



**MOTOROLA**

M68DOWNLD(D1)

**M6800/M6801/M6809  
DOWNLOAD PROGRAM**

**User's Guide**

**SYSTEMS**

**MICROSYSTEMS**

M68DOWNLD (D1)

AUGUST 1979

M6800/M6801/M6809

DOWNLOAD PROGRAM

USER'S GUIDE

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, Motorola reserves the right to make changes to any products herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

EXORdisk, EXORterm, EXbug, and MICRObug are trademarks of Motorola Inc.

First Edition

©Copyright 1979 by Motorola Inc.



# TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
PRODUCT DESCRIPTION	1
MEDIUM	1
HARDWARE REQUIREMENTS	2
HARDWARE CONFIGURATION	3
OPERATION	3
Start-Up	3
Transparent Mode	5
Change Duplex Mode	5
Start Download Process	5
Start Verify Process	6
Quit Current Process	7
ERROR MESSAGES	7
Transparent Mode	8
Downloading	8
Verifying	8
Downloading and Verifying	8
APPENDIX A    ABSOLUTE OBJECT RECORD FORMAT	 A-1
ILLUSTRATION	
FIGURE 1.    Communications Scheme Using the Download Program	 2



## INTRODUCTION

The Motorola Download Program permits the designer to load programs developed on a host system into his M6800/M6801, M6809, or Micromodule Development System. Key features of the Download Program are:

- . Keystroke entry to resident debug/monitor program.
- . Optional entry point for user-developed debug/monitor program, plus entry points for Motorola's EXbug and MICRObug debug/monitors.
- . Developer specification of ACIA addresses and control codes.
- . Software selection of transmission mode -- full or half duplex.
- . Position independent.
- . Comprehensive error checking and error checking of transmitted S-records.
- . Transparent, Download, and Verify modes.
- . Error reporting, including checksum error, bad byte count, memory error, invalid hexadecimal value in memory address, invalid hexadecimal value in byte, verification error.

## PRODUCT DESCRIPTION

Quite frequently, software development for a microprocessor system is done on a larger and more powerful computer than the microprocessor itself. This is done using a "cross assembler". A cross assembler is simply an assembler for one machine that runs on a different machine. The output of an assembler is usually a listing and a binary file of the object code generated. However, a cross assembler cannot generate a pure binary file of the object code, as it would not be readily transportable to the microprocessor system. Therefore, a cross assembler typically generates a file of binary encoded in ASCII. In other words, easily transportable ASCII characters are used to represent the binary object code of the program being developed. Motorola's cross assemblers generate such a file in a format known as S-records (see Appendix A).

Once this file is generated, the binary object represented by the ASCII file must be loaded into the memory of the development system. This may be done in several ways. By referring to the file of S-records, the user might key in the program by hand -- however, for anything but a short program, this would be tedious and error prone. The S-record file might be transferred to another medium such as cassette or paper tape, which could then be used to load the program -- but this requires the time to transfer the S-records from host computer storage to the second medium. The best way is for the host computer to communicate directly, automatically loading the program into development system memory. This capability is provided by the Download Program.

## MEDIUM

The Download Program, supplied in a 1K ROM, is totally position independent so long as it starts on a 256-byte boundary -- that is, an even multiple of \$100.

## HARDWARE REQUIREMENTS

The hardware requirements for the Download Program include a standard M6800/ M6801 or M6809 Development System, or equivalent Micromodule system. As shown in Figure 1, the system should include standard random-access memory, a terminal, an RS-232C interface (as provided by an MEX6850 ACIA Module for communications with the terminal), and a debug/monitor program. A 1K slot is needed for the Download Program ROM to reside in. This could be on a Micromodule 1A Monoboard Microcomputer Module, a Micromodule 4 ERAM/RAM Module, or an MEX68RR ERAM/ROM Module. In addition, a second ACIA is required for communication with the host computer.

Selection of a means of interfacing with the second ACIA depends on the type of host computer. Connection would typically be made via phone line and modem or acoustic coupler if the time sharing service of a large mainframe were being used to develop software. If an M6800 or M6809 Development System with an EXORdisk and EXORterm were serving as host, interface could be made by connecting a line from the second ACIA to the terminal ACIA of the host.

