

MiniMINC

MiniMINC Supplement

August 1979

This document contains supplementary information on MiniMINC routines and devices. The information in the *MiniMINC Supplement* is not documented in the other MiniMINC manuals.

Order Number AA-H789A-TC

MiniMINC

VERSION 1.1

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

NORTHEAST/MID-ATLANTIC REGION

Technical Documentation Center
Cotton Road
Nashua, NH 03060
Telephone: (800) 258-1710
New Hampshire residents: (603) 884-6660

CENTRAL REGION

Technical Documentation Center
1050 East Remington Road
Schaumburg, Illinois 60195
Telephone: (312) 640-5612

WESTERN REGION

Technical Documentation Center
2525 Augustine Drive
Santa Clara, California 95051
Telephone: (408) 984-0200

digital equipment corporation • marlboro, massachusetts

First Printing, August 1979

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright © 1979 by Digital Equipment Corporation

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL
DEC
PDP
DECUS
UNIBUS
COMPUTER LABS
COMTEX
DDT
DECCOMM
ASSIST-11
MINC-11

DECsystem-10
DECtape
DIBOL
EDUSYSTEM
FLIP CHIP
FOCAL
INDAC
LAB-8
DECSYSTEM-20
RTS-8
DECSYSTEM-2020

MASSBUS
OMNIBUS
OS/8
PHA
RSTS
RSX
TYPESET-8
TYPESET-11
TMS-11
ITPS-10
MiniMINC

CONTENTS

PREFACE ix

CHAPTER 1 SERIAL ASCII PROGRAMMING 1

- GENERAL DISCUSSION 1
- SET_SERIAL: SET ATTRIBUTES OF SERIAL LINE 3
- CIN: COLLECT CHARACTER STRING INPUT 6
- COUT: SEND CHARACTER STRING OUTPUT 11
- WORKING WITH SERIAL ASCII APPARATUS 14

CHAPTER 2 DATA PROCESSING ROUTINES 19

- FFT: PERFORM FAST FOURIER TRANSFORM 19
- POWER: CALCULATE POWER SPECTRUM
- COEFFICIENTS 23

CHAPTER 3 PROGRAM CONTROL ROUTINES 25

- SCHEDULE: SCHEDULE PROGRAM RESPONSE TO A TIME
- EVENT 25
- PAUSE: SUSPEND PROGRAM EXECUTION 28

CHAPTER 4: GRAPHIC ROUTINES 33

- FIND_CURSOR: CHANGE AND RECORD CURSOR
- POSITION 33

CHAPTER 5 MISCELLANEOUS ROUTINES 37

- GET_CHAR: ACQUIRE SINGLE CHARACTER FROM
- TERMINAL 37

CHAPTER 6 CHANGING OPERATING MODES ON THE MiniMINC TERMINAL (SETUP) 39

CONTENTS

WHEN TO USE SETUP MODE	39
HOW TO USE SETUP MODE	39
CHAPTER 7 TROUBLESHOOTING	49
TROUBLESHOOTING PROCEDURE	50
EXECUTING THE MiniMINC SELF-TEST	51
CHAPTER 8 SYSTEM SPECIFICATIONS	55
MiniMINC CHASSIS	55
SYSTEM RELIABILITY	56
DRIVE PERFORMANCE	56
POWER REQUIREMENTS	57
DEVICE SIGNALS	58
FLEXIBLE DISKETTES	58
KEYBOARD/MONITOR WITH GRAPHICS DISPLAY (VT105)	59
Alphanumeric Terminal Characteristics	59
CHAPTER 9 OPTIONAL SERIAL ASCII EQUIPMENT	61
APPENDIX A USING THE PDP-11/150 SYSTEM EXERCISER	63
APPENDIX B VT105 INTERACTIVE GRAPHIC TEST PROCEDURE	67
INTRODUCTION	67
PREPARE FOR INTERACTIVE GRAPHIC TEST	68
TEST PROCEDURES	69
Test Graph 0, Shaded Graph 0, and Graph 0 Brands	69
Test Graph 1, Shaded Graph 1, and Graph 1 Brands	71
Test Horizontal Lines	71
Test Vertical Lines	72
Test Shade Line 0	72
Test Shade Line 1	73
Test Strip Chart 0	74
Test Strip Chart 1	75
Exit Graphic Test	75
INDEX	77

