SHARP

RS-232C Serial Interface

MODEL CE-340R

INSTRUCTION MANUAL

FORWARD

Congratulations on your purchase of the Sharp RS-232C Serial Interface <CE-340R>.
Be sure to read this instruction manual in order to use the interface properly. Be sure to keep this instruction manual.
If during use there should be something that you do not understand or something is not functioning properly, it should prove useful.

CAUTION

1) This interface is made from LSI and other precision components and should not be used in places where there are rapid temperature fluctuations, high humidity or too much dust. Also, avoid using it in direct sunlight. The above may cause the interface to malfunction.
2) Do not hit or drop the interface.
3) When using the interface do not touch the pins of the components or the traces on the board directly with your hands since there is the danger of static electricity damaging the components.

TABLE OF CONTENTS

I . WHAT IS AN RS-232C INTERFACE?

The word RS-232C expresses a special standard of the EIA (Electronics Industry Association) and refers to the standard for interfaces used to transfer data between modems and transmission control devices and the binary serial data, control signals and timing signals between modems and data terminals.

The RS-232C interface is one of the devices generally used for the exchange of information between a computer and a peripheral device.

This interface was designed to conform to the EIA standard, but in particular it was designed for connecting to printers, plotters and similar devices.

However, just because a peripheral device has an interface that conforms to the RS-232C standard that does not necessarily mean that it can exchange information with all devices. Before using this interface be sure that both this instruction manual and the peripheral device specifications are fully understood.

Sharp cannot provide either hardware or software support for special customer applications. Moreover, Sharp cannot in any way be responsible for damages that arise as a result of customer misuse.

However, in so far as possible, all the information necessary for information exchange between the computer and a peripheral device is contained in this instruction manual.

II. SPECIFICATIONS

Model:	CE-340R
Input/output method:	RS-232C serial input/output
Number of channels:	2 channels*
Code used:	7 bit ASCII or 8 bit ASCII
Baud rate:	110 – 9600 bits/sec.
Transmission mode:	Half-duplex mode
Synchronization mode:	Asynchronous mode (not suited for synchronous mode)
Transfer control:	No protocols
Data format:	Stop bits ..... 1 / 1.5 / 2 Parity ..... even / odd / none
Command words:	CHANNEL, SEND, RCV, SENREV, OPCHNL, POLLING
Components:	Integrated circuits and discrete components
Operating temperature:	0°C to 40°C
Outer dimensions:	Width 137 mm x depth 152 mm x height 17 mm
Weight:	140 gm
Accessories:	Instruction manual (this manual), slot cover and channel number tabs

* This interface comes equipped with 2 channels and one computer can accomodate up to 2 interface cards which means that it can be equipped with up to 4 channels.

This interface can be put in slots 1, 2, 3 or 4 in the back of the computer (for 2 interface boards the slot combinations 1 and 3 or 2 and 4 cannot be used) and the channel number is determined by the slot number.

Channel number when put in slot 1 or slot 3 ..... channel number 0 and 1
Channel number when put in slot 2 or slot 4 ..... channel number 2 and 3

When 2 interface boards are used the following 4 combinations are possible:

Combination	Slot 1	Slot 2	Slot 3	Slot 4
1	Channel 0, 1	Channel 2, 3	X	X
2	Channel 0, 1	X	X	Channel 2, 3
3	X	Channel 2, 3	Channel 0, 1	X
4	X	X	Channel 0, 1	Channel 2, 3

Ⅲ. DATA INPUT/OUTPUT FORMAT

Set the data input/output baud rate, data length (7 bit code/8 bit code), stop bit length, parity format and other parameters to match the input/output format of the peripheral device being connected. The details are explained in section VI. Programming (p. 00).

1. BAUD RATE

The baud rate is the speed at which information is transferred and is expressed in units of bits/second.
It is possible to select baud rates of 110, 300, 600, 1200, 2400, 4800 and 9600 bits/sec. on this interface.

2. 7 BIT CODE AND 8 BIT CODE

7 bit code uses 7 bits to represent the data while 8 bit code indicates data represented by 8 bits.
Either 7 bit code or 8 bit code can be selected on this interface board.

3. START BIT AND STOP BITS

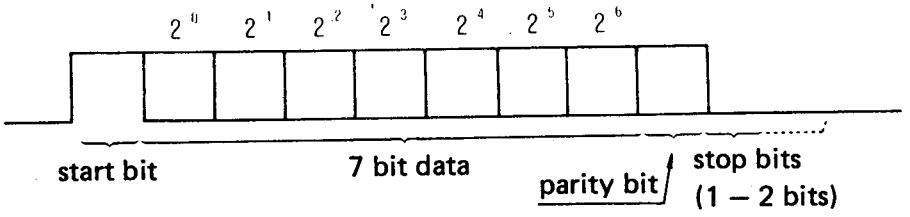
The start bit is the 1st bit represented by the rise from OFF (low level) to ON (high level) and the stop bit is the OFF bit, 1.5 bits or 2 bits following the 7 or 8 bits of data.
1, 1.5 or 2 bits may be selected as the stop bit length with this interface. However, the stop bit lengths selected here are the shortest and longest allowed.

4. PARITY

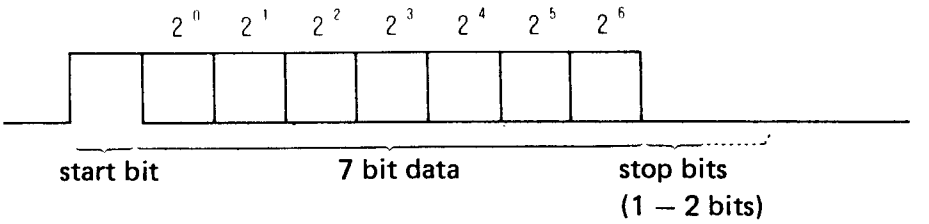
The parity bit is the bit that serves as a check of the 7 bit or 8 bit data. When the total of the data portion ON (high level) bits and the parity bit is an even number this is called even parity, the total being odd is called odd parity.
Either even parity, odd parity or no parity may be selected on this interface board.

5. DATA TIMING CHART

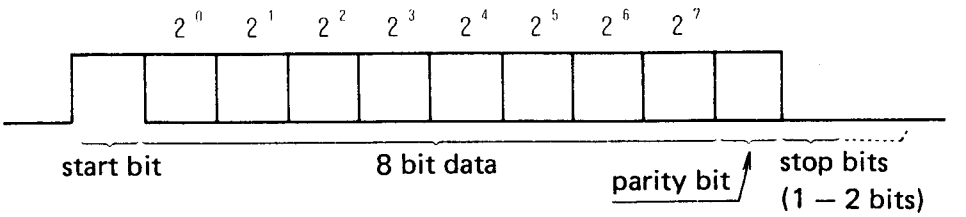
1) 7 bit code with parity



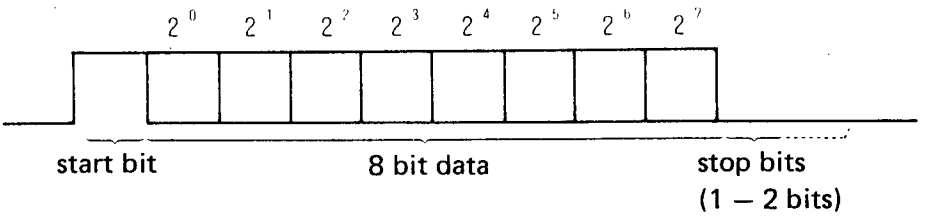
2) **7 bit code with no parity**



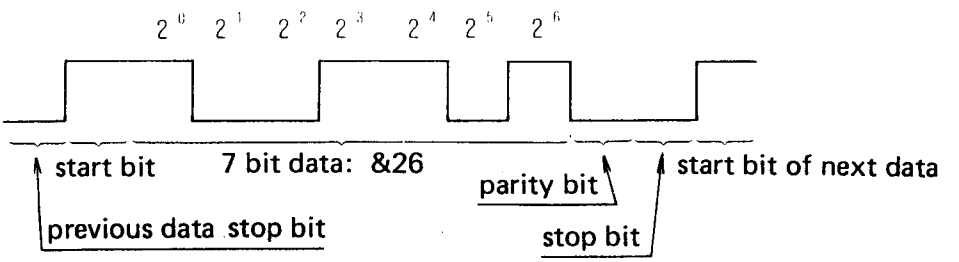
3) **8 bit code with parity**



4) **8 bit code with no parity**



5) **Example of 7 bit code with 1 stop bit and even parity (data: &26)**



Ⅳ. INPUT/OUTPUT SIGNALS AND CONTROL SIGNALS

1. INPUT/OUTPUT SIGNALS

Data cannot be input at an arbitrary time with this interface. Data input when the input command (RCV command, etc.) is not being executed is invalid.
The input signal (SD signal) is pin 2 on the connector and the output signal (RD signal) is pin 3 on the connector.
Reference section [III. DATA INPUT/OUTPUT FORMAT] (p. 3) for the data timing chart.

