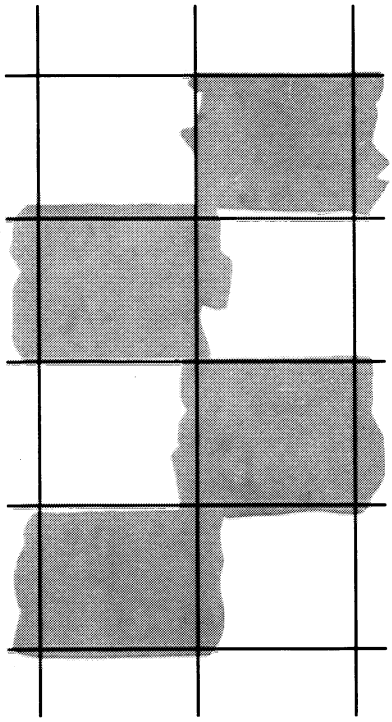
TOSHIBA

PROGRAMMABLE CONTROLLER

COMPUTER LINK-FUNCTION



EX100

USER'S MANUAL

INSTALLATION AND OPERATION

Important Information

Misuse of this equipment can result in property damage or human injury. Because controlled system applications vary widely, you should satisfy yourself as to the acceptability of this equipment for your intended purpose. In no event will Toshiba Corporation be responsible or liable for either indirect or consequential damage or injury that may result from the use of this equipment.

No patent liability is assumed by Toshiba Corporation with respect to the use of information, illustrations, circuits, equipment, or examples of application in this publication.

Toshiba Corporation reserves the right to make changes and improvements to this publication and/or related products at any time without notice. No obligation shall be incurred, except as noted in this publication.

Contents

How to read this manual

Overview

Cable connections

Transmission control

Control codes

Appendices

Inside this manual

This manual provides all the information you need to install, operate, and troubleshoot the EX100 computer link function.

This manual is divided into four main parts and an appendices section:

① Features and overview
② Cable connections
③ Data format
④ Transmission command and error code

Appendices: Application program
Specifications
Troubleshooting

Note and caution symbols

"Notes" call the reader's attention to information considered important for best understanding and operation of the equipment.

"Cautions" call the reader's attention to conditions or practices that could damage equipment or render it temporarily inoperative.

Related publications

EX100 User's Manual (UM-EX100 ** -E001)

Terminology

Many hardware terms particular to PC operation are defined in the glossary of "Guidebook to Manuals." You should familiarize yourself with these terms before proceeding. The following is a list of abbreviations and acronyms used in this manual.

ACO	ac output
A/D	analog/digital
AI	analog input
AO	analog output
AWG	American wire gage
CMOS	complementary metal oxide semiconductor
CPU	central processing unit
DI	dc input
DO	dc output
FD100	floppy disk drive unit
GP110	graphic programmer
GP110AP1	graphic programmer, stand-alone model
GND	ground
H	hexadecimal (when it appears in front of alphanumeric string)
Hz	Hertz
INP	input (ac)
I/O	input/output
LCD	liquid crystal display
LED	light-emitting diode
LSI	large-scale integration circuitry
MP100	miniprogrammer
NEMA	National Electrical Manufacturers Association
PC	programmable controller
PI	pulse input
PR100	PROM writer
PROM	programmable read-only memory
RAM	random access memory
RO	relay output
RS-232C, RS-422, RS-485	serial interfaces
RTD	resistance temperature detector (input)
UL	Underwriters' Laboratories, Inc.
Vac	ac voltage
Vdc	dc voltage

Features

Using the computer link module, you can connect a host computer (IBM PC, PC/XT, PC/AT™ or other compatible personal computer) to one or more EX100 unit and execute the following functions.

1) Supervising the EX100 (reading the RUN/HALT/ERROR status)
2) Controlling the EX100 operation mode (RUN/HALT)
3) Writing and reading data to/from the EX100 (including data on devices and registers)
4) Documenting and program storage
5) Programming (reading/writing/modifying program)

System configuration

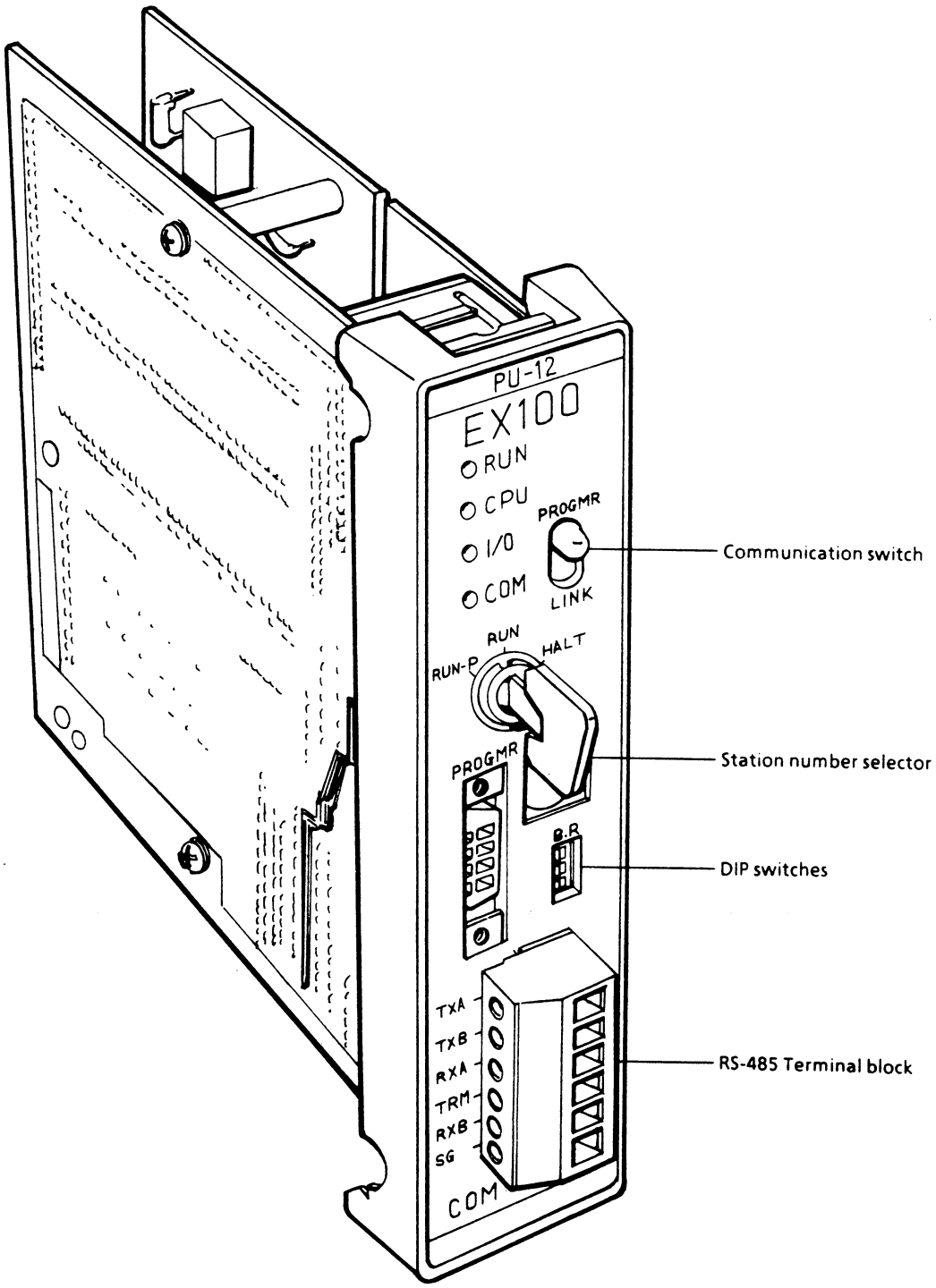
With the computer link system, one or more EX100 programmable controllers can be linked to a single host computer, using one of two transmission modes. In the **one-to-one** transmission mode, a single PC is linked to the host computer. In the **one-to-N** transmission mode, from two to 16 EX100 units can be linked to the host computer, using the computer link function.

In either mode, each computer link is linked to the host computer through an RS-485 interface. With this system the maximum transmission distance is 1 km (3,280 ft).

If the host computer uses an RS-232C interface, it is necessary to adapt it to a RS-485 connector by using Toshiba's RS-485/RS-232C conversion adapter (ADP-6237B).

