would store 03 hex in location 1 and E1 hex in location 2.

Note: SET will protect itself from being modified with this command.

INSTALLING SET FOR YOUR PRINTER

To reconfigure the program for a different printer (where you already know the control codes you need), you can use DDT. The program contains the strings to be sent to the printer in the following locations:

- 8C0 - string to initiate COMPRESSED printing
- 8D0 - string to initiate EMPHASIZED printing
- 8E0 - string to initiate EXPANDED printing
- 8F0 - string to restore printer to normal

Each string must contain a minimum of 2 bytes. The first byte indicates the number of bytes to be sent to the printer. Following this are the bytes that are actually sent. For example, to set an Epson to emphasized mode, two bytes must be sent to the printer. Therefore, the string starting at location 8FD0 (hex) is 02 (to indicate that two bytes are to follow) 1B 45. (Note: 1B is the hex code for ESC. See appendix B of any Epson MX printer manual for the Epson control codes. The manual for your printer should contain the control codes it needs.) A string can be up to 16 bytes, meaning that up to 15 bytes can be sent to the printer at any one time.

When you have made your changes, use the command

```
SAVE 8 SET.COM<cr>
```

to save the modified version.

While the SET program is currently prepared to set your printer to emphasized, compressed, and expanded modes, any string may be sent to the printer by making the appropriate modifications in the fashion outlined above.

APPENDIX A

A SMOOTHER, FASTER-SCROLLING WORDSTAR 2.26

Everybody's got a smoother, faster-scrolling WordStar for you. Of course, we think ours is the best; but we're naturally biased. Yet, this version will work on both single- and double-density machines, and on versions of CP/M that have been MOVCPM'ed. The only requirement is that you have the OZROM 1E.

If you want this version, you'll need your CP/M utilities diskette and a COPY of your WordStar diskette, one which does not have any "patches". (INSTALLED WordStar should work fine.)

NOTE: This patch, like all other smoother, faster, etc. patches, uses underlining in place of half-intensity. If you do not like this feature, set location 02B3 below to 0.

Put the CP/M utilities diskette in the A drive and the WS diskette in the B drive. Boot CP/M, so that you see the A> prompt. Enter in the information below that is underlined. You should see the computer print the rest on the screen. Parentheses enclose helpful information that we've provided.

```

A>DDT_B:WS.COM<cr>
DDT VERS 2.2
NEXT PC
3880 0100      (this may say 3900 0100 or 4000 0100)
-S249<cr>
0249 50 80<cr>
024A 02 .<cr>
-A264<cr>
    0264 JMP_2E0<cr>  (this will tell WordStar
    0267 <cr>        to clear the screen)
-S2AA<cr>
02AA 00 FF<cr>      (this tells WS to use the last
02AB 00 .<cr>        space on each line)
-S2B0<cr>
02B0 00 FF<cr>      (this tells WS to use our Memory-
02B1 00 00<cr>        Mapped Video)
02B2 00 F0<cr>      (F0 is not a typo)
02B3 00 FF<cr>      (for no underlining, use 00 instead)
02B4 00 FF<cr>
02B5 00 .<cr>
-A2BD<cr>
    02BD JMP_300<cr>
    02C0 <cr>
-S2D2<cr>
02D2 19 00<cr>      (may be 40. Make 0 anyway)
02D3 00 .<cr>
-A2E0<cr>
    02E0 MVI_A,1A<cr>
    02E2 JMP_106<cr>

```

```
02E5 <cr>
-A300<cr>          (this is the smoother, faster, etc.)
0300  LHL D,371D<cr>
0303  MVI A,E0<cr>
0305  ADD L<cr>
0306  STC<cr>
0307  RAR<cr>
0308  MOV L,H<cr>
0309  MOV H,A<cr>
030A  MVI A,0<cr>
030C  JNC 311<cr>
030F  MVI A,80<cr>
0311  ADD L<cr>
0312  MOV L,A<cr>
0313  SHLD EF5A<cr>
0316  LDA EF58<cr>
0319  MOV H,A<cr>
031A  MVI L,9<cr>
031C  PCHL<cr>
031D  <cr>
-^C
A>SAVE_56_B:WS.COM<cr>
```

When the A> prompt returns, you are ready to test your new WordStar.

If you want your WordStar to automatically log the B drive, do the following:

Right before you type the ^C (near the end), enter:

```
-A100<cr>
0100  JMP 3880<cr>
0103  <cr>
-A3880<cr>
3880  MVI C,E<cr>
3882  MVI E,1<cr>
3884  CALL 5<cr>
3887  JMP 2D10<cr>
388A  <cr>
```

and then type ^C and save 56 pages. (Always save 56 pages unless you've done some extended assembly-language patching; OCC saved too many pages on some versions of WS 2.26.)

PATCHES FOR DBASE II FUNCTION KEY OPERATION

Here is one method to allow dBase II to properly handle the function keys, both the normal ones and the ones provided with the OZROM 1E:

Put the CP/M utilities diskette in the A drive and a copy of your dBase II program diskette in the B drive. Boot CP/M, so that you see the A> prompt. Enter in the information below that is underlined. You should see the computer print the rest on the screen. Parentheses enclose helpful information that we've provided.