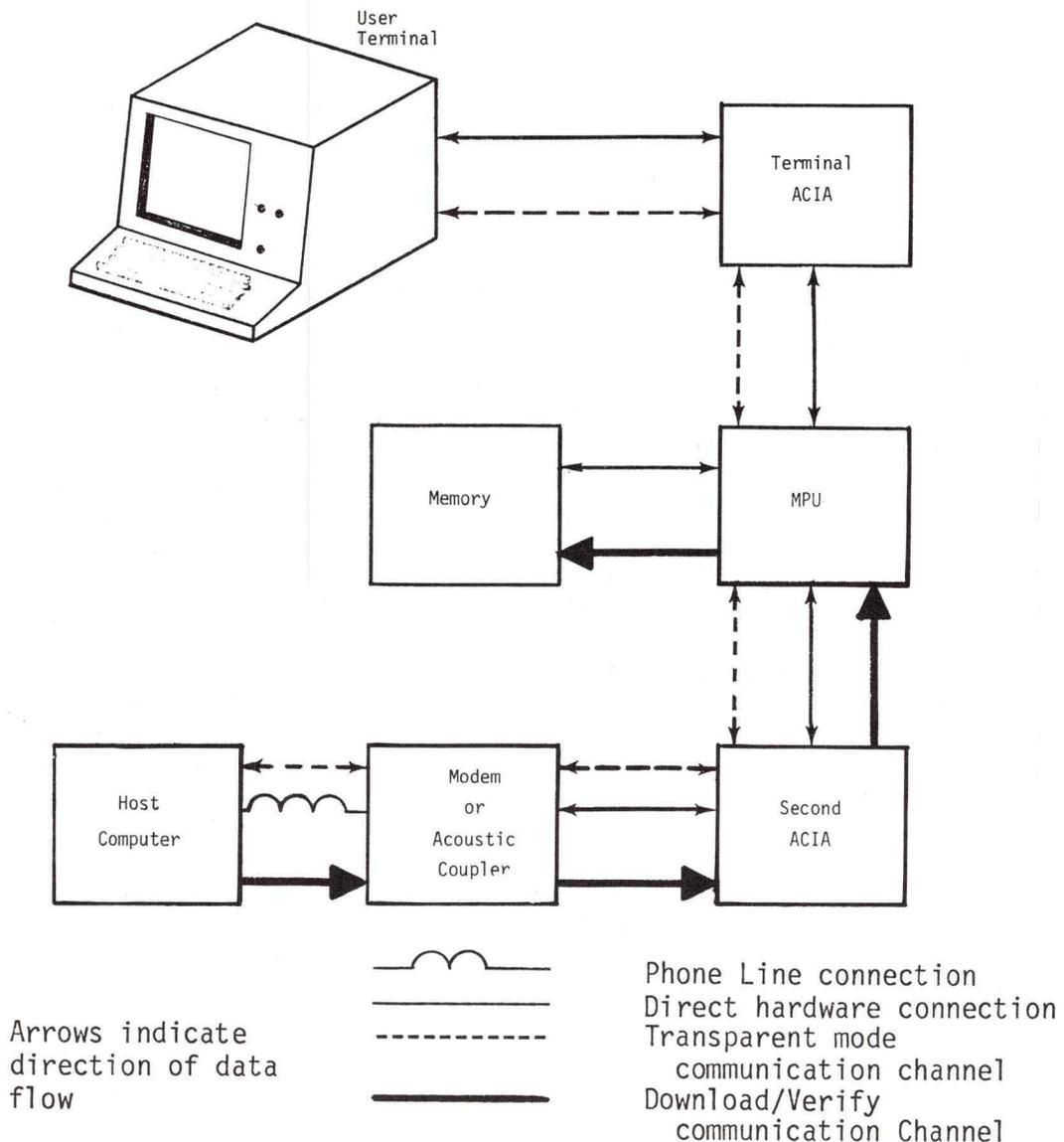


FIGURE 1. Communications Scheme using the Download Program

## HARDWARE CONFIGURATION

The following procedure should be performed to configure the development system for Download Program operations.