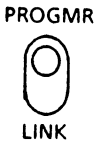
* IBM PC, PC/XT, PC/AT are registered trademarks of International Business Machines Corp.

Front panel


PU-12
EX100
RUN
CPU
I/O
COM
PROGMR
LINK
RUN
RUN-P
HALT
PROGMR
B.R
TXA
TXB
RXA
TRM
RXB
SG
COM
Communication switch
Station number selector
DIP switches
RS-485 Terminal block

Switch settings

Communication switch



Selects the device you'll use to communicate with the EX100 programmable controller.

PROGMR: Programmer; LINK: Computer link

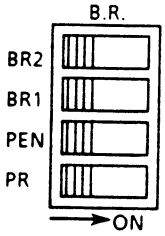
Station number selector

Selects the station number when the computer link has been selected via the communication switch.

Switch position	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Station number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

DIP switches

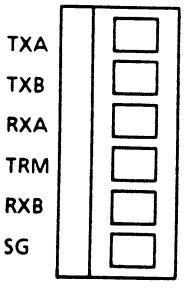
Baud rate:



Switch \ Baud rate	9600	4800	2400	1200
BR2	OFF	OFF	ON	ON
BR1	OFF	ON	OFF	ON

Parity mode:

Switch \ Parity mode	Non		Even	Odd
PEN	OFF	OFF	ON	ON
PR	OFF	ON	OFF	ON

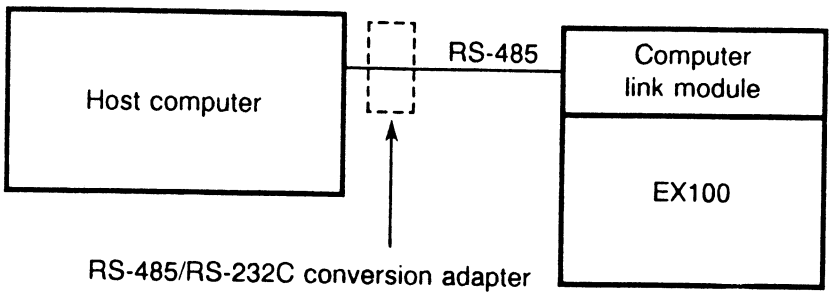


RS-485 transmission terminal

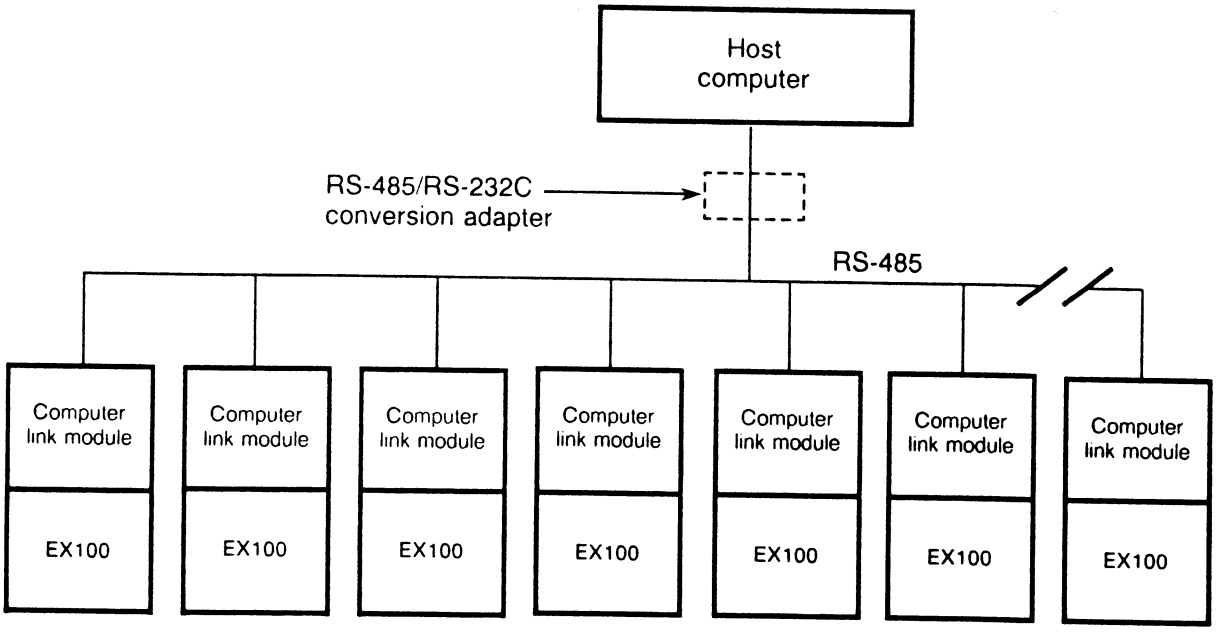
TXA: Sending line A (output)
TXB: Sending line B (output)
RXA: Receiving line A (input)
TRM: Terminal for termination resistor for one-to-one transmission.
RXB: Receiving line B (input)
SG: Signal ground

System configuration

Transmission type: One-to-one



Transmission type: One-to-N



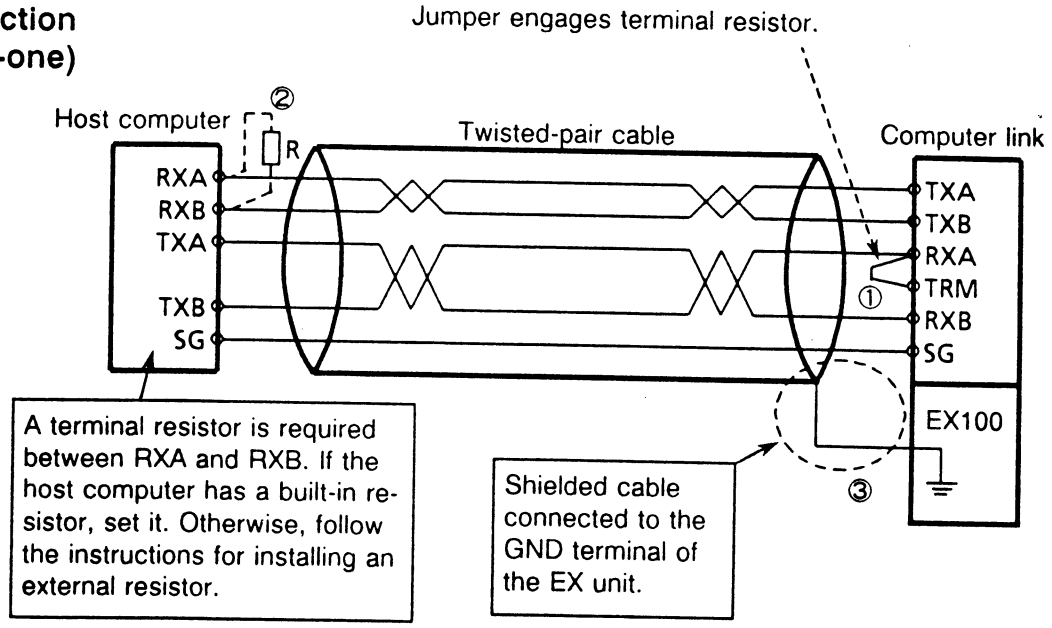
NOTE When computer-link modules are installed in several EX100 programmable controllers that are connected to a single host computer, a maximum of 16 stations (EX100 units) can be connected at one time.

One-to-one transmission mode

When the one-to-one transmission configuration is used, cable connections should be as follows:

① Jumper the RXA terminal on the computer link module to the TERM terminal. This utilizes the built-in terminal resistor.
② If the host computer does not have a built-in terminal resistor, connect an external ½-watt, 120-Ω resistor between the computer's RXA and RXB terminals.
③ Connect one end of the shielding wire of the shielded twisted-pair cable to the ground terminal (GND).

