iRMX 86™
TERMINAL HANDLER
REFERENCE MANUAL

Manual Order No.: 143324-001

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

REV.	REVISION HISTORY	PRINT DATE
-01	Original Issue	3/81

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BXP
CREDIT
i
ICE
iCS
im
Insite
Intel
Intel
Intelevision
Intellec
iRMX
iSBC
iSBX
Library Manager
MCS
Megachassis
Micromap
Multibus
Multimodule
PROMPT
Promware
RMX/80
System 2000
UPI
MScope

and the combination of ICE, iCS, iRMX, iSBC, iSBX, MCS, or RMX and a numerical suffix.

PREFACE

This manual documents the Terminal Handler, a subsystem of the iRMX 86 Operating System that supports the use of a terminal as an I/O device for an application system. It describes the effects of certain special keyboard characters as well as how to use the Terminal Handler for task input from, and output to, a terminal.

This manual was formerly a chapter in what was the iRMX 86 NUCLEUS, TERMINAL HANDLER, AND DEBUGGER REFERENCE MANUAL. Consequently, it has change bars even though it is a first-version manual.

READER LEVEL

This manual is intended for both application and system programmers who are familiar with the concepts and terminology introduced in the iRMX 86 NUCLEUS REFERENCE MANUAL.

RELATED PUBLICATIONS

The following manuals provide additional background and reference information.

Manual	Number
Introduction to the iRMX 86 Operating System	9803124
iRMX 86 Nucleus Reference Manual	9803122
iRMX 86 Configuration Guide	9803126

CONTENTS

PAGE

FIGURES

TABLE

CHAPTER 1. GENERAL INFORMATION

GENERAL INFORMATION

The Terminal Handler supports real-time, asynchronous I/O between an operator's terminal and tasks running under the iRMX 86 Nucleus. It is intended for use in applications which require only limited I/O through a terminal, and it generally is used in applications that do not include the iRMX 86 I/O System. The features of the Terminal Handler include the following:

- Line editing capabilities.
- Keystroke control over output, including output suspension and resumption, and deletion of data being sent by tasks to the terminal.
- Echoing of characters as they are entered into the Terminal Handler's line buffer.

An output-only version of the Terminal Handler is available for use in applications in which tasks send output to a terminal but do not receive input from the terminal.

Several Terminal Handler features, characteristics, and identifiers are configurable. For applications with unusual characteristics, such as a component hardware environment or multiple terminal handlers, configurability is important. For detailed information regarding configuration, see the iRMX 86 CONFIGURATION GUIDE.

The remainder of this manual is divided between two chapters. Chapter 2, Using a Terminal with the iRMX 86 Operating System, provides the information that is needed by an operator of the terminal. Chapter 3, Programming Considerations, contains the information that a programmer needs to write tasks that send data to, or receive data from, the terminal.

CHAPTER 2. USING A TERMINAL WITH THE iRMX 86™ OPERATING SYSTEM

While using a terminal that is under control of the Terminal Handler, an operator either reads an output message from the terminal's display or enters characters by striking keys on the terminal's keyboard. Normal input characters are those destined for input messages that are sent to tasks. Special input characters direct the Terminal Handler to take special actions. The special characters are RUBOUT, Carriage Return, Line Feed, ESCape, control-C, control-O, control-Q, control-R, control-S, control-X, and control-Z. The output-only version of the Terminal Handler does not support any of the special characters. In the remainder of this chapter, the handling of these two types is discussed, and the significance of each of the special characters is explained.

NOTE

This chapter contains several references to mailboxes and request messages used by tasks to communicate with the terminal. If you are puzzled by such a reference, look in Chapter 3 for an explanation.

HOW NORMAL CHARACTERS ARE HANDLED

The destination of a normal character, when entered, depends on whether there is an input request message at the Terminal Handler's input request mailbox. If there is an input request message, the character is echoed to the terminal's display and goes into the input request message. If there is not an input request message, the character is deleted.