Figure	1. MiniMINC Serial Input/Output Ports	14
	2. Serial Input/Output Connector Pin Identification	15
	3. How Alaising Occurs	22
	4. Setup A Display	40
	5. Setup B Display	43
	6. MiniMINC Cable Connections	51
	7. Attaching the Serial Input/Output Cable	62
	8. System Response to Configuration Input	64
	9. Typical System Exerciser Display	66
	10. Graph Test Pattern	69
	11. Shaded Graph Test Pattern	70
	12. Brand Test Pattern	70
	13. Horizontal Line Test Pattern	71
	14. Vertical Line Test Pattern	72
	15. Shade Line Test Pattern	73
	16. Strip Chart Test Pattern	74

FIGURES

CONTENTS

TABLES

Table 1.	Maximum Character Rates for MiniMINC Serial I/O Ports	15
2.	MiniMINC Serial Input/Output Port Connector Signals	16

PREFACE

This supplement describes certain features of the MiniMINC system that are not documented in other MiniMINC manuals. It contains material on new graphic routines, serial ASCII programming, and on the connection of serial ASCII devices to a MiniMINC system.

The supplement also supplies troubleshooting information for MiniMINC devices.

Most of the features described in this supplement apply to graphic programming and to the use of the serial ASCII routines. Before reading this supplement, you should understand general properties of the MiniMINC system, including the use of MINC routines in programs.

Therefore, you should read this supplement after you have read Books 1-4 of your MiniMINC documentation set.

CHAPTER 1

SERIAL ASCII PROGRAMMING

GENERAL DISCUSSION

Serial ASCII programs allow a MiniMINC system to communicate with serial ASCII instruments. These instruments respond to ASCII characters received on a communication line. Many such instruments can also return information to the MiniMINC system, and this information is also transmitted as a series of ASCII characters.

Serial ASCII instruments are used in a very wide range of applications and include such devices as interactive plotters and analytical balances. Not all instruments interpret particular ASCII characters the same way. The character LF (ASCII 15) is commonly used by line printers and terminals to control the movement of paper; on other, unrelated devices, this character might be used to activate a different function or it might be transmitted from the device to MiniMINC to indicate a condition peculiar to that device. A particular device's interpretation of characters forms the "protocol" for that device.

Because individual serial ASCII instruments have device-specific protocols, this manual cannot generalize about device protocols. Each type of instrument comes with its own user's manual, which defines the instrument's protocol.

The section "Working with Serial ASCII Apparatus" describes the physical connection of serial ASCII devices to a MiniMINC system. The only general requirement for connecting such a device to MiniMINC is that the device conform to the RS-232C standard for serial interfaces. The specifications and user documentation for serial ASCII devices will state explicitly whether

the device conforms to RS-232C. If a device conforms to this standard, "Working with Serial ASCII Apparatus" gives you specific procedures for connecting the device to MiniMINC.

As many as three serial ASCII devices can be connected simultaneously to a MiniMINC.

After you have connected a device to your MiniMINC system and have learned the device protocol, you can write serial ASCII programs with a set of special MiniMINC routines:

1. SET_SERIAL specifies the speed at which characters are transmitted to or from a particular device, the number of bits per character, and the parity scheme (ODD, EVEN, or NONE) used by the device to verify received characters. SET_SERIAL also defines the speed at which characters can be printed by a line printer device (LP:) connected to the MiniMINC.
2. CIN receives characters from a connected serial device.
3. COUT sends characters to a connected serial device.

NOTE

In setting transmission speeds (baud rates), the SET_SERIAL routine accepts baud rate parameters as high as 19200. However, as discussed in the section "Working with Serial ASCII Apparatus," the maximum speed at which characters move through the three serial ports is 240 characters per second.

A high baud rate, such as 9600, may be required to make an electronically valid connection to certain devices. In such a case, the SET_SERIAL argument must be 9600, and the signals that represent an individual character will be sent at that rate. However, no more than 240 of such characters can be sent in a second.

Therefore, baud rates do not translate consistently into a number of characters transmitted per second. A baud rate of 300 means that 30 characters are transmitted per second *while transmission is in progress*. If characters are transmitted in bursts, and not continuously, the net transfer rate (or "throughput") will be lower.

