890704

Customer Order Number 424420188-001
NSC Publication Number 424420188-001A
April 1989

HPC™

Communications Program (COMHPC)

User's Manual

2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, California 95052-8090

REVISION RECORD

REVISION	RELEASE DATE	SUMMARY OF CHANGES
Beta Site	04/89	Beta Site Release HPC^{TM} Communications Program (COMHPC) NSC Publication Number 424420188-001A

PREFACE

This manual provides information on the communications program (COMHPC) which supports the development of HPC™ microcontroller applications on National Semiconductor's microcontroller development systems.

COMHPC is a new communications program for downloading HPC COFF absolute object modules to the HPC MOLE microcontroller development system. COMHPC supports Bankswitch download, as well as terminal emulation and command file execution.

This manual assumes the user is already familiar with the host operating system. For example, the user needs to know how files are named and used under the operating system.

Other applicable microcontroller system manuals are the *Brain Board User's Manual*, Customer Order Number 420408188-001, and *HPC Personality Board User's Manual*, Customer Order Number 420410477-001. Also see *HPC Assembler/Linker/Librarian User's Manual* for information on generating COFF object modules, and for information on bankswitching linker.

The information contained in this manual is for reference only and is subject to change without notice.

CONTENTS

Chapter 1 INTRODUCTION

Chapter 2 COMMUNICATION PROGRAM (COMHPC)

Chapter 3 COMHPC COMMANDS

FIGURES

TABLES

INDEX

Chapter 1

INTRODUCTION

1.1 OVERVIEW

This manual provides information on the communications program (COMHPC) which supports the development of HPC microcontroller applications on the microcontroller development system. The invocation and operation of COMHPC vary depending on the operating system. In the following chapters, invocation and operation are described for the MS-DOS operating system. For information on COMHPC on UNIX and VMS operating systems, refer to Appendixes B and C.

Section 1.2 describes the communications program (COMHPC).

Section 1.3 summarizes the notation used to describe statement syntax and user input.

1.2 COMMUNICATIONS PROGRAM (COMHPC)

COMHPC is a new communications program for downloading HPC COFF absolute object modules to the HPC MOLE development system. COMHPC supports Bankswitch download, as well as terminal emulation and command file execution.

1.3 DOCUMENTATION CONVENTIONS

The following documentation conventions are used in text, syntax descriptions, and examples in describing commands and parameters.

1.3.1 General Conventions

Nonprinting characters are indicated by enclosing a name for the character in angle brackets < >. For example, <CR> indicates the RETURN key, <ctrl/B> indicates the character input by simultaneously pressing the control key and the **B** key.

1.3.2 Conventions in Syntax Descriptions

The following conventions are used in syntax descriptions.

Italics are used for items supplied by the user. The italicized word is a generic term for the actual operand that the user enters.

Spaces or blanks, when present, are significant; they must be entered as shown. Multiple blanks or horizontal tabs may be used in place of a single blank.

{ } Large braces enclose two or more items of which one, and only one, must be used. The items are separated from each other by a logical OR sign "|."

[ ] Large brackets enclose optional item(s).

| Logical OR sign separates items of which one, and only one, may be used.

... Three consecutive periods indicate optional repetition of the preceding item(s). If a group of items can be repeated, the group is enclosed in large parentheses "( )."

,,, Three consecutive commas indicate optional repetition of the preceding item. Items must be separated by commas. If a group of items can be repeated, the group is enclosed in large parentheses "( )."

( ) Large parentheses enclose items which need to be grouped together for optional repetition. If three consecutive commas or periods follow an item, only that item may be repeated. The parentheses indicate that the group may be repeated.

␣ Indicates a space. ␣ is only used to indicate a specific number of required spaces.

All other characters or symbols appearing in the syntax must be entered as shown. Brackets, parentheses, or braces which must be entered, are smaller than the symbols used to describe the syntax. (Compare user-entered [ ], with [ ] which show optional items.)

1.3.3 Example Conventions

In interactive examples where both user input and system responses are shown, the machine output is in regular type; user-entered input is in boldface type. Output from the machine may vary (*e.g.*, the date) and is indicated with italic type.

Chapter 2

COMMUNICATION PROGRAM (COMHPC)

2.1 INTRODUCTION

COMHPC is a system program which allows the user to download HPC COFF absolute object modules to the HPC MOLE microcontroller development system.

COMHPC is used by connecting a serial port on the host machine to a console port of the microcontroller development system. File download is initiated from the host system. The host is connected through an RS232 cable either directly to the microcontroller development system or indirectly through a modem. Appendix A contains information on the necessary cable connections.

This chapter presents an overview of invocation and operation of COMHPC. Also included are definitions of some terms used, a discussion of the two different operating modes (command mode and bypass mode), and a discussion of the system configuration.

2.2 OVERVIEW OF COMHPC

COMHPC is primarily intended to allow downloading of HPC microcontroller COFF object files from a host system to a HPC microcontroller development board. Additional features of COMHPC are:

- It provides automatic setup of the HPC microcontroller development system.
- It provides automatic download of greater than 16K code or bankswitch code to the HPC microcontroller development system.
- It allows the user to supply an alternate set of bankswitch commands for download.
- It does its own error checking and transmission retries.
- It allows the host system to act as the console for the HPC microcontroller development system.
- It allows recording of its command input and responses (or only its command input) into a history file. Such a history file, or a simple ASCII file, can be used as input for COMHPC; thus a session recorded can be re-run.

  It allows the user to specify data line and baud rate.

2.3 SYSTEM DEFINITIONS

''Console'' or ''CRT'' refers to the system console.

''Data'' port refers to a bidirectional, asynchronous RS232 port.

NOTE: The console and data ports are different physical devices.

As far as using COMHPC is concerned, host systems can be of two different types:

1. A system that has two RS232 ports; one is the system console, the other is a peripheral device. The system runs from an RS232 CRT terminal connected to the console port.

