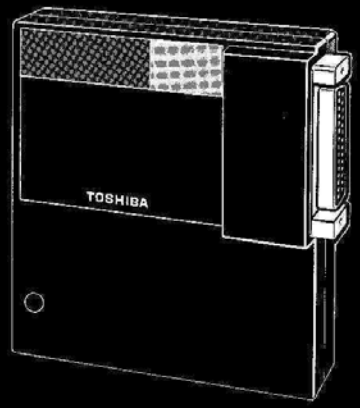
TOSHIBA

MSX

RS232C INTERFACE CARTRIDGE OWNER'S MANUAL



MODEL **HX-R700PE**

Scanned by abuur, converted to pdf by HansO, 2009

PREFACE

Your new Toshiba RS232C interface cartridge HX-R700 has been designed for long life and trouble-free service.

- Your new RS232C interface cartridge features built-in extended BASIC to support the RS232 communication functions in addition to the interface circuit. The RS232C, extended BASIC conforms with MSX standards and is compatible with that of other manufacturers.

- Therefore, your RS232C interface cartridge can be used not only with the Toshiba Home Computer HX-10 series but also with all other MSX computers.

- For details on the syntax and coding conventions of the RS232C communication functions, refer to the MSX BASIC REFERENCE MANUAL. Extended statements can be called using the CALL statement. The call statement or its underlined abbreviated form (-) is supported.

NOTICE

1. The contents of this manual are subject to change without prior notice.
2. When using a special application program or computation procedure, it is advisable that the execution sequence, intermediate results, and final results should be checked out carefully.
3. We are not responsible for any financial loss or lost profit which might result from the use of this computer.

MSX is a registered trademark of MICROSOFT CORPORATION
(c) 1984 by TOSHIBA CORPORATION

CONTENTS

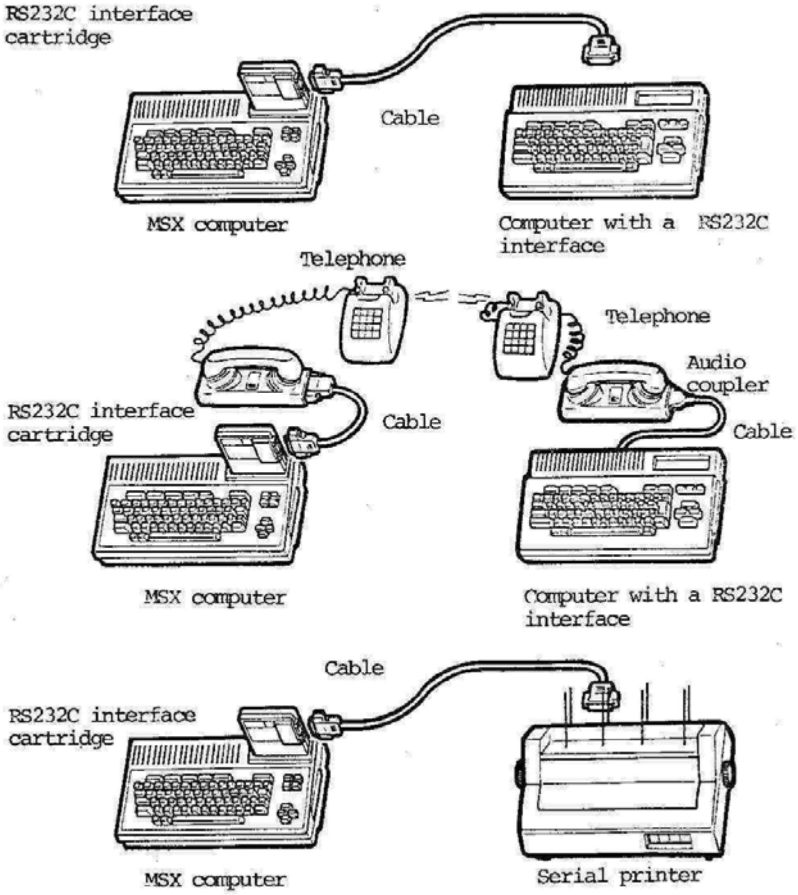
PAGE

CHAPTER 1

PREPARATIONS FOR USE

1. SYSTEM CONFIGURATION

The Toshiba RS232C interface cartridge is designed to enable MSX computers to communicate with other peripheral devices (computers, audio couplers, serial printers, etc).
Signal assignment and communication speed varies with the type of device to be connected as well as with the program. Even if the device at the destination supports communication speeds up to 19200BPS, such speeds may not be realized in actual practice due to limitations placed on communication methods, control signals etc. The specifications of both the devices should be checked to ensure optimum performance.


RS232C interface cartridge
Cable
MSX computer
Computer with a RS232C interface
Telephone
Telephone
Audio coupler
RS232C interface cartridge
Cable
Cable
MSX computer
Computer with a RS232C interface
Cable
RS232C interface cartridge
MSX computer
Serial printer

2. SPECIFICATIONS

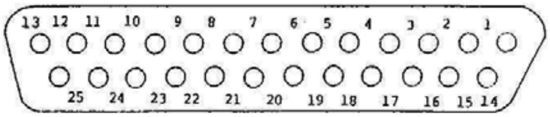
Standard specifications

Input voltage	+5V, +12V, -12V
Power consumption	150mA (+5V), 50mA (±12V)
Temperature and humidity	5～35°C . 20～80%
Dimensions	117mm x 26mm x 120mm
Weight	150g

Communication specifications (RS232C)

Synchronization method	Start-stop transmission method (Asynchronous method)
Communication speed	50. 75. 110. 300. 600. 1200. 1800. 2000. 2400. 3600. 4800. 7200. 9600. 19200 BPS (Transmission and reception speeds may be different)
Start bit length	1 bit
Character length	5, 6, 7, 8 bits
Parity bit	Even/Odd/No parity
Stop bit	1, 1.5, 2 bits
Signal level	ON +5～+12V OFF -5～-12V
Communication method	Full duplex
LSI	Programmable Communication Interface Conforms with I 8251 Programmable Interval Timer Conforms with I 8253