2. CONTROL SIGNALS

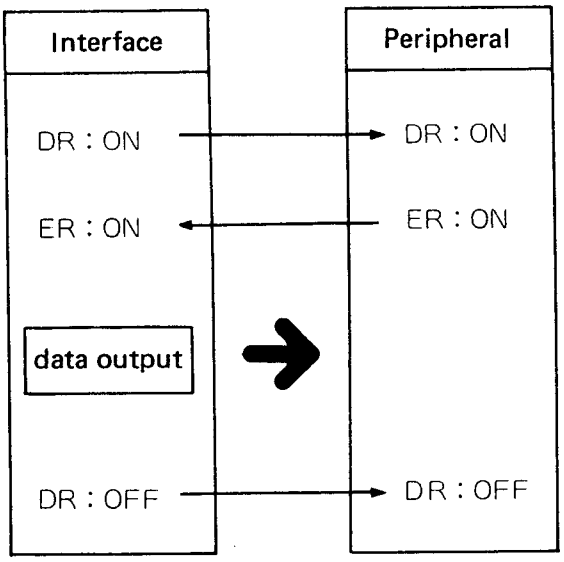
Signal name	Abbreviation	Pin number	Signal direction	Functional outline
Clear to Send	CS	5	CE-340R →peripheral	Signal which indicates that data may be input from a peripheral. Data from a peripheral is input when ON (high level) and is invalid when OFF (low level).
Data Set Ready	DR	6	CE-340R →peripheral	Signal indicating that the power is turned on at the interface (computer). It goes ON (high level) when the power is turned on.
Signal Ground	SG	7		
Carrier Detect	CD	8	CE-340R →peripheral	Signal indicating that the power is turned on at the interface (computer). (Signal indicating OK for carrier transmission in the case of an acoustic coupler.) It goes ON (high level) when data is transmitted from the interface (computer) if jumper wire* 04 (for channels 0 or 2) or jumper wire 14 (channel 1 or 3) is cut. It is not normally necessary to cut these jumper wires.
Ready	READY	19	CE-340R ← peripheral	Signal indicating that data may be output from the interface (computer). When ON (high level) data is output from the interface and when OFF (low level) data output is suspended. Even though this signal may go from ON (high level) to OFF (low level) up to 2 bytes of data may be output before output is suspended. When this signal is not used leave it open (unconnected).

Data Terminal Ready	ER	20	CE-340R ←peripheral	Signal indicating that the terminal (peripheral) device is ready for operation. If this signal is OFF (low level) when data is output, or open, the interface (computer) goes into an error condition. This signal is invalid if the jumper wires 01 (channels 0 or 2) or 11 (channels 1 or 3) are cut on the board. If jumper wire 02 (channels 0 or 2) or 12 (channels 1 or 3) are cut then data is output when this signal is ON (high level) and suspended when OFF (low level). However, even if this signal goes from ON (high level) to OFF (low level) up to 2 bytes may be output before data output is suspended.
Paper Out	PO	23	CE-340R ←peripheral	Signal indicating that the paper supply is low. (For an acoustic coupler it shows the state of the coupler, power on or off state). If it goes ON (high level) during data output the interface (computer) goes into an error state. Cutting jumper wires 03 (channel 0 or 2) or 13 (channel 1 or 3) on the board causes the polarity of this signal to be reversed such that an error state is enterd by the interface (computer) if it goes OFF (low level) during data output.
		Others		Not used.

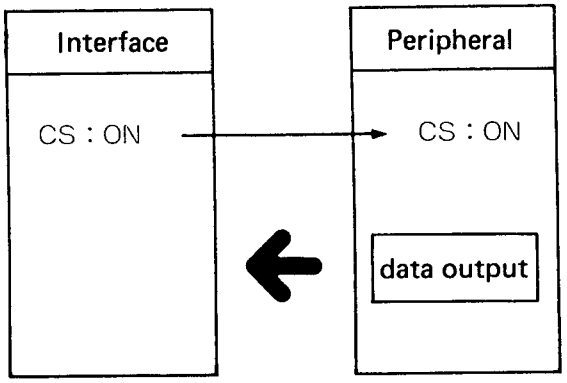
* Details of the jumper wires are explained in the next section [V. CONNECTING TO PERIPHERAL DEVICES] (p. 8).

3. GENERAL INPUT/OUTPUT SIGNAL FLOW FOR PRINTERS, X–Y PLOTTERS, ETC. (Example)

1) Output



2) Input



V. CONNECTING TO PERIPHERAL DEVICES

1. PERIPHERAL DEVICES THAT CAN BE CONNECTED

Printers, X–Y plotters and similar devices having an RS-232C interface may be connected. Using a special cable enables this computer to be connected to an acoustic coupler or a separate computer. Reference section [VII. SPECIAL CABLE] (p. 21) for details.

When making connections note that some printers, plotters, etc. vary in the way they process signals and that it is necessary to conform to the various specifications. Reference the next section for details.

2. CAUTIONS FOR CONNECTIONS

When connecting to peripheral devices the first thing that must be done is to make a thorough study of the specifications of the various devices. The cautions written here are for connecting common types of printers, X–Y plotters and acoustic couplers and depending on the device, it may be necessary to take special steps to connect it.

1) Printers and X–Y plotters

i) Processing the CR code

First, check to see whether the CR code is a carriage return (without linefeed) or carriage return and linefeed. Set it for the former carriage return and linefeed condition. This is done to prevent 2 linefeeds being output. Since after data output the CR code and LF code are automatically output, a CR code with the meaning of carriage return and linefeed plus a LF code would result in 2 linefeeds.

ii) Processing the ER signal (pin 20)

Check to see which of the following functions pin 20 (ER signal) has on the peripheral device (printer, X–Y plotter) interface.

① The signal shows that the peripheral device is prepared to operate, that power is ON, the device is ON LINE, etc. (When the device is ready to operate this signal goes ON (high level).)

② The signal shows the state of the remaining buffer memory in the peripheral device. (The signal goes ON (high level) when there is still memory remaining in the buffer and it can receive data.)

If the ER signal has the function of ①, then when the ER signal is OFF (low level) it is necessary for the CE-340R to be in an error state. If the ER signal has the function of ②, then when the ER signal is OFF (low level) it is necessary to momentarily halt data output from the CE-340R. When the CE-340R is shipped from the factory it is assumed that the ER signal will be as in ①. If the peripheral device has the ② ER signal function then cut jumper wire 02 (channel 0 or 2) or 12 (channel 1 or 3) on the CE-340R board. See the next section for the locations of the jumper wires. Because the CE-340R cannot know the state of the remaining buffer memory in the peripheral device when the ER signal has the meaning ①, there is the possibility that data may be lost if the baud rate is high or processing requires much time (FF, etc.). In order to deal with this it is necessary that the baud rate be lowered or that programming take into consideration the peripheral device buffer memory capacity, time needed for head movement, and similar factors. (For example, for a printer, execute WAIT commands after printing one line.) However, some devices use pin 19 (READY) with the function of ② which means that programming considerations are not necessary for these types of devices. Check the READY signal functions.

2) **Acoustic couplers**

A cable is needed to make the special connection necessary for an acoustic coupler. The details are given in section [VII. SPECIAL CABLE] (p. 21), but it is necessary to cut some jumper wires the same as in the case of 1).

Pins 19 and 20 on the CE-340R input respectively the CS signal (pin 5) and CD signal (pin 8) from the acoustic coupler. In order to temporarily halt data output (with no error) from the CE-340R when these signals are OFF (low level), cut jumper wires 02 (channels 0 or 2) or 12 (channels 1 or 3) on the CE-340R board.

See the next section for the locations of the jumper wires.