(For version 2.3b)

```
A>DDT_B:DBASE.COM<cr>      (or DDT_B:DO.COM<cr>)
DDT VERS 2.2
NEXT PC
4700 0100
-S34F6<cr>
34F6 CD Q<cr>
34F7 1C Q<cr>
34F8 35 Q<cr>
34F9 ^C
A>SAVE_70_B:DBASE.COM<cr>  (or SAVE_70_B:DO.COM<cr>)
```

(For version 2.4)

```
A>DDT_B:DBASE.COM<cr>      (or DDT_B:DO.COM<cr>)
DDT VERS 2.2
NEXT PC
4D00 0100
-S355A<cr>
355A CD Q<cr>
355B 7C Q<cr>
355C 35 Q<cr>
355D ^C
A>SAVE_76_B:DBASE.COM<cr>  (or SAVE_76_B:DO.COM<cr>)
```

Both these versions disable the ^S pausing for LIST STRUCTURE and LIST FILES, but ^S works fine when LISTing the contents of the database in use.

A similar, though lengthier, patch for version 2.4 is given in the July 1984 issue of the FOGHORN which was taken from the October 1983 issue of dNEWS. Again, this is for version 2.4 only.

Notes

APPENDIX B

INSTALLATION INSTRUCTIONS FOR THE OZROM 1E

We recommend that *before* you do anything else, make a back-up copy of the supplied diskette.

We also recommend that you read the following instructions before attempting to install the OZROM 1E ROM. Should the instructions leave the slightest doubt in the installer's mind about any part of the installation, or should any difficulty be encountered, please call Micro Management, Inc., at (214) 256-1960.

Please take extreme care with the electronic parts involved. Ground yourself and your tools immediately before handling the OZROM 1E EPROM to avoid any damage due to static electricity. (The screw holding the cover plate on those three-pronged electrical sockets is usually tied to "ground". Touch the screw before handling the OZROM or your computer.)

Note: If you have any additional circuit card installed which covers the Osborne 1 ROM, you will have to remove it temporarily. In the instance of the Osborne 80-column Screen-Pac, removing and replacing this circuit card can decrease the reliability of the connections to the card; we strongly suggest that you add a fan to your O1 (if you have not already done so) to reduce the thermal stress and subsequent failure of these connections.

We have enclosed some drawings to help you in the installation of the OZROM 1E inside your Osborne 1. There are currently three different combinations of Osborne cases and circuit cards:

If you have the tan and black case, start with drawing A near the end of this appendix. You will have to determine which of two possible main boards your computer has.

If you have the blue and white case, refer to drawing B.

TAN-CASE INSTALLATION INSTRUCTIONS:

You will need a Phillips screwdriver, a flat-blade screwdriver and a 1/20-inch Allen wrench for those !@##% knobs!

1. Unplug the computer from the wall socket.
2. Disconnect the keyboard from the front panel of the case.
3. Open up the Osborne computer case: Remove the screws from the front panel around the perimeter (6 screws), from the power switch panel (6), and from the handle area (2). If the screws are of different lengths,

remember which ones go where. With the Allen wrench, remove the contrast and brightness knobs.

4. Pull the front panel away from the computer, catching any diskettes you had left in the pockets and putting them aside. Put the panel where your dog or cat can't chew on it. Locate the system ROM about two inches **DIRECTLY BEHIND** the keyboard connector; if it is not there, you have a real **ANTIQU**E and will need two 2716 EPROMs instead of the single 2732 EPROM that we have provided -- contact us.
5. Remove the one or two screws that fasten the computer to the case at the front edge of each side (see fig. 1).
6. Carefully pull the computer out of the case toward you. It helps to have a friend to hold the case. **IF YOU HAVE NOT UNPLUGGED YOUR COMPUTER FROM THE WALL SOCKET, DO SO AT THIS TIME.** You may not be in any condition to do so later.
7. Turn the computer over and place it down on a surface you are not worried about scratching (newspapers help). You should see the bottom of the main circuit board.
8. Remove the four screws that hold this board down. Turn the board over very carefully; you should not have to disconnect anything.
9. The system ROM should be in the location indicated in figure 2. With a flat-blade screwdriver, carefully pry the ROM out of its socket, working it up a little at a time from both sides. When you have removed it, put it in a safe place for a moment.
10. Now remove the OZROM 1E from its protective tube and plug it into the just-emptied socket on the main circuit board, insuring that the notch on the OZROM 1E is pointing in the same direction as the notches on all the other **BIG** integrated circuits -- see Fig. 2. Make sure that none of the pins bend or stick out of the socket as you plug it in.
11. Replace any circuit card you may have removed earlier. Place the ROM that you removed from the board earlier into the protective tube for safekeeping.
12. Position the main circuit card back on its supports and replace the four screws that hold it in place.
13. At this time, *making sure there is no danger of shorting any components*, plug the AC power cord back in. Power the computer back on, after checking that there is a shorting plug or monitor adapter on the appro-

ropriate card edge connector. Wait a moment for the video monitor to warm up, then check the screen for a sign-on message. If you did not hear a beep at power-on, or the beep came on and stayed on, turn power off and check all connections. If you heard a beep at power-on, but no screen message appeared, check the contrast and brightness knobs, and the video connectors. If you still cannot get an appropriate sign-on message to appear, turn off power and replace the OZROM 1E with the original ROM and try again. If your original works but the OZROM 1E does not, inspect the OZROM 1E for bent pins. In the unlikely event everything checks out but the OZROM 1E still doesn't work, return it to us for replacement.