1. Select the speed (baud rate) of the second ACIA to correspond with that of the communications port or channel of the host computer.
2. Set the address of the second ACIA to the desired value. The Download Program contains default addresses for various debug/monitors. It is advisable to use the proper default value, if at all possible. These are as follows:

<u>MPU</u>	<u>Monitor</u>	<u>Value</u>
6800/6801	MICRObug	\$EC14
6800/6801	EXbug	\$EC14
6809	EXbug 09	\$EC14

3. Install the second ACIA in the development system and connect it to the communications link with the host computer.
4. Select the speed of the terminal ACIA to correspond with that of the terminal being used. The speed of the terminal must be at least as great as the speed at which communications are held with the host computer.
5. Set the address of the terminal ACIA to the desired value. The Download Program contains default addresses for various debug/monitors. It is advisable that the proper default value be used, if at all possible. These are as follows:

<u>MPU</u>	<u>Monitor</u>	<u>Value</u>
6800/6801	MICRObug	\$8408
6800/6801	EXbug	\$FCF4
6809	EXbug 09	\$FCF4

6. Install the terminal ACIA in the development system.
7. Finally, install the Download Program ROM in the 1K socket being used, and select its base address according to the hardware being used.

## OPERATION

The following paragraphs describe the operational procedures of the Download Program.

### Start-Up

The Download Program is started by entering the program at one of three points. These are described in the following chart:

ENTRY POINT ADDRESS	COMPUTER SYSTEM	
	6800/6801	6809
\$xx80	EXbug default entry point	EXbug 09 default entry point.
\$xx82	MICRObug default entry point.	Not used.
\$xx84	Entry point for any other debug/monitor.	Entry point for any other debug/monitor.

(where xx are the first two hexadecimal digits of the base address of the ROM)

The term "default entry point" is used in the context of this manual to mean an entry point that causes the Download Program to use the appropriate default values. These values are stored in locations \$20-\$29 in RAM, and are used to control such items as the ACIA addresses and the various control keys. Because some of the values are unique to a particular debug/monitor, the corresponding entry point should be used when that program is resident in the development system. Use of the default entry points is explained below.

Location(s)	Use	Default Values		
		EXbug	MICRObug	Exbug 09
\$20	Character to Start Load Process	←———— \$0C(↑L) —————→		
\$21	Character to Start Verify Process	←———— \$16(↑V) —————→		
\$22	Character to Quit Current Operation	←———— \$11(↑Q) —————→		
\$23	Character to Change Duplex Mode	←———— \$04(↑D) —————→		
\$24-\$25	Address of Terminal ACIA	\$FCF4	\$8408	\$FCF4
\$26-\$27	Address of Second ACIA	←———— \$EC14 —————→		
\$28-\$29	Return address to debug/monitor	\$F5C5	\$FC44	\$F564

Note that just two entry points are assigned to particular debug/monitors, the third being reserved for use with any other monitor. To provide for initialization with an unknown monitor, the Download Program does not provide default values when entered for execution at address \$xx84. To use this entry point, therefore, the user must manually load the appropriate values into memory locations \$20-\$29, using the memory examine and change function of the corresponding monitor. Should the user desire to alter any of the listed default values, then he must initiate execution of the Download Program via the \$xx84 entry point after manually entering all values, as described.

To summarize: execution of the Download Program is started by first choosing the appropriate entry point and then entering the program at that point. If use of \$xx84 is required, the appropriate values must be manually loaded into memory locations \$20-\$29 before the program is entered for execution.

## Transparent Mode

When the Download Program is first started, the following message will be printed:

```
DOWNLOAD X.XX
```

where X.XX is the version number.

At this point, the program is in the "transparent mode". This means that data entered at the user terminal is transmitted to the host computer, and data transmitted from the host computer is displayed on the user terminal, with the user terminal looking like any other to the computer. This form of terminal usage is represented by the dashed line of Figure 1. The user may now establish communication with the host computer and, for example, log on to the computer time-sharing service, enter a source program for use on the microprocessor development system, and assemble that program using the proper cross assembler. Unless a special command character is entered, communication will continue in the transparent mode.