HOW SPECIAL CHARACTERS ARE HANDLED

Table 2-1 lists the special characters and summarizes the effects of each of them. The following text comprises complete descriptions of the effects of the special characters. In these descriptions, there are several references to "the current line." The current line consists of the data, with editing, that has been entered since the last end-of-file charcter.

Table 2-1. Special Character Summary

SPECIAL CHARACTER	EFFECT
RUBOUT	Deletes previously entered character.
Carriage Return	Signals end of line.
Line Feed	Signals end of line.
ESCape	Signals end of line.
control-C	Calls an application program.
control-O	Kills or restarts output.
control-Q	Resumes suspended output.
control-R	Displays current line with editing.
control-S	Suspends output.
control-X	Deletes the current line.
control-Z	Sends empty message.

The following descriptions concern the special characters needed when entering data at the terminal. Most of these characters are for line-editing. Each description is divided into two parts: internal effects and external effects. The difference is that external effects are immediately shown on the terminal's display, whereas internal effects are those that are not directly visible.

RUBBING OUT A PREVIOUSLY-TYPED CHARACTER (RUBOUT)

Internal Effects: Causes the most recently entered but not yet deleted character to be deleted from the current line. If the current line is empty, there is no internal effect.

External Effects: If the current line is empty, the BEL character (07H) is sent to the terminal. Otherwise, the character is "rubbed out" in accordance with one of two available rubout modes. In the copying mode, the character being deleted from the current line is re-echoed to the display. For example, entering "CAT" and then striking RUBOUT three times results in the display "CATTAC". In the blanking mode, the deleted character is replaced on the CRT screen with the blanking character. For example, entering "CAT" and then striking RUBOUT three times deletes all three characters from the display. The copy mode is the default mode. The default blanking character for the blanking mode is a space (20H). If you wish to change either of these defaults, refer to the iRMX 86 CONFIGURATION GUIDE.

DISPLAYING THE CURRENT LINE (CONTROL-R)

Internal Effects: None.

External Effects: Sends a carriage return and line feed to the terminal, followed by the current line. If the current line is empty, the previous line is sent to the display, where it can be line edited and submitted as a new input message.

DELETING THE CURRENT LINE (CONTROL-X)

Internal Effects: Empties the current line.

External Effects: Causes the sequence (#, Carriage Return, Line Feed) to be sent to the terminal.

SENDING AN EMPTY MESSAGE (CONTROL-Z)

Internal Effects: Puts a zero in the ACTUAL field of the input request message currently being processed. The message is then sent to the appropriate response mailbox.

External Effects: None.

SIGNALLING THE END OF A LINE OF INPUT (CARRIAGE RETURN, LINE FEED, OR ESCAPE)

Internal Effects: Puts either the ASCII end-of-transmission character (0AH in the case of Carriage Return or Line Feed) or the ESCape character (1BH) in the current line. Each of these characters signals the end of a message, so the input request message currently being constructed is sent to the appropriate response mailbox.

External Effects: If the end-of-line indicator is either Carriage Return or Line Feed, both Carriage Return and Line Feed are sent to the terminal. If the indicator is ESCape, however, there is no effect on the display.

OUTPUT CONTROL

Output request messages that are sent to output mailbox (whose name is configurable; the default name is RQTHNORMOUT) can be processed in three ways:

- They can be output as described later in Chapter 3.
- They can be queued at the output mailbox where they remain until an operator at the terminal takes action to permit processing of the messages.
- They can be discarded.

In the descriptions that follow, these methods of dealing with output requests are called the normal mode, the queueing mode, and the suppression mode, respectively. Initially, output is in the normal mode.

SUSPENDING OUTPUT (CONTROL-S)

Puts output in the queueing mode.

RESUMING OUTPUT (CONTROL-Q)

Negates the effects of control-S by allowing the display of output requests that are sent to the output mailbox. The output that has been suppressed is displayed (very quickly) in the order in which it would have been displayed earlier if the control-S had not been pressed. If you are overwhelmed by this output, you can stop it again by pressing control-S.