SET_SERIAL: SET ATTRIBUTES OF SERIAL LINE

Operation

Some serial ASCII instruments require a specific set of attributes for their physical communication line to a computer. A familiar example of such an attribute is the baud rate.

SET_SERIAL is used to describe to the MiniMINC system the attributes of a particular serial line. The serial line ports are labeled TERMINAL #1, TERMINAL #2, and TERMINAL #3 on the back of your MiniMINC chassis.

You can also use SET_SERIAL to set the baud rate and parity of the serial port labeled PRINTER. Depending on the type of output device you connect to the PRINTER port, you may need to change the port's baud rate from its default speed of 300 baud.

A single SET_SERIAL statement sets the attributes of a single serial port. To set attributes for several ports, use an equal number of SET_SERIAL statements.

SET_SERIAL(option,port-designator,baud-rate,bits)

Statement Form

option This argument is a string expression consisting of a single option word enclosed in quotation marks, or consisting of several option words separated by commas, with the whole series enclosed in quotes. The valid option words are as follows.

Argument Descriptions

1. If you omit the option string, NONE and LP: are selected by default.

If you include a port-designator and omit the option string, the default becomes simply NONE.

LP: directs SET_SERIAL to the PRINTER port. If you specify LP:, you cannot specify a port-designator argument, because there is only one PRINTER port.

2. ODD sets the designated port to odd parity. Some instruments require characters to be transmitted and received with odd parity.
3. EVEN sets the designated port to even parity.
4. NONE sets the designated port to no parity.

port-designator This is an integer from 1 to 3, specifying a single port from the group labeled TERMINAL #1, TERMINAL #2, and TERMINAL #3. Recall that you cannot specify a port-

designator and the LP: option in the same SET_SERIAL statement.

If you omit the port-designator argument, a value of 1 is used by default.

baud-rate This is the baud rate at which the designated serial port will operate. The only valid values are 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, and 19200.

If you omit the baud-rate argument, a rate of 300 baud is set by default.

bits This gives the number of bits per character. The only valid values are 5, 6, 7, and 8. Note that this argument specifies the number of bits actually used to represent the character code.

If you omit the bits argument, the value 8 is used by default.

NOTE

At 110 baud, an additional two bits are appended to the character code as stop bits. At all other baud rates, one bit is used as a stop bit. Character codes are also preceded by a single start bit. Consequently, characters are ordinarily represented by 10 bits in a digital stream (1 start bit + 8-bit character code + 1 stop bit). This information may be useful in calculating the character throughput for a particular application. See the discussion of the CIN routine for further information.

Related Routines

1. CIN and COUT transmit characters through the TERMINAL ports. Use SET_SERIAL to designate the port's attributes before using CIN or COUT.
2. Because SET_SERIAL can change the baud rate of the PRINTER port, it can affect the speed with which information is printed. Commands that send information to the printer include SAVE, COPY, and DIR. The OPEN statement can also open a file called "LP:". Information subsequently output to the file "LP:" is directed to the PRINTER port and will be printed on a line printer if one is connected. For more information on SAVE, COPY, DIR, and OPEN, see *Book 3: MINC Programming Reference*. Further in-

formation on MiniMINC line printers is provided in this manual.

1. All of MiniMINC's serial ports have physical limitations on transmission speed. For instance, the three TERMINAL ports have a maximum character throughput of 240 characters per second. However, to form an electronically valid connection between MiniMINC and a particular instrument, you may need to specify, with SET_SERIAL, a high baud rate such as 9600.

A table of the maximum character rates is given in the section "Working with Serial ASCII Apparatus."

Restrictions

?MINC-F-Invalid or conflicting options requested

You included some option word other than LP:, EVEN, ODD, or NONE. Reenter the SET_SERIAL statement with valid options.

Errors

?MINC-F-Serial line number must be in range 1-3

You entered a port-designator that was neither 1, 2, nor 3. Reenter the SET_SERIAL statement with a valid port-designator.

?MINC-F-Illegal baud rate specified

You entered a baud-rate that was not a valid value. Reenter the statement with a valid baud rate.

?MINC-F-Illegal number of bits specified

You entered a bits argument that was neither 5, 6, 7, nor 8. Reenter the statement with a valid value.