Example: VT100 connected to VAX, with a separate data line to a microcontroller development system

2. A system that has a peripheral RS232 port (*i.e.*, a data port), but does not have an RS232 port for the console. Instead, the console is an integral part of the system, or is a video monitor and keyboard.

   Example: IBM PC

2.4 COMHPC MODES

COMHPC has two operating modes: command mode and bypass mode. COMHPC starts out in command mode and is switched to bypass mode by issuing the **BYPASS** command. Command mode is re-entered by issuing the bypass escape character (<ESC>).

When COMHPC is in command mode the user issues commands to download files, monitor the microcontroller development system, execute command files, and enable the history mechanism.

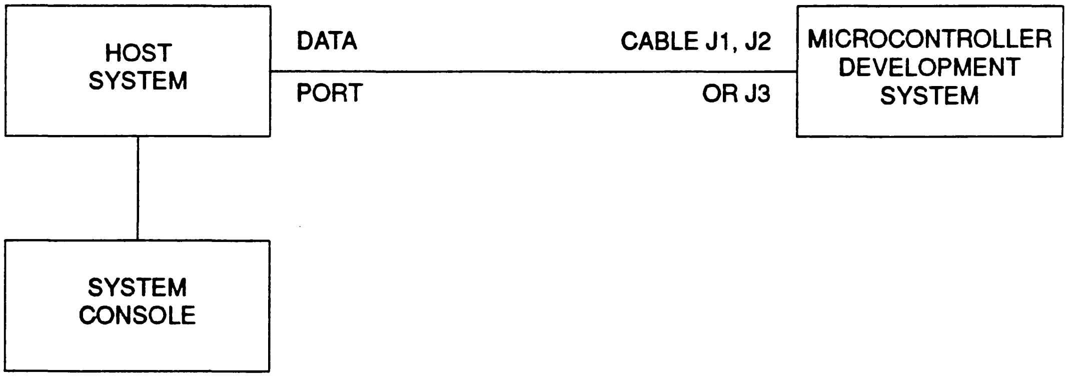
The bypass mode is COMHPC's "dumb terminal" function; it allows the system running COMHPC to act as the console for another system.

Specifically:

1. When the microcontroller development system is put into bypass mode, there is a simulated connection between the console and data ports. Characters entered at the console are sent out on the data port; characters sent in to the data port are sent out to the console.
2. COMHPC buffers both character streams — this allows for temporary differences in speed between the host and microcontroller development system.
3. While in bypass mode, COMHPC looks for a special escape character entered at the console. This character causes COMHPC to exit the bypass mode.

2.5 SYSTEM CONFIGURATION

This section describes how to hook up the host system to a microcontroller development system.



The host data port (*e.g.*, COM1 on IBM-PC) is connected to the microcontroller development system console port. On the microcontroller development system either J1, J2 or J3 can function as the console port. The choice should be made on the basis of port compatibility; the cable connecting the ports must be compatible with the way both ports are wired. Most combinations of ports can be covered by one of two basic cable wirings. See Appendix A for more information.

The host system is operated as normal. To operate the microcontroller development system, invoke COMHPC on the host system and enter bypass mode. The host system console acts as the console for the microcontroller development system. A separate CRT terminal for the microcontroller development system is not needed.

2.6 STARTUP AND DOWNLOAD DETAILS

On startup, COMHPC establishes initial handshake with the microcontroller development system and invokes the MONITOR program (the microcontroller development system prompt is M>). Before initiating download, COMHPC automatically invokes the COMM program on the microcontroller development system. During download, all data is initially loaded to SH (shared/on-board) memory and then, if required, moved to user (off-board) memory. (See *HPC Personality Board User's Manual* for details of emulation memory management.) COMHPC does multiple downloads of 16K each in case of bankswitch download or if code size is more than 16K. This is necessary because SH memory is limited to 16K. After completing download, COMHPC again invokes the microcontroller development system's MONITOR program. At this point the user can enter bypass mode to run and test his downloaded program.

The microcontroller development system SH blocks to be used for emulation after the download are defined by the user with the **/B** option on the invocation line or the **BLOCK** command on the command line. If **/B** or **BLOCK** is not specified, default SH blocks 12,13,14,15 are used. It is critical for proper download and emulation that the microcontroller development system SH blocks are properly specified to reflect the user's system memory configuration (see **BLOCK** command).

The user can also define alternate bankswitch commands for different bankswitch schemes. For this, the user may specify a bankswitch command file with the /S option (see Section 2.8.7).

The actual download sequence is as follows:

1. If there is user memory to load, the SH memory blocks are mapped to 12, 13, 14, 15. Code (of maximum size 16K) is downloaded to SH. The bank (if any) is selected, and the code is moved to user memory.
2. If there is more code to load to user memory, step 1 is repeated.
3. Finally, the SH blocks are mapped as specified by the user. The final download loads SH memory.

2.7 COMHPC INVOCATION AND OPERATION

This section discusses the invocation and operation of COMHPC.

2.7.1 Invocation

The invocation line varies depending on the operating system. See Appendix B for invocation on a UNIX operating system and Appendix C for invocation on a VMS operating system. For the MS-DOS operating system, the invocation line is as follows.

COMHPC [*options*] [*loadfile*] [*options*]

where: *loadfile* is the name of a COFF object file to be downloaded. Default extension is *.cof*. This is equivalent to command line **LOAD loadfile**. See **LOAD** command (Section 3.7) for more information.

options is a list of options. No options need be specified, in which case the defaults will be used. Section 2.7.2 describes the option syntax.

If no *options* nor *loadfile* are specified, Help information is displayed.

Any error detected on the invocation line causes COMHPC to stop execution and display the part in error with an appropriate message.

The following are sample invocation lines for MS-DOS systems:

```
COMHPC /B=2,8,15 /M TEST
COMHPC /H=THIS_SESSION \DEMO\SAMPLE
COMHPC /E=THIS_SESSION.HIS TEST
```

The first command line sets the microcontroller development system SH memory blocks 2,8, and 15 for download and loads the file *test.cof*. Monitoring of the microcontroller development system is also enabled.