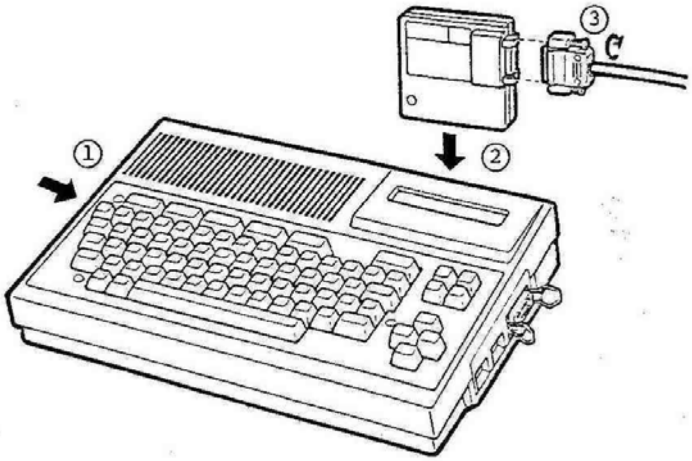
RS232C 25 pin D-type connections



Pin arrangement

Pin number	Signal number	Functions	Signal direction
1	FG	Chassis ground	
2	SD (TXD)	Transmit data line	Output
3	RD (RXD)	Receive data line	Input
4	RS (RTS)	Signal notifying the terminal device that the host computer is ready to start transmission	Output
5	CS (CTS)	Signal notifying the host computer that the terminal device is ready to receive data and control signals	Input
6	DR (DSR)	Signal notifying the host computer that the terminal device is ready to receive data and control signals	Input
7	SG	Signal ground	
8	CD	Signal notifying the host computer that the terminal device is receiving carriers	Input
20	ER (DTR)	Signal notifying the terminal device that the host computer is ready to transmit data and control	Output

3. CONNECTING THE RS232C INTERFACE



① Switch the computer OFF

② Insert the cartridge with the notched side to the rear.

③ Connect the RS232C cable

(Table 1 shows the connections required at the RS232C end.)

Note: As a number of different devices can be connected to the Toshiba RS232C interface, it is very important to check the manual of the peripheral device you intend to connect to the RS232C RTS.

CHAPTER 2

RS232C COMMUNICATION FUNCTIONS

1. INTRODUCTION TO THE RS232C

This cartridge features an interface which conforms with the RS-232-C standard and is designed for serial data transfer. The RS-232-C standard refers to the standard for communication with modems recommended by the Electronic Industries Association (EIA). This standard corresponds to JIS standard C6361-71 in Japan. Hereafter, this standard will be referred to as the RS232C for convenience.
The RS232C, in addition to being used for communication between computer systems through modems, is increasingly being used for direct communication between computers and with various peripheral devices.

BEFORE USING THE RS232C INTERFACE CARTRIDGE IT IS RECOMMENDED THAT THE USER SHOULD FULLY ACQUAINT HIMSELF WITH ITS FUNCTIONS AND ALSO CHECK THE SPECIFICATIONS AND REQUIREMENTS OF THE DEVICE WITH WHICH DATA COMMUNICATION IS TO BE CARRIED OUT.

The following are examples of communication via the RS232C:

- As a terminal

  By setting the computer to the terminal mode, it can be used as a display console and keyboard of another computer (Host computer).

- Program transfer

  Allows you to transfer programs to other computers.

- Data transfer

  Allows you to communicate (data transfer) with other computers using BASIC.

- Communications between computers

  Via a modem or acoustic coupler.

2. INITIALIZATION OF RS232C COMMUNICATION FUNCTIONS

(1) Synchronous mode

Synchronization (timing) is the convention between the send and receive sides with reference to the definition of a character (definition of a bit string between two specified bits which is considered to be a character). It is necessary to let the receive side know the format of the data being sent.
The synchronous method may be further classified into synchronous and start-stop (asynchronous) methods. As this interface supports the start-stop method, the terminal device must also be set to this method.

Synchronous method:

Prior to starting transmission, a synchronization code is sent to synchronize the two devices.

Start-stop method (Asynchronous method):

In this method, a start and stop bit is sent at the beginning and end of each character to synchronize the two devices.

The receive side assumes the bit string following the first bit (start bit; value 0) is one data item. The start bit consists of a single bit. However, the length of a character (bit number) is defined in the program. At the end of 1 character, a stop bit of specified length is received (value 1). The system now enters the Wait state, pending receipt of the next start bit. A slight delay between the above will not cause any problems. However, a bit with value 0, if transmitted between stop bits, will cause errors because it will be assumed to be a start bit.

(2) Communication methods

Two communication methods are available. They are full-duplex and semi-duplex. As this cartridge supports the full-duplex method, the terminal device must also be set to this method.
Semi-duplex: unidirectional transmission
Full-duplex: bi-directional transmission
There are considerable differences between the semi-duplex and full-duplex systems, particularly in systems using control signals such as audio couplers etc. In such cases, unless full-duplex mode is specified, normal communication cannot be guaranteed.
Semi-duplex method: in this method communication is carried out by controlling (ON-OFF) the control signals CS, RS etc.
Full-duplex method: in this method the control signals CS, RS etc. remain ON throughout the communication process.

(3) Character codes

Standard ASCII codes are used for communication. A user is, therefore, recommended to check the compatibility of the character codes of the two systems carefully .

CHAPTER 3

RS232C COMMUNICATION EXTENDED BASIC

1. RS232C COMMUNICATION EXTENDED BASIC LANGUAGE TABLE

Additional commands are automatically added to MSX BASIC when the RS232C is plugged in. These extra functions have been provided to enable efficient communication through the RS232C interface cartridge. Commands, statements and functions are identical to those in MSX BASIC or MSX DISK BASIC.

Extended BASIC statements (using the CALL statement)

COMINI:	initializes communication
COMTERM:	selects the terminal mode
COMDTR:	controls (ON/OFF) the ER signal
COMBREAK:	transmits the break signal
COMSTAT:	checks the RS232C port status
COM GOSUB:	sets the interrupt routine start line from the RS232C port
COMON:	enables the interrupt signal from the RS232C port
COMOFF:	disables the interrupt signal from the RS232C port
COMSTOP:	holds the interrupt signal from the RS232C port

Commands

SAVE:	sends a program
LOAD:	receives a program
RUN:	executes the received program
MERGE:	merges the received with existing programs in the memory

Statements

OPEN:	opens an RS232C file
CLOSE:	closes an RS232C file
PRINT#:	sends characters and numeric data strings
PRINT# USING:	sends characters and numeric data strings in specified format
INPUT#:	receives numeric and character data strings and assigns them to variables
LINE INPUT#:	receives numeric and character data strings and assigns them to variables

Functions

EOF: checks if the EOF code (&H1A) has been received
INPUT$: receives character strings of 2 or more characters
LOC: calculates the number of data bytes in the receive buffer
LOF: calculates the number of remaining bytes in the receive buffer

Device number

This cartridge has only one RS232C port and therefore the device number has been permanently set to 0.

CLOSE Statement

Functions: closes an RS232C file

Format: CLOSE [[#] file number [, [#] file number..]]