?MINC-F-Specify either LP: or serial line number, but not both

You specified the LP: option and a port-designator in the same statement. Reenter the statement. If you want to set attributes for a printer, omit the port-designator. If you want to set attributes for TERMINAL #1, TERMINAL #2, or TERMINAL #3, omit the LP: option; you may also need to specify the ODD or EVEN option in this case.

SET_SERIAL("LP:",1200)

sets PRINTER port to operate with a 1200-baud printer (no parity).

SET_SERIAL("LP:")

resets PRINTER port to 300 baud.

SET_SERIAL

same as SET_SERIAL("LP:")

Examples

SET_SERIAL("ODD",1,9600,7)

sets TERMINAL #1 port to operate with a 9600-baud device, using odd parity and 7 bits per character. Note that because of physical limitations of TERMINAL ports, character transmission will actually occur at a maximum of 240 characters per second.

SET_SERIAL(.1)

sets TERMINAL #1 port to no parity, 300 baud, and 8 bits per character.

CIN: COLLECT CHARACTER STRING INPUT

Operation

CIN receives characters transmitted in serial ASCII format via a serial transfer channel.

MiniMINC systems have five serial channels. The ports for these channels are located on the back of the MiniMINC chassis, and they are labeled CONSOLE, PRINTER, TERMINAL #1, TERMINAL #2, and TERMINAL #3. The CONSOLE port should be used only to connect the MiniMINC VT105 terminal to the chassis. The PRINTER port should be used only to connect a line printer. The PRINTER port is identified by the logical name LP: in such statements as SAVE LP:.

Use the TERMINAL ports to connect serial ASCII instruments to MiniMINC, following the instructions in the section "Working with Serial ASCII Apparatus." Use the CIN routine to collect characters from the TERMINAL ports only.

MiniMINC systems also have a port labeled MODEM. This port is not used by MiniMINC; do not connect instruments to it. The MODEM port is used by optional communication software not discussed in the MiniMINC manuals.

Statement Form

CIN(option,string-name,string-length,port,timeout-interval)

Argument Descriptions

option The only valid option is the word RETRIEVE. The RETRIEVE option retrieves any characters arriving since the last CIN statement. The retrieved characters are assigned to the input string variable named by the string-name argument. If there was no previous CIN statement, there will be no characters to retrieve.

When you use the RETRIEVE option, a maximum of 128 previous characters can be retrieved. The first character assigned to the input string is the first one received since the last CIN statement. If more than 128 characters have arrived since the

last CIN statement, CIN retrieves 128 characters and then halts with a fatal error.

If you omit the option word, the first character assigned to the string is the first character that arrives after the current CIN statement begins executing. Characters arriving prior to the current CIN statement are lost.

In RETRIEVE mode, CIN collects as many characters as specified by the string-length argument. For example, if the string length is 10, then CIN collects 10 characters starting with the first character after the most recent CIN statement. If more than 10 characters have already arrived, CIN still collects only 10 characters; the surplus characters can be retrieved by the next CIN statement. If fewer than 10 characters have arrived, CIN collects all the characters and waits for the remaining characters necessary to make a string of 10.

When you specify a variable-length string (string-length=0) in RETRIEVE mode, CIN continues collecting characters until one of the following conditions occurs:

- A carriage-return character (ASCII 13) is received.
- The timeout interval expires.
- The MINC BASIC string limit of 255 characters is reached.
- The limit of 128 previous characters is reached (error condition occurring when more than 128 characters have arrived since the last “non-RETRIEVE” CIN statement).

string-name The name of the string variable to which collected characters are assigned. This argument must be either a string variable, such as S\$, or an element of a string array, such as S\$(1).

You cannot collect an array of strings with a single CIN statement.

You cannot omit the string-name argument from a CIN statement.

string-length The length of the character string to be collected. Valid values are 0, and the range of integers from 1 to

255. The use of **RETRIEVE** mode imposes certain restrictions on the length of a received string. See “Restrictions” for this routine.

A value of 0 specifies a variable string length. For variable strings, CIN collects characters until the timeout interval is reached, or the maximum of 255 characters is reached, or the carriage return character is received. The carriage return character itself does not become part of the collected string, nor is it included in the character count. Note that if your instrument terminates its messages with a carriage return and a line feed, your program will always see the line feed as the beginning of the next message.

