THE MICRO WORKS

# CBUG MONITOR

# OWNER'S MANUAL

# CBUG MONITOR
# OWNER'S MANUAL

## TABLE OF CONTENTS



## COPYRIGHT NOTICE



## WARRANTY STATEMENT

CBUG monitor is provided as is without warranty. Reasonable care has been taken to insure that the program operates as described in this manual. If you find discrepancies or errors, please notify us. We will attempt to correct any errors brought to our attention, however, we make no guarantee to do so.

## INTRODUCTION TO THE CBUG MONITOR

This monitor comes either on a cassette tape or in a 2K ROM. The same program is used in both media, as it is entirely relocatable (position-independent) code. Except where specifically noted, this manual refers equally to both products.

If CBUG is supplied in a 2716 EPROM, it should be installed in either the Extended Basic socket to the left of the Basic ROM on the main board, or else in an outboard ROM pack. If installed in the main board, it is executed with the command EXEC 36864 (which is $9000 in hex), and if it is in an external ROM pack it can accessed with EXEC 53248 (which is $D000 in hex). See page 8 for installation.

The cassette tape version of the monitor should be loaded with the CLOADM command. It will load at $0600 and wipe out any Basic program that may have been there. By typing EXEC, the monitor is cold started and it will automatically move Basic's pointers so that new Basic programs will be loaded just beyond the end of the monitor and the monitor can be restarted any time without affecting Basic programs. High-resolution graphics under Basic control should not be used with CBUG loaded at $0600.

The tape may be offset-loaded, and it will run wherever it is loaded. Typing EXEC will run it and readjust Basic's pointers as usual; to avoid changing the pointers, CBUG may be run by jumping to the address twelve bytes beyond where it was loaded.

## RUNNING THE MONITOR

When the monitor issues its prompt, a single-letter command may be typed, along with any parameters needed. With the exceptions listed below, parameters are in hex and any non-hex character (including backspace) will cause the command to abort and return to the monitor prompt level. Commands are shown as they appear on the screen, with spaces between the parameters, but these spaces are added by the monitor and should not be typed in by the user.

The following are the commands:

G Go back. Will return to Basic unless the monitor was entered from some machine-language program. The registers, including the PC, are loaded from the register list as displayed by the R command. (See section on stack frame for more details.)

R Display register list. Shows what was in the registers when the monitor was entered, and what will be loaded by G or J commands. (See section on stack frame for more details.) The condition codes register is listed not in hex but in binary, as each of its bits has meaning. A letter indicates a one bit, and a dash a zero bit.

M Memory examine & change. "M 1234" will display a line of eight bytes, with the cursor by address 1234. The cursor may be moved up, down, left, or right with the arrow keys (or the space bar) to display more memory. Typing hex numbers will enter data into memory. A carriage return will exit the command. Another "M" has the same effect as a carriage return followed by an "M". Inverted numbers indicate a write to an area where there is no RAM.

This command is the primary tool for working with machine language.

I Insert hex to memory. "I 1234 2345 67" will set locations 1234 through 2345 (inclusive) to hex 67.

T Transfer block of memory. "T 0123 1234 2345" will copy the contents of locations 0123 through 1234 (inclusive) to the area of memory starting at 2345. Note that the format of both the "I" and "T" commands could be called "FROM - THRU - TO".

J Jump to machine-language subroutine. The registers A, B, X, Y, and U are loaded out of the register list shown by the R command, and a JSR instruction is done to the location specified by the parameter. "J 1234" will jump to location 1234, and when an RTS is encountered control will return to the monitor.

C Change register list. This is followed by a single-letter parameter which designates which register to change. Admissible register names are A, B, X, Y, U, D (direct page), C (condition codes), and P (program counter). This transfers control to the memory examine / change function ("M") at the address on the stack where the specified register is stored.

S Save to cassette. "S 1234 2345 1357 MYFILE" (followed by a carriage return) will write a file on cassette with the name MYFILE and containing the data from 1234 through 2345 inclusive. 1357 is the transfer address which is loaded into the EXEC pointer. The tape is loaded with Basic's CLOADM command and if an EXEC is then done control will transfer (in this example) to location 1357.