Description:

(1) Closes the RS232C file specified by the file number. The file number, once closed, can be used to open another file.

(2) When multiple file numbers are specified, closes all the specified files.

(3) When the file number is omitted, closes all the files which are currently open.

(4) If the CLOSE statement is executed for a file which has been opened for output, the EOF code (&H1A) is output through the RS232C port.

(5) In addition to the CLOSE statement, files may be closed using the END, RUN, NEW, and CLEAR statements.

(6) The RS signal is turned OFF when the CLOSE statement is executed.

Example: CLOSE#1, #2

COM GOSUB Extended Statement

Functions: specifies a line number, the interrupt processing routine of which is to be executed when data are transferred to the RS232C port.

Format: CALL COM (["device number:"] , GOSUM line number)

Description:

(1) Specifies the interrupt processing routine start line when data are received at the RS232C port. When the CALL COMON statement is executed, data are transferred to the RS232C port causing an interruption. The interrupt routine of the specified line number is executed by this statement.

(2) An RS232C file must be opened in the input mode (using the OPEN statement) prior to executing the CALL COM GOSUB statement.

(3) Default device number is 0.

(4) Interrupt processing routine

The RETURN statement is used to return from the interrupt processing routine.
The system remains in CALL COMSTOP status while interrupt processing is being executed (Hold interrupt). It is reset to CALL COMON status when the RETURN statement is executed.

(5) The system is held in CALL COMSTOP status while the interrupt processing routine specified in the following statement is being executed. Interrupt priority is lower than the interrupt specified below:

```
ON ERROR GOTO
ON STOP GOSUB
ON STRIG GOSUB
ON KEY GOSUB
ON SPRITE GOSUB
ON INTERVAL GOSUB
```

Example: CALL COM (, GOSUB 1100)

COMBREAK Extended Statement

Functions: Generates the break signal

Format: CALL COMBREAK [(["device number"] [, integer expression])]

Description:

(1) Resets the time and the send data line (SD) for sending the number of characters specified in the integer expression to 0.
The time T for resetting the signal to 0 is calculated as follows:
T = (1 + character length + stop bit length)*(Value of integer expression)/communication speed

(2) The value of the integer expression may be specified in the range 3 to 32767. The default value is 10.

(3) The default device number is 0.

(4) When all data in parentheses are omitted, the parentheses themselves are omitted.

(5) An RS232C file must be opened prior to executing this statement. File OPEN mode is optional.

Example:
```
CALL COMBREAK (, 10)
CALL COMBREAK
```

COMDTR Extended Statement

Functions: controls (ON/OFF) the ER signal (DTR signal)

Format: CALL COMDTR (["device number"], integer expression)

Description:

(1) Turns the ER signal OFF when the integer expression is 0 and turns the signal ON when it is not zero. The ER signal is ON at POWER-ON.

(2) The default device number is 0.

(3) An RS232C file must be opened prior to executing this statement. File OPEN mode is optional.

Example: CALL COMDTR (,0)

COMINI Extended Statement

Functions: Initializes the RS232C communication functions

Format:

```
CALL COMINI (["[device number:] character length
parity    stop bit    X on/off control
CS control     receive autoLF     send autoLF
SI/SO"][, receive speed ,[ send speed][, wait
time]])
```

For details on omitting parameters, see Item 2 in the **following** description.

Description:

(1) Initializes the RS232C port communication functions. You may specify functions using either upper case or lower case characters.

Device number

The default device number is 0.

Character length

Specify a character length which matches the character length of the device with which data is to be communicated. You may specify character lengths (bits/character) in the range 5 to 8 using integers from 5 to 8.

5: 5 bit character
6: 6 bit character
7: 7 bit character
8: 8 bit character

The initialization value is 8. This is also the default value designation.

Parity

Specifies parity type.
Specify parity by one of the following alphabetic characters.

E: Even parity
O: Odd parity
I: Ignore parity in data receive operation. Sets parity bit to 0 in a data send operation. However, the character length is limited to 5-7 bits.
N: Communicates without adding a parity bit

The initialization value is N. The default value designation is also N.

Stop bit length

Specifies the length of the stop bit (number of bits).
Specify stop bit length in the range from 1 to 3.

1: 1 bit
2: 1.5 bits
3: 2 bits

The initialization value is 1. The default value designation is also 1.

X on/off control

Specifies the X on/off option (prevents receive buffer overflow using the X on and X off codes).

X on code &H11 X off code &H13

When the X off code is received in a data send operation, transmission is delayed. Data send operation is resumed when the next X on code is received. In a receive operation, the X off code is transmitted when the data in the receive buffer (capacity 127 bytes) reaches 113 bytes, requesting the send side suspend to data transmission. When the amount of data in the receive buffer falls to 2 bytes, the X on code is transmitted requesting the send side to resume transmission of data.
In this case, the RS signal is turned OFF after the X off signal has been transmitted and is turned ON immediately before the X on signal is turned ON.
Specify this option by one alphabetic character X or N.

X: Enable flow control
N: Disable flow control

CS Control

Specifies the CS signal check option. Executes CS check on X on/ off code, even in a data receive operation, when the X on/ off option is selected.
Specify this option by one alphabetic character H or N.

H: When executing a PRINT# statement, a SAVE command or when sending data in terminal mode, stops transmission until the CS signal is turned ON.
N: Does not check the CS signal.

The initialization value is H. The default value designation is also N.

Receive autoLF

When the terminal device data delimiter code (including the line delimiter code in BASIC) is the CR code (&H0D), it specifies that the LF code is to be added to the CR code as required to the MSX BASIC delimiter code (CR and LF (&H0A) codes).
Specify this option by one alphabetic character A or N.

A: Adds the LF code to the incoming CR code.
N: Does not add the LF code to the incoming CR code.

Send autoLF

When the terminal device data delimiter code is the CR code, specifies that the LF code be deleted, as required, from the MSX BASIC delimeter code when both the CR and LF codes are sent.
Specify this option by one alphabetic character A or N.

A: Does not send the LF code following the CR code.
N: Sends both the CR and LF codes.

The initialization value is N. The default value designation is also N.

Send and receive speeds.

You may specify different speeds for receive and send operations.
Communication speed is specified as number of bits per unit time (bits/second, BPS). The following 14 communication speeds are supported by this interface cartridge.

```
  50,   75,  110,   300,  600
1200, 1800, 2000,  2400, 3600
4800, 7200, 9600, 19200
```

The initialization value is 1200 BPS. The default value designation is also 1200 BPS.
If the parameter up to the comma (including send speed and Wait time) is omitted, the send speed is assumed to be equivalent to the receive speed by default.