Values in the range 1-255 specify fixed-length strings. When collecting characters for a fixed-length string, CIN recognizes no particular character as a terminator. Therefore, such characters as the carriage return can become part of the collected string and therefore are included in the character count.

The timeout interval is especially useful in collecting fixed-length strings. When collecting characters for a fixed-length string, CIN continues executing until the number of requested characters has arrived and been collected. Therefore, the timeout interval assures that CIN finishes executing in case, for some reason, the routine never receives a sufficient number of characters.

port The serial port carrying input characters. Valid values are 1, 2, and 3, designating the ports labeled **TERMINAL #1**, **TERMINAL #2**, and **TERMINAL #3**, respectively.

Before using CIN, be sure to use the **SET_SERIAL** routine to select the correct attributes for the designated port.

If you omit the port argument, a value of 1 (designating **TERMINAL #1**) is used by default.

timeout-interval The time interval after which CIN considers the current string to be complete. Valid values are 0 and the range of decimal numbers greater than or equal to 0.1.

A value of 0 specifies no timeout.

A value greater than or equal to .1 specifies the timeout interval in seconds. The interval .1 seconds is the smallest interval

guaranteed to be precise. The resolution of the timeout interval is one tick of the system clock.

If you omit the timeout-interval argument, a value of 0 (no timeout) is used by default.

1. COUT sends character strings to an instrument connected to a serial port. COUT can send strings of fixed or variable length.
2. SET_SERIAL defines the attributes of a particular serial port. These attributes are the baud rate, the number of bits per character, and the type of parity (ODD, EVEN, or NONE). These attributes must be set to match the specifications of a particular instrument.

Related Routines

In a program that transfers serial ASCII characters, the attributes of the serial port must be set *before* a CIN statement attempts to receive characters from the port. In such a program, be sure that the SET_SERIAL statement is executed before any CIN or COUT statement.

SET_SERIAL has default values for all its arguments. However, it is good practice to include the required values explicitly in the SET_SERIAL statement. If you include the values explicitly, the resulting program will be easier for you and other users to interpret and modify.

Include a SET_SERIAL statement in any program that uses CIN and/or COUT. Do not assume that the attributes of a port are “already” set to the correct values, even if you have set the attributes in a previous program.

3. The SAVE, OPEN, and COPY statements allow you to list programs or print data on a line printer. In these statements, the line printer is identified by the logical name LP:. To work with these statements and with the logical name LP:, the line printer must be connected to the PRINTER port. CIN and COUT cannot send characters to the PRINTER port.

SAVE, OPEN, and COPY are related to CIN and COUT in that they, too, allow the serial transfer of ASCII characters to a connected device. However, they cover the special case of line printers. CIN and COUT should be used instead for other types of serial transfer.

Restrictions

1. A CIN statement with the RETRIEVE option can retrieve up to 128 characters received since the completion of the most recent CIN statement. Therefore, you can use CIN to collect an essentially continuous stream of character input because you can process strings as they arrive without losing intervening characters.

The 128-character maximum also imposes a practical limit on the sustainable rate at which characters can be input. At 9600 baud, 128 characters can arrive in approximately 135 milliseconds. At 1200 baud, 128 characters can arrive in approximately 1 second.

To establish the actual character throughput for a given application, divide the baud rate by the number of bits per character plus stop bits (usually 10)—this gives the number of characters that the TERMINAL port can handle per second (“hardware” throughput). Figure the time taken by your program to process 128 characters and divide the time by 128—this gives the number of characters that the program can handle per second (program throughput). The character throughput is the hardware throughput, program throughput, or 240, whichever is lowest.

2. CIN does not echo received characters. That is, CIN does not automatically transmit a received character back to the sender for display or verification.
3. If you direct a CIN statement to a TERMINAL port to which no device is connected, the system may appear to be malfunctioning or inactive. Type two CTRL/C commands to exit from the CIN statement.

Errors

?MINC-F-Channel or unit # not in system for the routine

?MINC-F-Data lost—transfer rate too high

You specified an invalid channel number (the port argument) in the statement.

More than 128 characters have arrived since the last CIN statement finished executing (RETRIEVE mode).