B Set baud rate. It is followed by a valid baud rate (0110, 0300, 0600, 1200, 2400, 4800, or 9600) and the single letter P or C. To use the comm link for upload/download, Auto mode, or with the X command, "C" should be used; "P" is for setting a baud rate for use with a printer under Basic. (The baud rate may not exceed 2400 with the "P" option.)

L Load hex. This puts data into memory without the formatting done by the "M" command. "L 1234 23 45 67 89" and a carriage return to exit will set memory starting at 1234 to the values 23, 45, 67, and 89.

Convert hex to decimal. "S 1234" will print out 4660, which

is the decimal equivalent.

Convert decimal to hex. ". 12345" (carriage return) prints out 3039, which is the hex equivalent. Unlike most parameters, the decimal number may be any number of digits (so long as the result fits in two bytes) and therefore must be terminated by a carriage return.

Move display page. Any 512-byte block of RAM may be displayed on the screen. Normally the block starting at 0400 is displayed. (Prove this to yourself by typing "L 0400 11 11 11 11" or something similar.) Now type "P 0000". The upper left corner of the display will flash furiously; more on that in a second. Type carriage return, and the display will change abruptly. You are now actually looking at memory starting at location zero. You will notice one byte changing rapidly; this is the counter which keeps track of the color of the cursor. The monitor is running normally, but you just can't see the screen. To restore order, type "P" again and the display will be back where it belongs (at 0400); now either type a carriage return or another address. If you want to watch the stack, type "P 3FFF" (or "P 0FFF" on a 4K machine).

To look at a particular byte of memory (say, to watch a variable at location 1234) type "P 1234" and note carefully the spot on the screen that is going nuts. When you hit return, that spot will display the variable.

Upload. The baud rate should first be set (for example, "B 1200 C"). U 1234 2345 will send to the comm link and to the screen the contents of 1234 through 2345 inclusive, in Motorola's checksummed format. See page 7.

Download. The baud rate should first be set. If 9600 baud is used for download, there should be at least two stop bits (as with any software UART) to allow the software to process the character after it is read. Pressing any key on the keyboard will abort the operation. Data is expected on the comm link in the same format as it is dumped by Upload. If no cursor appears when "D" is first pressed, this means there is a break condition on the RS232 input. This can sometimes be caused by a joystick pushed all the way to one side.

Take over software interrupt. Until this command is executed, it is undefined what will happen if a SWI instruction ($3F) is encountered by the 6809. This instruction sets the vector so that control will return to the monitor. Machine language debugging may then be accomplished by inserting SWIs into the program, at which point the monitor will be entered and the register contents dumped. You may now change the registers ("C" command) or examine or alter memory ("M" command), after which execution may be resumed at the point after the SWI by typing "G". An SWI may be put in place of an existing instruction in RAM and, upon its execution, the instruction put back and PC decremented by one before typing "G".

CBUG makes certain calls to the Basic ROM which assume that the DP register contains zero. CBUG should not be entered (via SWI or otherwise) with DP not zero.

AU Auto mode. Once auto mode is entered, only a reset will remove it. The baud rate must have been set. Commands are entered not from the keyboard but from the comm link. The response from the monitor which normally goes to the screen is suppressed. All keypresses are sent on the comm link. This mode is useful for using the computer as an intelligent terminal connected to a host system.

Terminal Mode. This causes the computer to emulate a CRT terminal. The baud rate must have been set to 110. Two forms of the command, X F and X H stand for full- and half-duplex mode respectively; that is, without and with local echo. Lower-case letters are sent as control characters, since the ability to send control characters is necessary on many timesharing systems but lower-case letters are not. The break key will generate a line break, and shift-backarrow will send an escape ($1B). The only escape from terminal mode is by typing control-M (shift-O, M, shift-O). The speed is limited to 110 baud; this is so that the the display processor will have time to scroll on line feeds before the next character starts coming in. Higher baud rates may be used if the sending device pauses after any character which causes a scroll.