The next command line opens file *this_session.his* for recording this session. File *sample.cof* which resides in the directory *demo* is loaded.

The last command loads file *test.cof* and then executes history file *this_session.his*.

2.7.2 Options

An invocation line option is specified in a manner consistent with the operating system. See Appendix B for information for a UNIX system and Appendix C for a VMS system.

The invocation line options for an MS-DOS operating system start with a ``/'', which may be preceded by blank space. Options are not case-sensitive. Options and their arguments are separated by ``='', which may be preceded or followed by blank space. Option arguments are separated by ``,'', which may be preceded or followed by blank space. Numeric option arguments may be in hex (start with 0x), octal (start with 0), or decimal (not start with 0).

See Section 2.8 for a description of all the options.

2.7.3 Default File Extensions

Default file extensions depend on the operating system. See Appendixes B and C for information on UNIX and VMS operating systems. For MS-DOS, a default extension is always placed on a file unless one is explicitly specified.

2.7.4 Help File Search Order

When searching for help file *COMHPC.HLP*, directories are searched in the following order:

1. current directory
2. default directories
    a. directory specified by environment variable COMHPC, if it exists
    b. directory specified by environment variable HPC, if it exists
    c. directory *\HPC*

2.8 INVOCATION OPTIONS

This section describes the options that may be used on the invocation line.

2.8.1 Block List

Syntax: **/B=***block list* | **OFF**

Description: The **/B** option sets the microcontroller development system SH memory blocks to use for emulation after the download. The argument block list is up to four blocks (0 to 15) or OFF.

This option is equivalent to command **BLOCK** *block list* | **OFF** in command mode. See **BLOCK** command (Section 3.2) for more information on the microcontroller development system SH blocks.

NOTE: It is critical for proper download and emulation that the SH blocks are properly specified to reflect the user's system memory configuration. Default SH blocks are 12,13,14,15.

Example: /B=2,8,15
This sets SH blocks 2,8 and 15.

2.8.2 Execute File

Syntax: /E=*file*

Description: The /E option executes input file. All COMHPC commands including bypass commands contained in the input file are executed, after the load file, if any, is loaded. The input file is typically a history file, but may also be a simple ASCII file. The default extension of the input file is *.his*. This option is equivalent to the command @*file* in command mode.

Example: /E=setup.his

2.8.3 History File

Syntax: **/H=***file*

Description: The **/H** option generates a history file. The default extension is *.his*. All commands entered as well as output from COMHPC will be written to the history file. This option is equivalent to the command **HISTORY ON, OUT, F=***file* in command mode.

Example: /H=this_session

2.8.4 Line

Syntax: **/L=***line* [*,baud*]

Description: The /L option defines line and baud rate to use for download. The argument line can be COM1, COM2, COM3, or COM4 for MS-DOS system; default is COM1. The argument baud can be 300, 600, 1200, 2400, 4800, 9600, or 19200; default is 19200. No parity, 8 data bits, 1 stop bit are set.

Example: /L=COM2,4800

This sets COM2 port for communication with the microcontroller development system at 4800 baud.

2.8.5 Monitor

Syntax: **/M**

Description: The **/M** option monitors the microcontroller development system. COMHPC generates microcontroller development system commands which are hidden to the user. This option echoes all microcontroller development system commands and responses to the console, preceded by "*". This option is useful in debugging COMHPC interaction with the microcontroller development system. This option is equivalent to the command **MONITOR ON** in command mode.

2.8.6 Restart

Syntax: **/R**

Description: The **/R** option restarts COMHPC without loading the COFF file, if specified. The microcontroller development system is initialized to the MONITOR program (microcontroller development system prompt M>). Default parameters are used for download unless they are specified on the invocation line along with **/R** option. This option is equivalent to the command **LOAD =R** in command mode.

2.8.7 Bankswitch Command File

Syntax: */S=file*

Description: The /S option defines alternate bank switch command file for microcontroller development system. This command file contains commands that are used by COMHPC to select the bank while downloading bankswitch code. Each line in file is a **MONITOR** command; the only permitted commands are **ALTER**, **AW**, or **MODIFY** (see *HPC Personaltiy Board User's Manual* for command usage). COMHPC will reset the upper bit of each character, and convert it to upper case for use by the microcontroller development system. "%" in file is replaced by active bank address during download (the bank address is passed from the COFF file). The default set of bankswitch commands is:

```
MODIFY DIRB,6300    ;define bit 8,9,13,14 of Port B as output
MODIFY BFUN,0       ;disable Port B alternate function
MODIFY M E3,%       ;bank address to upper byte of Port B
```

The default file extension is *.bsw*.

NOTE: Before the above bankswitch commands are executed, the MONITOR commands RESET and SINGLESTEP are executed to breakpoint the HPC chip.

Example: /S=bankfile.bsw

Chapter 3

COMHPC COMMANDS

3.1 INTRODUCTION

After the COMHPC program is invoked, the prompt

```
comhpc>
```

indicates the program is ready to accept a command and is in command mode.

The command formats are listed alphabetically in Table 3-1. The definitions used in the command formats are listed in Table 3-2.

The minimum abbreviation for a command name is indicated by the uppercase letters of the command name in the syntax description. For example, BYPASS may be specified as BY, BYP, BYPA, BYPAS, or BYPASS. Commands are not case sensitive.

The commands may be optionally preceded by blank space. The commands and their arguments are separated by blank space. The command arguments are separated by ``,'', which may be preceded or followed by blank space. The default file extensions are dependent on the operating system (see Section 2.7.3).