The characters are arriving too quickly for CIN to receive them.

Characters are arriving at the wrong baud rate. (Be sure that you have set the baud rate correctly with a SET_SERIAL statement.)

?MINC-F-Invalid or conflicting options requested

You entered an invalid option word in the CIN statement. If you include an option word, RETRIEVE is the only valid option.

?MINC-F-No workspace available for the string specified

The string you specified in the string-name argument requires more memory than is available in the workspace. Reduce the size of the program if possible, or use the EXTRA_SPACE command.

?MINC-F-Use array element instead of array for argument #

You specified an array name for the string-name argument. The string-length argument can accept input for a single string only; therefore, the string-length argument must be either the name of a single string or the name of an *element* of a string array, such as C\$(1).

None.

Examples

COUT sends characters in serial ASCII format via a serial transfer channel.

**COUT: SEND
CHARACTER
STRING OUTPUT**

MiniMINC systems have five serial ASCII channels. The ports to the channels are labeled CONSOLE, PRINTER, TERMINAL #1, TERMINAL #2, and TERMINAL #3 on the back of the MiniMINC chassis. The chassis also has a port called MODEM that is not used by MiniMINC. The MODEM port is used by optional communication software that is not discussed in the MiniMINC manuals.

Operation

The CONSOLE and PRINTER ports are reserved for use by the MINC VT105 terminal and by line printers, respectively. COUT sends characters to instruments via the TERMINAL ports.

See section "Working with Serial ASCII Apparatus" for instructions on how to connect serial ASCII devices to the TERMINAL ports.

COUT(option,string-name,string-length,port)

Statement Form

option The character string selecting an optional mode for COUT.

**Argument
Descriptions**

If you include the option word WAIT, COUT pauses and waits for the transfer to finish before allowing the next program statement to execute.

If you do not include an option word, COUT starts the transfer and passes control to the next statement when at most 32 characters remain to be sent. In this mode, the transfer will finish while the program continues.

string-name The variable name for the data sent. COUT requires that the string-name argument be a string expression. You cannot send a complete string array with a single COUT statement.

You cannot omit the string-name argument from a COUT statement.

string-length The number of characters to send. Valid values are 0 and integers in the range 1-255.

A value of 0 sends as many characters as are contained in the string, up to the string maximum of 255 characters. COUT appends a carriage return to the transmitted string when the string-length is 0.

Values in the range 1-255 send a fixed number of characters from the string; no carriage return is appended. The transmission begins with the first character in the string. If the string does not contain enough characters, COUT sends only as many as are in the string.

If you omit the string-length argument, a value of 0 (variable-length string) is used by default.

port The port to the serial channel that carries output characters. Valid values are 1, 2, and 3, designating the ports labeled TERMINAL #1, TERMINAL #2, and TERMINAL #3, respectively.

Be sure that the port you designate has been assigned the correct attributes by a SET_SERIAL statement.

If you omit the port argument, TERMINAL 1 is designated by default.

Related Routines

1. CIN collects character strings from an instrument connected to a serial channel. CIN can receive strings of fixed or variable length.
2. SET_SERIAL defines the attributes for a serial port that is then designated in a COUT statement. When you use COUT

in a program, include the appropriate `SET_SERIAL` statement, and be sure that `SET_SERIAL` is executed before `COUT`. Do not assume that the designated port is "already" set to the required attributes, even if you have done so in a previous program.

3. The `SAVE`, `OPEN`, and `COPY` statements allow you to list programs or print data on a line printer. In these statements, the line printer is identified by the logical name `LP:`. To work with these statements and with the logical name `LP:`, the line printer must be connected to the `PRINTER` port. `CIN` and `COUT` cannot send characters to the `PRINTER` port.

`SAVE`, `OPEN`, and `COPY` are related to `CIN` and `COUT` in that they also allow the serial transfer of ASCII characters to a connected device. However, they cover the special case of line printers. `CIN` and `COUT` should be used instead for other types of serial transfer.

1. When you omit the option word from a `COUT` statement, `COUT` can finish executing when up to 32 characters remain to be transmitted. A second `COUT` statement, again without the option word, can start executing before the first transfer is complete. However, the second statement cannot finish executing until the total number of characters remaining from both statements is less than or equal to 32.