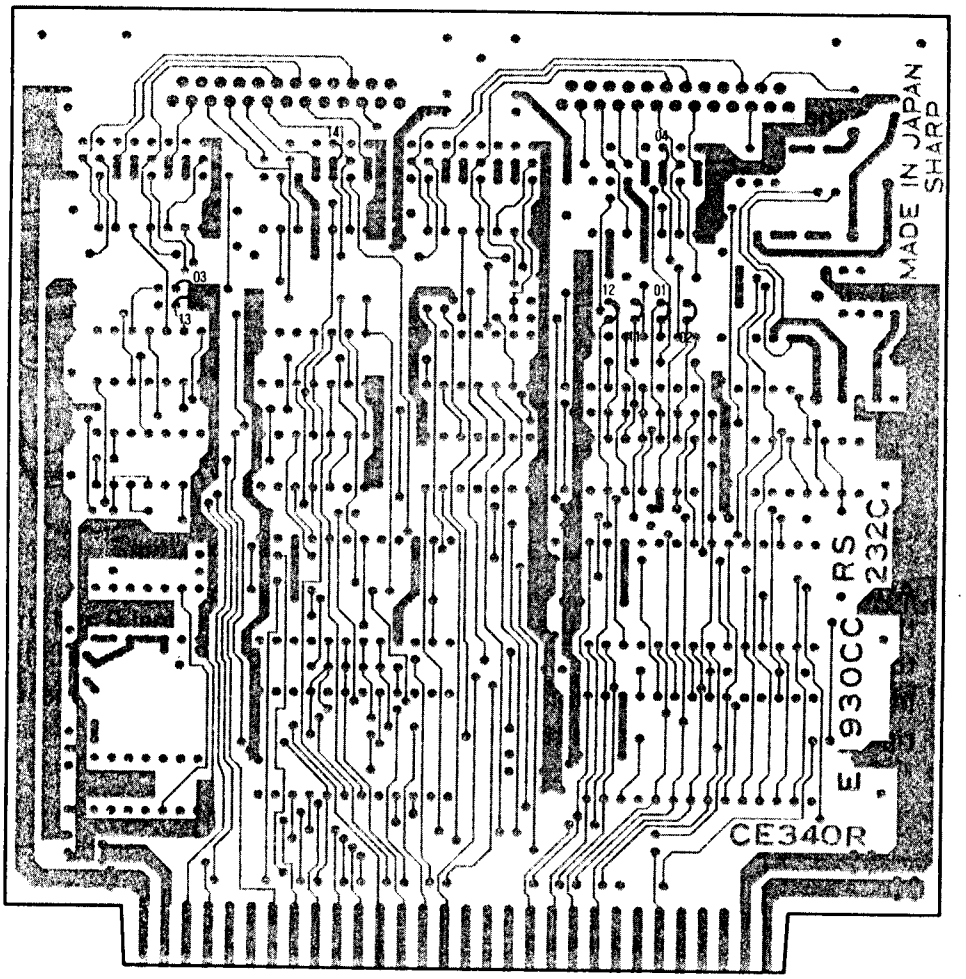
Pin 23 on the CE-340R is used to input the DR signal (pin 6) from the acoustic coupler.

In order to cause an error condition in the CE-340R when the DR signal is OFF (low level), cut jumper wires 03 (channels 0 or 2) or 13 (channels 1 or 3) on the CE-340R board.

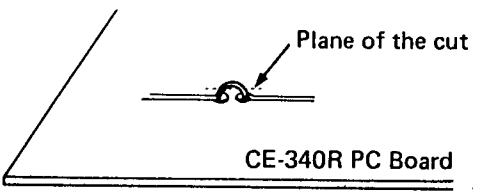
3) **Others**

Connect other devices having special signals only after a careful study of the peripheral device specifications has been made and they are fully understood. Reference section [IV. INPUT/OUTPUT SIGNALS AND CONTROL SIGNALS] (p. 5) for details on the CE-340R signals.

3. JUMPER WIRE DIAGRAM



Note) When cutting jumper wires be careful to cut them with the wire-cutters held parallel to the board. (See picture below.)

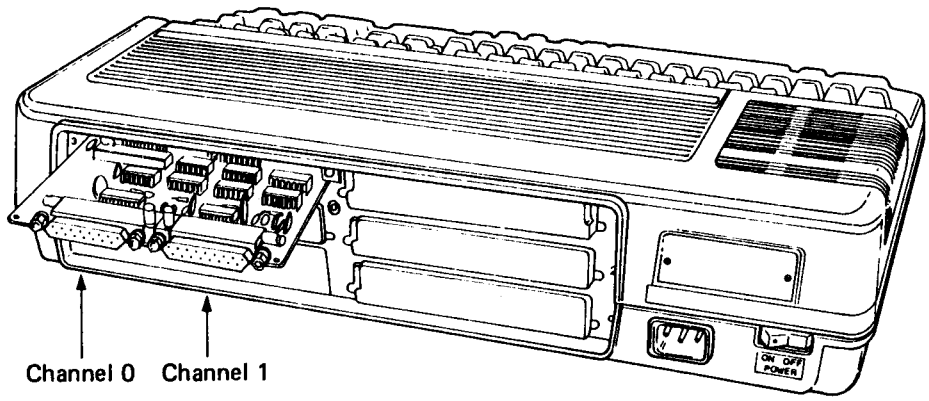


4. HOW TO CONNECT

CAUTION) **Both computer and peripheral device must have power turned off before connections are made.**

1) Installing the CE-340R

The CE-340R may be installed in any of slots 1, 2, 3 or 4 in the back of the computer in the direction as shown in the picture below (it is installed in slot 3 in the picture below). (See page 2.)
Insert the CE-340R board firmly, then insert the slot cover supplied.



2) Attaching the connector

If installed in slot 1 or 3, the connector on the left is channel 0 and the connector on the right is channel 1 when facing the back of the computer. If installed in slot 2 or 4, the connector on the left is channel 2 and the connector on the right is channel 3. (See the table below.)

Slot number	1	2	3	4
Left connector	0	2	0	2
Right connector	1	3	1	3

Channel number table

Attach the connector to the CE-340R as shown in the diagram below. After attachment use the holes provided and screws to hold it in place.

3) Power on and off

After the interface and the peripheral device have been connected turn the power on. As a rule, the computer should have power turned on first and then the peripheral device. When turning power off first turn power off at the peripheral device and then at the computer. However, details as given in the various peripheral device instruction manuals should be followed.

Ⅵ. PROGRAMMING

In order to make the command word syntax simply understood, the syntax symbols are defined and their uses explained. These symbols are for syntatical use in explaining the commands and are not related to those used in actual examples.
Note that some of the symbols used here differ from the symbols as used in the computer instruction manual.

Symbol	Meaning
|	Separates the parts on the left and right.
[ ]	Parts in brackets may be omitted.
{ }	Parts in braces may be repeated with a comma (,).
{{ }}	Parts in double braces may be repeated with a semi-colon (;).
n	Represents an integer (Example: 10).
S, T	Represents a literal constant (Example: "SUB").
A	Represents a numerical variable (Example: NO).
A%	Represents an integer type numerical variable, including dimensioned integer type numerical variables (Example: V%).
A$, B$	Represents a literal variable, including dimensioned literal variables (Example: DA$).
X, Y	Represents an equation (Example: I + 3).
C, D, E	Represents literal strings (Example: M$ + N$).

Note 1) The [ENTER] key is omitted in the syntatical explanation, but during actual execution it must be the last key pressed.
(For multiple statements a colon (:) must be entered last, followed by the next statement.)

Note 2) Enter the program with the CE-340R interface installed.

1. CHANNEL

Performs channel selection and baud rate and data format selection.

Setting the data format means selecting 7 bit or 8 bit code, stop bit length and parity.

Syntax) CHANNEL X [, Y, C]

General form: CHANNEL channel number [, baud rate, data format]

The channel is selected by the equation X, the baud rate by equation Y, and the data format by literal string C, with the channel numbers selected in the SEND command, RCV command and SENREV command being expressed by the equation X.

The baud rate (equation Y) may be 110, 300, 600, 1200, 2400, 4800 or 9600.

The date format (literal string C) is given as follows.