Reset. This causes the computer to be reset and return to Basic.

## THE STACK FRAME

When the monitor is entered, either from Basic or an SWI (see the "I" command), the registers are saved on the stack. (When the monitor is first cold started, some registers may be changed before being stacked, but subsequent entries will not change anything.) This stack frame is what is displayed by the "R" command. The "G" command is really an RTI instruction, which automatically loads all the registers off the stack. The operation of the stack frame can be seen by this experiment: Type "R". Now restart the monitor by typing "J 060C" (for tape) or "J 900C" (for inboard ROM) or "J C00C" (for outboard ROM). Type "R" again and the stack pointer will have decreased by 000C. This is because a new stack frame was made when the monitor was reentered. These may be changed by the "C" command. Now type "G". We are still in the monitor, but back to the first stack frame; any changes made to the register list are no longer there. Another "G" should return to Basic. This is worth thinking about, anyway.

## Jump table

Jumps to various routines are included in the beginning of the monitor. References by programs should be made to these rather than into the body of the monitor code so as to insure compatibility with future releases. The addresses shown in the table below are relative to the starting address of the monitor; each jump takes 3 bytes.

0000 9 bytes - Basic tokens in tape version; NOPs in ROM version.
0009 Hard start - fixes Basic pointers, prints banner, then Firm start.
000C Firm start - builds stack frame, clears Auto mode, then Warm start.
000F Interrupt entry - assumes stack frame exists, dump registers and do Warm start.
0012 Warm start - save stack pointer and go execute commands
0015 Inboth - looks for input from either keyboard or comm link. Keyboard input will have bit 7 set, comm input bit 7 clear.
0018 Inney - Keyboard input. Returns with bit 7 clear.
001B Echo - Inney, then Outey.
001E Outey - Output to screen. The control characters CR, LF, and BS are interpreted as they would be on a CRT.
0021 RS232O - Output 8 bit byte to comm link.
0024 RS232I - Input 8 bit byte from comm link.
0027 Outhex - Print 2 digit hex number to screen. Number is found in location 00FB. This is useful in Basic: POKE 251,N : EXEC 36903 (or other address depending on where the monitor base address is).
002A Print data - Will print the data string (using Outey) whose address is in the X register. The operation is ended by a null in the string.
002D Out decimal - Prints the decimal value of the number in the D register (A:B) to the screen with left zero supression. The value zero will produce no output.
0030 Out2h - This prints a one byte hex value to the screen which is pointed at by X, incrementing X. It is the version of Outhex designed to be called by machine-language programs.

## Variable table

The following variables are the only ones of interest to the user and therefore the only ones which can be counted upon for future compatibility. They are in an area of page 0 which the Basic interpreter does not use.

param - $00FB - 251 - the parameter to the Outhex function.

mode - $00FC - 252 - if nonzero, the monitor will be in auto mode (under control of comm link).

upf - $00FD - 253 - upload flag. If nonzero, Outey will also send to the comm link.

NOTES ON THE DIFFERENCES BETWEEN THE TAPE VERSION
AND THE ROM VERSION OF CBUG

The listing shown in this manual is for the tape version of CBUG. The same source program is used for both, with the differences accomplished through conditional assembly. Both versions are the same size, so addresses shown in the listing apply to both versions. The differences are as follows:

(1) The first nine bytes of the tape version are the tokens for the Basic program "10 EXEC". In the ROM version, these bytes are NOP's ($12) so that the monitor may be executed by jumping to its first byte. The tape version must be started by jumping to one of the branch instruction which are after the first nine bytes.

(2) The first jump, LBRA HSTART, jumps to two different places. In the tape version, it jumps to a routine which does a Basic "New" statement to reset the pointers around the monitor. In the ROM version, it jumps around that routine as it is not necessary (although the code is still there).

To make a cassette version out of a ROM version, you may do the following:

(1) Move the monitor down to 0600 using the "T" command.

(2) Replace the Basic tokens in the first nine bytes, referring to the listing in this manual.

(3) Change the first jump, again referring to the listing.