RS-485 connection (one-to-one)

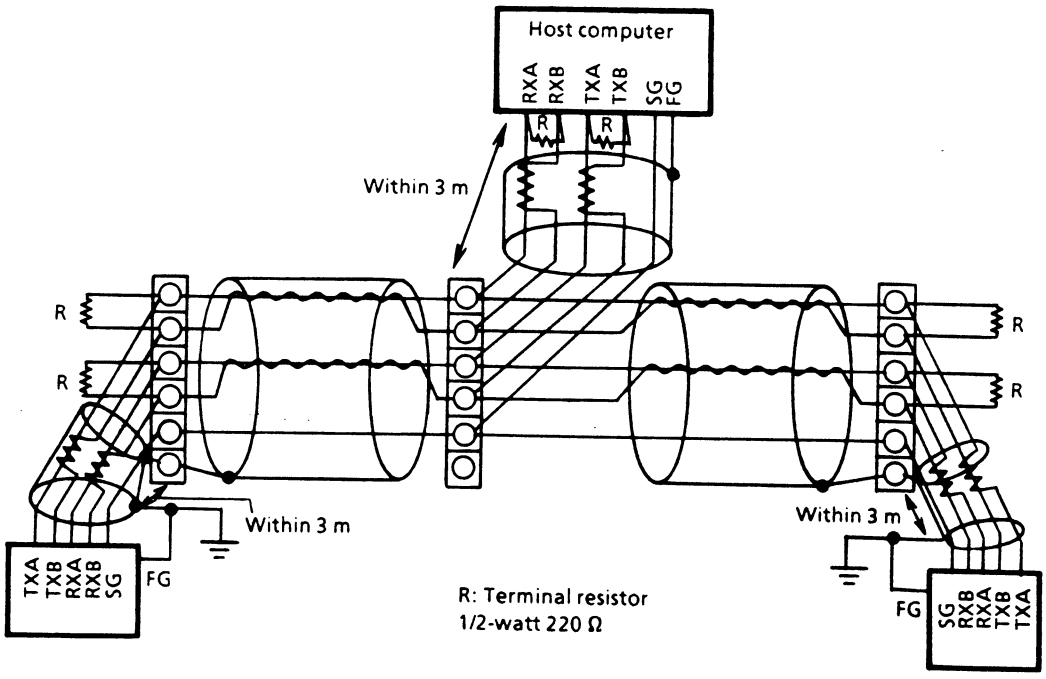


One-to-N transmission mode

When two or more computer links (2 to 16) are linked to the host computer, the basic cable connections are made as follows.

① Connect terminal resistors between terminals RXA-RXB and TXA-TXB at each end of the cable run. (The combined terminal resistance must be ½-watt, 220Ω.)
② Connect the cable shield to frame ground at one end only. Use the GND terminal for the ground connection on the EX100 side.
③ Install a terminal block to branch off to the host computer or computer link. The branch should not exceed 3m (10 ft) cable length from the terminal block to the computer or module.

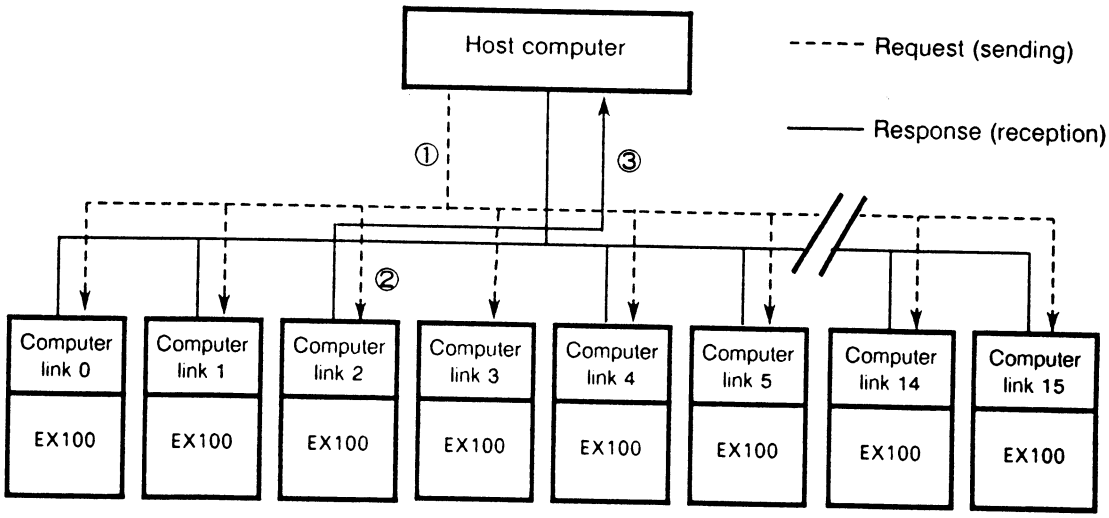
RS-485 connection (one-to-N)



Transmission overview

In a multiple-computer link configuration, each computer link module constantly waits for requests issued by the host computer. When a request is generated, each computer link module checks the station number of the request. The module with the matching station number processes the request while the other link modules continue to stand by. This is why each computer link must have a unique station number. Otherwise more than one link may attempt to process the request, resulting in faulty data.

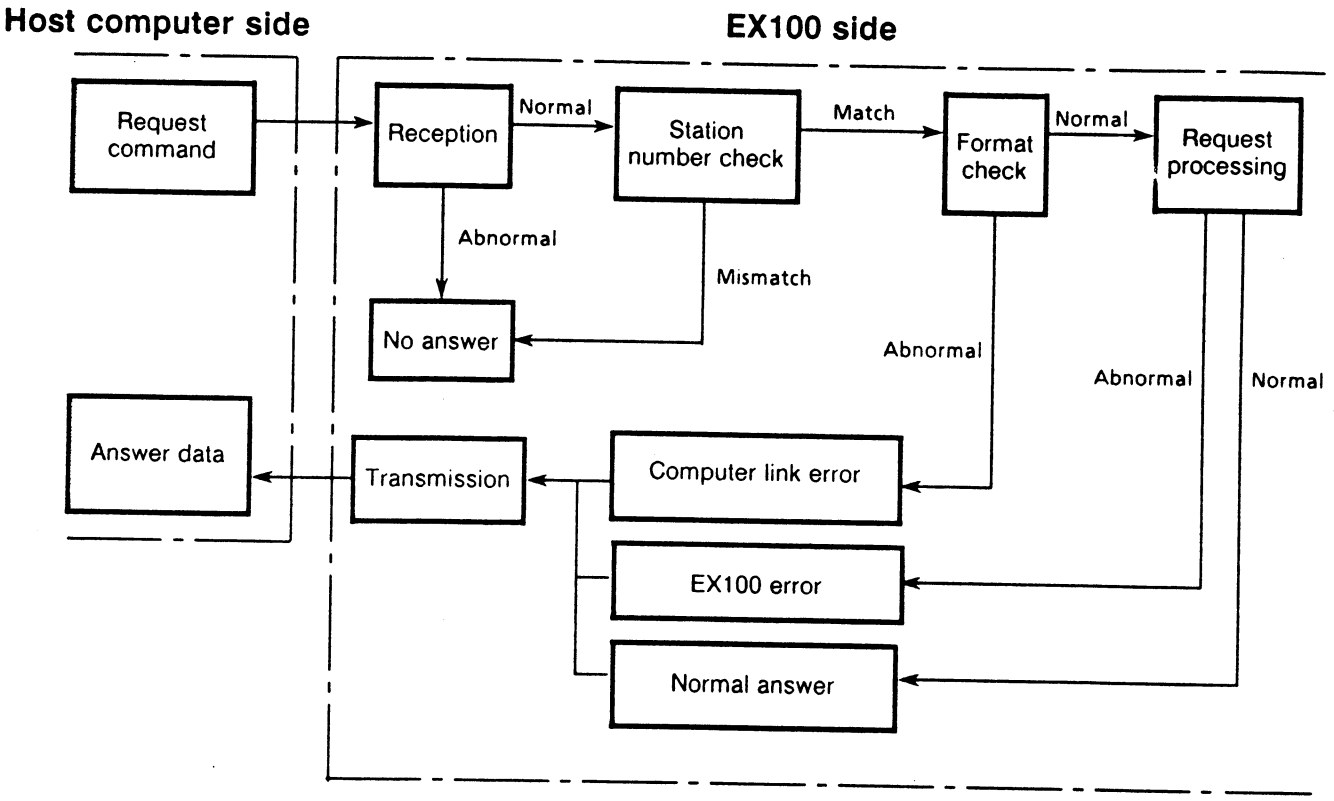
The following diagram illustrates the processing sequence executed when a request to station 2 is issued:



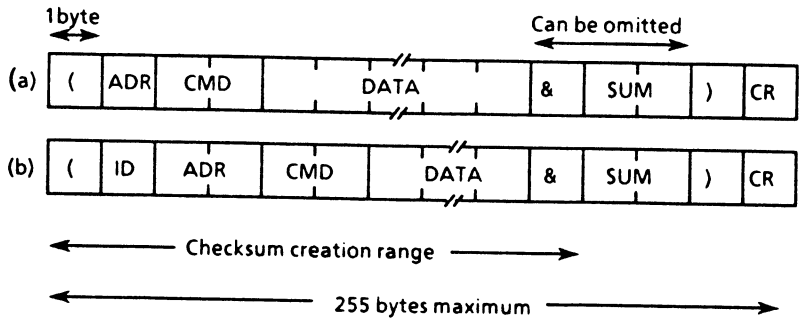
Processing sequence executed when a request to station 2 is issued by the host computer:

① Request is sent from host computer to computer links (request to station 2).
② Request is executed by station with same number as request (only station 2 executes request).
③ Processing result is returned as response to host computer (response from station 2).