14. If an appropriate sign-on message appeared, turn off the machine and unplug the power cord. Reverse the disassembly procedure, making sure that when you replace the computer inside the tan case, you do not crimp or screw any screws into any cables at the back of the machine. Check all your cable connections at this time to make sure none are loose.
15. After re-assembly, power the computer back on again and verify operation again. Turn off the computer immediately if anything occurs which you do not associate with normal power-on, like a continuous beep or pure garbage on the display. In this instance, perform a partial disassembly and check your connections again.
16. Finally! You're done with the installation and can proceed with the operating instructions.

BLUE-CASE INSTALLATION INSTRUCTIONS

You will need a Phillips screwdriver at least 4" long and a flat-blade screwdriver.

1. Unplug the computer from the wall socket.
2. Twist off the contrast and brightness knobs and put them in a safe place. Remove the 6 screws around the perimeter of the front cover and put them in the same safe place. Detach the keyboard from the computer at the front panel connector.
3. Pull the front panel toward you and put it in a safe place, away from any pets, kids, and inquisitive neighbors. Turn the computer upside down to expose the bottom 5 screwholes. Loosen all five screws and pull the bottom straight up and away from the computer. You can leave the screws in the case for ease of reassembly, if you desire. Put the case bottom down in a safe place, also.
4. You should now be looking at the bottom of the main

circuit card. Remove the four OUTSIDE screws and put them in a safe place, being careful to distinguish them from the front panel screws.

5. Lift the main card up and back. You may have to disconnect the internal monitor cable connector to lift the card.
6. The system ROM should be in the location indicated in figure 2. With a flat-blade screwdriver, carefully pry the ROM out of its socket, working it up a little at a time from both sides. When you have removed it, put it in a safe place for a moment.
7. Now remove the OZROM 1E from its protective tube and plug it into the just-emptied socket on the main circuit board, insuring that the notch on the OZROM 1E is pointing in the same direction as the notches on all the other BIG integrated circuits -- see Fig. 2. Make sure that none of the pins bend or stick out of the socket as you plug it in.
8. Replace any circuit card you may have removed earlier. Place the ROM that you removed from the board earlier into the protective tube for safekeeping.
9. Position the main circuit card back on its supports and replace the four screws that hold it in place. Make sure that the internal monitor cable is plugged back in CORRECTLY if you had to disconnect it earlier. If you have double density, be sure to keep this monitor cable away from the double-density circuit card.
10. At this time, *making sure there is no danger of shorting any components*, plug the AC power cord back in. power the computer back on, after checking that there is a shorting plug or monitor adapter on the appropriate card edge connector. Wait a moment for the video monitor to warm up, then check the screen for a sign-on message. If you did not hear a beep at power-on, or the beep came on and stayed on, turn power off and check all connections. If you heard a beep at power-on, but no screen message appeared, check the contrast and brightness knobs, and the video connectors. If you still cannot get an appropriate sign-on message to appear, turn off power and replace the OZROM 1E with the original ROM and try again. If your original works but the OZROM 1E does not, inspect the OZROM 1E for bent pins. In the unlikely event everything checks out but the OZROM 1E still doesn't work, return it to us for replacement.
11. If an appropriate sign-on message appeared, turn off the machine and unplug the power cord. Reverse the disassembly procedure, making sure that when you re-

place the computer cover, do not crimp or screw any screws into any cables. Check all your cable connections at this time to make sure none are loose.

12. After re-assembly, power the computer back on again and verify operation again. Turn off the computer immediately if anything occurs which you do not associate with normal power-on, like a continuous beep or pure garbage on the display. In this instance, perform a partial disassembly and check your connections again.
13. Finally! You're done with the installation and can proceed with the operating instructions.

