

Marinchip 9900 Debug Monitor  
User Guide  
by John Walker

(C) Copyright 1978 Marinchip Systems  
All Rights Reserved

Revised July 1978

**Marinchip Systems** 16 St. Jude Road  
Mill Valley, CA 94941 ■ (415) 383-1545

Table of contents

1.	Introduction	- 2
2.	Using the monitor	- 3
2.1.	Entering the monitor	- 3
2.1.1.	ROM-based monitors	- 3
2.1.2.	Disc-based monitors	- 3
2.2.	Monitor console handler	- 3
2.2.1.	Delete character - Backspace (Control H)	- 3
2.2.2.	Delete line - Cancel (Control X)	- 3
2.2.3.	Retype line - Control R	- 3
2.3.	Monitor commands	- 3
2.3.1.	A - Dump memory in ASCII	- 3
2.3.2.	B - Boot from external storage	- 4
2.3.3.	G - Go to address	- 4
2.3.4.	H - Dump data in hexadecimal	- 4
2.3.5.	L - Load memory	- 4
2.3.6.	P - Patch memory	- 5
2.3.6.1.	Null - display next address	- 5
2.3.6.2.	< - display previous address	- 5
2.3.6.3.	<number> - patch memory	- 5
2.3.6.4.	+<number> - add number to memory	- 5
2.3.6.5.	-<number> - subtract number from memory	- 5
2.3.6.6.	><number> - go to new address	- 5
2.3.6.7.	Control C - terminate command	- 5
2.3.6.8.	Sample Patch command use	- 6
2.3.7.	R - Read from CRU	- 6
2.3.8.	W - Write to CRU	- 6
2.3.9.	T - Terminal emulation	- 6
2.3.9.1.	Memory load format	- 7
2.4.	Interrupting a command	- 7

## 1. Introduction

The Marinchip 9900 Debug Monitor is a small (less than 2K bytes), ROM-based monitor that provides the functions performed by a hardware front panel in traditional minicomputer systems. Since the user interacts with the Debug Monitor via an interactive display terminal rather than switches and lights, it can provide a much more convenient interface for machine language debugging. The most unique feature of the monitor is its ability to receive a memory load down a communication line from a remote development system. This allows very small M9900-based systems to use other larger systems for their software development. In disc based systems, the Debug Monitor may also be used as a conventional program under the operating system. This mode of operation avoids the need for ROM to hold the Monitor.

## 2. Using the monitor

### 2.1. Entering the monitor

The Debug Monitor can be used either as a ROM-based monitor or as a program under a Marinchip operating system. The monitor is entered differently depending upon how it is being used.

#### 2.1.1. ROM-based monitors

When power is applied to the M9900 CPU, it will automatically execute a LOAD function, which will place the Debug Monitor in control. When the monitor is entered, it will respond with the message:

Marinchip 9900 Monitor.

and prompt the user for a command with a period (.). The monitor can be re-entered at any time by pressing the RESET switch on the front panel of the computer. This will halt any executing program, reset the computer and any peripherals connected to it, and place the Debug Monitor back into control. It will respond with the standard sign on line, described above.

#### 2.1.2. Disc-based monitors

A Disc-based monitor is called from operating system command level simply by typing its name, "DEBUG", when the operating system command prompt appears. The operating system will load the Debug Monitor and transfer control to it. The Debug Monitor sign on line and prompt will appear as described above for the ROM-based version of the monitor.

## 2.2. Monitor console handler

The Debug Monitor contains a handler for the system console. It echoes the characters typed by the user back to the console, and assembles characters into lines. The line is the unit processed by the Debug Monitor. A line of input is sent to the monitor by pressing the RETURN key. Before RETURN is typed, several keys may be used to correct errors in the information typed in. The available editing keys are described below. Any character other than RETURN or one of the special characters described below will be echoed to the console and placed in the input buffer. ASCII control characters will be echoed to the console as an up-arrow (^) followed by the corresponding letter on the keyboard.

2.2.1. Delete character - Backspace (Control H)

The Backspace key (Control H on some terminals) will erase the last character typed. It will move the terminal cursor back one character so that the erroneous character may be corrected. Successive depressions of the Backspace key may be used to rub out as many characters as desired.

2.2.2. Delete line - Cancel (Control X)

The Cancel key (Control X on most terminals) will erase the entire line entered by the user. The characters ^X will be typed to indicate that the line was deleted, and the console will advance to a new line so that the user can re-enter the input from the start.

2.2.3. Retype line - Control R

The Control R key will retype the current input line as it stands, and leave the carriage extended at the end so that further characters may be added or more corrections may be made. This allows users of hard-copy terminals to see the current contents of the input line when many backspaces and overtypes have obscured the printed copy.