Table 3-1. COMHPC Commands

COMMAND	OPERANDS	DESCRIPTION
BLock	[*block list* | OFF]	Specify/Display microcontroller development system SH Blocks.
BYpass		Enter bypass mode.
End		Exit COMHPC and return to host
Help	[*command*]	List the syntax of all commands or a specific command to the console.
HIstory	[(ON|OFF),] [(OUT|NO),] [F=*file*]	Specify/Display history parameters.
LOad	[*file* | =R]	Download file or restart microcontroller development system.
MONitor	[ON | OFF]	Specify/Display microcontroller development system monitoring status.
@	[*file*]	Execute commands from input file.
;	*text*	Comment line.
!	*system command*	Invoke operating system command.

Table 3-2. Command Format Definitions

OPERANDS	DESCRIPTION
ON	Enable
OFF	Disable
OUT	Output Enable
NO	Output Disable
=R	Restart microcontroller development system only
block list	List of up to 4 microcontroller development system SH blocks, 0 to 15
file	Valid filename
system command	Operating system command
text	Character string terminated by a carriage return

✓

3.2 BLOCK COMMAND

Syntax: **BLock** [*block list* | **OFF**]

after the download

Description: The **BLOCK** command specifies/displays the microcontroller development system SH Blocks to use for emulation. The argument block list is up to four blocks or OFF. If no argument is specified, this command displays SH Blocks that were set with previous /B option or **BLOCK** command.

The microcontroller development system SH memory provides emulation memory on the microcontroller development system. The user may optionally provide his own user memory in addition to SH memory, or in place of SH memory. SH is mapped in 4K blocks; therefore, as SH is limited to 16K size, the user can specify a maximum of four blocks. Each block starts on a 4K page boundary and has a block number that equals the most significant nibble of the address space; for example, 5000/5FFF has the block number 5. See the *HPC Personality Board User's Manual* for more information on SH Blocks and usage.

If the user has bankswitched memory, SH blocks cannot fall into the switchable bank address range. This is because SH blocks are always visible to the system and cannot be bankswitched.

NOTE: It is critical for proper download and emulation that the microcontroller development system SH Blocks are properly defined to reflect the user's system memory configuration. The default SH blocks are 12,13,14,15.

Example:

```
comhpc> BLOCK 2,8,15
MOLE SH Blocks = 2,8,15
```

In this example, SH Blocks 2,8,15 are set, then displayed.

```
comhpc> BLOCK
MOLE SH Blocks = 2,8,15
```

In this example, the current SH Blocks are displayed.

3.3 BYPASS COMMAND

Syntax: **BY**pass

Description: The **BYPASS** command puts the host machine into bypass mode, where the host acts as a dumb terminal for the microcontroller development system.

When the host is put into bypass mode, there is a simulated connection between the console and the data port. Characters entered at the console are sent out on the data port; characters sent in to the data port are sent out to the console. Temporary differences in speed between the host and the microcontroller development system are taken care of by COMHPC buffering.

The bypass mode allows the user to interact directly with the microcontroller development system in a transparent manner. Note that this obviates the need of a separate CRT terminal for the microcontroller development system. All work is done from the host system console.

The bypass mode is exited when <ESC> is entered at the console.

Example: Assume the microcontroller development system is in MONITOR program (microcontroller development system `M>` prompt).

```
comhpc>BYPASS

M> LIST

...

...

...

M> <ESC>

comhpc>
```

In this example, entering **BYPASS** puts the host into bypass mode. The user then issues some microcontroller development system MONITOR commands and when <ESC> is pressed, the host exits the bypass mode, returning to COMHPC prompt (leaving the microcontroller development system in MONITOR program).

3.4 END COMMAND

Syntax: **End**

Description: The **END** command exits COMHPC and returns to the host operating system.

Example: `comhpc>` **END**

3.5 HELP COMMAND

✓

Syntax: **Help** [*command*]

Description: The **HELP** command, with no argument, lists the syntax of all commands to the console. If a command argument is given, specific information is given on that command.

Example:

```
comhpc> HELP
BLock      [<block list> | OFF]   specify/display MOLE SH blocks
BYpass                            enter bypass mode
End                               exit COMHPC
Help       [<command>]            gives command help
HIstory    [(ON|OFF),] [(OUT|NO),] [f=<file>]
                                  specify/display history parameters
LOad       [<file> | =R]          load file or restart MOLE
MONitor    [ON | OFF]             specify/display monitor status
@          [<file>]               execute commands from file
;          <comment>              ignore this line
!          <system command>       invoke operating system command
```

comhpc> **HELP** **!** ~~blank~~ space

Syntax: !<operating system command>

Description: ~~Execute~~ The ! command invokes a shell or executes an operating system command.

3.6 HISTORY COMMAND

Syntax: **HI**story [(ON|OFF),] [(OUT|NO),] [F=*file*]

Description: The **HISTORY** command specifies and controls the history file generation. When ON, all commands entered will be written to the history file. If OUT is specified, the command output will also be written to the history file, preceded by the comment character.

The history command, without any arguments, will display current history status.

Once a history file has been selected, either on the invocation line or with this command, the recording may be turned ON and OFF without renaming the file.

The History ON and OFF commands will be written to the History file with the comment character prepended.

The default extension of the history file is *.his*.

This command is useful for saving a COMHPC session on a file. It is also useful for debugging purposes. Such a history file can be used as input for the COMHPC; thus, a session recorded can be re-run.

Example:

```
comhpc> HISTORY ON,NO,F=this_session
History ON,NO,F=this_session.his
```

This and the following commands entered will be written to the file *this_session.his*. The specified history parameters are displayed.

```
comhpc> HISTORY OUT
History ON,OUT,F=this_session.his
```

This and the following commands as well as the command output will be written to the history file. The modified history parameters are displayed.

```
comhpc> HISTORY
History ON,OUT,F=this_session.his
```

The command **HISTORY**, when invoked without any arguments, displays the current history parameters.