Drawing A -- Tan and Black Cases

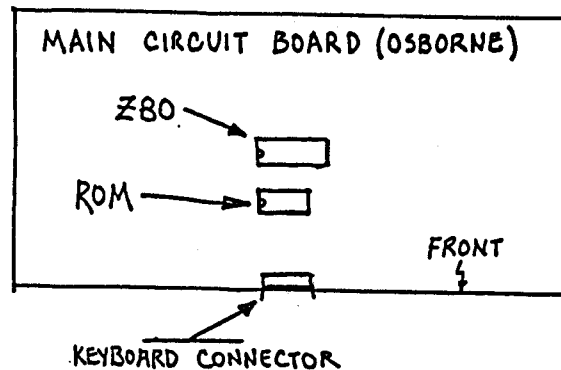
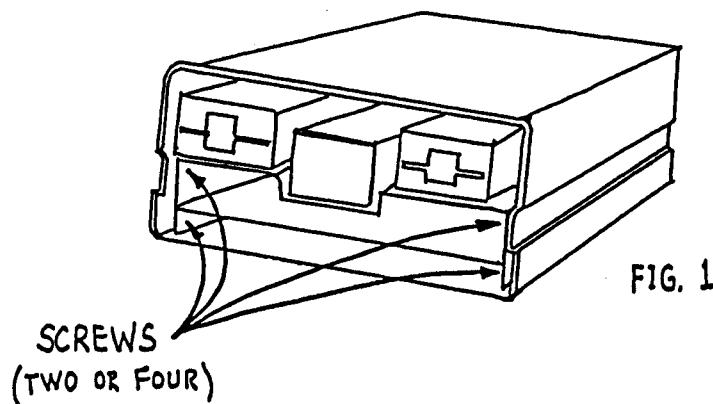


FIG. 2

Drawing B -- Blue and White Cases

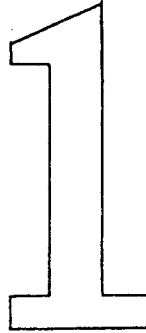


FIG. 1

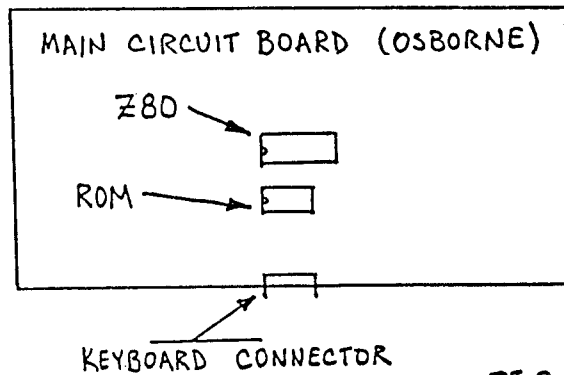


FIG. 2

APPENDIX C

TECHNICAL INFORMATION ON THE OZROM 1E

Differences Between the OZROM 1E and other ROMs

IEEE-488 functions

The OZROM 1E does not support the IEEE-488 functions. Although Centronics printer operation is not affected, some printers (like those manufactured for the Commodore PET computers) and some data acquisition and analysis devices (like those produced by Hewlett-Packard) require the IEEE-488 routines to operate. In this instance, extra software would be required to allow the operation of such devices with the OZROM 1E. We may offer such software in the future if the demand warrants it, but brief surveys indicate that less than five per cent of O1 owners use the IEEE-488 port for IEEE-488 devices.

Vertical Scrolling

Since other ROMs regard the top 4K of RAM (F000-FFFF hex) to be dedicated to the video screen, they can scroll vertically very easily, merely by moving the "window" down through video memory. When a linefeed is issued at the bottom of the screen, the O1 "window" is itself moved down one line. In the OZROM 1E, only 3K of RAM is used for the video (F000-FBFF hex) since the last 1K is used for the redefinable keyboard and extra function keys. Since the last 1K is reserved, a linefeed at the bottom of the screen results in the moving up one line of the last 23 lines displayed, first the characters themselves, then the half-intensity attribute (due to the original design). This double block-move requires about 35 milliseconds, resulting in a visibly slower scrolling of those programs which scroll vertically instead of using cursor addressing.

1200-baud modem operation

This last is also the reason for the unreliability of modem operations at 1200 baud. Although the electrical design of the O1 does allow interrupt-driven serial I/O, using this interrupt would require the modification of almost every modem program that runs on the O1, not to mention the extensive rewrite of the video output routines to support the interrupt. The only alternative requires the polling of the serial port to determine if a character is waiting at the input. At 1200 baud, which translates roughly to 120 characters per second, the serial port must be polled at least every 8.33 msec in order to avoid missing a character (the port in the Osborne requires that you "read" any character in its input register before it will accept another character; if you aren't quick enough, you'll miss subsequent characters).

When a linefeed is issued at the bottom of the screen, the computer is tied up in a video block move for about 35 msec, resulting in a loss of a couple of characters (at 1200

baud) near the beginning of every line. The best method of dealing with this problem is to answer "4" to the question of "How many nulls?" that a remote computer would ask.

If the remote computer does not ask the question, you are just about out of luck. IF YOU CAN RECEIVE A FILE WITHOUT ECHOING IT TO THE SCREEN, you can safely transfer files with a 1200-baud modem, or with some file-exchange programs like MOVE-IT. Another method of transferring files that get echoed to the OI screen is to pause the display before it gets to the bottom with control-S (which should get sent to the remote computer, telling it to pause also), clear the screen with control-shift-Z, and send a control-Q to tell the remote computer to resume. If you have OZPATCHed your system, you can assign ctrl-shift-Z followed by ctrl-Q to a function key, then operate your modem using ctrl-S to pause the display and the function key to continue (it will clear the display, then send ctrl-Q to the remote computer).