2.3. Monitor commands

When the monitor prompt appears, the user may enter any monitor command. After the command has completed, the monitor prompt will appear again. All monitor commands are single characters. Any parameters required by the command should be separated from the command character by one or more spaces. All input to the monitor may be either upper or lower case. The following paragraphs describe the commands available in the monitor.

2.3.1. A - Dump memory in ASCII

A <start>, <length>  
or: A <start>: <end>

The ASCII dump command dumps memory words in ASCII. Each word is dumped as two characters representing the high order and low order bytes in the word. If a byte is an ASCII control character, an up-arrow followed by the corresponding letter will be typed. In the first format, the two parameters are the start address and the number of words to dump. In the second format, the first parameter is the start address, and the second parameter is the ending address. Both parameters are hexadecimal numbers. If only one parameter is supplied, only one word will be dumped. If the <start> address is odd, it will be rounded down to the previous even address. Examples of the "A" command are:

A 12C0  
12C0: Th

A 12C0,6  
12C0: Th is i s it ^A^M

A 12C0:12C4  
12C0: Th is i

If the word count exceeds 16, the dump will be continued onto as many extra lines as are required, with 16 words dumped per line. The dump may be terminated by typing Control C (see the section "Interrupting a command" below).

# Marinchip 9900 Debug Monitor User Guide

## 2.3.2. B - Boot from external storage

This command causes the Debug Monitor to transfer control to the Marinchip Boot ROM, which loads the operating system from external storage. The actual loading is not part of the Debug Monitor, but the "B" command is included to activate it for convenience. When using the operating system, bringing up the system consists simply of applying power to all components, inserting the system disc in the proper drive, and typing the "B" command.

## 2.3.3. G - Go to address

G <address>

The Go to command transfers control to a program stored in memory at the specified address. The program is given control with its own set of workspace registers. If <address> is omitted, the <address> used on the last Go to command will be used.

## 2.3.4. H - Dump data in hexadecimal

H <start>, <length>  
or: H <start>: <end>

The Hex command dumps memory words in hexadecimal. In the first format, the two parameters are the start address and the number of words to dump. In the second format, the second parameter is the last address to dump. Both parameters are hexadecimal numbers. If only one parameter is supplied, only that single word will be dumped. If the <start> address is odd, it will be rounded down to the previous even address. Examples of the Hex command are:

```
.H 340
0340: 14BC

.H 340,5
0340: 14BC 10FF 1687 0640 5F20

.H 340:342
0340: 14BC 10FF
```

Eight words will be dumped per line. The dump will be continued onto as many lines as necessary. The dump may be terminated by typing Control C (see the section "Interrupting a command" below).

## 2.3.5. L - Load memory

L <address>

The Load command will prompt the user with the <address> at which data is to be loaded. The user may then enter hexadecimal data, separated by commas. Numbers scanned will be stored in successive word locations. ASCII characters may be entered by enclosing either one or two characters in double quotes (""). A single quoted character will be right justified, while two characters will occupy one word, with the first character in the high byte and the second character in the low byte. If the input line ends with a trailing comma, the user will be re-prompted with the next address to be loaded, and the data entry may continue. Leaving off the trailing comma will end the Load command. Some typical uses of the Load command are as follows:

```
.L 1240
1240: 123,2356,12CE,
1246: 18CF,BE04

.L C010
C010: "Th","is","i","s","te",
C01A: "xt",FFFF
```

## Marinchip 9900 Debug Monitor User Guide

The Load command is provided only in the disc-based version of the Debug Monitor. In ROM-based versions, the Patch command (see below) may be used in place of the Load command.

### 2.3.6. P - Patch memory

P <address>

The Patch command allows convenient interactive display and modification of the contents of memory. The <address> parameter specifies the first address to be examined. The Patch command will respond with the current address, followed by a colon, and the contents of the location in hexadecimal. The carriage will be left extended following the current value so that the user can respond. Valid responses are as follows:

2.3.6.1. Null - display next address

Simply typing return in response to the prompt will leave the current location unmodified and display the next address and its contents. An area of memory can be displayed by successive depressions of the return key.

2.3.6.2. ^ - display previous address

An up arrow character (^) followed by a return will cause the previous word address and contents to be dumped.

2.3.6.3. <number> - patch memory

Entering a hexadecimal number, or a one or two character string enclosed in double quotes ("), will replace the contents of the current location with the value entered. The next location and contents will be displayed. If desired, more than one number may be entered, separated by commas. The numbers will be stored in successive words, and the location following the area modified will be displayed on the next prompt.

2.3.6.4. +<number> - add number to memory

The number (which can be one or two ASCII characters enclosed in double quotes) is added to the contents of the current memory location. The location is then retyped as modified.

2.3.6.5. -<number> - subtract number from memory

The number is subtracted from the current location, and the current location is retyped following the subtraction.

2.3.6.6. ><number> - go to new address

The Patch command continues at the address specified by <number>. This may be used to patch a totally different area of memory than that specified by the <address> on the initial command.