(4) Save the program to cassette, using the "S" command.

(5) In moving and saving the monitor, be sure to include the last byte, which is a zero. This byte is necessary for Basic after doing the "New" command.

S1 FORMAT

The format used for sending data in the Upload and Download commands is as follows:

S1CCAAAAdddddd ... ddKK

where

S1 is the header for each record;
CC is a two-digit hex number indicating how many bytes are in the record (counting two for the address and one for the checksum, but not counting the count itself);
AAAA is a four-digit hex address where the data is stored;
dd... is a series of two-digit hex numbers which are the data to be stored;
KK is a two-digit hex number which is the checksum.

All bytes, including the count, address, data, and checksum, but not counting the S1, are added together (ignoring carry). The sum should be $FF. If CBUG detects a checksum error on download it will print a question mark and return to command mode.

Example record:

S1071000FF00FF00EA

This would load the four bytes FF, 00, FF, 00 into memory starting location 1000. The count (07) includes four data bytes, two address bytes, and the checksum. The checksum is correct since

07 + 10 + 00 + FF + 00 + FF + 00 + EA = xxFF

A series of records of the above type are sent by Upload and expected by Download. They are separated by carriage return / line feed for readability only. The Download command may be terminated by sending the end-of-file record:

S9

This record is not sent by the Upload command.

Locate and remove the Phillips head screw under the label in the middle of the top of the ROM pack. Pry the pack open from the connector end.

4. Remove the PC card. Using a solder sucker or solder wick, remove the solder from the 24 holes corresponding to the empty ROM space.

5. Install a 24-pin low profile socket in the holes that were just cleared in step 2. Make sure that the pin 1 orientation mark is in the same direction as pin 1 on the Tandy ROM.

6. Cut the trace to the Tandy ROM's pin 20, about 1/2 inch away from the chip. This is the chip select. Pin 20 of a ROM must be pulled high to disable it. If you intend not to use the Tandy ROM, simply wire its pin 20 to the +5 volt trace as in figure 1. Otherwise follow step 5 to install a switch.

b. To be able to switch between the two ROMs, use a double-pole double-throw switch as in figure 2. The chip-select signal which originally went to both pin 20s must be switched between them, while the other pin 20 must be connected to +5 volts.

6. Cover with tape the trace shown in figure 1. This trace causes the computer to jump to the ROM pack on power-up; covering the trace (or drilling out the plate-thru hole) will allow the computer to start in Basic.

7. Install CBUG in the socket, replace the PC card in the plastic case and tighten the screw. Plug the pack into the Color Computer ONLY WITH POWER OFF, turn on the computer and EXEC 53248.