3.7 LOAD COMMAND

Syntax: LOad [*file* | =R]

Description: The **LOAD** command loads the specified COFF object file into the microcontroller development system. The default extension of the file is *.cof*. After the load, the SH blocks are mapped as specified by the **/B** option or **BLOCK** command; default SH blocks are 12,13,14,15.

If no argument is specified, the load uses the file specified on the last **LOAD** command or on the invocation line (it is an error if no load file was previously specified). If the argument is "=R", no load is done; the microcontroller development system is initialized to the MONITOR program (microcontroller development system prompt `M>`).

When the **LOAD** command is invoked, first the SH blocks specified are displayed. Next, during the download, COMHPC continuously displays retry count and the number of transmission data blocks sent to the microcontroller development system. Upon successful completion of the download, message `TRANSMISSION COMPLETE` is also displayed.

COMHPC determines alternate bank switch commands and line parameters from the /S and the /L options on the invocation line. When these options are not specified, their default values are used.

The error messages and their likely causes are discussed in Section 4.

NOTE: It is critical for proper download and emulation that the SH blocks be defined properly to reflect the user's system memory configuration. See **BLOCK** command.

Example:

```
comhpc> LOAD TEST.COF
MOLE SH block = 12,13,14,15

block sent: ####
retry:      ####

TRANSMISSION COMPLETE
```

This command initiates download of the object file *test.cof*. The microcontroller development system SH blocks are indicated. During the download, the number of blocks sent and retry count are continously updated. The message `TRANSMISSION COMPLETE` indicates that the download was successful.

3.8 MONITOR COMMAND

Syntax: **MONitor** [ON | OFF]

Description: The **MONITOR** command specifies/displays the microcontroller development system monitoring status.

COMHPC generates microcontroller development system commands which are hidden to the user. If the argument is "ON", all microcontroller development system commands and responses are echoed to console, preceded by "*". This command is useful in debugging COMHPC interaction with the microcontroller development system, particularly when errors occur in the system.

If no argument is specified, the current status is shown.

Example:

```
comhpc> MONITOR ON
Monitor on
```

This example sets the microcontroller development system monitoring.

3.9 @ (EXECUTE) COMMAND

Syntax: **@** [*file*]

Description: The @ command executes the COMHPC commands from the input *file*. Files may nest to the limit of memory. Commands being executed are also echoed on the console. On error, execution of the input file is aborted and control transferred to console. Execution of the input file can be resumed with entering @ without a filename.

A simple ASCII or a history file (for example, as generated earlier by the **HISTORY** command) can be used as an input file. Thus, this command in conjunction with the **HISTORY** command, can be used to re-run frequently executed commands, or commands saved earlier, in "batch" mode.

The default extension of file is *.his*.

Example:

```
comhpc> @COM.HIS
...
...
LOAD test.cof
        ^
error: file not found
input file aborted.
comhpc> LOAD tst.cof
...
...
...
comhpc> @
...
...
...
comhpc> END
```

In this example, COMHPC starts executing commands from the input file *com.his* in response to **@com.his** entered by the user. When an error occurs while executing the command **load test.cof**, the file *com.his* is aborted and control returns to the COMHPC prompt. The user then loads the proper file and resumes execution of *com.his* by entering @. Finally, COMHPC exits when the user enters the command **END**.

✓

3.10 ; (COMMENT) COMMAND

Syntax: *;comment*

Description: The ; command au c~~ea~~ses COMHPC to ignore all characters including and after ";" till the end of line. This allows commenting history files and also capturing the output of a session without interfering with reuse of the session as an input file. Comment lines will be echoed to the history file if appropriate, but with an additional comment character prepended.

Example:

```
comhpc> HISTORY ON,OUT,F=this_session.his

comhpc> ;Now execute input file

comhpc> @last_session.his
```

In this example, the user first enables history recording and then issues the command to execute commands from the file *last_session.his*. The comment line in this case enhances the readability of the history file.

3.11 ! (SYSTEM) COMMAND

Syntax: *!operating system command*

Description: The ! command invokes a shell or executes an operating system command.

Example: **!rename a.his b.his**

In this example, the file *a.his* is renamed to *b.his*.

Chapter 4

COMHPC ERROR MESSAGES

4.1 INTRODUCTION

Several categories of errors are possible: the option or command specification could be incorrect; the download may have errors; the loading may be aborted by the user, or the microcontroller development system encountered errors. The error messages and their likely causes are discussed in this chapter.

4.1.1 Option and Command Error Messages

Option and command error messages are listed in Table 4-1.

✓

Table 4-1. Operand Error Messages

MESSAGE	DESCRIPTION
Bad argument	Argument specified is not valid.
Bad bankswitch command file	Bankswitch command file must contain alter, aw, or modify commands only.
Bad object file	File specified for download is not a valid COFF object file.
Bad octa~~l~~l constant	Invalid octa~~l~~l characters.
Bad option	Illegal option.
Block not available	The microcontroller development system SH Block specified is not available for download and emulation.
Cannot open file	Error in opening file
Closing history file	Output history file closed.
Disk directory full	Disk directory full.
Disk error	Error in opening, closing or renaming file.
Excess argument	Excess argument on line.
File not found	File specified not found in the directory.
Help file not found	Help file not found.
Input file aborted	Input history file aborted.
Initial handshake aborted	Initial handshake aborted.
Invalid filename	Illegal filename.
Line error	Error opening line.
Missing argument	No argument specified for the option.
Missing file name	Filename required.
MOLE not respond	No response from microcontroller development system; check line or microcontroller development system.
No file specified	No file specified where a filename was required.
Not an object file	File specified for download is not a valid COFF object file.
Object file contain no data	No data to download.
Out of memory	Not enough memory is available to execute the command.
Read error	Error while reading file.
Syntax error	Badly formed command line.
Too many files	Too many files open.
Unknown command	Unknown command.
Unknown option	Illegal option.
Unsupported baud rate	Baud rate specified is not supported.
Unsupported data line	Data line specified is not supported.
Write error	Error while writing to file.