2.3.6.7. Control C - terminate command

Typing Control C will terminate the Patch command and return to the Debug Monitor command level. The command prompt will reappear.

2.3.6.8. Sample Patch command use

The following is a typical use of the Patch command. Note that the initial address and contents is typed by the Debug Monitor, and the information following it is entered by the user.

```
P 3000
3000: 1820
3002: 10FE
3004: 0204 460
3006: 5F20 3D0, 380
300A: C1D0 ^
300B: 0380 >5000
5000: 1635 0
5002: 5524 ^
5000: 0000 F"a"
5000: 0061 ^C
```

The user enters the Patch command. The first location is dumped. The user simply presses return. The next location is dumped. The user again presses return. The next location is displayed. The user enters the hex value "460". That value is placed in 3004, and the command advances to the next location. The user overlays two words starting at 3006. The user wishes to see the last location patched. It is displayed. The user decides to patch at address 5000. The user zeroes 5000. The user backs up to 5000. ... and adds lower case "A". Just to see what it is. The patch is accomplished, the patch is terminated by Control C. The monitor prompt reappears.

2.3.7. R - Read from CRU

R <start bit>, <bit count>

Data is read from the Communication Register Unit (CRU) I/O bus starting at <start bit>. The number of bits to be read is specified by <bit count>. Note that both <start bit> and <bit count> are hexadecimal numbers. If <bit count> is omitted, a single bit will be read. The information read will be dumped in hexadecimal on the console. If <bit count> is 8 or less, the information will be dumped in the top two digits of the four digit hexadecimal value dumped, otherwise it will appear right justified in the four digit value.

2.3.8. W - Write to CRU

W <data>, <start bit>, <bit count>

The specified <data> is sent to the Communications Register Unit starting at <start bit>, and continuing for the number of bits specified by <bit count>. If the <bit count> is 8 or less, the data to be sent should appear in the upper byte of a 16 bit quantity. If <bit count> is omitted, 1 will be assumed. If <start bit> is omitted, 0 will be assumed.

2.3.9. T - Terminal emulation

T

This command may be used only in systems equipped with two serial I/O interfaces, and assumes that the second interface is connected to a serial data link to another computer system, called the host. After entering the "T" command, the Debug Monitor will act as a transparent link between the user terminal and the host system. Characters typed by the user on the terminal will be buffered and sent to the host, and characters received from the host will be displayed on the user terminal. It is assumed that the host echoes the characters typed by the user back to the user terminal. If it does not, the user terminal must be placed in local echo (half-duplex) mode. The only exception to the complete transparency of data transmission is that the Control B key on the terminal may be used to

## Marinchip 9900 Debug Monitor User Guide

escape from the terminal emulator back to debug monitor command mode. As a result, the Control B character may not be sent to the host system.

When in this mode, the host system can send a memory load down-line to the Debug Monitor by sending the sentinel:

```
//LOAD MEMORY
```

followed by encoded information which specifies the locations to be loaded and the data to be placed there. At the completion of the memory load, the loaded program may be placed into execution directly, or control may be returned to the Debug Monitor.

### 2.3.9.1. Memory load format

The following is the format of the information received by the Debug Monitor during a remote memory load. Marinchip software permits generation of information in this format. This information is provided to aid implementors of cross-development tools on other systems.

Information sent during memory load consists of a string of 8-bit ASCII characters. Each byte received is decomposed as follows:

```
PTTTXXXX
```

where P is the parity bit, which is ignored, TTT is the type of this byte (see table below), and XXXX is a four bit information field whose interpretation depends upon the TTT field. During the execution of a memory load, an 8 bit accumulator and a 16 bit memory pointer register are maintained. The description of the TTT field below will make reference to these two registers.

TTT	Meaning
0	Format control. Ignore byte.
1	End load. Return to monitor
10	Clear accumulator, put XXXX in the upper 4 bits.
11	Store accumulator + XXXX field into memory at memory pointer.
100	Increment memory pointer by 1.
101	Set high byte of memory pointer to accumulator + XXXX.
101	Set low byte of memory pointer to accumulator + XXXX.
110	Jump to memory pointer address.
111	Delete code. Ignore byte.

Since codes with TTT fields of 000 and 111 are ignored, the memory load handler is insensitive to those ASCII codes that timesharing systems are fond of inserting into output text without being asked. These ignored characters include carriage return, line feed, form feed, null, and delete.

### 2.4. Interrupting a command

Whenever the monitor is typing output on the terminal, or accepting input, pressing the Control C key will interrupt the monitor and return control to the command processor. The command prompt will appear, and the user may enter any Debug Monitor command. The Control C key may be used to stop a Dump command, or to escape from a Patch or Load command. Note that the Control C does not interrupt the monitor when it is emulating a terminal (following the "T" command), as Control C, like all other characters, is passed through transparently to the host system).