DELETING OR RESTARTING OUTPUT (CONTROL-O)

If output is in the normal mode, control-O puts it in the suppression mode. If output is in the suppression mode, control-O restores it to the normal mode. If output is in the queueing mode, control-O has no effect. Internally, the request messages that tasks send while output is being suppressed are returned to those tasks just as if the output had not been suppressed.

PROGRAM CONTROL

The remaining control character affects system behavior.

CALLING A USER-WRITTEN PROCEDURE MANUALLY (CONTROL-C)

Control-C invokes a parameter-less, user-written procedure named RQ$ABORT$AP. This procedure, which must be compiled under either the COMPACT or SMALL control, can perform any actions that suit the application. Often, as its name suggests, RQ$ABORT$AP aborts an application. If it is present (and it need not be), RQ$ABORT$AP is not required to have a RETURN statement.

Control-C also causes the effects produced by control-Z; that is, it returns the current input request message with its ACTUAL field set to zero. This is the case even if the application system does not contain an RQ$ABORT$AP procedure.

SETTING A BAUD RATE

The Terminal Handler can be set to operate at any of the following baud rates:

110
150
300
600
1200
2400
4800
9600
19200

The rate is set during software configuration, when an iRMX 86 system variable, RQRATE, is assigned one of these baud rate values.

CHAPTER 3. PROGRAMMING CONSIDERATIONS

The iRMX 86 Terminal Handler supports terminal input and output by providing mailbox interfaces. Figure 3-1 shows the use of these mailboxes. In the figure, an arrow pointing from a task to a mailbox represents an RQ$SEND$MESSAGE system call. An arrow pointing from a mailbox to a task indicates an RQ$RECEIVE$MESSAGE system call.