When an error occurs in the invocation or command line, the error message is displayed with an arrow pointing to the position where the error occurred. For example:

1. comhpc> **@comnd.his** ; error on command line

```
                ^
   error: file not found
```

2. **COMHPC /R /L=COM1,110 test.cof** ; error on invocation

```
                       ^
   error:  unsupported baud rate
```

4.1.2 Abort Error Messages

Table 4-2 lists the transmission abort error messages.

To understand the abort messages better, it is important to know that there are two phases to the transmission. The probable cause of an error message depends upon whether the error occurs during phase 1 or phase 2.

Phase 1 is an initialization transmission; this part of the transmission occurs after the load command is invoked, but before any data blocks are sent. Aborts, during this phase, are generally due to a bad communications line or to incompatible UART settings.

During phase 2 of the transmission, the actual file download occurs. During this part of the transmission, aborts are mostly due to a line with marginal quality or actual loss of the line.

4.1.3 User Abort

The user who invoked the transfer may abort it anytime by hitting **<CTRL/C>.** An abort message is sent to the microcontroller development system so that the microcontroller development system will abort cleanly.

4.1.4 Error Messages

Whenever the microcontroller development system responds unexpectedly, the microcontroller development system responses are shown preceded by '*'. This includes responses during download; for example, read/write verify errors while moving code from SH memory to user memory will be shown. As appropriate, the microcontroller development system command which caused the error will be shown in parentheses following the error message. To further determine the cause of a microcontroller development system error, the user may rerun the **COMHPC** command sequence with microcontroller development system monitoring enabled (see **MONITOR** command).

Table 4-2. Transmission Abort Error Messages

MESSAGE	DESCRIPTION AND CAUSES
Bad link up	Error in invoking download. Try again or check line.
Bad map list	Error in **MAP** command on development system.
Bad transmission	Bad transmission. Line may be lost or bad, or the microcontroller development system aborted.
Code in on-chip RAM area	Code in any bank cannot fall in on-chip RAM area.
Bank address cannot fall in SH block	The microcontroller development system SH blocks are always visible for emulation and cannot be bank switched.
Illegal transmission	Illegal transmission detected. Transmission cannot continue.
MOLE aborted	The microcontroller development system requested an abort after having a fatal error.
Retry error	Illegal transmission retry.
Transmission timeout	Timeout in waiting for reply. Line may be lost or bad, or the microcontroller development system aborted.
User abort	User aborted transmission.

Appendix A

RS232 INTERFACING

A.1 RS232 CABLES

This appendix describes the RS232 cable required to connect the host to the microcontroller development system. The type of cable needed depends on the two RS232 ports being hooked together.

RS232 ports can be of two types — Data Terminal Equipment (DTE) or Data Communication Equipment (DCE). Table A-1 shows the relevant RS232 signals. (''To DCE'' means that the signal is an input to a DCE-type port and, therefore, an output of a DTE port.)

Table A-1. RS232 Signals (25-Pin Connector)

PIN NUMBER	SIGNAL NAME	DIRECTION
2	Transmitted Data	To DCE
3	Received Data	From DCE
7	Signal Ground	-

Note that the RS232 standard assumes: a DTE sending data to a DCE through pin 2 and receiving data from the DCE through pin 3 of the standard connector and, correspondingly, a DCE receiving data on pin 2 and sending data on pin 3. The cable used to connect a DTE to a DCE is a standard RS232 cable, referred to here as a straight through cable. DTEs can be connected to DTEs and DCEs can be connected to DCEs, but a straight through cable cannot be used to make the connection; a special null modem cable which transposes signals is required.

A straight through cable is shown in Figure A-1. This cable will connect a host DTE port to J1 port of the microcontroller development system, or a host DCE port to J2 port of the microcontroller development system.

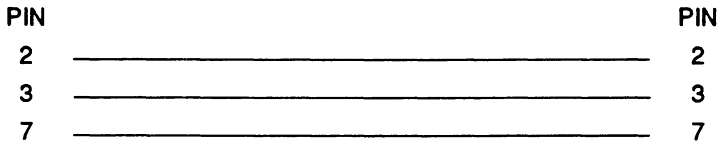


Figure A-1. Straight Through Cable

Figure A-2 shows a null modem cable. This cable will connect a host DTE port to J2 port of the microcontroller development system, or a host DCE port to J1 port of the microcontroller development system.

Refer to the manufacturers's schematics to determine how a particular host RS232 port has been wired. Table A-2 shows which of the microcontroller development system ports are DCE or DTE.

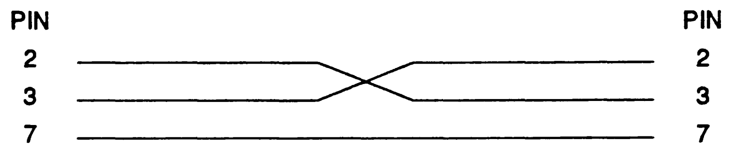


Figure A-2. Null Modem RS232 Cable

Table A-2. RS232 Ports On The Microcontroller Development System

RS232 CONNECTOR	IS WIRED AS
Port J1	DCE
Port J2	DTE
Port J3	DCE*

* This port can be reconfigured using jumpers. Refer to the *Brain Board User's Manual*. The wiring as shipped is indicated.

Appendix B

UNIX INVOCATION AND OPERATIONS

B.1 INTRODUCTION

This appendix describes the invocation and operations for COMHPC on a UNIX® operating system. See the release letter for installation instructions.

B.2 COMHPC INVOCATION AND OPERATION

This section describes the operation of COMHPC on the UNIX operating system.

B.2.1 Invocation

The COMHPC invocation for a UNIX operating system is as follows:

`comhpc` [*options*] [*loadfile*]

where: *loadfile* is the name of a file to load. There is no default file extension assumed.

options is a list of options. ‘‘-l’’ option must be specified; line and baud are as appropriate to the system. No other options need to be specified, in which case the defaults are used. Section B.2.2 describes the option syntax.