One more possibility: If you are into assembly-language programming, you may want to modify the I/O section of your modem program to test for a line-feed character at the serial input port. If you get one, send an XOFF (control-S) character to the remote computer, process the linefeed, and then send an XON (control-Q) to tell the remote computer to resume communication.

Truer Televideo 912/920 Emulation

It was impractical to do a true emulation of the TV 912/920 terminals, as most of the programs already written for the OI make use of the current OI definitions which stray somewhat from the TV definitions. However, some minor bugs have been corrected, most notably the "phantom lines" in WordStar and the character-delete bug when line length is set at 128. Also, the following commands have been added:

ESC j (1Bh 6Ah) will turn on both underlining and half-intensity at the same time.

ESC k (1Bh 6Bh) will turn off both underlining and half-intensity at the same time.

ESC Y (1Bh 59h) will clear the display from the cursor to the end of the screen.

The y-coordinate in ESC S sequences is now ignored.

An Interrupt "Hook" for Programmers

Another feature of the OZROM 1E is the interrupt "hook" we've provided to make systems programming a little easier. The ALARM program uses this hook to run a "task" in the "background mode" while you continue to compute in your usual manner.

When the video circuitry generates its 60-hertz interrupt, the SYSINT system interrupt routine saves the stack pointer, program counter and CPU registers, then it calls address OEFF2h. Under standard operating conditions, a RET instruction will be encountered there, and the interrupt

routine will continue with updating the clock and counters, scanning the keyboard, resetting the interrupt, and restoring the previous "operating environment" after which control is returned to the interrupted program.

With this hook, a gut-level programmer can install his interrupt-level routine without having to save or restore any registers -- he merely puts a jump-to-his-routine at location OEFF2h and ends his routine with a RET instruction. There are 26 bytes of interrupt stack at his disposal (he can push or call up to 13 levels), and he can use RAM from OFCC0h to OFD7Fh for his routine in this version of the OZROM 1E (192 bytes) without having to "do a MOVCPM."

Serious recommendations to this would-be gutsy programmer: You should disable interrupts before moving your code and the jump-to-your-code into position, then enable them afterwards. If you use the video RAM space mentioned above, do not use any instructions which output to port 2 (the half-intensity bank port). Also do not use this space if you will be running one of the programs which walks on the reserved video space. In any case, your code must be above 4000h. It also must be below 0C000h if you plan on using port 2 to access the half-intensity bit of video memory. It would also be a very good idea to insure that your code won't be "walked on" by normal OI programs.

1793/1797 Floppy Disk Controller Operation

The OZROM 1E floppy disk I/O routines are compatible with both the WD1793 (MM8877) and WD1797 floppy disk controller chips. Double-sided disk drive operation is supported, and the OZROM 1E is the ROM supplied with the excellent G2 780K "quad-density" drive upgrade (that's 780K per DRIVE!) At the time of publication of this document, only the OZPATCH program was incompatible with the G2 operating system software (and later versions of the G2 software may include the features obtained through OZPATCH).

Control-shift-key Notes

There are three levels at which ctrl-shift keys are interpreted: the interrupt level, the ROM console input level, and the RAM BIOS console input level.

Interrupt-level keys are those keys that are immediately processed upon being accepted by the keyboard scanner. These keys are the screen-positioning keys; the time, key-click, blink, and auto-horizontal-scroll toggles; the Screen-Pac 52-, 80-, and 104-column select keys; and the Assign-Parallel-Printer-to-Null-Device and Reassign-Parallel-Printer keys.

Actually, it's not the keys themselves, exactly, but the codes they generate, which determines the level at which they are processed. The interrupt-level keys are keys which generate keycodes from OF0h to OFCh, which are the "process immediately" codes. The ctrl-shift keys are decoded through the table in ROM (not RAM) and cannot therefore be redefined, but any of the other keys can be defined with the code for the ctrl-shift keys and become "interrupt-level"

keys.

Codes OFDh through OFFh cannot be generated from the keyboard, and if they are found in the RAM keytable they are dealt with harshly, resulting in the keytable pointer being changed to point back to the table in ROM. (The keytable pointer will also point back to the table in ROM if the value at [KTBPTR] + 2 changes from its zero value. See the Memory Map.)

Codes OE8h through OEFh are mapped into OF8h through OFFh. Again, these are codes that cannot be generated through the keyboard -- someone would have to poke them into the keytable in RAM.

Codes OEOh through OE7h are used to "expand" a corresponding MyKey II function key. They are generated and assigned during the MyKey II programming via ctrl-shift-R.

Codes ODDh, ODEh, and ODFh are permanently assigned to ctrl-shift-Z, ctrl-shift-X, and ctrl-shift-R, respectively. They can, however, be also assigned to other keys through ctrl-shift-X. These codes are trapped and interpreted at the ROM console input level. All other codes are passed to the RAM BIOS, where an OZPATCHed system would trap the keycodes corresponding to screen-print and printer control.