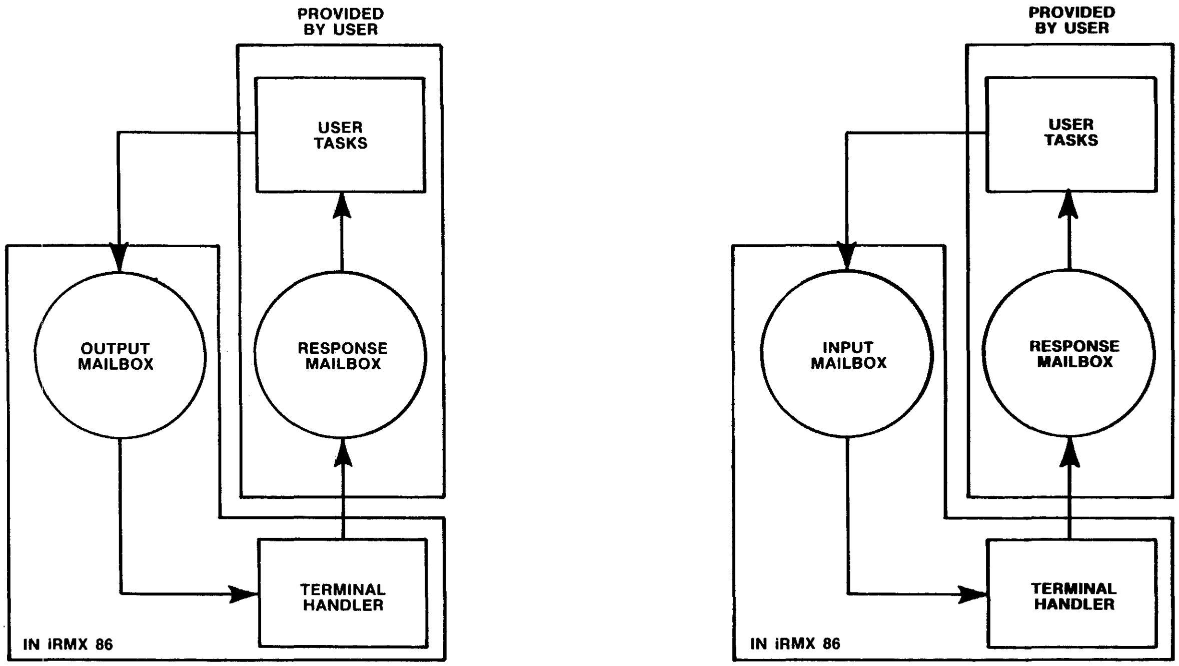

Figure 3-1. Input and Output Mailbox Interfaces

The protocol that tasks observe is much the same for input and output. In each case, the task initiates I/O by sending a request message to a mailbox. An input request mailbox (default name RQTHNORMIN) and an output request mailbox (default name RQTHNORMOUT) are provided. The Terminal Handler processes the request and then sends a response message back to the requesting task. The task waits at a response mailbox for the message. Thus, whether a task does input or output, it first sends and then receives. The full details of the input and output protocols are described later in this chapter. Output is discussed fir3t because it is somewhat simpler and therefore is easier to understand.

For both input and output, a task sends a message segment to the Terminal Handler. The format of a request message is depicted in Figure 3-2. The numbers in that figure are offsets, in bytes, from the beginning of the segment. The field names have different meanings for input and for output. For both input and output, the first four fields are WORD values. The MESSAGE CONTENT field can be up to 132 bytes in length for input and up to 65527 bytes in length for output.

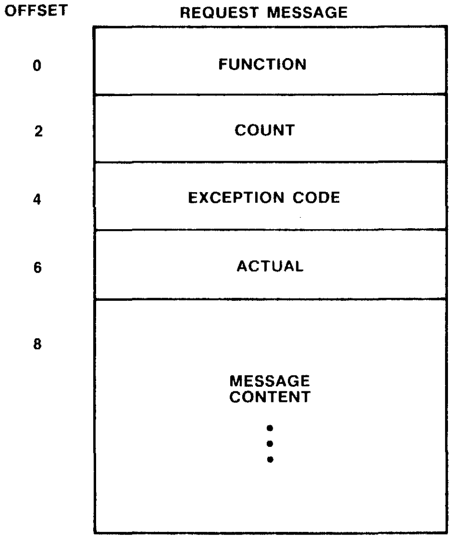

Figure 3-2. Request Message Format

In the following discussions, the names F$WRITE and F$READ are literal names for the particular WORD values 5 and 1, respectively.

OUTPUT

The first thing a task does when transmitting output is prepare an output request message. The task must fill in the following fields prior to sending the message:

FUNCTION --- F$WRITE.

COUNT --- the number of bytes (not to exceed 65527) in the MESSAGE CONTENT field.

MESSAGE CONTENT --- the bytes that are to be output.

Having prepared the message segment, the task must send it to the output request mailbox. Messages sent to this mailbox are processed in a first-in-first-out manner. Processing a message involves sending the characters in the MESSAGE CONTENT field to the terminal until a total of COUNT characters have been sent. There is one exception; when the Terminal Handler encounters the end-of-transmission character (OAH), it sends a Carriage Return and a Line Feed to the terminal.

When sending the output request message, the task specifies a user-supplied response mailbox. If no response mailbox is specified, the Terminal Handler will delete the segment that contained the message. In addition to transmitting the message to the terminal, the Terminal Handler fills in the remaining fields in the output request message. The requesting task can wait indefinitely at the response mailbox (that is, it can call the RQ$RECEIVE$MESSAGE system call with a time limit of OFFFFH) immediately after sending the output request. By observing this protocol, the task can learn of the success or failure of the output attempt. The fields that provide this information are the following:

- EXCEPTION CODE --- the encoded result of the output operation:
    - E$OK --- the operation was successful.
    - E$PARAM --- the FUNCTION field in the message did not contain F$WRITE.
    - E$BOUNDS --- the COUNT field in the message is too big for the segment, that is, COUNT + 8 is greater than the length of the segment containing the message.
- ACTUAL --- the actual number of bytes output.