Signal flow The following diagram illustrates the signal flows during normal and abnormal communication between the host computer and an EX100 programmable controller.



Transmission protocols



Either transmission format (a) or (b) can be used. However, when (a) is used a maximum of 10 EX100s (station numbers 0 through 9) can be accessed because of the one-digit ADR limitation of (a). When (b) is used, all 16 stations (station numbers 00 through 15) can be accessed because this format can use two digits in its ADR sector.

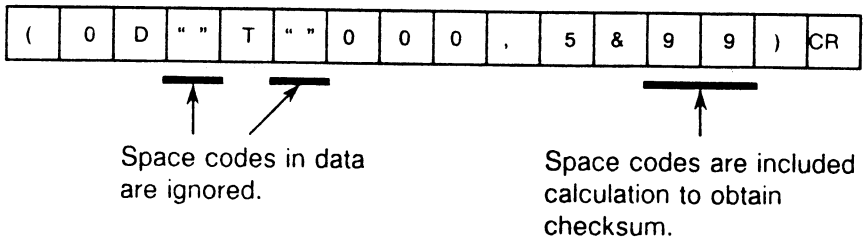
Text contents

- "(": Transmission begins (H28) — 1 byte
- ID: Format identification code"A" (H41) — 1 byte
- ADR: Station number
  - Format (a) — 1 byte
    - "0" (H30) through "9" (H39)
  - Format (b) — 2 bytes
    - "00" (H30, H30) through "15" (H31, H35)
- CMD: Command — 2 bytes
- DATA: Data field
- "&": Checksum delimiter (H26) — 1 byte
- SUM: Checksum — 2 bytes
  8-bit ASCII code of the lowest-order byte of the sum obtained by adding the bytes from transmission-begin to the checksum delimiter
- ")": Ending code (H29) — 1 byte
- CR: Carriage return code (H0D) — 1 byte

The following transmission protocols apply to the computer link functions of the EX100:

(1) When on and functioning normally, the EX100 always is waiting for a request command from the host computer. The EX100 will not transmit any data without a request.

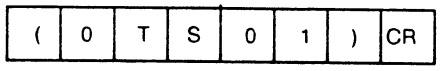
(2) All space codes (H20) are ignored.
If space codes (H20) are included in the data received from the host computer, they are ignored. However, when checksum is obtained, the space codes are included in the calculation.



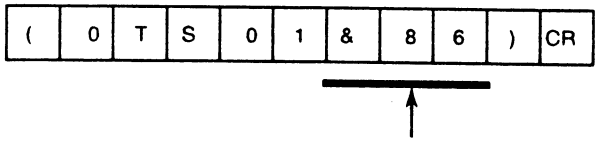
(3) Checksum can be omitted in the transmission text.
Data can be transmitted from the host computer whether or not checksum is attached to the data.

(4) Checksum is attached to the answer text. Checksum is always attached to the answer data from the EX100.

Transmission text

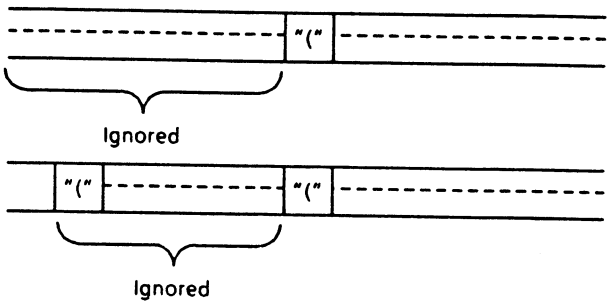


Answer text



(5) 8-bit ASCII code is used. All transmission/reception data consists of 8-bit ASCII code.

(6) Data received before the "(" (open parenthesis mark) code is ignored. The EX100 ignores all the data received before "(" (H28).



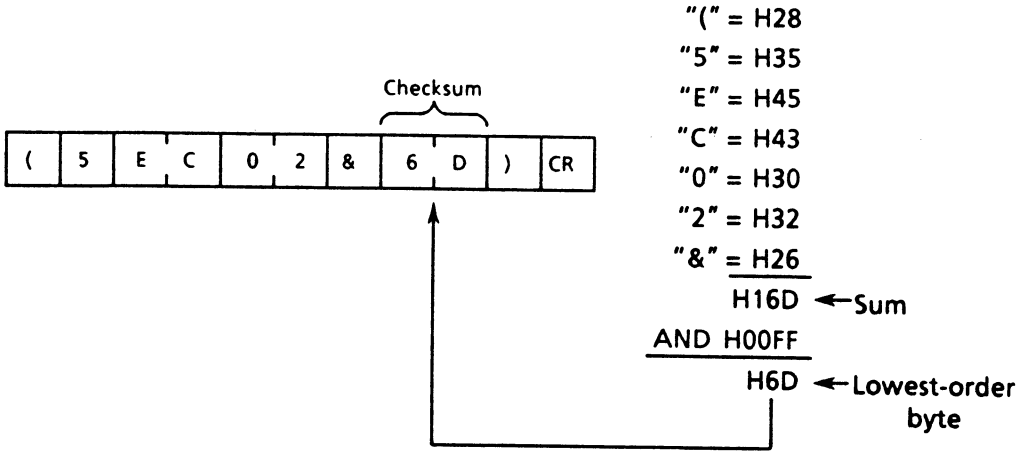
(7) ")" CR terminates reception.
Reception is considered to be terminated only after ")" (closed parenthesis mark) and CR have been received. If a ")" without a CR (carriage return), or vice versa, is received, it is considered to be a transmission error.

NOTE In this manual, data enclosed within " " (quotation marks) indicates data for 8-bit ASCII code.

Creation of checksum

Checksum is the ASCII code of the lowest-order byte of the sum obtained by adding the data from the transmission-begin code "(" to the checksum-delimiter code "&".

Example:



Control codes for send/receive data

The following control codes for send/receive data are available with the computer link network.

Send code	Contents	Function	Receive code	Remarks
—	Computer link error	Indicates a transmission error has been detected in computer link	CE	Occurs when processing terminates abnormally
—	EX main-frame error	Indicates error has occurred in EX100	EE	Occurs when processing terminates abnormally
ER	EX error status read	Reads EX100 error status	ER	
TS	Test	Returns transmitted text as it was received by computer link	TS	
ST	EX status read	Reads EX100 mode (RUN/HALT/ERROR)	ST	
DR	Device/ register read	Reads device or register contents	DR	
DW	Device/ register write	Writes device or register contents	DW	
BR	Program block read	Reads programs block-by-block	BR	
BW	Program block write	Writes programs block-by-block	BW	
EC	EX control	EX100 operation status control	EC	
IR	Power failure storage area read	Reads starting register of power failure memory retention area	IR	
SR	System parameter read	Reads EX100 system information	SR	
TR	Diagnostic table read	Reads user-defined error information	TR	

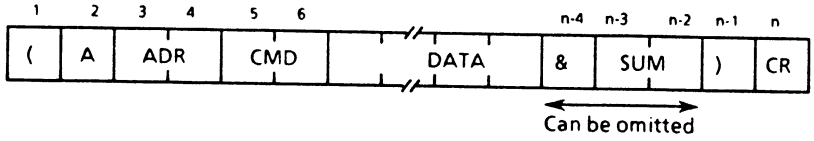
NOTE The command descriptions given in the following pages are for transmission format (b).