More on System Crash Damage Protection

With the OZROM 1E, all RST instructions are trapped in the ROM to produce a controlled system crash. All IEEE-488 jump vectors are likewise controlled and produce the same result -- which shows the top two lines of the display going like gangbusters hog-wild on a shopping spree.

Programmers (and non-programmers who have experienced "the screaming lock-out") will appreciate this, as the damage to their diskettes due to this problem (as well as the need to turn off the machine to reset it) will be greatly reduced.

NZCPR Application Note

For those installing NZCPR or similar CCP replacements, the OZROM 1E does not require a specific jump to CCP + 35Ch. Although a jump to the third page is still required, it can be anywhere from CCP + 300h to CCP + 3FFh, as the OZROM 1E does not check the low-order byte in the jump address (unlike previous ROMs).

CRC Checking on Disk Access

Previous ROMs did not make use of the CRC error detection capability of the floppy disk controller chip unless you told them to, and it is very likely that you never did, since you probably didn't know about the situation. As a consequence of the CRC error checking built into the FDC routines, you will not get the wrong impression from file copy programs when you ask for verification of the copy (they only verified what they wrote to the destination diskette, because it was assumed that a "successful read" of the source diskette meant no errors.)

THE OZROM 1E MEMORY MAP

The following are some of the memory locations reserved by the OZROM 1E for its own purposes. In all toggles, 0 = OFF, non-zero = ON. Refer to the O1 Technical Manual for a more complete list of reserved memory locations.

EF07	Parallel Printer Control -- 0 = Normal, Non-zero = null device (Double density only)
EF57-8	Current BIOS starting address. May be used to determine start of the O1's special BIOS, regardless of MOVCPM or operating system patches.
EF5C-D	KTBPTR -- Keyboard Pointer. Points to start of keyboard definition table. The OZROM 1E defines this to be FC00h initially, but if FC02 (or [KTBPTR] + 2) changes from its initial value of zero, the OZROM 1E changes the contents of KTBPTR to point to the original table in ROM.
EF5F	Keyboard matrix value of the last key pressed.
EF66	Start of system clock. Hours in 2-digit BCD.
EF67	Minutes in 2-digit BCD.
EF68	Seconds in 2-digit BCD.
EF69	Sixtieths of a second in 2-digit BCD.
EFDF	Time display toggle byte.
EFE0	Keyclick toggle byte.
EFE1	Keyclick counter.
EFE2	Blink toggle byte.
EFE3	Blink counter.
EFE4	Blink rate value. Gives 30/X blinks per sec. Default value of 20 (decimal) for 1.5 blinks/sec.
EFE5	KEYTYP tells whether current key is normal, shift, control, or control-shift key.
EFE6-7	Temporary address pointer used by the keyboard redefinition routines.
EFE8	Clock position value. Clock position is given by F000 + ((value + 43) MOD 256), which can position clock display anywhere on first two display lines.
EFE9	Temporary storage of current character @ cursor position. Used by blink routine.
EFF2-4	Interrupt routine hook. Preset with RET (0C9h).
F000	Beginning of video RAM.
FC00	Start of RAM keycode table. FC00-FC37 -- NORMAL FC38-FC6F -- SHIFT FC70-FCA7 -- CONTROL
FCA8-BF	Reserved for future use.
FCC0-FD7F	Used by the ALARM and TIMER programs. Can be used by the programmer for special interrupt routines (see Interrupt Hook). 192 bytes available.
FD80-FDBF	Temporary storage for bottom video line (64 chars) during special MyKey II reprogramming.
FDC0-F	MyKey II function key pointers.
FDE0	MyKey II cyclic counter. Indicates next function key space to use.
FDE1-FDFF	MyKey II stack space.
FEE0-FFFF	MyKey II function key definition space (8 defs).