If a negative T (seconds) is specified, the communication speed B (BPS) given by the following expression is assumed by default.
B = -1843200/(T*16)

Wait Time

When the CS control option is selected, if the CS signal wait time is exceeded, the send operation is terminated and an I/O error is notified.
When the CS control function is not selected, the Wait time is disabled.
Specify the Wait time by an integer in the range from 0 to 255. The specification unit is approximately 1 second. When 0 is specified as the Wait time, the system enters the Wait state awaiting the CS signal.
The initialization value is 0 and the default value designation is also 0.

(2) In addition to the above the following may also be omitted.

Example: CALL COMINI

Entries placed in quotation marks may be omitted beginning from the right end.

Example: CALL COMINI ("0:8N")

To omit entries to the right of a colon (:) which appears in a set of quotation marks ("), a space must be entered to indicate omitted entries.

Example: CALL COMINI ("0:8 XH", 1200)

When omitting the Speed and Wait Time parameters, omit the entry up to the following comma.

Example: CALL COMINI ("0:", 4800)

Example: CALL COMINI ("0:",,, 100)

(3) When the CALL COMINI statement is executed, the ER signal is turned on and the RS signal remains unaffected.

(4) The CALL COMINI statement is executed automatically at POWER ON. Initialization values are used to execute this statement

Example: CALL COMINI ("0:8N3XNNNN", 4800,, 5)

The above statement specifies the following conditions:

Character length: 8 bits
No parity check
Stop bit length: 2 bits
X on/off control: Enable
CS control: Disable
Receive autoLF: Disable
Send autoLF: Disable
Receive speed: 4800 BPS
Send speed: 4800 BPS
Wait time: 5 seconds

COMON Extended Statment

Functions: Enables interrupt from RS232C.

Format: CALL COMON ("device number:")

Description:

Executes the interrupt routine specified in the CALL COM GOSUB statement when data are received at the RS232C port after executing the CALL COMON statement.

COMOFF Extended Statement

Functions: Inhibits interrupt from RS232C

Format: CALL COMOFF ("device number:")

Description:

Inhibits an interrupt generated at the RS232C port when data are received.

COMSTOP Extended Statement

Functions: Holds interrupt from the RS232C port.
Format: CALL COMSTOP ("device number:")
Description:

(1) Holds the interrupt generated when data are received at the RS232C port until the following CALL COMON statement is executed.
When data are received at the RS232C port after the CALL COMSTOP statement has been executed, the interrupt processing routine is executed only after the next CALL COMON statement.

(2) The CALL COMSTOP is a statement to hold the execution of the CALL COMON statement. It is therefore invalid except when executed in CALL COMON status.

COMSTAT Extended Statement

Functions: Checks the RS232C port status.
Format: CALL COMSTAT ("device number:", numeric variable)
Description:

(1) Checks the RS232C port status and assigns it to a numeric variable.

(2) The status is checked by reference to numeric variable bit 8

Bit 15:
Checks buffer error. If data continues to be received in the buffer full condition, the incoming data is not recorded and an error is reported.
Bit settings are as follows:
Error: 1, Normal: 0

Bit 14:
Checks time out errors. When the CS control option is selected, an error occurs if CS remains OFF when the specified Wait time expires.
Bit settings are as follows:
Error: 1, Normal: 0

Bit 13:
Checks framing errors. Reports an error if a 0 bit is received between Stop bits.
Bit settings are as follows:
Error: 1, Normal: 0

Bit 12:
Checks overrun errors. Reports an error if data is received while the receive buffer is full.
Bit settings are as follows:
Error: 1, Normal: 0

Bit 11:
Checks parity errors. Reports an error if parity does not match.
Bit settings are as follows:
Error: 1, Normal: 0

Bit 10:
Checks errors when processing stops due to the control stop condition.
Bit settings are as follows:
Error: 1, Normal: 0

Bit 9:
Permanently set to 0

Bit 8:
Permanently set to 0

Bit 7:
Indicates CS signal status.
Bit settings are as follows:
CS signal ON: 1, CS signal OFF: 0

Bit 6:
Indicates timer status. Set the timer to 10 sec. to monitor the CS signal.
Bit settings are as follows:
Timer set: 1, Timer not set: 0

Bit 5:
Permanently set to 0

Bit 4:
Permanently set to 0

Bit 3:
Indicates DR signal status.
DR signal ON: 1, DR signal OFF: 0

Bit 2:
Indicates Break signal status.
This bit is set (1) if the RD signal line goes into the Break condition (ON for a few seconds) after the CALL COMSTAT statement has been executed. Otherwise this bit is not set (0).

Bit 1:
Permanently set to 0.

Bit 0:
Indicates CD signal status.
CD signal ON: 1, CD signal OFF: 0

(3) An RS232C file must be opened, using the OPEN statement, prior to executing the CALL COMSTAT statement. File OPEN mode is optional.
Example: CALL COMSTAT ("0:", F): PRINT BIN$(F)

COMTERM Extended Statement

Functions: Selects Terminal mode.
Format: CALL COMTERM ("device number:")
Description:

(1) Sets the system computer to Terminal mode.

(2) To exit from Terminal mode, hold the CTRL key and press the STOP key.

(3) CLOSE the RS232C port, if open, prior to exit from the Terminal mode.

(4) The RS signal remains ON while the system is in Terminal mode. When the CS control option is selected in the CALL COMINI mode, data are transmitted after confirming that the CS signal is ON.

(5) Using the F6, F7, F8, and STOP keys, you can select the following functions in the Terminal mode:

Literal mode
Hold the SHIFT key and press the F1 key to turn ON/OFF the Literal mode.
When the Literal mode is selected, a control code (codes lower than &H1F) will be displayed as follows:
A two character control code (&H40) following the hat mark (^).

Echo back
Hold the SHIFT key and press the F2 key to turn ON/OFF the Echo back mode.
When the Echo back mode is selected, data entered are sent to the host computer and simultaneously displayed (echoed back) on the screen.

Printer echo back
Hold the SHIFT key and press the F3 key to turn ON/OFF the Printer echo back mode. When this mode is selected, the data currently being displayed on the screen are output to the printer.

Break code
Press the STOP key. A Break signal will be sent **through** the SD signal line.
Example: CALL COMTERM ("0:")

EOF (End of file) Function

Functions: Indicates the end of a file by checking the EOF code (&HIA).
Format: EOF (file number)
Description:

(1) Receives data through the RS232C port and assigns it to a variable. If an EOF code is received, it stores the code in the receive buffer. It also checks if read data remains in the receive buffer before setting the EOF code. The status is indicated by -1 or 0.
End of file: -1
End of file not reached: 0
Example: A EOF (1)

INPUT# Statement

Functions: Reads data items from a file and assigns them to program variables.
Format: INPUT# file number, variable [, variable ...]
Description:

(1) Received data are stored in the receive buffer. The INPUT# statement reads data items (numeric values or character strings) from the file specified by the file number, and assigns them to specified variables.