Literal string C: " 7 E 1 "
① ② ③

① 7 for 7 bit code and 8 for 8 bit code

② E for even parity, O for odd parity and N for no parity

③ 1 for 1 stop bit, 1.5 for 1.5 stop bits and 2 for 2 stop bits

If, Y, C are omitted, the SEND command, RCV command and SENREV command take the channel number of equation X. The baud rate and data format remain as previously set. If it was not previously set the default setting is baud rate = 300 bits/sec, data format = 7 bit code, parity = even and stop bits = 1 bit. The condition after power is turned on is that among the interfaces installed the lowest channel number of the installed channels is selected.

Addendum)

For the case of a 7 bit ASCII code input and output data format.

When the operand is X, Y, C, the execution of this command sets the SI/SO (shift in/shift out) status to SI (for the equation X channel only).

When the operand is X alone (the Y and C operands are omitted) the SI/SO status is unaffected by the execution of this command.

Example) `10 CHANNEL 1,600,"8N1"`

The above example sets the baud rate to 600 bits/sec for channel 1 with a data format of 8 bit code, no parity and 1 stop bit. At the same time the following SEND command, RCV command and SENREV command have their channels set to channel 1.

Addendum) Concerning SI/SO (shift in/shift out)

7 bit ASCII code systems normally have an SI state and an SO state allowing 7 bit data to be matched with and processed as 8 bit data.

The code to switch from an SI state to an SO state is the SO code (CHR$ &0E).

The code to switch from an SO state to an SI state is the SI code (CHR$ &0F).

(The SI code and SO code may be given regardless of the state.)

Therefore, since both the SI code and the SO code are included in the 7 bit ASCII code, attention should be given when the amount of data is a consideration in a program.

The SI state and SO state are processed independently for input and output on each channel.

2. SEND

Output data to the channel using the baud rate and data format as specified by the CHANNEL command.

Syntax 1) SEND ⟪C⟫ [;]
General form: SEND ⟪literal string⟫ [;]

Output the literal data as given in literal string C.
The literal string may be given by a dimensioned string variable. (Example: A$ (*)). Putting ; (a semicolon) after the last literal string supresses the output of a CR code (CHR$ &0D) and LF code (CHR$ &0A) after the data. The omission of the semicolon after the last literal string causes a CR code and LF code to automatically be output after the data.

Note) **The number of characters output (number of bytes) is the number of characters in the literal string. However for a dimensioned literal variable the number of characters is that as specified by the DIM statement. In this case NULL code is output after the actual characters.**

Example) `10 SEND "ABC"`

The above example outputs the literal string (literal constant) "ABC" followed by CR code and LF code.

Example) `10 SEND DM$(*)`

If in the above example the contents of DM$(*) are as shown in the table below,

[The declaration statement is DIM DM$(3,5)*4]

A A	B B B B	C C C
D	E E E	F F
G G G	H	I I
J J J J	K K	L L L
M	N N N N	

then the data is output as shown below (for 8 bit ASCII). $^{N}_{L}$ means NULL code, $^{C}_{R}$ means CR code and $^{L}_{F}$ means LF code.

A A $^{N}_{L}$ $^{N}_{L}$ B B B B C C C $^{N}_{L}$ D $^{N}_{L}$ $^{N}_{L}$ $^{N}_{L}$ E E E $^{N}_{L}$
F F $^{N}_{L}$ $^{N}_{L}$ G G G $^{N}_{L}$ H $^{N}_{L}$ $^{N}_{L}$ $^{N}_{L}$ I I $^{N}_{L}$ $^{N}_{L}$ J J J J
K K $^{N}_{L}$ $^{N}_{L}$ L L L $^{N}_{L}$ M $^{N}_{L}$ $^{N}_{L}$ $^{N}_{L}$ N N N N $^{N}_{L}$ $^{N}_{L}$ $^{N}_{L}$ $^{N}_{L}$
$^{C}_{R}$ $^{L}_{F}$

Syntax 2) SEND USING n | S | (X) | A$ | IMAGE T ;[Y | C] [;]

General form: SEND USING line number | line label name literal variable | IMAGE image signal ;[equation | literal string] [;]

Output character data as given in the euation Y and the literal strinc C according to the IMAGE statement format. Syntax is according to the PRINT statement syntax 2 as given in the computer instruction manual. Reference it along with the IMAGE statement. Placing a semicolon after the last equation or literal string suppresses the output of CR code (CHR$ &0D) and LF code (CHR$ &0A) after the last data is output. The omission of the semicolon after the last literal string causes an LF code to automatically be output after the data. (The CR code is not output.) Omission of Y | C; causes both the CR code and the LF code to not be output. The CR code and LF code can be selected by the IMAGE statement.

Example)
```
10 SEND USING 500;X$,Y$
```

The above example outputs the literal strings X$ and Y$ in the format as given by the IMAGE statement in line 500.

Example)
```
 10 H$="TRANSISTORS":K=1320
 20 SEND USING 100;H$,K
 30 END
100 IMAGE '11A 5X "PRICE" 4X $$,### C'
```

The above example outputs data in the following format. S_P is the SP (space) code.

T R A N S I S T O R S S_P S_P S_P S_P S_P P R I C

E S_P S_P S_P S_P S 1 , 3 2 0 C_R L_F

Syntax 3) SEND A% [, X]

General form: SEND integer type variable [, number of bytes output]

The low order byte of the 2 binary bytes that represent the integer type variable A% is output unaltered. The high order byte is ignored.

An integer type variable may also be a dimensioned integer type variable. (Example: B%(*))

The CR code (CHR$ &0D) and LF code (CHR$ &0A) are not output. Equation X gives the number of bytes or elements of an indexed variable that are output. When this operand is omitted all the elements of the integer variable A% are output.

However if A% is not a general indexed variable only one byte is output.

Example)
```
10 SEND M%(*),10
```

In the above example 10 bytes (10 elements) of the integer variable M% are output.

Note) Use 8 bit code with this command.

Addendum)

NULL code output is possible.

Example) 10 SEND 0%

The above example outputs NULL code.

3. RCV . . . . . RECEIVE

Input data to the channel using the baud rate and data format as specified by the CHANNEL command.

Syntax 1) RCV [/] A$, X, C [, | ; D]

General form: RCV [/] literal variable, wait time, end code 1 [, | ; end code 2]

First, if the slash (/) after the mnemonic is omitted, data is input to the literal variable A$ unit end codes C, D are reached (details given later). The end code is also input into the variable.

If the limits of the variable are exceeded before the end code is input, then an error occurs (ERROR 109). Also, NULL code is ignored and not input into the variable. The literal variable A$ may also be a dimensioned variable.

(Example: A$ (*))

For the literal strings C and D, only the first characters are taken as valid end code.

i) **Concerning the slash (/)**

Putting the slash after the mnemonic gives the host side echo back function. The host side echo back function outputs each byte of data after it is input and is only valid for low baud rates (under 1200 bits/sec).

ii) **Concerning the X equation (wait time)**

The X equation is the data wait time in units of 0.1 seconds. Therefore, if the equation is given a value of 150, the wait time is 15 seconds. If data is not input within the allowed waiting time then an error occurs (ERROR 103). Equation X must have a value in the range 0 to 255. If the equation is given the value of 0, it waits until data is input (wait time = ∞). Whatever the wait time, the wait condition can be cleared using the HALT key.

iii) **Concerning literal strings C and D (end code)**

The following is an explanation of the end codes C and D. If there is one end code (, | ; D is omitted) then data is input until the end code given by literal string C is reached. If there are 2 end codes with a comma (,) between the literal strings C and D, then data is input until either the end code given by literal string C or D is reached.

If there are 2 end codes with a semicolon (;) between the literal strings, then data is input until both end codes are reached in that order.

For example, ["A", "B"] markes either "A" or "B" the end code, while ["A" ; "B"] makes "AB" the end code.

Example) 10 RCV A$,0,"@"

In the above example data is input into the literal variable A$ until the end code " @ " is reached. (wait time = ∞)

Note) **In the case of 7 bit code, if the end code is set to CHR$ & 01 – &1F or CHR$ &81 – &9F, only 7 bit data is used regardless of the condition of SI/SO. And if set to CHR$ & 20 – &7F or CHR$ & AO – &FF, it is identified by the SI/SO condition and 7 bit data.**

Syntax 2) RCV [/] A%, X, C [, | ; D]

General form: RCV [/] integer variable, wait time, end code 1 [, | ; end code 2]

Input numeric data (binary data) into the integer variable A% 1 byte for 1 variable (2 bytes) until the end codes in literal strings C or D are reached. In other words, a single binary byte of data is input into the low order byte of an integer variable, with the high order byte being made 00 hex.