THE ROM JUMP TABLE

This table is provided for the programmer's convenience. It is strongly suggested that direct ROM calls be made through this jump table, since actual ROM addresses called by this table may vary according to which version of the OZROM 1E is being used.

```

0100  JMP COLD_BOOT
0103  JMP WARM_BOOT
0106  JMP INPUT_STATUS ;01 KEYBOARD
0109  JMP CONSOLE_INPUT ;01 KEYBOARD
010C  JMP CONSOLE_OUT  ;01 DISPLAY
010F  JMP LIST          ;SERIAL OUTPUT
0112  JMP PUNCH        ;SERIAL OUTPUT
0115  JMP READER       ;SERIAL INPUT
0118  JMP DRIVE_RESET
011B  RET!NOP!NOP      ;NO ROM-LEVEL SELECT DISK
011E  JMP READ
0121  JMP WRITE
0124  JMP READ_TRACK_INFO
0127  JMP READ_SECTOR
012A  JMP WRITE_SECTOR
012D  JMP LIST_STATUS ;STATUS OF SERIAL PORT
0130  JMP SENSE_DISK_DENSITY
0133  JMP ROM_JUMP_1  ;EXIT FROM INT. ROUTINE FOR BANK 1
0136  RET!NOP!NOP      ;THERE IS NO ROM_JUMP_2
0139  JMP FORMAT
013C  JMP INIT_SERIAL
;
; 013F-0156 WERE IEEE-488 ROUTINES
; CALLING THESE FORMER JUMPS RESULTS IN A
; CONTROLLED SYSTEM CRASH (NO DAMAGE TO DISK DATA)
;
; JUMPS FROM 0157 THROUGH 0186 PERFORM SAME
; FUNCTIONS AS DEFINED FOR 01 ROM VERSION 1.4
; (SEE THE 01 TECHNICAL MANUAL FOR DETAILS)
;
; IEEE-488 AS IOBYTE DEVICE 3 NOW NULL DEVICE
;
0187  OR OFFH!RET      ;IEEE PRINTER INPUT STATUS
018A  OR OFFH!RET      ;IEEE PRINTER OUTPUT STATUS
018D  XOR A!RET!RET    ;IEEE PRINTER INPUT
0190  RET!RET!RET      ;IEEE PRINTER OUTPUT
;
0193  JMP PARALLEL_INPUT_STATUS
0196  JMP PARALLEL_OUTPUT_STATUS
0199  JMP PARALLEL_INPUT
019C  JMP PARALLEL_OUTPUT
;
; THE FOLLOWING IS THE HOOK FOR DIRECT FDC COMMAND EXECU-
; TION.  REQUIRES COMMAND IN RWCMD (OEFD1H) AND ANY OTHER
; REQUIRED DISK I/O RAM LOCS (LIKE SAVTYP) DEFINED.
;
019F  JMP READ/WRITE_SEC

```

THE OZROM 1E KEYS

Here are the ROM key definitions. The normal, shift, and control keytables are moved up into RAM at power-up or after the RESET button is pushed. The lower-case "h" after a value indicates hexadecimal.

Normal:	ESC	TAB	0h	0h	0h	CR	'	[
	1	2	3	4	5	6	7	8
	q	w	e	r	t	y	u	i
	a	s	d	f	g	h	j	k
	z	x	c	v	b	n	m	,
	8Ah	8Dh	0	SP	.	p	o	9
	8Bh	8Ch	-	/	;	\	l	=
Shift:	ESC	TAB	0h	0h	0h	CR	"]
	!	@	#	\$	%	^	&	*
	Q	W	E	R	T	Y	U	I
	A	S	D	F	G	H	J	K
	Z	X	C	V	B	N	M	<
	8Ah	8DH)	SP	>	P	O	<
	8Bh	8Ch	_	?	:	;	L	+
Control:	ESC	TAB	0h	0h	0h	CR	'	FOh
	81h	82h	83h	84h	85h	86h	87h	88h
	^Q	^W	^E	^R	^T	^Y	^U	^I
	^A	^S	^D	^F	^G	^H	^J	^K
	^Z	^X	^C	^V	^B	^N	^M	{
	F1h	F4h	80h	SP	}	^P	^O	89h
	F2h	F3h	^_	^	;	^\	^L	'
Ctrl-Shift:	0h	0h	0h	0h	0h	CR	'	^]
	FBh	0h	0h	0h	F9h	^^	FCh	FAh
	D1h	D7h	0h	DFh	F8h	0h	D5h	C9h
	C1h	F5h	C4h	C6h	0h	C8h	CAh	F7h
	DDh	DEh	0h	D6h	F6h	CEh	CDh	0h
	0h	0h	FBh	SP	0h	D0h	CFh	0h
	0h	0h	0h	7Fh	0h	DCh	CCh	0h

Equivalents:

ESC = ^I = 1Bh	^\ = 1Ch
TAB = ^I = 09h	^] = 1Dh
CR = ^M = 0Dh	^^ = 1Eh
SP = space = 20h	^_ = 1Fh
80h through 89h = ^1 through ^9	
8Ah = up-arrow key	
8Bh = right-arrow key	
8Ch = down-arrow key	
8Dh = left-arrow key	

Only the control-shift table cannot be changed. See the control-shift-key notes in this section for additional information.

Notes

APPENDIX D

ROM ERROR MESSAGES

The supplied programs have their own error messages; consult the program documentation if the error message seems to have been generated by the program.

BOOT ERROR -- You are not using an OZROM 1E.

NO SYS! -- You attempted to boot a diskette that had been formatted, but did not have a "bootable" system on it. Either use SYSGEN to put the CP/M operating system on the diskette or use the diskette you meant to use in the first place.

OH, NO! -- This means essentially the same as BOOT ERROR. If you get this once, and only once, each time you boot with the arrow keys, refer to the section on booting with the arrow keys, page 12. Otherwise, you may have a problem with the diskette or the disk drive. First try a different diskette to determine if the problem "is not with your set" but due instead to a bad sector on the system tracks of your diskette.

OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOH, NO! -- Either you left your disk drive door open or you have a real problem. Say a prayer and try a different bootable diskette -- you may be trying to boot an unformatted diskette.

RAM TEST_

UUUU -- This is not a valid error message. If you get this in the middle of your program, then your ROM is defective. If you get it when you turn on your machine, check to see if your TAB key is stuck.