(2) The file specified by the file number must first be opened in input mode, using the OPEN statement.

(3) Data separated by the following delimiters are assumed to be a single item.
Numeric variables:
Ignores initial blanks. The character string from a character other than a blank up to the following blank, comma (,), CR code (&H0D), CR and LF code combination (&H0A) is assumed to be a single data item.
Character variables:
Ignores initial blanks. The character string from a character other than a blank, up to the following comma (,), CR code (&H0D), CR and LF code combination is assumed to be a single data item.
A combination of the CR and CF codes in this order is assumed to indicate end of a data item. However, the reverse value is not interpreted as a delimiter.

(4) The variable types must match the corresponding data item types.

(5) If the X on/off control option is selected in the CALL COMINI statement, the system transmits the X off code, when the amount of data in the buffer reaches 113 bytes, to turn off the RS signal. Received data are assigned to variables and as soon as the amount of data in the buffer drops to 2 bytes, the system outputs the X on signal to turn on the RS signal.

(6) If the X on/off as well as the CS control options are selected in the CALL COMINI statement, the X on/off codes are held untill the CS signal is turned on.
Example: INPUT#1, A,B

LINE INPUT# Statement

Functions: Reads a character string from a file and assigns it to a string variable.
Format: LINE INPUT# file number, variable.
Description:

(1) Received data is stored in the buffer.
The LINE INPUT# statement reads a character string from the file specified by the file number and assigns all characters up to the CR (&H0D) and LF (&H0A) code combination or only the CR code to the variable.

(2) Only a combination of CR and LF codes, arranged in the above order, or a CR code are regarded as delimiters. A combination of LF and CR codes appearing in this order is not regarded as a delimiter.

(3) The specified file number must have been opened using the OPEN statement.

(4) If the X on/off option is selected in the CALL COMINI statement, the system transmits the X off code when the amount of data in the buffer rises to 113 bytes. The received data are assigned to variables and as soon as the amount of data in the buffer falls to 2 bytes, the system transmits the X off signal to turn ON the RS signal.

(5) If the X on/off as well as the CS control options are selected in the CALL COMINI statement, the X on/off codes are held untill the CS signal is turned ON.
Example: LINE INPUT#1, A$

LOAD	Command

Functions: Receives a program
Format: LOAD "COM [device number]:" [,R]
Descriptions:

(1) The LOAD command receives programs through the RS232C port, and loads them to the memory.

(2) The LOAD command stops with the EOF code (&H1A).

(3) The LOAD command clears previous programs or variables and closes files currently open.

(4) The default device number is 0.

(5) If the R option is selected, a program is executed automatically. In this case currently open files are not closed.

(6) The LOAD command turns ON the RS signal.

(7) If the X on/off control option is selected in the CALL COMINI statement, the system transmits the X off code upon receipt of the CR code to turn OFF the RS signal. One line of the received program is loaded to the program area and when the amount of data in the receive buffer falls to 2 bytes, the system sends the X on code to turn ON the RS signal.

(8) If both the X on/off and the CS control options are selected in the CALL COMINI statement, the X on/off code is held untill the CS signal is turned ON.
Example: LOAD "COMO:"

LOC	Function

Functions: Calculates the number of bytes stored in the receive buffer

Format: LOC (file number)
Description: Calculates the number of bytes currently stored in the receive buffer.
Example: A = LOC (1)

LOF	Function

Functions: Calculates the number of bytes remaining in the receive buffer.
Format: LOF (File number)
Description: Calculates the number of bytes currently remaining in the receive buffer.
Example: A = LOF (1)

MERGE	Command

Functions: Receives a program and merges the received program with the program currently stored in the memory.
Format: MERGE "COM [device number]:"
Description:

(1) Receives a program (ASCII code program) through the RS232C port and merges it with the program currently stored in the memory to create a single new program.

(2) The MERGE command stops when the EOF code (&H1A) is received.

(3) If the incoming program contains line numbers identical to the program currently stored in the memory, the incoming program has priority.

(4) The default device number is 0.

(5) The RS signal is turned ON when this command is executed.

(6) When the X on/off option is selected in the CALL COMINI statement, the X off code is sent to turn OFF the RS signal when the CR code is received. When one line of the received program is entered to the program area and the remaining buffer data is reduced to 2 bytes, the X on code is transmitted to turn ON the RS signal.

(7) When the X on/off option and CS control options are selected in the CALL COMINI statement, the X on/off code is kept held until the CS signal is turned ON.
Example: MERGE "COMO:"

OPEN Statement

Functions: Opens a RS232C file.
Format: OPEN "COM [device number] :" [For mode] AS [#] file number
Description:

(1) The OPEN statement opens a file for data (numeric and character) communication through the RS232C port and assigns a file number to it. Once a file has been opened, it may be used for data input/output operations using the INPUT# and PRINT# statements. Control signals (COMDTR, COMSTAT, COM GOSUB, COMBREAK, etc.) are supported. States may also be checked.
The file is closed after processing, using the CLOSE signal.

(2) The mode selection feature allows you the option of opening files in the INPUT or OUTPUT modes.

INPUT: Input mode
When a file is opened in the INPUT mode, the same receive buffer area is assigned (127 bytes). Even if files are opened several times with different file numbers, only one receive buffer is assigned. Therefore, this method of opening files is not recommended.

OUTPUT: Output mode
The CLOSE statement generates a EOF code (&H1A)

Default mode: SEND/RECEIVE mode
Both send and receive operations may be specified for the same file number. In this case, the EOF code (&H1A) will not be transmitted when the file is closed (CLOSE statement is executed).

(3) The file number is specified by an integer in the range from 1 to 15, but not exceeding the value specified in the MAXFILES statement. A file number already assigned to another file may not be specified.

(4) The default device number designation is 0.

(5) The RS signal is turned ON by this statement
OPEN "COM:" FOR OUTPUT AS#1
OPEN "COMO:" AS#1

PRINT#	Statement

Functions: Outputs data to a specific file.
Format: PRINT# file number
PRINT# file number, expression [;expression ...] [;]
PRINT# file number, expression [, expression ...] [,]
A question mark (?) may be used in place of PRINT.
Description:

(1) Outputs specified data items by expressions through the RS232C port.

(2) The file number must specify a file opened in the output mode, using the OPEN statement.