CE
Computer link error

When the computer link detects a parity error, checksum error, etc., or when a format mismatch is detected as the computer link is receiving data, the host computer receives a "CE" error message from the computer link.

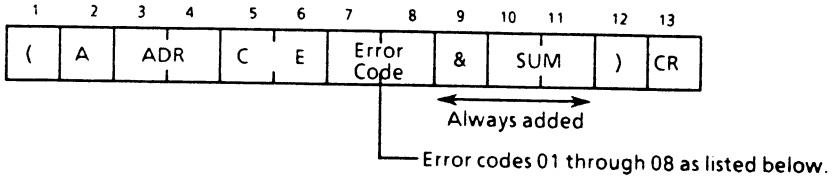
Transmit data

Host computer → Computer link



Receive data

Host computer ← Computer link



Codes for the CE error

Computer link error code	Error	Explanation
01	Command error	Appropriate command is missing
02	Format error	Transmission format mismatch detected
03	Checksum error	Checksum mismatch detected
04	Ending code error	Ending code ")" and CR have not been received
05	Excessive text length	Text exceeds 255 bytes
08	Time-out 1	Time gap of 1 second or longer occurs during reception of series of data from host computer

Execution examples

The following two execution examples demonstrate how you check the RECEIVE DATA line to determine which computer link error (e.g., CE01) has occurred. The following "input" and "send" data was input to demonstrate computer link Test 0 (described in Appendix A).

Command error

```
INPUT DATA  =(A00TT)
SEND DATA   =(A00TT&97)
RECEIVE DATA=(A00CE01&D8)
```

Computer link error (CE) with error code 01: **appropriate command is missing.**

Format error

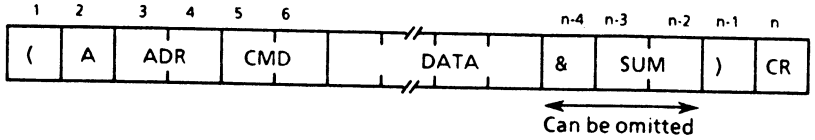
```
INPUT DATA  =(A00DRRW60,8)
SEND DATA   =(A00DRRW60,8&F8)
RECEIVE DATA=(A00CE02&D9)
```

Computer link error (CE) with error code 02: **register to be read is missing.**

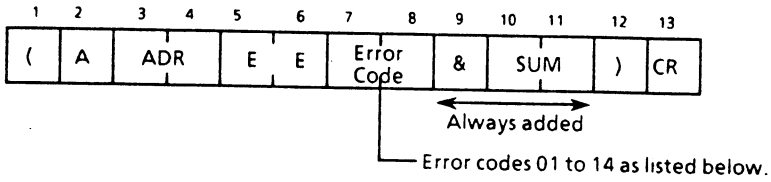
EE
EX error

When the EX100 receives an erroneous request, it sends an "EE" error to the host computer.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



Codes for the EE error

Error code	Error	Specification
01	No END instruction	END instruction not detected upon execution of RUN start program check
02	Illegal pair instruction	One of following detected upon RUN start program check: 1. More than three nesting levels exist 2. Same pair instruction exists more than once in single nest 3. Reset instruction located prior to set instruction 4. Two pair instructions overlap
03	Program failure	1. Program has been destroyed 2. Program checksum failure detected
04	Memory full	Memory became full due to repetitive program write and/or insertion
05	Illegal page/circuit number	Requested page or circuit does not exist
06	Mode mismatch	Invalid mode command received

Error code	Error	Specification
07	PROM error	Attempt to rewrite program made after PROM was installed
08	Operand error	Operand in program does not match I/O device assignment information (Example: when only output modules are used, XW00 is used)
09	Erroneous register number/size	Undefined register specified, or register specified was larger than range of EX100
10	I/O mismatch	I/O programmed did not match the installed modules
11	I/O NO SYNC	No response returned from I/O module
12	Erroneous transmission	EX100 produced text reception processing error, or undefined command was received
13	Type mismatch	Memory and I/O type recorded on cassette tape did not match those of EX100
14	Page full	Page in program contains more than 255 instructions

Execution examples

Three execution examples are provided to demonstrate how you check the RECEIVE DATA line to determine which computer link error (e.g., EE09) has occurred. The following data are input to computer link Test 0 (described in Appendix A).

No END instruction

```
INPUT DATA   =(A00EC02)
SEND DATA    =(A00EC02&D9)
RECEIVE DATA=(A00EE01&DA)
```

EX100 error (EE) with error code 01: **END instruction is missing.**

Erroneous mode

```
INPUT DATA  =(A00EC02)
SEND DATA   =(A00EC02&D9)
RECEIVE DATA=(A00EE06&DF)
```

EX100 error (EE) with error code 06: **mode mismatch.** In this case, the data indicate that the program was executed with the RUN/HALT switch in the HALT position.

I/O mismatch

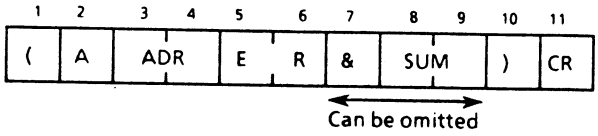
```
INPUT DATA  =(A00EC02)
SEND DATA   =(A00EC02&D9)
RECEIVE DATA=(A00EE10&DA)
```

EX100 error (EE) with error code 10: **I/O mismatch.** The data indicate that the I/O module assignments made in the program differ from the I/O modules' actual configuration.

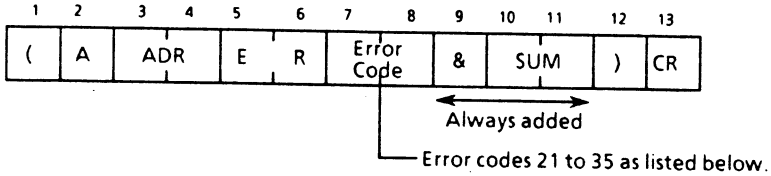
ER
EX error status read

If an error occurs in the EX100 status, the "ER" command can be used to read the error's cause and determine why the EX unit shut down.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



CE or EE is returned when a transmission error occurs.

ER command error codes

Error code	Error	Remark
00	Normal (no errors)	Normal operation
21	No END instruction	END instruction not detected when program check was executed during initialization
22	Illegal pair instruction	During program check at initialization, at least one of the following errors was detected: 1. More than three nesting levels exist 2. Same pair instruction detected more than once in single nest 3. Reset instruction precedes set instruction 4. Two pair instructions overlap
23	Program failure	1. Program was destroyed 2. Program checksum failure detected
24	Illegal instruction	Undefined instruction (illegal instruction) detected during program execution

Error code	Error	Remark
25	Scan time-out	Scan time exceeded 200ms
26	I/O SYNC error	I/O module failed to respond
27	I/O module mismatch	When power supply was turned on with RUN/HALT switch in RUN position, programmed I/O did not match installed I/O
28	Operand error	Operand in program did not match I/O module assignment information (e.g., When the first slot contains output module, XW00 is used)
29	dc power supply voltage failure	Drop in dc supply voltage of EX100 to below required level
30	Watchdog timer error	Occurs when watchdog timer is not reset within required time (350ms)
31	CPU error	Peripheral LSI error detected
33	Extension I/O power failure	Drop in supply voltage of extension I/O unit to below required level
34	PROM failure	Failure detected in PROM contents of EX100
35	I/O bus failure	I/O bus failure detected immediately before initialization or batched I/O processing

Execution examples

Three execution examples are provided to demonstrate how to check the RECEIVE DATA line to determine which computer link error (e.g., ER33) has occurred. The following data are input to computer link Test 0 (described in Appendix A).

Normal operation

```
INPUT DATA    =(A00ER)
SEND DATA     =(A00ER&86)
RECEIVE DATA =(A00ER00&E6)
```

EX100 error status read (ER) with error code 00: **no error.**

No END instruction

```
INPUT DATA    =(A00ER)
SEND DATA     =(A00ER&86)
RECEIVE DATA =(A00ER21&E9)
```

EX100 error status read (ER) with error code 21: **no END instruction.**

I/O mismatch

```
INPUT DATA    =(A00ER)
SEND DATA     =(A00ER&86)
RECEIVE DATA =(A00ER27&EF)
```

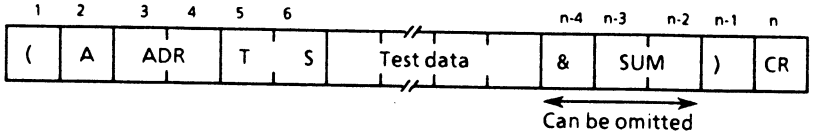
EX100 error status read (ER) with error code 27: **I/O registration mismatch.**

TS
Test

This command tests each computer link connected to your host computer. The computer link returns the transmitted text unchanged to the host computer. If the transmitted data contains no checksum, a checksum is added to the receive data.