The special command characters and their uses are as follows. (In the following paragraphs, the up arrow (↑) indicates a control character.)

↑D - Change Duplex Mode - When the user is in transparent mode, he may change the duplex characteristics of the terminal by striking the "character to change duplex mode" (default character is ↑D). This will change the transmission mode of the terminal and a message indicating the new mode will be printed.

When the program is first started, the terminal is in full duplex. Thus, striking the ↑D immediately after starting the program will put the terminal in half duplex. Striking the ↑D again will then put the terminal back in full duplex, and so on.

The change duplex mode is provided for those users whose host computer does not have echo capability and must communicate with a terminal in the half duplex mode. Since most development system terminals normally operate in full duplex because the majority of debug/monitor programs have echo capability, the change duplex mode feature of the Download Program provides a convenient way of changing the transmission mode to communicate with a half duplex host computer.

↑L - Start Download Process - The ↑L command is used to start a download, as follows:

With the Download Program in the transparent mode, type in the host computer command to list the desired S-record file, but do not strike the carriage return at the end of the command line. Instead, enter the "character to start a download" (default character is ↑L). At this time, the message "LOADING" will be printed on the terminal. Also, if an S0 record (header record) is found in the download process, the name in the record will be printed on the terminal. During the download, extensive error checking is performed, and any errors that occur will be reported on the terminal (see paragraph on error messages). The download will continue until an S9 record (end of file) is encountered. At that time, the following message will be printed:

```
XXXX ERRORS
```

where XXXX is the hexadecimal number of errors that occurred during the download. A number of 0000 indicates a successful download. Finally, the user is placed back into the transparent mode.

## Examples

If the host computer were IBM's TSO time-sharing service, a command to start a download process might appear as follows (all user entries are underlined):

```
READY

LIST TEST.OBJ.DATA NONUM †L
LOADING TEST
0000 ERRORS

READY
```

If the host were a 6800 or 6809 development system running MDOS, an example would be:

```
=COPY SORT.LX,#CN †L
LOADING BSORT
** B AT $2357
** C AT $2350
0002 ERRORS

=
```

### CAUTION

The Download Program uses RAM locations \$20-\$FF, inclusive, for its working variables, stack, etc. Therefore, the program should not be used to load these locations.

†V - Start Verify Process - The procedure to verify a program that has already been downloaded is exactly the same as that for downloading the S-record file, with one exception. Instead of typing the "character to start a download" in place of the carriage return on the command line, type the "character to start a verify" (default is †V). Rather than "LOADING", the message "VERIFYING" will be printed on the terminal. The rest of the process is exactly the same as that for a download.

Examples:

TSO:

```
READY

LIST GAME.OBJ.DATA NONUM †V
VERIFYING TIC-TAC-TOE
** V AT $72F3
** V AT $72F4
** V AT $72F5
** V AT $72F6
** V AT $9321
0005 ERRORS

READY
```

6800/6809 MDOS:

```
=COPY PAYROLL.LX,#CN ↑V  
VERIFYING PAYROLL  
0000 ERRORS
```

=

↑Q - Quit Current Process - Use of the ↑Q command causes immediate exit from the operation in progress. From the download and verify modes, exit is made to the transparent mode, and the message \*\*ABORTED is printed. From the transparent mode, control is returned to the resident debug/monitor, and no message is printed. The latter use of the ↑Q command provides a convenient way for the user to begin debugging a program which has been downloaded into his development system RAM. To return to the transparent mode, the user must restart the Download Program, as previously described.

Examples:

TS0:

READY

```
LIST MEMTEST.OBJ.DATA NONUM ↑L  
LOADING MEMORY-TEST
```

\*\* M AT \$E000

\*\* M AT \$E001

\*\* M AT \$E002

\*\* M AT \$E003

(User types ↑Q)

\*\* ABORTED

3457DF9621

S118E010265922345FD3C4:!

(User hits BREAK key)

READY

6800/6809:

```
=COPY NEWPROG.LX,#CN ↑V  
VERIFYING TESTPROG  
0000 ERRORS
```

= ↑Q

EXBUG 1.2 MAID

\*

ERROR MESSAGES

The following list presents, in the sequence in which they would normally occur, a description of the error messages generated by the Download Program.