If no *options* or *loadfile* specified, help is given.

Any error detected on the invocation line causes COMHPC to stop execution and display the part in error with an appropriate message.

The following are sample invocation lines for UNIX systems:

```
comhpc -b 2,8,15 -m test.cof

comhpc -h this_session.his /demo/sample.cof

comhpc -e this_session.his test.cof
```

The first command line sets the microcontroller development system SH memory blocks 2,8, and 15 for download and loads the file *test.cof*. Monitoring of the microcontroller development system is also enabled.

The next command line opens file *this_session.his* for recording this session. File *sample.cof* which resides in the directory *demo* is loaded.

The last command loads file *test.cof* and then executes the history file *this_session.his*.

B.2.2 Options

An invocation line option is specified in a manner consistent with the UNIX operating system. Refer to Section 2.8 for a detailed description of the options.

Options for a UNIX operating system follow the System V Interface Definition. Each option is preceded by a "-". Options are case sensitive. There are no optional arguments; the option either requires an argument, or it doesn't. Multiple options without argument may be appended together or listed separately. For example:

-mr is equivalent to **-m -r**

See Section B.2.4 for mapping between UNIX and MS-DOS invocation line options.

B.2.3 Default File Extensions

For UNIX operating systems, there is no default file extension assumed.

B.2.4 Help File Search Order

When searching for help file *comhpc.hlp*, directories are searched in the following order:

1. current directory
2. default directories
   a. directory specified by environment variable COMHPC, if it exists
   b. directory specified by environment variable HPC, if it exists
   c. directory */hpc*

B.2.5 Option Table

Table B-1 shows the mapping between UNIX and MS-DOS invocation. UNIX options as well as the arguments are case sensitive.

Table B-1. Mapping Between MS-DOS and UNIX Invocation

MS_DOS	UNIX
/B=block list	-b block list
/B=OFF	-b OFF | off
/E=input file	-e input file
/H=history file	-h history file
/L=line	-l line
/L=line,baud	-l line,baud
/M	-m
/R	-r
/S=file	-s file

Appendix C

VMS INVOCATION AND OPERATIONS

C.1 INTRODUCTION

This appendix describes the invocation and operations for COMHPC on a VMS operating system. See the release letter for installation instructions.

C.2 COMHPC INVOCATION AND OPERATION

This section describes the operation of COMHPC on a VMS operating system.

C.2.1 Invocation

COMHPC invocation for a VMS operating system is as follows:

```
COMHPC      [options] [loadfile] [options]
```

where: *loadfile* is the name of a file to load. Default extension is *.cof*.

options is a list of options. ''/L'' option must be specified; line and baud are as appropriate to the system. No other options need to be specified, in which case the defaults are used. Section C.2.2 describes the option syntax.

If no *options* or *loadfile* are specified, help is given.

Any error detected on the invocation line causes COMHPC to stop execution and display the part in error with an appropriate message.

The following are sample invocation lines for VMS systems:

```
COMHPC /B=(2,8,15) /M TEST
COMHPC /H=THIS_SESSION [DEMO]SAMPLE
COMHPC /E=THIS_SESSION TEST
```

The first command line sets the microcontroller development system SH blocks 2,8, and 15 for download and loads the file *TEST.COF*. Monitoring of the microcontroller development system is also enabled.

The next command line opens file *THIS_SESSION.HIS* for recording this session. File *SAMPLE.COF* which resides in the directory *DEMO* is loaded.

The last command loads file *TEST.COF* and then executes the history file *THIS_SESSION.HIS*.

C.2.2 Options

An invocation line option is specified in a manner consistent with the VMS operating system. Refer to Section 2.8 for a detailed description of the options.

Options for the VMS operating system follow the rules of the DEC command line interpreter. Each option is preceded by a ''/''. Options are not case sensitive.

See Section C.2.4 for mapping between VMS and MS-DOS invocation line options.

C.2.3 Default File Extensions

For VMS operating systems, a default extension is always placed on a file unless one is explicitly specified.

C.2.4 Help File Search Order

When searching for help file *COMHPC.HLP*, directories are searched in the following order:

1. current directory
2. default directories
    a. directory specified by logical name COMHPC$, if it exists
    b. directory specified by logical name HPC$, if it exists
    c. directory specified by logical name SYSHPC$

C.2.5 Option Table

Table C-1 shows the mapping between VMS and MS-DOS invocation.

Table C-1. Mapping Between MS-DOS and VMS Invocation

MS_DOS	VMS
/B=block list	/B=(block list)
/B=OFF	/B=OFF
/E=input file	/E=input file
/H=history file	/H=history file
/L=line	/L=line
/L=line,baud	/L=(line,baud)
/M	/M
/R	/R
/S=file	/S=file

Appendix D

SETTING UP FOR COMHPC

D.1 INTRODUCTION

This appendix describes the step-by-step procedure for setting up and verifying the hardware required to operate COMHPC on an MS-DOS system.

D.2 PROCEDURE FOR CONFIGURATION

The procedure is divided into two parts:

- Connecting the MS-DOS and the microcontroller development system with an RS232 cable.
- Verifying the hardware connection.

D.2.1 Hardware Setup

The MS-DOS COM1 port and the microcontroller development system are connected by an RS232 cable. Appendix A discusses the RS232 cable and the microcontroller development system port to use.

D.2.2 Hardware Verification

The hardware is tested by creating a HPC COFF file, loading the file from the MS-DOS system to the microcontroller development system and verifying that the file was received correctly.