Therefore the values input into the integer variable A% must be in the range 0 to 255.

The integer variable may also be a dimensioned integer variable (Example: C% (*)). The end code is also input into the variable (element of the dimensioned variable). The slash (/) and equation X follow the same rules as in Syntax 1). In Syntax 2) NULL code can be input into the variable.

Example) `10 RCV B%(*),100,"A"`

In the above example numeric data (binary data) is input into the dimensioned variable B%(*) 1 byte at a time until data with the value of end code "A" (41 hex) is reached. (wait time = 10 seconds respectively)

4. SENREV ..... SEND & RECEIVE

Output data, then immediately thereafter input data using the channel, baud rate and data format as given by the CHANNEL command.

This command is a combination of the SEND and RCV commands.

Syntax 1) SENREV C, A$, X, D [, | ; E]

General form: SENREV literal string , literal variable, wait time, end code 1 [, | ; end code 2]

Output literal data as given by literal string C and immediately thereafter input data to the literal variable A$ until end codes D or E are reached.

The details are as given in the sections on SEND command Syntax 1) and RCV command Syntax 1).

However, CR code (CHR$ &0D) and LF code (CHR$ &0A) is not output after the end of data output. Also, there is no echo back function.

Example) `10 SENREV "READY",D$,0,CHR$ &0D`

In the above example, immediately after "READY." is output, literal data is input to the variable D$ until the CR code (end code) is reached. (wait time = ∞)

Addendum)

In case literal string C is used as a dimensioned variable, end code (for the output) should come at the end of output data.

Syntax 2) SENREV C, A%, X, D [, | ; E]

General form: SENREV literal string, integer variable, wait time, end code 1 [, | ; end code 2]

Output the literal data as given by literal string C then immediately thereafter input numeric data (binary data) 1 byte per variable (2 bytes) into the integer variable A% until the end codes of literal strings E or D are reached.
The details are as given in the section on the SEND command Syntax 3), the RCV command Syntax 2) and Syntax 1) of this command.

5. OPCHNL . . . . . OPEN CHANNEL

Opens the input buffers (1 byte in length) on a given channel.

This command is valid with respect to the POLLING command mentioned later. For details use the POLLING command as a reference for comparison.

Syntax 1) OPCHNL {X}

General form: OPCHNL {channel numbers}

Opens the input buffer memory on the channels as given in equation X. The contents of the given channels' buffer memories are cleared.
The open/closed condition of channels other than those specified by the X equation are not affected.
The command is valid with respect to the POLLING comand mentioned later and is unrelated to the SEND, RCV, or SENREV commands.

Note) **The SI/SO (shift in/shift out) state (use for input commands) of the channels opened with this command are put into the SI state by the execution of this command (for the case of 7 bit codes).**

Example) 10 OPCHNL 0,3

In the above example channels 0 and 3 are opened.

Addendum)

There is a one byte buffer memory on each channel for input use. If input of 2 or more bytes occurs only the last byte remains in buffer memory.

Syntax 2) OPCHNL

All the open channels are closed.

They are also closed by the SEND, RCV and SENREV commands (only those channels doing input and output).

6. POLLING

Check the contents of the buffer memories in the channels opened by the OPCHNL command.

Syntax) POLLING C, A

General form: POLLING comparison code, numeric variable

Check the contents of the buffer memories used for input on the channels opened by the OPCHNL command starting with the lowest channel number and input into the numeric variable A the channel number of the channel whose contents match the comparison code in literal string C. Set the channel number of the input and output commands to the value as given by the numeric variable A. (See CHANNEL command.)

If a channel having data matching the comparison code is found, channels having larger channel numbers are not checked. If a channel having data matching the comparison code is not found, the numeric variable A = −1 and the channels in the commands used for input and output are not affected.

Only the first character in the literal string C is used as comparison code. If 2 or more characters are given, the second and following characters are ignored.

Note) A channel having code matching the comparison code is automatically closed.

Addendum)

The OPCHNL and POLLING commands may for example be used in the following way.

When connecting several peripheral devices to one computer, it is possible to pass information between the computer and a specific peripheral device when the given peripheral device sends a request code to the computer that matches a previously specified code and the computer checks for that code in the channels to the peripheral devices.

Example)

```
100 CHANNEL 0,300,"7E1"
110 CHANNEL 1,1200,"7E2"
120 CHANNEL 2,110,"8N1"
130 CHANNEL 3,600,"701"
140 OPCHNL 0,1,2,3
150*"POL":POLLING CHR$ &05,A
160 IF A<0 THEN "POL"
170 SENREV "READY",D$,0,CHR$ &0D
    :
500 GO TO "POL"
```

In the above example the statement in line number 150 sets the channel number to that of the channel found to have the ENQ code (CHR$ &05) and the statement in line number 170 performs data input and output.

7. A SIMPLE PROGRAMMING EXAMPLE

The following example assumes that a printer with keyboard (KSR type) is connected. (A mini-floppy disk is also used.)

.....

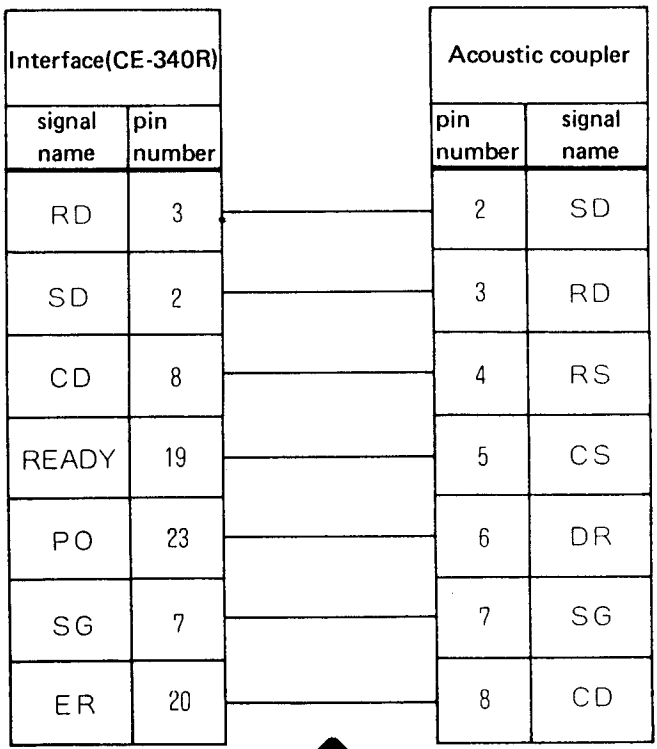
Example)

```
100 !! REGISTER
110     MAXFILE 1
120     OPEN "0",1,"TEL:A1"                                  File name "TEL" open
130     CHANNEL 0,300,"7E1"                                  Channel setting
140     SEND :SEND "* REGISTER":SEND :SEND
150     SENREV "NUMBER OF MEMBERS ?          ",NM$,0,CHR$ &0D,CHR$ &7F   Inputting
160     IF RIGHT$ (NM$,1)=CHR$ &7F SEND "(DELETE)":GO TO 150          the number
170     NM=VAL LEFT$ (NM$,LEN NM$-1)                          of members
180     SEND USING 190;NM                                    Delete code processing
190     IMAGE 'C L "*  NUMBER OF MEMBERS IS " ### C'
200     BPRINT #1,NM                                         Writing into the disk
210     FOR I=1 TO NM
220     SEND "NO.";STR$ I;"          ";                      Outputting the number
230     SENREV "       NAME ?          ",NA$,0,CHR$ &0D,CHR$ &7F   Inputting the name
240     IF RIGHT$ (NA$,1)=CHR$ &7F SEND "(DELETE)":GO TO 230      Delete code processing
250     SENREV "       TEL ?          ",TE$,0,CHR$ &0D,CHR$ &7F   Inputting a telephone
260     IF RIGHT$ (TE$,1)=CHR$ &7F SEND " (DELETE)":GO TO 250          number
270     SEND USING 280;I,LEFT$ (NA$,LEN NA$-1),LEFT$ (TE$,LEN TE$-1)   Delete code
280     IMAGE 'C L "*  NO." ### 10X "NAME : " 15A 10X "TEL NO." 15A C'   processing
290     DISP USING 300;I,LEFT$ (NA$,LEN NA$-1),LEFT$ (TE$,LEN TE$-1)
300     IMAGE '"NO." ### ":" 15A "...." 15A'
310     BPRINT #1,NA$,TE$:NEXT I                             Writing into the disk
320     SEND USING 330;
330     IMAGE 'C L L "* END OF REGISTER" C L L L L L L L L L L'
340     CLOSE 1                                              Closing the data file
350     DISP "END OF LIST":END
```