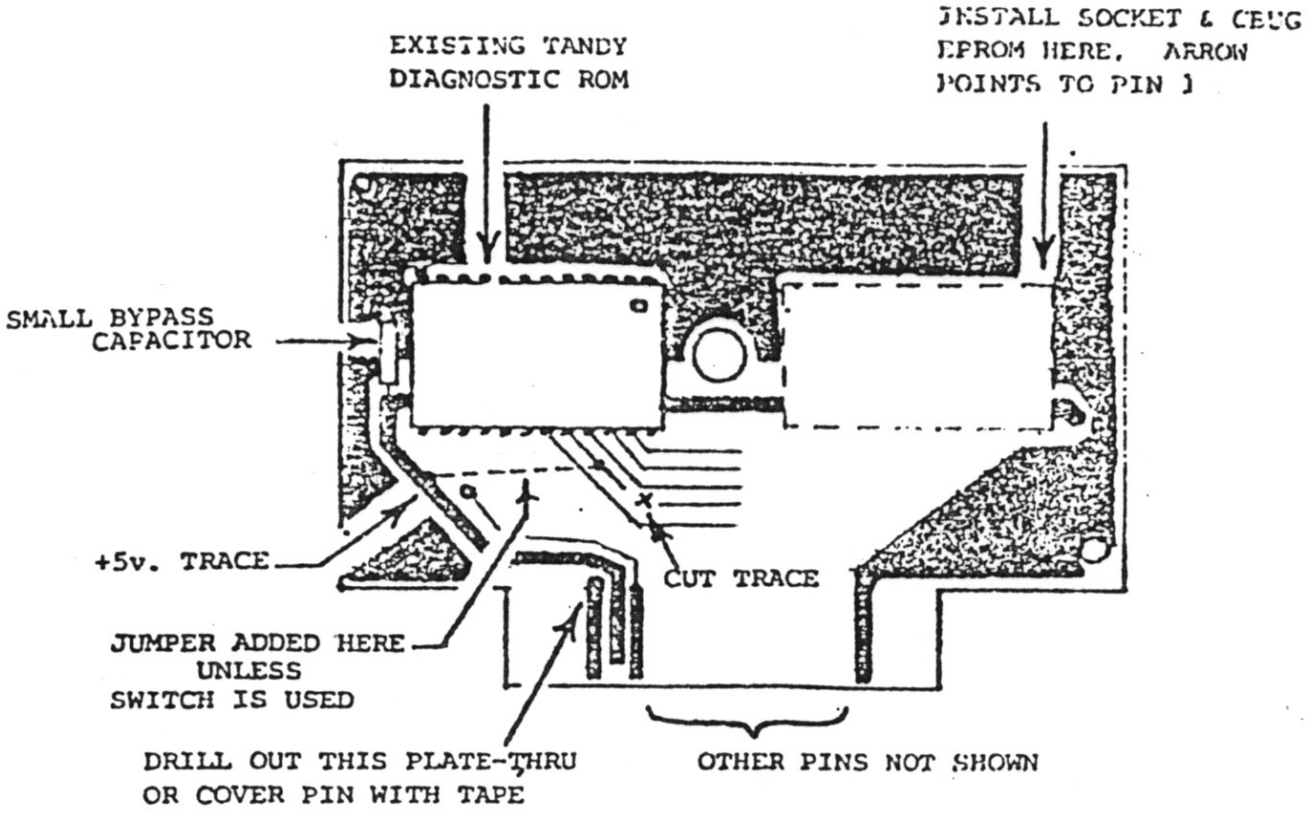
EXISTING TANDY
DIAGNOSTIC ROM
INSTALL SOCKET & CBUG
EPROM HERE. ARROW
POINTS TO PIN 1
SMALL BYPASS
CAPACITOR
+5v. TRACE
CUT TRACE
JUMPER ADDED HERE
UNLESS
SWITCH IS USED
DRILL OUT THIS PLATE-THRU
OR COVER PIN WITH TAPE
OTHER PINS NOT SHOWN