## Transparent Mode

### \*\* BUFFER OVERFLOW

In the transparent mode, all data transmitted from the host computer are buffered because even though the terminal may be set at the same baud rate as the host computer, it is possible the actual baud rates differ, with the host sending characters at a slightly faster rate than the terminal can handle, resulting in loss of characters. To prevent such loss, a buffer is used. During a very long transmission, should the buffer be filled to capacity and overflow, the overflow message is printed, the buffer is cleared, and communication is resumed. Buffer overflow can be prevented by avoiding long transmissions or using a baud rate higher at the terminal than at the computer.

## Downloading

### \*\* M AT \$XXXX

During a download, whenever a byte is stored in a memory location, the location is immediately read back to verify the byte was stored properly. This message indicates that a byte was not stored properly at hexadecimal location XXXX, possibly due to one of three items:

1. Memory is bad, in which case a memory test should be run for verification.
2. An attempt was made to store a byte in non-existent memory, in which case more RAM should be added to the system or the program should be relocated to load into existing RAM.
3. An attempt was made to store a byte where ROM is residing, in which case the ROM should be removed and RAM installed in its place, if possible, or the program should be relocated.

## Verifying

### \*\* V AT \$XXXX

This message indicates that a verification error occurred at hexadecimal location XXXX. In other words, the byte read from the S-record file for location \$XXXX does not equal the byte already stored in memory at location \$XXXX.

## Downloading and Verifying

### \*\* C AT \$XXXX

This message indicates that a checksum error occurred in reading the S-record whose starting address was hexadecimal XXXX. This would most likely be due to a transmission error.

### \*\* B AT \$XXXX

This message indicates that a bad hexadecimal byte (meaning a non-hexadecimal digit) was encountered in the S-record file at hexadecimal location XXXX. The byte in memory will not be changed/verified, and the program will continue. This is most likely caused by a transmission error.

\*\* A AT \$XXXX

This message indicates a bad address (one with a non-hexadecimal digit) was found in the S-record file at the hexadecimal XXXX'th S-record in the file. The remainder of the S-record will be ignored. The probable cause for this error is a transmission error.

\*\* K AT \$XXXX

This message indicates that a bad byte count (one with a non-hexadecimal digit) was encountered in the hexadecimal XXXX'th S-record of the file being downloaded/verified. The remainder of the S-record will be ignored. This error would most likely be due to a transmission error.