Note) End code: CHR$ &0D, CHR$ &7F (for delete code); wait time: ∞

Ⅶ. SPECIAL CABLE(ACOUSTIC COUPLER CONNECTION CABLE)

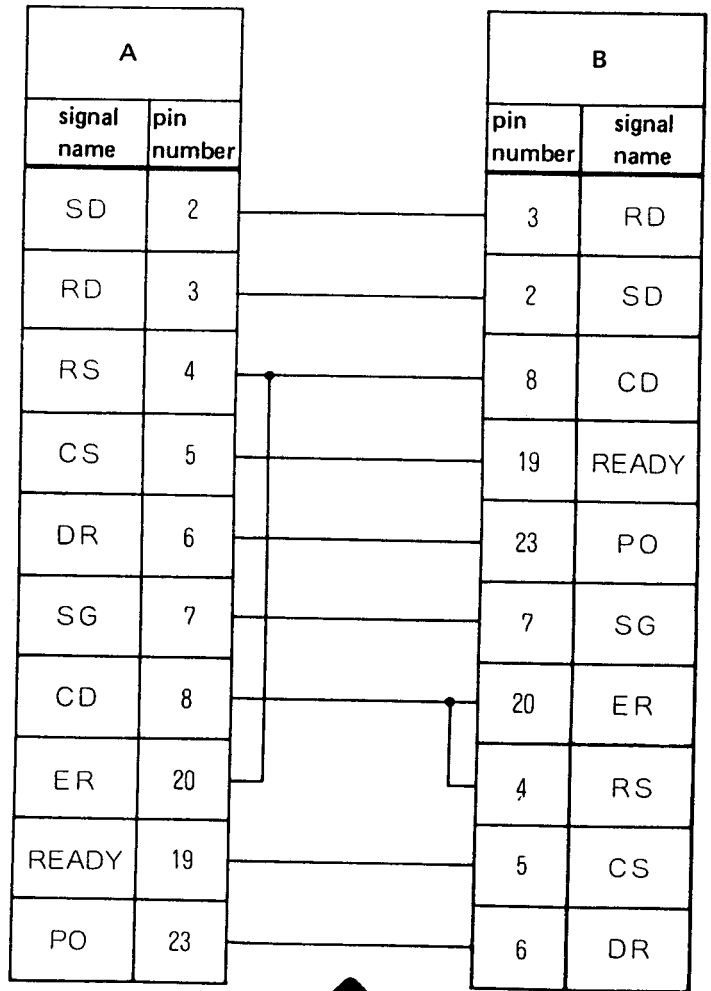
When connecting this interface to an acoustic coupler prepare a cable having the connections as shown below.



Be sure to cut the correct jumper wires. (See page 00.)

Addendum) Shown here is another way of making the connection.

Using a cable wired as follows eliminates the distinction between the interface side and the acoustic coupler side. It will operate when hooked up to any connector and it is possible for two personal computers having CE-340R interfaces installed to exchange information.



* unused pins are open.

Special cable

* * * MEMO * * *

APPENDIX—1 RLIST COMMAND

This command is valid with respect to some peripheral devices (printers). Use it with those devices where applicable.

[Function]

For a printer, output the program list in the personal computer when specified.

Syntax) RLIST S [, X [, Y]]

General form: RLIST title [, line number 1 [, line number 2]]

A valid command for a printer, it causes the page number, month/day, hour: minute and the title expressed by the literal constant S to be output on each page and lists the program from line number X to Y.
The details are as written in the Syntax 3 LIST command section of computer instruction manual.

Note 1) **Special symbols (CHR$ &8O — &9F and CHR$ &E) — &FB) appearing in the program are all converted to the underline code (CHR$ &5F) when output.**

Note 2) **Lowercase characters are output in their respective codes. Printers not having these codes will fail to print them when output or may, depending on the printer, perform special operations.**

Addendum)

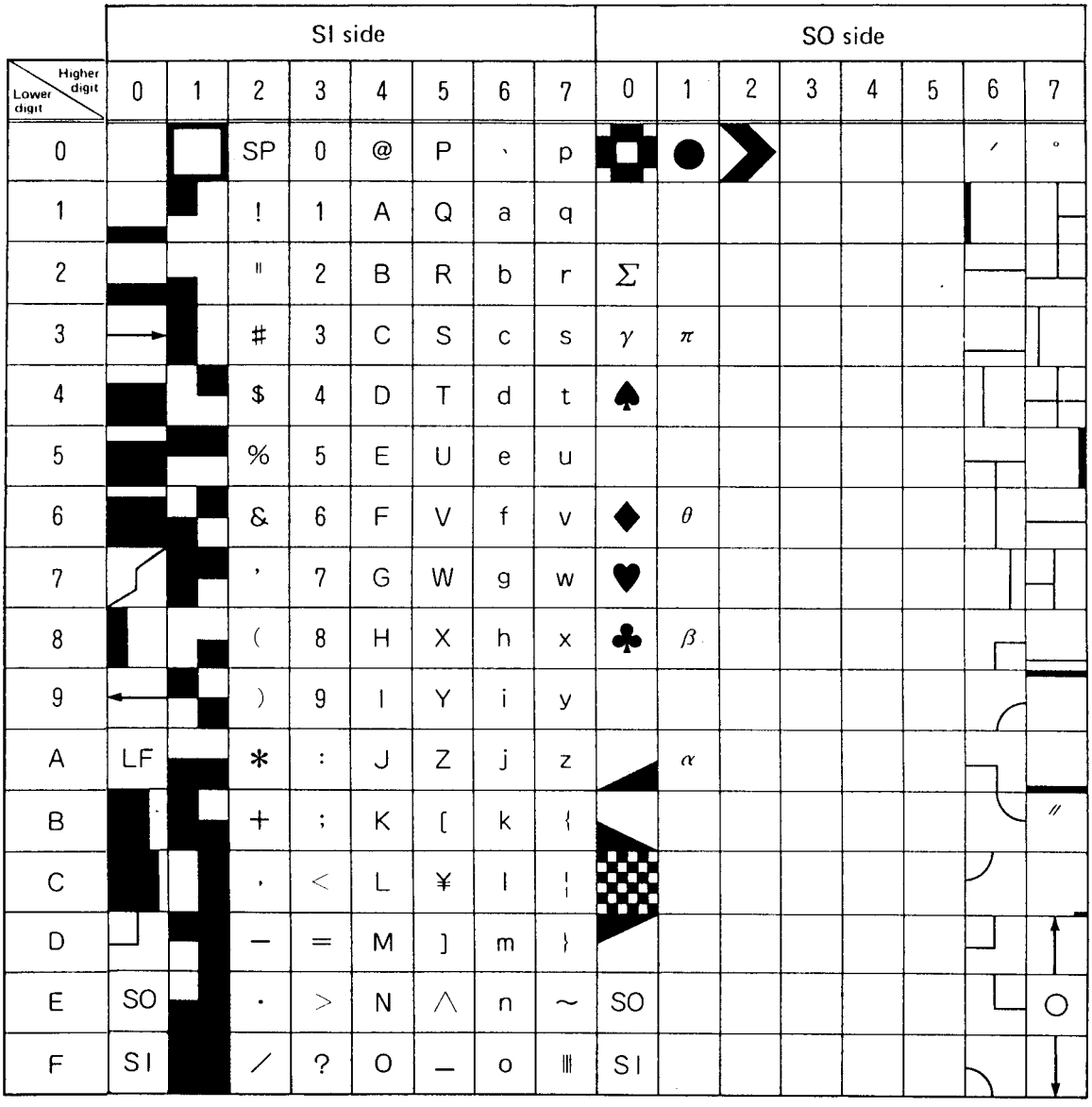
The execution of this command does not use the Formfeed (FF) function. Formfeed is performed by using the CR code and LF codes.

Addendum)

The number of columns output by RLIST command is 80 columns or less from the left margin.