(3) Output formats are identical to those in the PRINT statement:
A numeric data item is preceded by a blank (for a positive value) or a minus sign (-) (for a negative value), and the numerals are converted to a string, which is followed by a blank.
For a string, all the characters in the string are output.

(4) More than one expression may be specified in a PRINT# statement, by separating them with commas (,) or semicolons (;).
When commas are used as delimiters, blanks are added to justify the preceding data to 14 digits. Then following data are output. These are again followed by blanks to justify the data to 14 digits (28 digits total). The data transmission speed is inversely proportional to the number of blanks. When strings are separated by semicolons, they are output serially to a file.

(5) When multiple characters are read by an INPUT# or LINE INPUT# statement, only one variable is read as a character string. Use commas to separate character strings for data read using the INPUT# statement. The same can be done for character string data read with the LINE INPUT# statement. In this case the combination of CR (&H0D) and LF (&H0A) codes is used as the delimiter.

(6) When a comma or semicolon does not appear at the end of PRINT# statement, the output data will be followed by CR (&H0D) and the LF (&H0A) codes. If the autoLF option is selected in the CALL COMINI statement, only the CR code is output. When a comma or semicolon is placed at the end of a PRINT# statement, no CR or LF code will be output. The output format is identical to that of the PRINT statement, except that the PRINT# statement outputs data not to the screen but to a file.

(7) If the CS control option is specified in the CALL COMINI statement, transmission is delayed untill the CS signal is turned on.

PRINT# USING Statement

Functions: This statement outputs data (numbers or strings) to a specified file and in the specified format.

Format: PRINT# file number, USING format control string; expression [; expression ...] [;]
PRINT # file number USING format control string; expression [, expression ...][,]
A question mark (?) may be used in place of PRINT#.

Description:

(1) The PRINT# USING statement outputs the specified data (numbers or strings) using a specified format, through the RS232C port.

(2) The data output format is identical to that of the PRINT# statement, except that the PRINT# USING statement requires format specifications.

(3) The format specifications are identical to that of the PRINT USING statement. For details, see the "MSX BASIC REFERENCE MANUAL".

RUN Command

Functions: Executes a received program

Format: RUN "COM [device number]:"

Description:

(1) An ASCII format program is received through the RS232C format. The RUN command loads the program to the memory and executes it automatically. When this command is executed, all variables are cleared, and all files are closed.

(2) The RUN program stops loading received data when the EOF code (&H1A) appears and executes the received program.

(3) The default device number is 0.

(4) This command turns ON the RS signal.

(5) If the X on/off control option is selected in the CALL COMINI statement, the system transmits the X off code when the amount of data in the buffer reaches 113 bytes to turn OFF the RS signal. Received data are assigned to variables. As soon as the amount of data in the buffer drops to 2 bytes, the X on code is transmitted to turn ON the RS signal.

(6) If both the X on/off and the CS control options are selected in the CALL COMINI statement, the X on/off codes are held untill the CS signal is turned ON.

Example: RUN "COM0:"

Functions: Saves a program to a specified file

Format: SAVE "COM [device number]:"

Description:

(1) The SAVE command is used to save an ASCII code program through the RS232C port.

(2) Sends the EOF code (&H1A) at the end of a program SAVE operation.

(3) A combination of the CR (&H0D) and LF codes (&H0A) is sent at the end of each line as a line delimiter. However if the Send autoLF option is selected in the CALL COMINI statement, only the CR code is sent as a delimiter at the end of each line.

(4) The RS signal remains ON while a program is being sent.

(5) If the CS control option is selected in the CALL COMINI statement, the system enters the wait state untill the RS signal is turned ON.

Example: SAVE "COM:"

CHAPTER 4

COMMANDS AND OPERATIONS

1. INITIALIZATION BY CALL COMINI

FORMAT

```
CALL COMINI ("device number:  character length    parity
stop bit    X on/off control    CS control   receive autoLF
send autoLF    SI/SO", receive speed, send speed, wait time)
```

Example: `CALL COMINI ("0: 8N3XNNNN", 4800, 4800, 30)`

The above statement specifies the following:

Character length 8 bits
No parity
Stop bit: 2 bits
X on/off control required
CS control not required
No receive autoLF
No send autoLF
No SI/SO control
Receive speed 4800BPS
Send speed 4800BPS
Wait time 30 secs

1) Device number

The device number is set permanently to 0 in this cartridge. Therefore, the device number must be specified as 0.

2) Character length

Normally a character is 8 bits long. In certain cases 7 bit characters are required. Seven bit characters are common in TSS (Time Sharing Systems) etc. Depending on the requirements of the device with which data are to be communicated, character lengths of 5, 6, 7, and 8 bits may be selected. The required character length is specified by an integer in the range from 5 to 8.

5: 5 bit character
6: 6 bit character
7: 7 bit character
8: 8 bit character

3) Parity

There is the possibility of data errors occurring during communication. This is likely to reverse bit values from 1 to 0. The parity check function has been provided to check if the bit values have been changed (1 to 0) accidentally. For this purpose, a bit is added at the end of each character. This is called a parity bit. The number of bits of level 1 is set to either an odd or even number. Then a check is made at both the send and receive ends to determine whether the number of bits is even (even parity) or odd (odd parity).
It is necessary that the same parity check type be specified for the devices between which communication is to be carried out.
Parity is specified using one of the following alphabetic characters:

E, O, I, or N
E: Even parity
O: Odd parity
I: Ignore parity in receive operation and set the parity bit to 0 in data transmission Character length varies between 5 and 7 bits.
N: No parity bit

4) Stop bit length

A stop bit is transmitted at the end of each character. On the receive side a check is made to determine if the length of the stop bit exceeds a predefined length. Identical stop bit lengths must be specified for both the send and receive sides. The stop bit length is specified by an integer from 1 to 3.

1: 1 bit
2: 1.5 bits
3: 2 bits

5) **X on/off control**

Specifies the flow control option (prevents receive **buffer overflow by** specifying the X on code and X **off codes).**

X on code...&H11, X off code...&H13

Data transmission is delayed when the X off code is received and is resumed when the next X on code is received.
On the receive side, the X off code is sent out as soon as 113 bytes of data are recorded in the receive buffer (capacity 127 bits) to request the send side to suspend data transmission. When the amount of data in the receive buffer falls to 2 bytes, the receive side generates the X on code signal to request resumption of data transmission. When a LOAD command is executed, the X off signal is generated upon receipt of the CR code (&H0D) code.
When the amount of data in the receive buffer falls to 2 bytes, the X on code is transmitted. The flow control option is specified by one alphabetic character X or N.