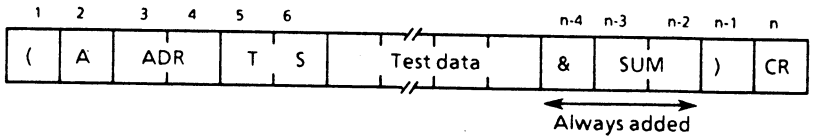
Transmit data

Host computer → Computer link



Receive data

Host computer ← Computer link



Eight-bit ASCII codes can be used as test data, except for the following five special characters, which are reserved for system use.

"(" H28
"&" H26
")" H29
CR H0D

} Error is indicated if one of these codes is used.

" " H20 — Ignored (space)

CE or EE is returned when a transmission error occurs.

Execution examples

Two execution examples are provided to demonstrate how to check the RECEIVE DATA line to determine which computer link error (e.g., CE02) has occurred, or to indicate that the test has been completed with no errors found. The following data are input to computer link Test 0 (described in Appendix A).

Normal TS

```
INPUT DATA    =(A00TSABCDEFGHIJKLMNOPQRSTUVWXYZ)
SEND DATA     =(A00TSABCDEFGHIJKLMNOPQRSTUVWXYZ&75)
RECEIVE DATA =(A00TSABCDEFGHIJKLMNOPQRSTUVWXYZ&75)
```

Data show that the test (TS) has completed normally.

TS error

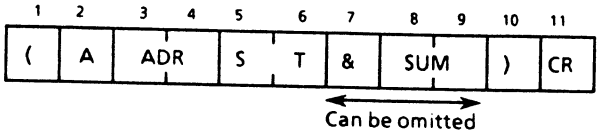
```
INPUT DATA    =(A00TS)
SEND DATA     =(A00TS&96)
RECEIVE DATA =(A00CE02&D9)
```

Computer link error (CE) with error code 2: **format error; test (TS) has no data.**

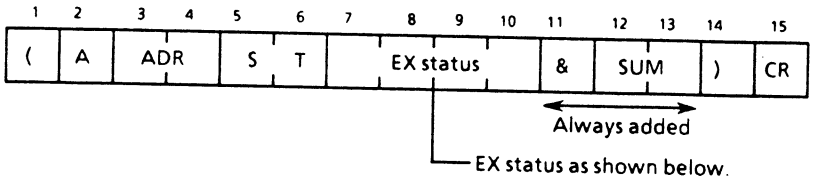
ST
EX status read

This command (ST) reads the EX100 unit's status (RUN, HALT, ERROR) and receives error codes when errors occur. When ST is executed, EX100 operation continues — unless an error down occurs.

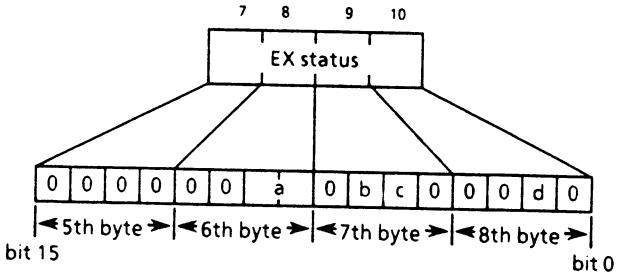
Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



EX status



a = EX mode
01 (1): HALT
10 (2): RUN/RUN-F
11 (3): ERROR

b = 1: TOSLINE error
c = 1: Computer link error
d = 1: Request for diagnostic display

Execution examples Three execution examples are provided to demonstrate how to check the RECEIVE DATA line to determine the EX100's status (RUN, HALT, ERROR). The following data are input to computer link Test 0 (described in Appendix A).

Reading HALT

```
INPUT DATA    =(A00ST)
SEND DATA     =(A00ST&96)
RECEIVE DATA =(A00ST0100&57)
```

Mode 01 in EX status: **HALT mode.**

Diagnostic display in RUN

```
iNPUT DATA    =(A00ST)
SEND DATA     =(A00ST&96)
RECEIVE DATA =(A00ST0202&5A)
```

The contents of a user-defined error can be received by reading the diagnostic table when the diagnostic display request appears.

Reading error status

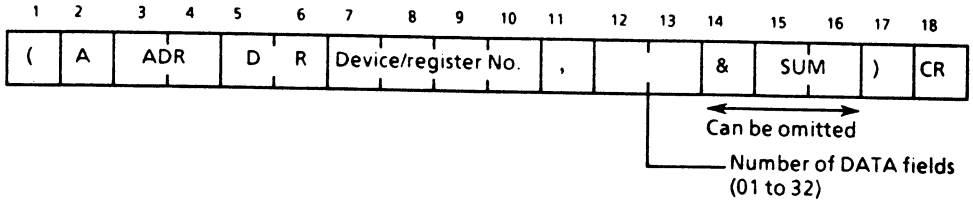
```
INPUT DATA    =(A00ST)
SEND DATA     =(A00ST&96)
RECEIVE DATA =(A00ST0300&59)
```

EX mode 3 of EX status indicates an error.

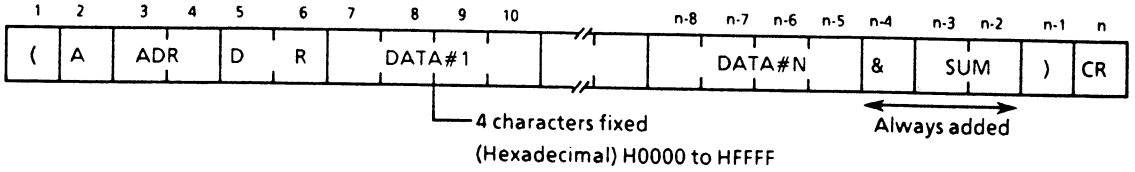
DR
Device/register read

The device/register read command (DR) is used to read the contents of devices or registers consecutively.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



CE or EE is returned when a transmission error occurs.

When devices are read, either 0000 (OFF) or 0001 (ON) is read out.

By executing this command once, data can be read from 1 to 32 consecutive devices or registers.

Instead of R001, either R01 or R1 can be specified in the device/register number field. Likewise, 1 to 9 can be specified in the DATA fields instead of 01 to 09.

Express all values in the DATA fields in hexadecimal format.

NOTE If data registers D1000 to D1535 are used, omit the D from each register number. Specify the remaining digits in the device/register number field.

Execution examples Three execution examples are provided to demonstrate use of the "DR" code. The following data are input to computer link Test 0 (described in Appendix A).

Reading RW, R

To read auxiliary relay registers/devices (RW, R):

```
  INPUT DATA = (A00DRRW00,1)
   SEND DATA = (A00DRRW00,1&EB)
RECEIVE DATA = (A00DR682A&66)  →  RW00=H682A

  INPUT DATA = (A00DRR0,16)
   SEND DATA = (A00DRR0,16&9A)
RECEIVE DATA = (A00DR0000 0001 0000 0001 0000 0001 0000
                   R0   R1   R2   R3   R4   R5   R6
                 0000 0000 0000 0000 0001 000C 0001
                   R7   R8   R9   RA   RB   RC   RD
                 0001 0000&8B)
                   RE   RF
```

Reading XW, YW, X, Y

To read I/O registers/devices (XW, YW, X, Y):
(The same data is read from XW and YW (X and Y)).

```
  INPUT DATA = (A00DRXW0,2)
   SEND DATA = (A00DRXW0,2&C2)
RECEIVE DATA = (A00DR0000F3D5&37)

  INPUT DATA = (A00DRYW0,2)
   SEND DATA = (A00DRYW0,2&C3)
RECEIVE DATA = (A00DR0000F3D5&37)
```