1. Create the following file *test.asm* using your text editor:

```
.SECT       TEST,ROM16,ABS=0F000
LOOP:        JP     .
            .FB     0800,0     ; data
            .END    LOOP
```

2. Assemble and link file using current revision software.

```
C> ASMHPC test.asm
C> LNHPC test.obj
```

3. Power up (or reset) the microcontroller development system and invoke COMHPC

```
C> COMHPC /R
```

4. COMHPC initializes the microcontroller development system and responds with its prompt

```
comhpc>
```

5. Enter bypass mode

```
comhpc> BYPASS

M>
```

At this point, the MS-DOS console is acting as the microcontroller development system console. This shows that connection between the microcontroller development system and MS-DOS system is correct, and the microcontroller development system prompt `M>` shows that the microcontroller development system was initialized properly.

6. Press <ESC> to escape bypass mode

   ```
   M> <ESC>

   comhpc>
   ```

7. Download the test file to the microcontroller development system.

   ```
   comhpc>   LOAD test.cof

   MOLE SH Blocks = 12,13,14,15

   block sent: 13
   retry:      0

   TRANSMISSION COMPLETE
   ```

8. The microcontroller development system SH blocks defined for emulation are shown. During download, the number of blocks sent to the microcontroller development system and a count of retries is displayed (Retries should be zero for a clean connection). Upon successful download, the message `TRANSMISSION COMPLETE` is displayed. Otherwise, an appropriate error message is echoed on the system console.

The check-out is done.

Appendix E

A SAMPLE COMHPC SESSION

This is a brief example of a download session with COMHPC on an MS-DOS host system connected to a the microcontroller development system. This assumes a COFF object module *sample.cof* with bankswitched code at address 4000/4FFF. In this session, the user first loads the object file and then enters bypass mode to verify the code. History is enabled to record the session.

Create file *setup.his*. This command file is used to provide setup for bankswitch.

```
[setup.his]
BYPASS              ;enter bypass mode
RESET               ;Reset before bankswitch
S                   ;Breakpoint microcontroller development system
MODIFY BFUN,0       ;disable port B alternate function
MODIFY DIRB, 6300   ;define bit 8,9,13,14 of port B as output
```

Create file *swbank0.his*. This command file will be used to switch to bank 0.

```
[swbank0.his]
@setup            ;do setup for bankswitch
MODIFY M E3, 0    ;bank 0 user defined as upper byte of port B = 0
```

Create file *swbank1.his*. This is used to switch to bank 1.

```
[swbank1.his]
@setup            ;do setup for bankswitch
MODIFY M E3, 1    ;bank 1 user defined as upper byte of port B = 1
```

Assume the console port of the microcontroller development system is connected to COM2 port of the MS-DOS system. The user sets SH blocks 14 and 15 to use for emulation; he is using SH memory as the non-bank switched memory at address E000/FFFF. The user has provided bankswitched user memory at address 4000/4FFF, for bank 0 and bank 1. The microcontroller development system is powered up, and the following invocation line is entered. COMHPC loads the file *sample.cof*.

```
C>COMHPC /B=14,15 /L=COM2 /E=SwBank0 /H=this_session sample
```

```
MOLE SH blocks = 14,15

block sent: ####
retry:      0

TRANSMISSION COMPLETE
```

Next, COMHPC starts executing commands from file *swbank0.his*.

```
comhpc>@setup           ;do setup for bank switch
comhpc>BYPASS           ;enter bypass mode

M>RESET                 ;Reset before bankswitch
M>S                     ;Breakpoint microcontroller development system
M>MODIFY BFUN, 0        ;disable port B alternate function
M>MODIFY DIRB, 6300     ;define bit 8,9,13,14 of port B as output

M>MODIFY M E3, 0        ;bank 0 user defined as upper byte of port B = 0
```

User then enters command to check code at address 4000/4FFF.

```
M>LIST 4000/4FFF     ;show bank 0 at 4000/4FFF
...
...
...
```

Next, the user returns to COMHPC by entering <ESC> and issues the command to execute file *swbank1.his* to switch to bank 1. ~~The user then enters commands to verify code at address 4000.~~

```
M><ESC>

comhpc>@swbank1          ;switch to bank 1   ok
                         ;COMHPC starts executing commands from file "swbank1.his"
comhpc>@setup            ;do setup for bank switch
comhpc>BYPASS            ;enter bypass mode

M>RESET                  ;Reset before bankswitch
M>S                      ;Breakpoint microcontroller development system
M>MODIFY BFUN, 0         ;disable port B alternate function
M>MODIFY DIRB, 6300      ;define bit 8,9,13,14 of port B as output

M>MODIFY M E3, 1         ;bank 1 user defined as port B address = 0
```

User then enters command to check code at address 4000/4FFF.

```
M>LIST 4000/4FFF     ;show bank 1 at 4000/4FFF
...
...
...
```

Now the user returns to COMHPC to exit the session.

```
M><ESC>

comhpc> END
```

For more information on bankswitch^ing linker, see *HPC Assembler/Linker/Librarian User's Manual*.

INDEX

National
Semiconductor

MICROCOMPUTER
SYSTEMS GROUP

READER'S COMMENT FORM

In the interest of improving our documentation, National Semiconductor invites your comments on this manual.

Please restrict your comments to the documentation. Technical Support may be contacted at:

(800) 759-0105 — U.S. and Canada only
((0)8141) 103-330 — Germany only

Please rate this document according to the following categories. Include your comments below.

	EXCELLENT	**GOOD**	**ADEQUATE**	**FAIR**	**POOR**
Readability (style)	□	□	□	□	□
Technical Accuracy	□	□	□	□	□
Fulfills Needs	□	□	□	□	□
Organization	□	□	□	□	□
Presentation (format)	□	□	□	□	□
Depth of Coverage	□	□	□	□	□
Overall Quality	□	□	□	□	□

NAME ______________________________ DATE ____________

TITLE ______________________________

COMPANY NAME/DEPARTMENT ______________________________

ADDRESS ______________________________

CITY ____________________ STATE ____________ ZIP __________

Do you require a response? □ Yes □ No PHONE ______________________________

Comments:

____

____

____

____

____

Communications Program (COMHPC)

CUT, FOLD, STAPLE, AND MAIL