APPENDIX—2 CODE TABLE FOR INPUT

1. 7 BIT ASCII CODE TABLE

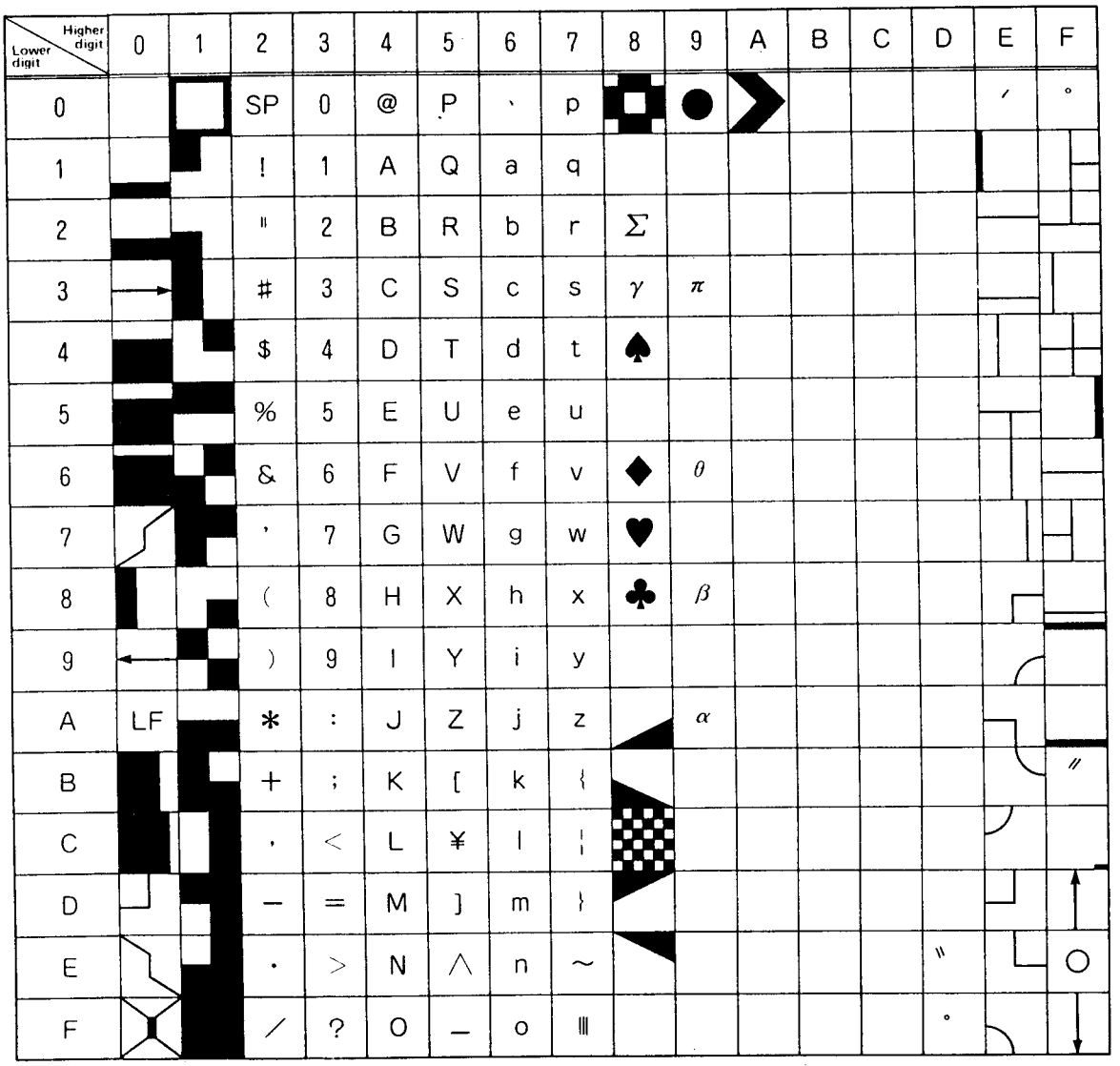


Lower digit \ Higher digit	SI side 0	1	2	3	4	5	6	7	SO side 0	1	2	3	4	5	6	7
0			SP	0	@	P	`	p							´	°
1			!	1	A	Q	a	q								
2			"	2	B	R	b	r	Σ							
3			#	3	C	S	c	s	γ	π						
4			$	4	D	T	d	t	♠							
5			%	5	E	U	e	u								
6			&	6	F	V	f	v	◆	θ						
7			'	7	G	W	g	w	♥							
8			(	8	H	X	h	x	♣	β						
9			)	9	I	Y	i	y								
A	LF		*	:	J	Z	j	z		α						
B			+	;	K	[	k	{								//
C			,	<	L	¥	l	¦								
D			−	=	M	]	m	}								
E	SO		.	>	N	^	n	~	SO							○
F	SI		/	?	O	_	o	▒	SI							

Note 1) **LF : Carriage Return Line Feed**
SO : Shift Out
SI : Shift In

Note 2) **Character codes which are left blank in the above table are used for Japanese characters, except for the character code 00 on the SI side.**

2. 8 BIT ASCII CODE TABLE



Lower digit \ Higher digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	`	p								
1			!	1	A	Q	a	q								
2			"	2	B	R	b	r	Σ							
3			#	3	C	S	c	s	γ	π						
4			$	4	D	T	d	t	♠							
5			%	5	E	U	e	u								
6			&	6	F	V	f	v	◆	θ						
7			'	7	G	W	g	w	♥							
8			(	8	H	X	h	x	♣	β						
9			)	9	I	Y	i	y								
A	LF		*	:	J	Z	j	z		α						
B			+	;	K	[	k	{								
C			,	<	L	¥	l	¦								
D			−	=	M	]	m	}								
E			.	>	N	^	n	~								
F			/	?	O	_	o	▒								

Note 1) LF : Carriage Return Line Feed

Note 2) Character codes which are left blank in the above table are used for Japanese characters, except for the character code 00.

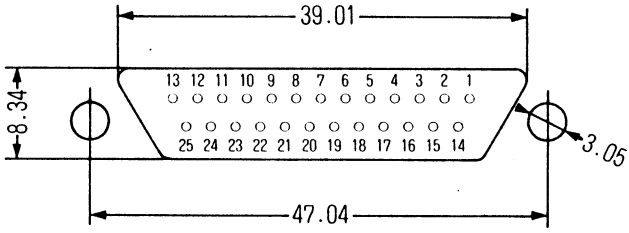
APPENDIX—3 ERROR CODE TABLE

ERN	Error condition
(odd)	
101	Peripheral error during data output.
103	No data input after the wait time has passed.
105	Parity error or framing error (stop bit incorrectly positioned) during data input.
107	Overrun error during data input (baud rate is too high).
109	Variable used for input overflows during data input.

Note) The odd numbered error code numbers (ERN) give a branch when the ON ERROR statement is used.

ERN	Error condition
(even)	
100	3 or more interface boards are installed. The positions in which the interface boards are installed (slot numbers) are inappropriate.
102	The specified channel number has no interface installed.
104	The channel number specified is inappropriate. The wait time of the input command is inappropriate.
106	The operand of a CHANNEL command is inappropriate.
108	The number of bytes output as specified by a SEND command (integer variable output) is inappropriate.
110	Execution of a special command (inappropriate command).
112	Interrupt error due to external noise, etc.

APPENDIX—4 CE-340R CONNECTOR PINOUT



The size above is that of connector with cable.
[units: mm]