Registers XW and YW have the same contents.

```
  INPUT DATA = (A00DRXA,16)
   SEND DATA = (A00DRXA,16&B1)
RECEIVE DATA = (A00DR0000 0000 0000 0000 0000 0000 0001
                   XA   XB   XC   XD   XE   XF   X10
                 0000 0001 0000 0001 0000 0001 0001
                   X11  X12  X13  X14  X15  X16  X17
                 0001 0001 &8C)
                   X18  X19

  INPUT DATA = (A00DRYA,16)
   SEND DATA = (A00DRYA,16&B2)
RECEIVE DATA = (A00DR0000 0000 0000 0000 0000 0000 0001
                   YA   YB   YC   YD   YE   YF   Y10
                 0000 0001 0000 0001 0000 0001 0001
                   Y11  Y12  Y13  Y14  Y15  Y16  Y17
                 0001 0001 &8C)
                   Y18  Y19
```

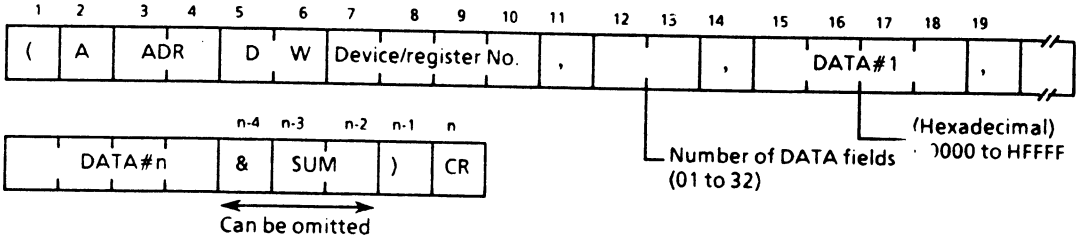
Reading D To read data registers (D):

```
  INPUT DATA = (A00DRD999,8)
   SEND DATA = (A00DRD999,8&D8)
RECEIVE DATA = (A00DR0000 0003 0003 1234 0000 0000 0000
                     D999   ~    ~    ~    ~    ~    ~
                      0000 &95)
                      D1006

  INPUT DATA = (A00DR1100,8)
   SEND DATA = (A00DR1100,8&AB)
RECEIVE DATA = (A00DR0000 0001 0002 0003 0004 0005 0006
                    D1100   ~    ~    ~    ~    ~    ~
                      0007 &A1)
                      D1107
```

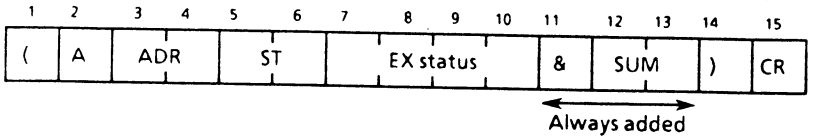
DW
Device/register write

The computer link function's device/register write (DW) command is used to write the contents of consecutive devices or registers.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link

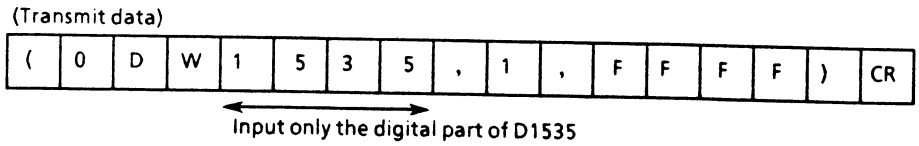


Writing 0000 turns the specified device OFF; writing 0001 turns it ON. The DW command can be used in write-enable mode by setting the write-enable switch to the ON (upper) position prior to operation.

This command allows data to be written into one to 32 consecutive devices or registers. Instead of R001, either R01 or R1 may be specified in the device/register number field. Likewise, 1 to 9 may be specified as the number of DATA fields instead of 01 to 09. Input the DATA field value in hexadecimal format.

If data registers D1000 to D1535 are used, omit the D when requesting data from these registers and use only the numerical portion.

Example: Transmit data used to write HFFFF into D1535



Execution examples Two execution examples are provided to demonstrate the use of the "DW" code. The following data are input to computer link Test 0 (described in Appendix A).

Writing data into RW

To write data into auxiliary registers (RW) and confirm the written data:

In the following examples, first a value is written to two successive registers. Then the values of the registers are read back two ways – first using the register-read command, then using the device-read command.

```
INPUT DATA   =(A00DWRW00,2,FFFF,FFFF)···Writing into
               RW00 and RW01
SEND DATA    =(A00DWRW00,2,FFFF,FFFF&79)
                           RW00 RW01
RECEIVE DATA=(A00ST0102&59)

INPUT DATA   =(A00DRRW00,2)···Reading from RW00 and
               RW01
SEND DATA    =(A00DRRW00,2&EC)
RECEIVE DATA=(A00DRFFFF FFFF&B5)
                     RW00 RW01

INPUT DATA   =(A00DRR000,32)···Reading from R000 to
               R01F
SEND DATA    =(A00DRR000,32&F8)
RECEIVE DATA=(A00DR0001 0001 0001 0001 0001 0001 0001
              0001 0001 0001 0001 0001 0001 0001 0001
              0001 0001 0001 0001 0001 0001 0001 0001
              0001 0001 0001 0001 0001 0001 0001 0001
              0001&A5)
```

All 32 devices = 0001 (ON)

Writing data into T

To write data into timer registers (T_n) and confirm the written data:

The values are written to five successive timer registers, and the values are then read back using the device/register read command.

```
INPUT DATA   =(A00DWT0,5,AAAA,BBBB,CCCC,DDDD,EEEE)
                ···Writing into T0, T1, T2, T3, and T4
SEND DATA    =(A00DWT0,5,AAAA,BBBB,CCCC,DDDD,EEEE&87)
                          T0   T1   T2   T3   T4
RECEIVE DATA=(A00ST0102&59)


INPUT DATA   =(A00DRT0,5)···Reading from T0, T1, T2, T3,
                and T4
SEND DATA    =(A00DRT0,5&6A)
RECEIVE DATA=(A00DR7FFF 7FFF 7FFF 7FFF 7FFF&B2)
                    T0   T1   T2   T3   T4
```

All four timers are indicated as being set to the maximum value of H7FFF.

NOTE

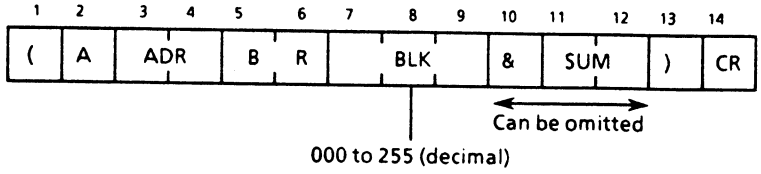
The limit of the timer value is H7FFF.

BR
Program block read

The program block-read (BR) command is used to read a program a block at time.

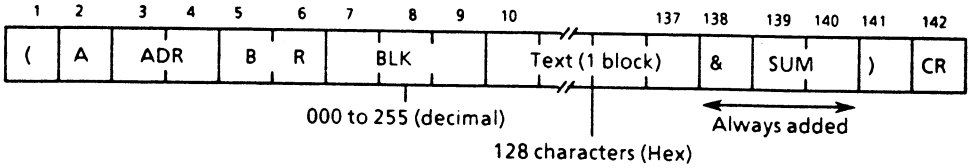
Transmit data

Host computer → Computer link

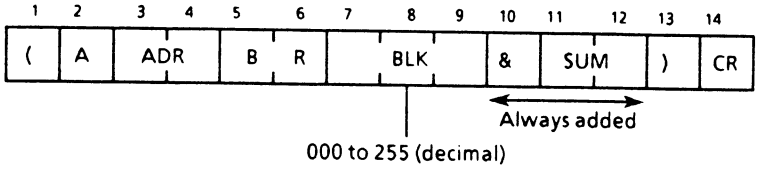


Receive data

Host computer ← Computer link
Appropriate block is detected:



Appropriate block is not detected:



Program block read cannot be executed during RUN. Use it only after the EX100 has been placed in the HALT mode.

Always start program block read beginning with block 000 and read the blocks in order. An error will occur if you attempt to begin with any other block. However, if an error occurs while a block other than 000 is being read, try reading that block again. If that doesn't work, go back to the starting block and restart program block read from the beginning.

After program block read is begun, do not execute another command before the end block is read, otherwise program block read will be aborted. If it aborts, it is necessary to start the operation again at block 000.

To read the register contents along with the program data, execute the device/register-read command immediately after completion of program block-read.

Execution examples

The following execution example is provided to demonstrate use of the "BR" code. The following data are input to computer link Test 0 (described in Appendix A). In this example, BR reads the program four blocks at a time for a total of eight blocks.