X: Flow control enable
N: Flow control disable

6) CS control

Specifies the CS signal check option for data transmission. This option is required if the terminal device uses program software which turns ON the CS signal when data are received.
This option is specified by one alphabetic character H or I.

H: Holds data transmission untill the CS signal is turned ON when transmitting data with PRINT #, a SAVE command or in terminal mode.

(For details, see Section 11)

NB: Does not check CS signal during data transmission.

7) Receive autoLF

Data delimiter codes vary with computer systems (including the line delimiter in BASIC). In certain systems, only the CR code (&H0D) is the delimiter code while in other systems the combination of CR and LF codes (&H0D) is used as the delimiter. In MSX BASIC the CR and LF codes are used as the delimiters.
Other computers and printers may use CR or CR and LF codes as delimiters.
If the terminal device uses the CR code as the delimiter, and the LF code is added to the incoming CR code, a combination of the CR and LF codes is created allowing it to satisfy MSX BASIC format requirements.
The autoLF option is specified by one alphabetic character A or N.

A: Adds a autoLF code to an incoming CR code
N: Does not add a LF code to an incoming CR code.

8) Send autoLF code

When the terminal device supports only the CR delimiter code, the CR and LF delimiter codes used in MSX BASIC are not sent together. Only the CR code is sent and the LF code is omitted.
This option is specified by one alphabetic character A or N.

A: Does not send a LF code following the CR code
N: Sends a LF code following the CR code

9) Receive and send speed

Specifies identical communication speeds at the send and receive sides. Specifies the speed supported by the terminal device.
Communication speeds up to 9600BPS may be specified in SAVE commands. Accurate communication at speeds exceeding 9600BPS are not guaranteed and errors may occur.
This computer allows you to select different speeds for send and receive operations. Communication speed is specified as the number of bits per unit time (BPS: Bits per second). A total of 14 speeds are supported:

50	75	110	300	600
1200	1800	2000	2400	3600
4800	7200	9600	19200	

10) Wait time

When the CS control option is selected, the send operation is not started until CS is turned ON. Therefore, errors will occur unless CS is turned ON within the specified time limit.
The wait time is specified by an integer in the range from 0-255. The time unit used is the second. When 0 is specified, the system will continue to wait until CS is turned ON.
The default value is 0. Therefore the wait time should be specified as at least 4 to 6 seconds.

2. TERMINAL MODE

(1) Switching terminal modes

Terminal mode allows you to use your computer as an I/O (input/outout) terminal of another computer system (host computer).
Terminal mode allows you to access host computer programs. In this mode, your computer can be used simply as a display console to display incoming data or as a keyboard to enter data from the keyboard.

1) Determine the host computer's specifications

Check the specifications of the host computer, when the host is not connected directly, but through an audio coupler. The specifications of the audio coupler must also be checked carefully.

2) Initialize the system

For details on the initialization process, see Section 2. In addition to initializing the computer to which the interface is being connected, the host computer must also be initialized.

3) Switch to Terminal mode

Select Terminal mode using the CALL COMTERM statement.

Format: CALL COMTERM ("device number :")

In this cartridge, the cartridge number is 0.
If the RS232C port is open before entering Terminal mode, CLOSE the port.
Control signal functions are as follows. Assign control signals in conformity with the host computer programs and functions.

ER signal: Turns ON the interface cartridge at computer POWER-ON.
RS signal: Turns ON the interface cartridge during terminal mode.
CS signal: When the CS control option is selected in the CALL COMINI statement, it transmits data after confirming that CS is ON.

If initialization is incorrect or if the control signal operations do not match, unpredictable displays may result or the system may not respond at all. In such cases hold the CTRL key and press the STOP key. Then re-initialize the system. In certain computer systems, the system enters the Wait state pending data input from the RS232C port. In this case, the keyboard is disabled and the system cannot recover. In such cases, switch OFF the power supply and repeat operations from the beginning. It is, therefore, recommended that the user should check initialization conditions carefully as well as control signals before using the RS232C interface cartridge.

- Exit from Terminal mode

  Hold the CTRL key and press the STOP key to exit from Terminal mode.

(2) Operations in terminal mode

Using the SHIFT, F1, F2, F3, and the STOP keys, you can select the following functions:

1) Literal mode

This mode is switched ON/OFF everytime you hold the SHIFT key and press the F1 key. When the Literal mode is selected, the control code (codes below &H1F) appears as a hat mark followed by a 2 character code beginning with &H40. When the X on/off command is entered, the X on or X off code will not be displayed on the screen.

2) Echo back

This function is switched ON/OFF everytime you hold the SHIFT key and press the F2 key. When this function is selected, the data entered from the keyboard are transferred to the host computer and displayed on the computer at the same time (echoed back).

3) Printer echo back

This function is switched ON/OFF everytime you hold the SHIFT key and press the F3 key. When this function is selected, the data being displayed on the screen are simultaneously output to the printer.

4) Break code

Press the STOP key to send the Break signal. This means that the data line (SD) goes low (0) while the STOP key is pressed.

- DEL code processing

  Certain host computers send a DEL code (&H7F) at the beginning of a line. In a MSX computer, the BS key executes the DEL function. In this case the cursor will move back to the upper line. In such cases, a function to inhibit the host computer DEL code should be set, and a program should be created to carry out data communication which will dispense with the need to operate in the Terminal mode. In the latter case the DEL code should be processed in such a program.

3. PROGRAM TRANSFER

(1) Program transfer

The RS232C cartridge will allow you transfer an ASCII code program from a computer to which this cartridge is connected to another program. It will also allow you to load ASCII code programs.
When the computer to which you computer is linked is also a MSX computer, you can access the programs in such a computer. Similarly any programs transmitted to such a computer can also be executed by the other computer.
However, if the computer to which your computer is linked is not a MSX computer, even if the programs in the other computer can be accessed, they cannot be executed. Similarly any MSX programs sent to such a computer cannot be executed by that computer. The MSX programs transmitted to such a computer are received by it as data. Similarly MSX programs stored in such a computer can be received as data.

Example:

Communication between MSX computers

Send as a program	→	Receive as a program	(Stored in the program area)
SAVE command		LOAD command	
Send as a program	→	Receive as data	(Stored in variable area)
SAVE command		INPUT# statement	
Send as data	→	Receive as a program	(Stored in the program area)
PRINT# statement		LOAD command	

(In this case the data must be coded as a program.)

Send as data	→	Receive as data	(Stored in the variable area)
PRINT# statement		INPUT# statement	

(2) Sending a program

1) Check the specifications of the terminal device

Check the specifications of the terminal device carefully. If the terminal device is not connected directly but through an audio coupler, the specifications of the audio coupler must also be checked.