Additional Error Conditions

No error message, but the keys cannot be redefined from the keyboard -- See the Weird Programs section under MyKey, page 6.

No error message, but the program screwed up when ctrl-shift-R was used -- Again, see the Weird Programs section under MyKey, page 6, and contact Micro Management about the program you were using.

Two lines of rapidly-changing garbage -- You have run a program that has crashed, or attempted to make use of one of the IEEE-488 functions which the OZROM 1E does not have.

Notes

APPENDIX E

ASCII CHARACTER CODE TABLE

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

A CONTROL-SHIFT-KEY QUICK REFERENCE GUIDE

Toggles

blinking cursor -- ctrl-shift-B
keyclick -- ctrl-shift-K
scrolling -- ctrl-shift-S
time display -- ctrl-shift-T

With the Osborne Screen-Pac

52-column display -- ctrl-shift-5
80-column display -- ctrl-shift-8
104-column display -- ctrl-shift-1 or ctrl-shift-0

With Parallel Printer -- Double Density only

Assign printer to null device -- ctrl-shift-esc
Re-assign printer -- ctrl-shift-7

With OZPATCHed system

Print screen, 80 x 24 -- ctrl-shift-P
Print screen, 128 x 24 -- ctrl-shift-W
Send next char to printer -- ctrl-shift-O, followed
by character
Query ^P printer echo (beeps if on)
-- ctrl-shift-Q

Key Redefinition

Exchange key definition with another -- ctrl-shift-X,
followed by two keys: First key is key to be redefined,
second is new key definition.

Redefine key to be one of eight MyKey II functions --
ctrl-shift-R, followed by key to be function key, followed
by function (up to 63 characters), followed by ctrl-shift-R.
Use ctrl-shift-X to restore original key definition.

Restore original definition -- ctrl-shift-X, key, key.
Control-shift keys cannot be redefined.

Other

Clear Osborne 1 screen -- ctrl-shift-Z
Generate 1E hex (^ ^) -- ctrl-shift-6
Generate 1D hex (^ |) -- ctrl-shift-[
Generate 7F hex (DEL) -- ctrl-shift-/

INDEX

- ALARM program 15
 arrow keys, booting with 12
 with control key 2
 with FK program 22
 with shift key 5
 ASCII keycodes ... 3,C-7,E-1
 autostarting w/MEMORIZE . 14
 backspace 7
 Bdos errors 11,C-4
 beep length 1
 BIOS 6,17,C-3
 blinking cursor 3,8
 block cursor w/MyKey II .. 7
 booting 12
 BOOT ERROR D-1
 caps-lock effect 5
 clear-screen 9
 clock 9
 clock accuracy 15
 control -arrows 2
 keyboard 4
 of system parameters .. 8
 -shift keys 2
 cursor, blinking 3
 cursor, block 7
 CRASH program 9,16
 crash protection 9,C-4
 CRC error checking C-4
 dBase function key patch A-3
 decimal-hex-ASCII table E-1
 display control codes 11,C-2
 Dvorak 5
 80-column 2,8
 error messages D-1
 exit MyKey II 7
 FAST program 12
 features of OZROM 1E 1
 52-column 2,8
 FK program 19
 floppy disk operation C-3,-4
 formfeed to printer 9
 function keys, dBase II A-3
 MyKey II 7
 original 19
 guide, quick reference . E-2
 hex-ASCII table E-1
 IEEE-488 functions ... 1,C-1
 installation 2,B-1
 interrupt "hook" C-2
 jump table C-6
 keyboards 4,5
 keyclick 8
 key redefinition 4-7
 key repeat 1
 key, restore 5
 keys, control-shift .. 2,C-7
 keytable C-7
 MyKey 4-6
 MyKey II 5,6,7
 MEMORIZE program 5,6,7,13,14
 memory map C-5
 memory tester 10,11
 modem operation 11,C-1
 normal keyboard 5
 NO SYS! error message .. D-1
 NZCPR C-4
 OH, NO! error message .. D-1
 original key restore 5
 104-column mode 8
 OZPATCH 8,17
 parallel port 8
 printer control . 8,9,17,C-5
 printer lockout 8
 printscreens 8
 programs, support 13
 programming
 MyKey II function keys 8
 keyboard redefinition 4-8
 O1 function keys 19
 quick reference guide .. E-2
 QWIKKEY 5
 RAM test 10
 reference, quick E-2
 RESET 6,10,16
 restore original key 5
 Screen-Pac 8,E-2
 screen positioning 2
 screen print 8
 scrolling, horizontal 8
 vertical 2
 SET program 27
 shift keyboard 5
 SMARTKEY program 5
 smoother, faster WS A-1
 SuperCalc 4,6
 support programs 13
 system parameter control . 8
 technical information .. C-1
 Televideo emulation C-2
 test, memory 10
 time display toggle 8
 TIME program 14
 TIMER program 15
 toggles 2,8,E-2
 1200-baud modem 11,C-1
 vertical scrolling ... 2,C-1
 video control codes 11
 weird programs 6
 WordStar 1,3,8
 smoother, faster ver. A-1