In summary, the protocol observed by tasks doing output is as follows:

- Prepare the output request message segment, filling in the FUNCTION, COUNT, and MESSAGE CONTENT fields.
- Send the segment, via the RQ$SEND$MESSAGE system call, to the output request mailbox. It is advisable, but not necessary, to specify a response mailbox in the system call.
- Wait indefinitely, via the RQ$RECEIVE$MESSAGE system call, at the response mailbox. When received, the message contains the results of the transmission in the EXCEPTION CODE and ACTUAL fields.

INPUT

The protocol for obtaining input is much the same as that for outputting. A message is prepared and sent to a request mailbox; then, after the data has been input, the message is received at a response mailbox. There is a significant difference, however, between input and output protocols. Because the input is contained in the message segment at the response mailbox, it is necessary to designate a response mailbox and then wait there.

CAUTION

> When multiple tasks use the same mailbox for input from the terminal, it is possible for a task to get input that is intended for another task.

A task needing input first prepares an input request message. It must fill in the FUNCTION and COUNT fields prior to sending its request. The FUNCTION field must contain F$READ. The COUNT field reflects the maximum possible number of input characters in the input message. The value of COUNT must not exceed 132; moreover, COUNT + 8 must not exceed the length of the input request message segment.

When sending the input request message, the task must specify the response mailbox in its call to RQ$SEND$MESSAGE. The Terminal Handler obtains characters from the terminal and places them in the MESSAGE CONTENT field. The message is terminated by an end-of-line character (Carriage Return, Line Feed, or ESCape). The lone exception is when the end-of-line character has been "normalized" by being preceded by a control-P; then the end-of-line character is treated as a normal character.

NOTE

> If more than COUNT characters are entered prior to the end-of-line character, the extra characters are ignored, and the terminal emits a beep tone.

After the message is complete, the Terminal Handler fills in the EXCEPTION CODE and ACTUAL fields as follows:

- EXCEPTION CODE --- the encoded result of the input operation, which is one of the following:
  - E$OK --- the operation was successful.
  - E$PARAM --- either the FUNCTION field in the message did not contain F$READ or the COUNT field was greater than 132.
  - E$BOUNDS --- COUNT + 8 is greater than the length of the message segment.
- ACTUAL --- the number of bytes actually entered and placed in the MESSAGE CONTENT field.

The requesting task must wait indefinitely (that is, it must make a RQ$RECEIVE$MESSAGE system call with a time limit of OFFFFH) at the designated response mailbox immediately after sending the input request.

In summary, the input protocol is as follows:

- Prepare the input request message segment, filling in the FUNCTION and COUNT fields.
- Send the segment, via the RQ$SEND$MESSAGE system call, to the input request mailbox. In the call, specify a response mailbox.
- Wait indefinitely, via the RQ$RECEIVE$MESSAGE system call, at the response mailbox. When received, the message segment will contain the results of the input operation in the MESSAGE CONTENT, EXCEPTION CODE, and ACTUAL fields.

INDEX

The primary reference of each multiple-page topic is underscored.

iRMX 86™ Termin
Referen
1

REQUEST FOR READER'S COMMENTS

Intel Corporation attempts to provide documents that meet the needs of all Intel product users. This
you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completenes
document.

1. Please specify by page any errors you found in this manual.

2. Does the document cover the information you expected or required? Please make suggest
improvement.

3. Is this the right type of document for your needs? Is it at the right level? What other types of docun
needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating. ____________

NAME ____________________ DATE ________
TITLE ____________________
COMPANY NAME/DEPARTMENT ____________________
ADDRESS ____________________
CITY ____________________ STATE ________ ZIP CODE ____

Please check here if you require a written reply. □

-IKE YOUR COMMENTS . . .

;ument is one of a series describing Intel products. Your comments on the back of this form) us produce better manuals. Each reply will be carefully reviewed by the responsible All comments and suggestions become the property of Intel Corporation.

Intel Corporation
5200 N.E. Elam Young Pkwy.
Hillsboro, Oregon 97123

S. Technical Publications

intel®

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) **987-8080**

Printed in U.S.A.