2) Initialize the system

For details on the initialization process, see Section 2. In addition to initializing the computer to which this cartridge is connected, initialize the computer at the terminal also. Transmission speeds up to 9600BPS may be selected.

3) Set the terminal computer to the receive state

The control signals of the computer to which the cartridge is connected are controlled as follows:

ER signal:	Turned ON at POWER-ON
RS signal:	Turned ON before program transmission and turned OFF after transmission
CS signal:	When the CS control is selected in the CALL COMINI statement, data are sent only after confirming that the CS is ON.

The EOF code (&H1A) is sent at the end of a program transferred using the SAVE command. The terminal computer should be programmed so that it will interpret the EOF code as the end of the program being transferred. Use the LOAD command to transfer programs between MSX computers.

4) Sending a program

Send a program through the RS232C port using the SAVE statement. In this case the program will be transferred in ASCII code.

Format: SAVE "COM device number:"

In this cartridge, the device number is set permanently to 0.
If initialization is incorrect or when the control signals do not match, the system may not respond at all. In such a case, hold the CTRL key and press the STOP key. Then re-initialize the system.
In certain computers, the system will lock into the Wait state pending data input from the RS232C port. In such a case, keyboard input is disabled and the system cannot recover. Switch OFF the power and restart from the beginning. The user is therefore recommended to check initialization and control signals carefully before using the RS232C cartridge.

(3) Receiving a program

1) Check the specifications of the terminal computer carefully

If the terminal computer is not connected directly, but through an audio coupler, the specifications of the coupler must also be checked.

2) Initialize the system

For further details, see Section 2. In addition to initializing the computer to which this cartridge has been connected, the terminal computer must also be initialized.

3) Set the computer to program receive status

A program is sent through the RS232C port using the LOAD command. The program is transferred in ASCII code.

Format: LOAD "COM device number:"

The device number is set permanently to 0 in this cartridge.
Programs not written in ASCII code are likely to cause errors. Make sure that the programs to be received are in ASCII code.

If initialization is incorrect or control signals do not match, the system may not respond at all. In such a case, hold the CTRL key and press the STOP key. Then re-initialize the system.
In certain types of computers, the system may enter the Wait state pending input from the RS232C port.
In this case the keyboard is disabled and the system cannot recover. Turn OFF the system POWER and restart from the beginning.

(2) Sample program

Using the RS232C, you can communicate between two computers. First, enter F1 from one of the systems. The system entering F1 becomes the transmission side while the other system automatically becomes the receive side.
Enter the character to be sent followed by the return key. The entered data will be sent to the receiving side. Press the F2 key to change receive and send sides. To complete communication, press the F3 key.

```
1Ø  CLEAR 2ØØØ:CLS:CALL COMINI(.,,4)
2Ø  ON ERROR GOTO 41Ø:DIM A$(2Ø)
3Ø  A$(Ø)="YES":F=Ø:T=7Ø
4Ø  A$(1)="HELLO":A$(2)="PLEASE":A$(3)="GOOD BYE"
5Ø  FOR I=1 TO 3
6Ø  KEY I,A$(I)+CHR$(13)
7Ø  NEXT I
8Ø  PRINT"RS232C DEMO PROGRAM"
85  LPRINT"RS232C DEMO PROGRAM"
9Ø  OPEN"comØ:"AS#1
1ØØ CALL COM(.GOSUB34Ø):CALL COMON
11Ø D$=INKEY$
12Ø IF F=2 THEN 26Ø
13Ø IF D$=""THEN 11Ø ELSE PRINT D$;:LPRINT D$;
14Ø LINEINPUT E$:LPRINT E$:D$=D$+E$
15Ø IF D$<>A$(1) THEN 11Ø
16Ø PRINT#1.D$:F=1
17Ø FOR J=Ø TO T
18Ø IF F=Ø THEN J=T+1
19Ø NEXT J
2ØØ IF J=T+1 THEN 41Ø
21Ø PRINT"++ SENDING MESSAGE ++"
215 LPRINT"++ SENDING MESSAGE ++"
22Ø LINEINPUT B$:LPRINT B$
23Ø PRINT#1.B$
24Ø IF B$=A$(3) THEN 3ØØ
25Ø IF B$<>A$(2) THEN 22Ø
26Ø PRINT"** RECEIVING MESSAGE **"
265 LPRINT"** RECEIVING MESSAGE **"
27Ø INPUT#1.C$:PRINTC$:LPRINTC$
28Ø IFC$<>A$(3)ANDC$<>A$(2) THEN27Ø
29Ø IF C$=A$(2) THEN 21Ø
3ØØ FOR J=Ø TO T:NEXT J
31Ø CLOSE:PRINT:PRINT:PRINT"THE END"
315 LPRINT:LPRINT:LPRINT"THE END"
32Ø KEY1,"color":KEY2,"auto ":KEY3,"goto "
33Ø ON ERROR GOTO Ø:END
34Ø INPUT#1,C$
35Ø IF F=1 THEN GOTO 39Ø
36Ø IF C$<>A$(1) THEN RETURN
37Ø PRINT#1.A$(Ø)
38Ø CALL COMOFF:F=2:RETURN
39Ø IF C$<>A$(Ø) THEN RETURN
4ØØ CALL COMOFF:F=Ø:RETURN
41Ø PRINT"CAN NOT COMMUNICATE !"
415 LPRINT"CAN NOT COMMUNICATE !"
42Ø GOTO31Ø
```

CHAPTER 4

BEFORE CALLING FOR SERVICE

Problem	Corrective measures and trouble shooting
Communication disabled Terminal device unpredictable	1. Do the specifications of the host and terminal match? Hold the CTRL key and press the STOP key. Then check the above mentioned specifications. 2. Do the pin assignments of the host and terminal match? Switch OFF power and check cable pin assignment.
Data errors	1. Do the communication specifications of the host and terminal match? Hold the CTRL key and press the STOP key. Then check the above mentioned specifications. 2. Check host and terminal character codes.
Terminal mode display abnormal	1. Do the communication specifications of the host and terminal match? Hold the CTRL key and press the STOP key. Then check the above mentioned specifications. 2. Check host and terminal character codes.

Due to differences in the communication methods and control signals of the host and the terminal, it may become impossible to communicate through the RS232C interface cartridge.
Even if the terminal supports communication speeds up to 19200BPS, such high speeds may not be possible due to limitations placed by communication methods and control signals.
A user is recommended to check the specifications of both the host and the terminal carefully.

TOSHIBA CORPORATION
TOKYO JAPAN
PRINTED IN JAPAN
A 0581