FIG. 1

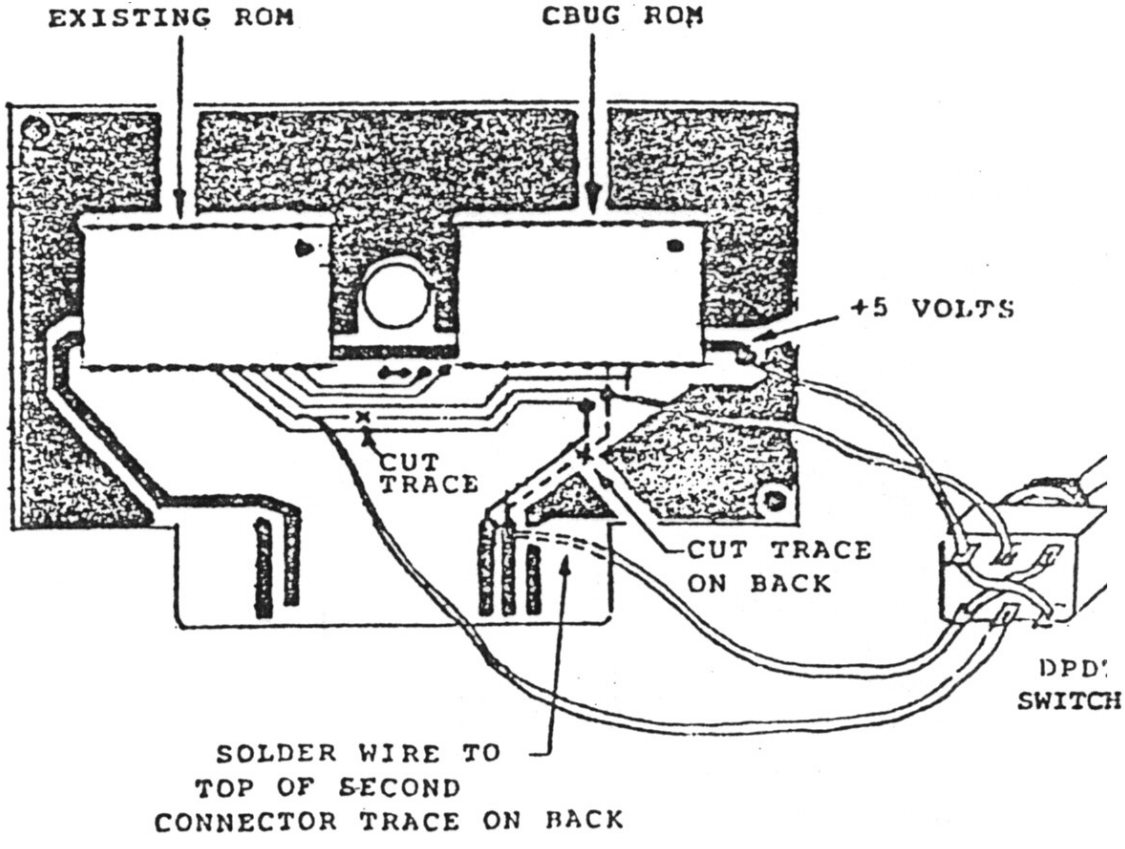
EXISTING ROM
CBUG ROM
+5 VOLTS
CUT
TRACE
CUT TRACE
ON BACK
DPDT
SWITCH
SOLDER WIRE TO
TOP OF SECOND
CONNECTOR TRACE ON BACK

FIG. 2

## NOTES ON HI-RES GRAPHICS ON THE COLOR COMPUTER

The Micro Works CBUG monitor gives the user a chance to investigate many features of the color computer, including Hi-res graphics. The following notes may be of some use to anyone experimenting with this.

The display at any point on the screen depends on three things: The byte at the address corresponding to that picture element (or "pixel"), a byte at location $FF22, and three bits of information about the size of the block to display which are controlled by writing to $FFC0 through $FFC5. Also, the address of the start of the display block is controlled by $FFC6 through $FFD3 and may be changed with the "P" command in CBUG.

To set the three size bits (or any of the registers in the range $FFC0 through $FFDF), write to an odd address to set a bit and to an even address to clear a bit. The data written is ignored; only the address is important. Smaller address pairs are mapped to the less significant bits. For example, to write a 001 (base 2) to the size bits, write data to $FFC1, $FFC2, and $FFC4.

The area displayed on the screen may be 1/2 K, 2K, 3K, or 6K, depending on the size bits. Each byte may represent one, four, six, or eight pixels, of eight, four, or two colors, depending upon the upper five bits at $FF22. For example, the most significant bit at $FF22 is the "Full Graphics" bit and enables various graphics modes determined by the other bits.

And remember: Once you change any bits, you won't be able to see what you are doing until you reset (which resets everything), so be familiar with the CBUG commands and remember where you are. Have fun!

## NOTES ON COMPUTER GENERATED MUSIC

The Color Computer contains a six-bit DAC at location $FF20, the output of which may be directed to the audio of the TV set. Using this it is possible to play music which sounds much better than that which can be obtained using Basic's SOUND command; in fact algorithms exist for playing four-part harmony. For those who wish to experiment with this output, the following is an example of machine-language sound:

```
CBUG:M 0700
0700  86 3F B7 FF  23 1F 89 F.
0708  FF 20 12 12  12 5C 26 F7
0710  4C 2A 01 4F  20 EF 3F 3F
CBUG:J 0700
```