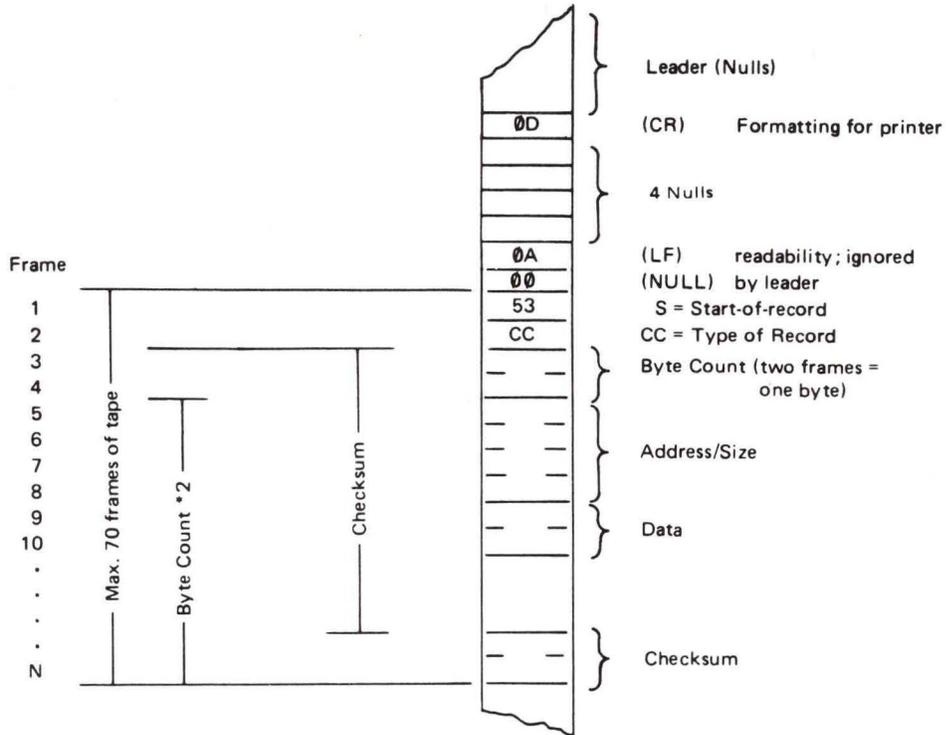
\*\* ABORTED

This message is printed whenever the user strikes the ↑Q during a download or verify operation. The program is then returned to transparent mode.



# APPENDIX A

## ABSOLUTE OBJECT RECORD FORMAT



Frames 3 through N are hexadecimal digits represented by a 7-bit ASCII character. Two hexadecimal digits are combined to make one 8-bit byte.

The checksum is the one's complement of the summation of 8-bit bytes.

Frame	CC = 30 Header Record	CC = 31 Data Record	CC = 39 End-of-File Record
1. Start-of-Record	53	53	53
2. Type of Record	30	31	39
3. Byte Count	31	31	30
4.	32	36	33
5.	30	31	30
6. Address/Size	30	31	30
7.	30	30	30
8.	30	30	30
9. Data	34	39	46
10.	38	38	43
.	34	30	
.	34	32	(Checksum)
.	35		
.	32		
.		38	
.		41	8A (Checksum)
N. Checksum	39		
	45		



# SUGGESTION/PROBLEM REPORT

Motorola welcomes your comments on its products and publications. Please use this form.

To: Motorola Microsystems  
P.O. Box 20912  
Attention: Publications Manager  
Mail Drop M374  
Phoenix, Az. 85036

Comments

Product: \_\_\_\_\_

Manual: \_\_\_\_\_

*Please Print*

\_\_\_\_\_  
Name

\_\_\_\_\_  
Title

\_\_\_\_\_  
Company

\_\_\_\_\_  
Division

\_\_\_\_\_  
Street

\_\_\_\_\_  
Mail Drop

\_\_\_\_\_  
Phone Number

\_\_\_\_\_  
City

\_\_\_\_\_  
State

\_\_\_\_\_  
Zip

HARDWARE SUPPORT: (800) 528-1908  
SOFTWARE SUPPORT: (602) 962-3935





ADDENDUM

TO

M6800/M6801/M6809

DOWNLOAD PROGRAM

USER'S GUIDE

1. In the second table, page 4, the EXbug default value for memory locations \$28-\$29 should be changed from \$F5C5 to \$F564.

2. Refer to pages 3 and 4, "Start-Up" -

The DOWNLOAD program initializes both the terminal and host ACIA with the same configuration status byte, \$15. This sets up 8 bits, no parity, 1 stop bit, and divide by 16. If this is not desired, the program can be patched as follows:

Read, change, and blow new ROM with PROM Programmer III.

or

Write a small program to transfer contents to RAM area and patch in RAM.

or

Use ROLLOUT command of MDOS to create disk file of DOWNLOAD ROM. Then patch disk file.

6800 VERSION 1.01

```
$xxC3      JMP  PATCH1      (was $DE,26,86,03)
$xxF9      JMP  PATCH2      (was $86,03,A7,00)
```

```
          ORG  <your choice>
PATCH1  EQU  *
          <code to initialize ACIA's>
          JMP  $xxD3
PATCH2  EQU  *
          <code to initialize host ACIA after BREAK>
          JMP  $x101
```

6809 VERSION 1.00

```
$xxBF      JMP  PATCH1      (was $86,03,A7,9F)
$xxFE      JMP  PATCH2      (was $86,03,A7,9F)
```

```
          ORG  <your choice>
PATCH1  EQU  *
          <code to initialize ACIA's>
          JMP  $xxD3
PATCH2  EQU  *
          <code to initialize host ACIA after BREAK>
          JMP  $x10A
```

NOTE

"xx" is first two hexadecimal digits of the base address of the ROM.