Restrictions

?MINC-F-Channel or unit # not in system for the routine

You specified an invalid channel number (the port argument) in the routine statement. The port argument must have the value 1, 2, or 3.

?MINC-F-Invalid or conflicting options requested

You entered an invalid option word in the `COUT` statement. If you include the option word, `WAIT` is the only valid choice.

?MINC-F-Use array element instead of array for argument #

The string-name must be a string expression.

?MINC-F-Serial output channel # is not ready.

You specified a channel (port) that was not ready, such as a port to which no device is connected. This message may also occur if a connected device is not asserting the required DTR (Data Terminal Ready) signal.

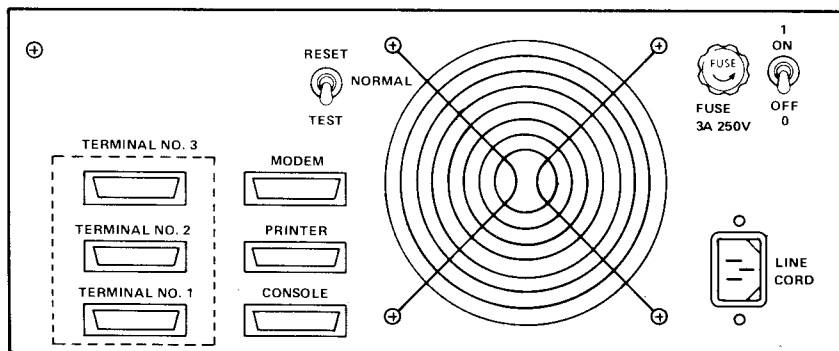
None.

Errors

Examples

WORKING WITH SERIAL ASCII APPARATUS

All communication between the MiniMINC system and external apparatus occurs through the serial input/output ports on the rear of the MiniMINC chassis (see Figure 1). These ports provide EIA standard signals and are not suitable for communicating with ASR 33/35 teletypewriters and other equipment requiring 20 milliamperes current loop interfaces.



MR-S-196-79

Figure 1. MiniMinc Serial Input/Output Ports

One of these ports, labeled MODEM, provides all the control lines necessary to transmit and receive data via modulator/demodulator apparatus (modems) of the kind provided by common carriers and others for long distance transmission of digital data.

NOTE

The MODEM port is not used by the MiniMINC system. It cannot be addressed by the routines SET_SERIAL, CIN, and COUT. The MODEM port is used by optional communication software that is not discussed in the MiniMINC manuals.

A second port, CONSOLE, is dedicated to the MiniMINC terminal.

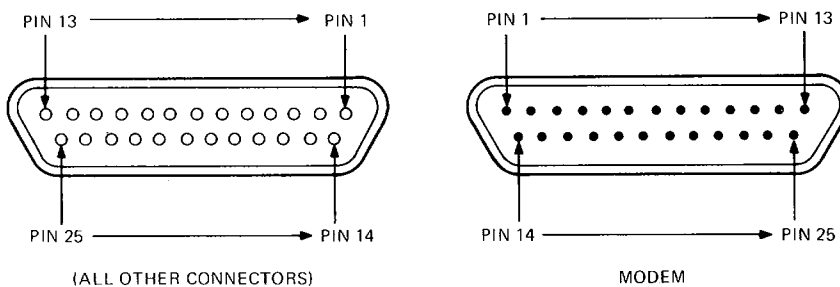
The remaining four ports (PRINTER, TERMINAL #1, TERMINAL #2, and TERMINAL #3) are functionally identical except for their maximum operating speeds and are capable of communicating with most terminals, printers, and other apparatus designed to operate with serial ASCII code. The speed of all input/output ports (except MODEM and CONSOLE) can be set to any permitted rate with the SET_SERIAL routine.

The serial input/output ports are capable of communicating in either full- or half-duplex mode at the rates specified in Table 1. Note, however, that each rate assumes no competition from other devices simultaneously communicating with the system. The actual maximum rates for all devices on a system will depend on how many other devices are competing for attention, how fast the other devices are operating, what mode they are operating in (half or full duplex), whether they operate at a continuous pace or in bursts, and so on.