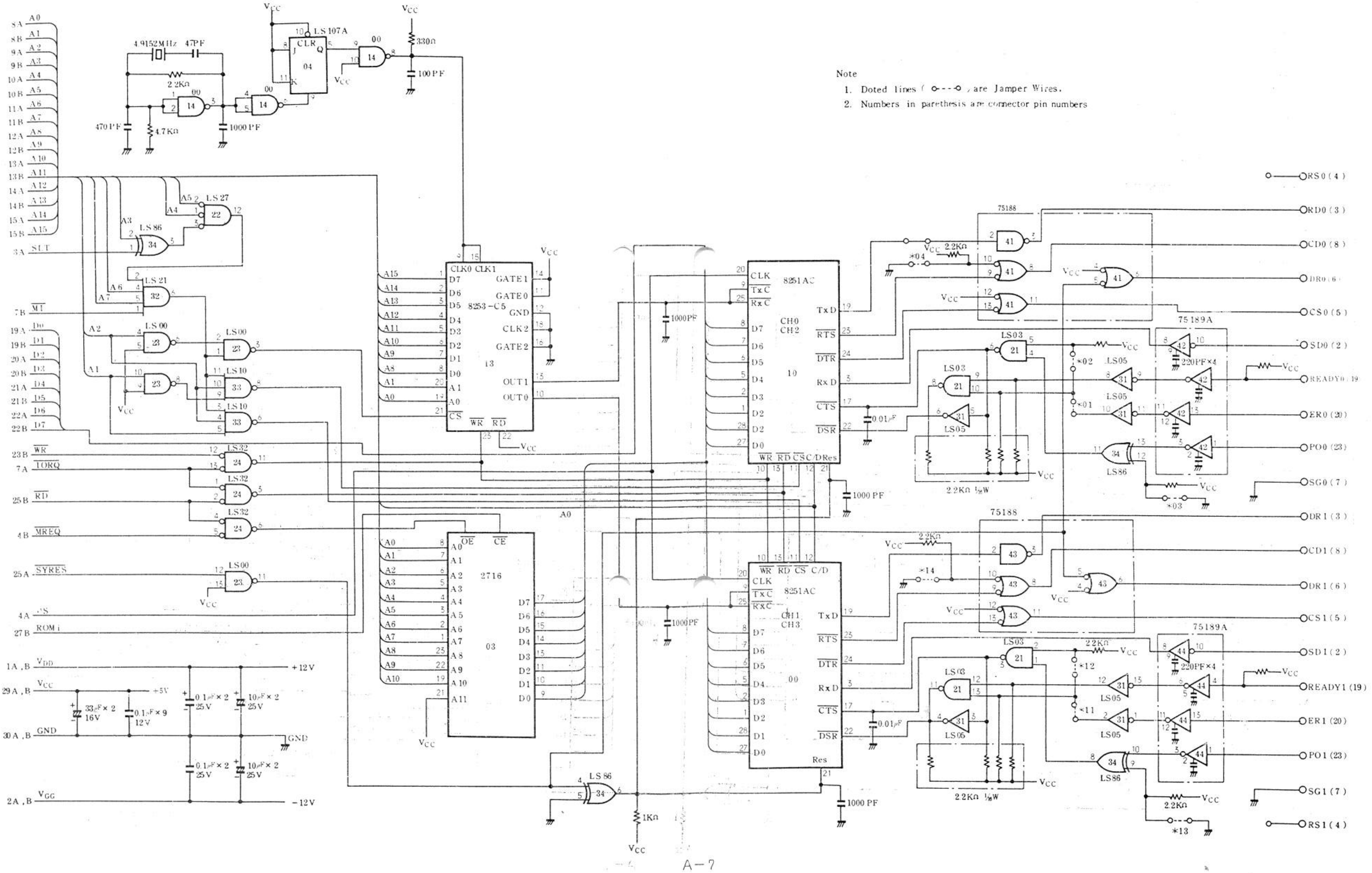
Pin number	Signal name	Abbreviation	EIA code
1	Unused	(FG)	(AA)
2	Send data	SD	BA
3	Receive data	RD	BB
4	Unused	(RS)	(CA)
5	Clear to send	CS	CB
6	Data set ready	DR	CC
7	Signal ground	SG	AB
8	Carrier detect	CD	CF
9	Unused		
10	Unused		
11	Unused		
12	Unused		(SCF)
13	Unused		(SCB)
14	Unused		(SBA)
15	Unused		(DB)
16	Unused		(SBB)
17	Unused		(DD)
18	Unused		
19	Ready	READY	SCA
20	Data terminal ready	ER	CD
21	Unused		(CG)
22	Unused		(CE)
23	Paper out	PO	CH/CI
24	Unused		(DA)
25	Unused		

( ) are unused pins.

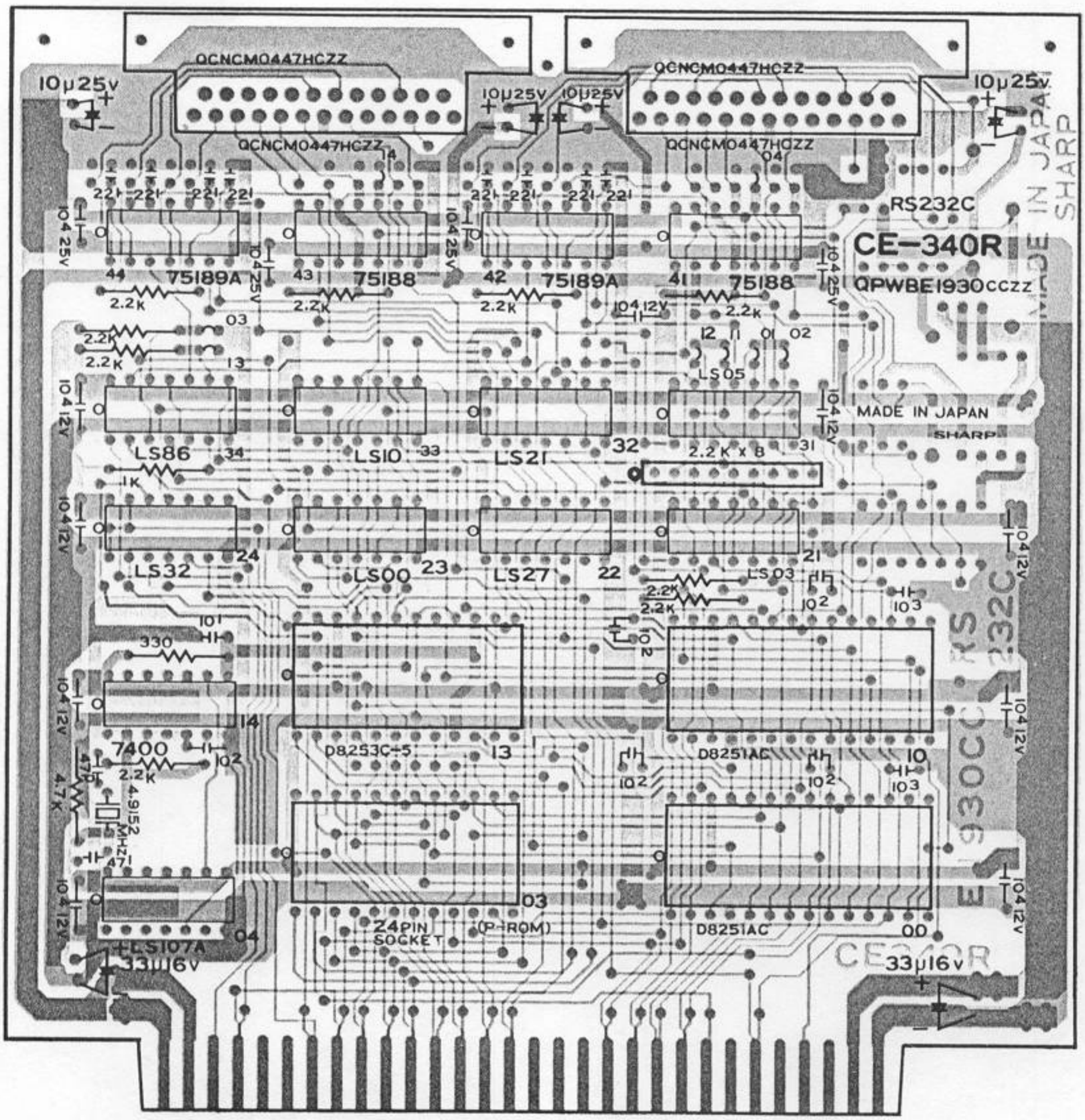
APPENDIX—5 JUMPER WIRE STATUS LIST

Jumper wire		Jumper wire not cut (as shipped)	Jumper wire cut
Channel 0, 2	Channel 1, 3		
01	11	Error occurs when the ER signal (pin 20) goes OFF (low level) or is open during data output.	ER signal (pin 20) is ignored.
02	12	Same as above.	Waits (suspends) data output when ER signal (pin 20) is OFF (low level) or open.
03	13	Error occurs when the PO signal (pin 23) is ON (high level) during data output.	Reverses the polarity of the PO signal to the left.
04	14	The CD signal (pin 8) is always ON (high level) when the interface (personal computer) has power on.	The CD signal (pin 8) is ON (high level) only during data output.

APPENDIX—6 CE-340R CIRCUIT DIAGRAM



APPENDIX–7 CE-340R PARTS POSITION DIAGRAM



SHARP CORPORATION
OSAKA, JAPAN

Printed in Japan
1D030(3175)