Table 1. Maximum Character Rates for
MiniMINC Serial I/O Ports

<i>I/O Port</i>	<i>Maximum Rate (char/sec)</i>
Console	960
Printer	960
Terminal 1	240
Terminal 2	240
Terminal 3	240

When printers and terminals manufactured by Digital Equipment Corporation are involved, no special cables are required. If the cable provided with the unit mates with the desired input/output port, it should make the necessary connections without further user manipulation. When apparatus not manufactured by Digital is to be used, it may be necessary to make up custom cables. Figure 2 and Table 2 provide the required information.



MR-S-197-79

Figure 2. Serial Input/Output Connector Pin Identification

Table 2. MiniMINC Serial Input/Output
Port Connector Signals

Connector	Signal Designation	Device Group
Console (female)		
pin 2	Term Xmit Data	LA38 DECwriter IV,
pin 3	Term RCV Data	LA34 DECwriter IV,
pin 20	Term RDY	LA36 DECwriter II,
pin 1	Chassis GND	VT50 Decscope,
pin 7	Signal GND	VT100 video terminal,
		VT105 graphic terminal,
		VT52 DECscope
Printer (female)		
pin 2	LP Xmit Data	LA35 DECwriter II,
pin 3	LP RCV Data	LA36 DECwriter II,
pin 20	LP RDY	LA38 DECwriter IV,
pin 1	Chassis GND	LA34 DECwriter IV
pin 7	Signal GND	
Modem (male)		
pin 2	Modem RCV Data	
pin 3	Modem Xmit Data	
pin 12	SEC Carrier Detect	
pin 5	PRI CTS	
pin 22	Ring Indicator	
pin 8	Carrier Detect	Bell System type:
pin 6	Data Set RDY	103, 113, 202, 212
pin 11	SEC RTS	or equivalent
pin 4	PRI RTS	
pin 20	DTR	
pin 17	SYN CLK R	
pin 15	SYN CLK T	
pin 1	Chassis GND	
pin 7	Signal GND	
Terminal 1,2,3 (female)		
pin 2/3	Term 1 Xmit Data	LA36 DECwriter II,
pin 3/2	Term 1 RCV Data	VT50 DECscope,
pin 20	Term 1 RDY	VT52 DECscope,
pin 7	Signal IN (GND)	LA34 DECwriter IV,
pin 1	Chassis IN (GND)	LA38 DECwriter IV,
		VT100 video terminal
		VT105 graphic terminal

is per bevestiging
volgens -7

NOTE

A typical serial ASCII protocol includes the signals DSR (Data Set Ready), DTR (Data Terminal Ready), CTS (Clear to Send), and RTS (Request to send). Please note the following exceptions for MiniMINC:

1. MiniMINC requires DTR but does not supply DSR.
2. MiniMINC neither requires nor provides CTS or RTS.

After a reverse transform, you must divide each element of the resulting arrays by the data-length argument.

data-length The number of points in each of the component arrays that are used as input to FFT. Valid values are powers of 2 that range from 8 to 2048. The real-component and imag-component arrays must each contain at least this number of elements. (The actual maximum array length depends on the amount of available workspace; see the LENGTH command in Book 3.)

real-component The name of the integer array containing the real component of the (complex) input data that is to be transformed. FFT replaces the input data in this array with the calculated frequency coefficients.

The nature of the FFT algorithm requires that the data satisfy certain conditions. See “Restrictions” for the FFT routine.

imag-component The name of the integer array containing the imaginary components of the (complex) input data to be transformed. FFT replaces the input data with the calculated frequency coefficients.

The nature of the FFT algorithm requires that the data satisfy certain conditions. See “Restrictions” for the FFT routine.

If the input data are real, then all elements of the imag-component array should be 0.

scale-factor The integer scale factor used to scale the real and imaginary components of the result to provide the final result.

For efficient calculations, FFT requires integer input arrays. However, because integer values are limited in range, FFT requires a mechanism for keeping the data values within the integer range during its calculations. It does this by dividing all the values by 2 whenever necessary to keep within the range. Therefore, the results are proportionally correct but need to be scaled in order to restore the original absolute values. After FFT has finished, the scale-factor argument contains the number of divisions. Create final results in two real arrays by multiplying each element of the integer result arrays by 2 to the power n (where n is the scale factor).

Related Routines

1. POWER calculates the power spectrum of a set of data by using the real and imaginary coefficient arrays calculated