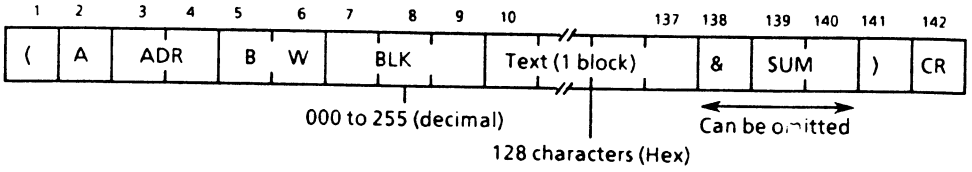
```
SEND DATA    =(A00BR000)···Single command initiates read.
RECEIVE DATA=(A00BR0002810040004000100000000000000000000000
             0080808084000000000000000000000000000000000000
             0000000000000000000000004FFFFFFFFFFFFFFFFFFFF
             FFFFFFF&8B)

SEND DATA    =(A00BR001)
RECEIVE DATA=(A00BR001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
             FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
             FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
             FFFFFFF&14)

SEND DATA    =(A00BR002)
RECEIVE DATA=(A00BR002FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
             FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
             FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF06
             80000300&4A)

SEND DATA    =(A00BR003)
RECEIVE DATA=(A00BR003030003000300030003000300030003000300030
             003000300030003000300030003000300030003000300030
             003000300030003000300030003000300030003000300030
             0030003&76)

SEND DATA    =(A00BR004)
RECEIVE DATA=(A00BR004010438004000700000000000000000000000
             000000000000000000000000000000000000000000000000
             000000000000000000000000000000000000000000000000
             0000000&32)

SEND DATA    =(A00BR005)
RECEIVE DATA=(A00BR005000000000000000000000000000000000000
             000000000000000000000000000000000000000000000000
             000000000000000000000000000000000000000000000000
             0000000&18)
```

```
SEND DATA    =(A00BR006)
RECEIVE DATA=(A00BR00600000000000000000000000000000000000000
              00000000000000000000000000000000000000000000
              00000000000000000000000000000000000000000000
              0000000&19)

SEND DATA    =(A00BR007)
RECEIVE DATA=(A00BR00700000000000000000000000000000000000000
              00000000000000000000000000000000000000000000
              00000000000000000000000000000000000000000000
              0000000&1A)

SEND DATA    =(A00BR008)
RECEIVE DATA=(A00BR008&1B)
```

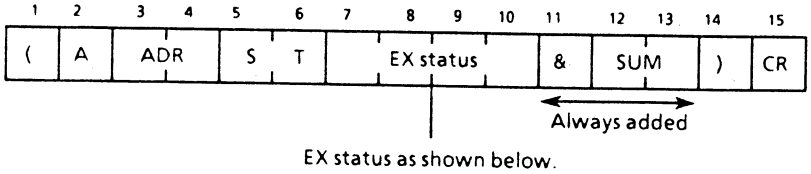
BW
Program block-write

The program block-write (BW) command is used to write a program block-by-block.

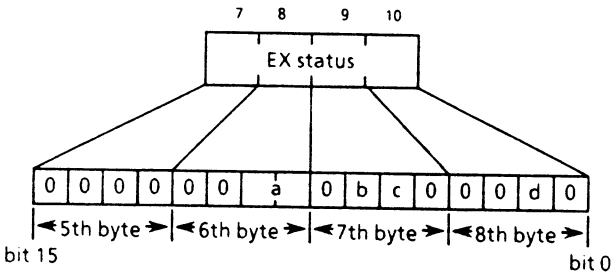
Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



EX status



a = EX mode
01 (1): HALT
10 (2): RUN/RUN-F
11 (3): ERROR

b = 1: TOSLINE error
c = 1: Computer link error
d = 1: Request for diagnostic display

CE or EE is returned when a transmission error occurs.

NOTE The format for the BW command is the same as that used for the EX status read (ST) command.

Program block-write must be started in block 000, with the blocks written in order, or an error will occur. However, if an error occurs while a block other than 000 is being written, try writing that block again. If that doesn't work, go back to the starting block and restart program block-write from the beginning.

After program block-write is started, do not execute another command before the ending block is written because it will cause program block-write to abort. If it does abort, it is necessary to start the operation again at block 000.

To write the register contents along with program data, execute the device/register-write command immediately after completion of program block-write.

Execution example

The following execution example is provided to demonstrate use of the "BW" code. The following data are input to computer link Test 0 (described in Appendix A). In this example, BW writes the program four blocks at a time for a total of eight blocks.

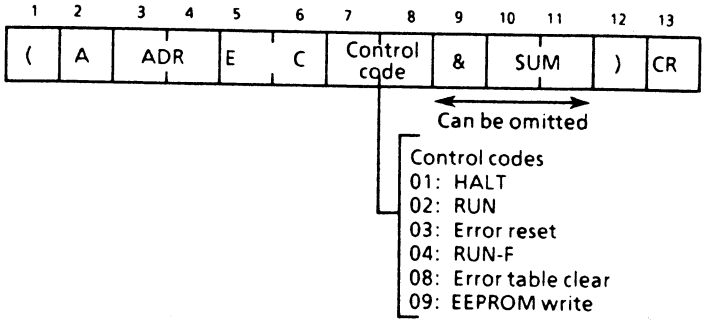
```
SEND DATA    =(A00BW00028100400040001000000000000000000000
               0808080840000000000000000000000000000000000000
               00000000000000000000004FFFFFFFFFFFFFFFFFFF
               FFFFFFFFFF)
RECEIVE DATA=(A00ST0100&57)

SEND DATA    =(A00BW001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
               FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
               FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
               FFFFFFFFFF)
RECEIVE DATA=(A00ST0100&57)

SEND DATA    =(A00BW002FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
               FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
               FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
               0680000300)
RECEIVE DATA=(A00ST0100&57)

SEND DATA    (A00BW00303000300030003000300030003000300030003000300
               03000300030003000300030003000300030003000300030003000300
               03000300030003000300030003000300030003000300030003000300
               030003000)
RECEIVE DATA=(A00ST0100&57)

SEND DATA    =(A00BW00401043800400070000000000000000000000
               00000000000000000000000000000000000000000000000000
               00000000000000000000000000000000000000000000000000
               0000000000)
RECEIVE DATA=(A00ST0100&57)

SEND DATA    =(A00BW00500000000000000000000000000000000000000
               00000000000000000000000000000000000000000000000000
               00000000000000000000000000000000000000000000000000
               0000000000)
RECEIVE DATA=(A00ST0100&57)
```

```
SEND DATA    =(A00BW006000000000000000000000000000000000000
              000000000000000000000000000000000000000000000
              000000000000000000000000000000000000000000000
              0000000000)
RECEIVE DATA=(A00ST0100&57)

SEND DATA    =(A00BW007000000000000000000000000000000000000
              000000000000000000000000000000000000000000000
              000000000000000000000000000000000000000000000
              0000000000)
RECEIVE DATA=(A00ST0100&57)
```

EC
EX control

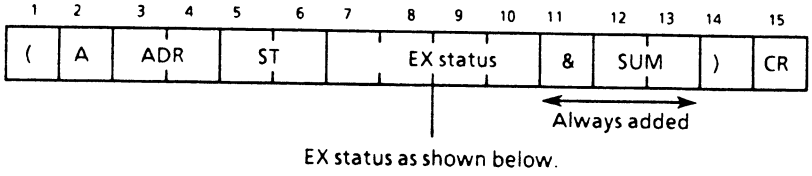
The computer link's EX control (EC) command is used to control the status of the EX100 programmable controller.

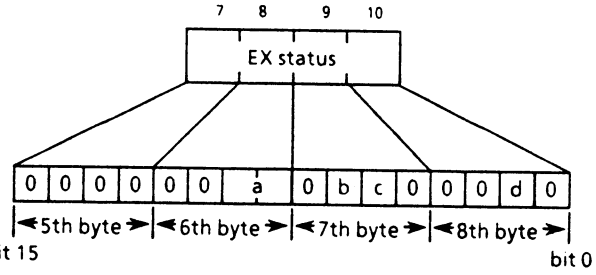
Transmit data Host computer → Computer link



Receive data Host computer ← Computer link

EX control (EC) is used to control the EX100.





a = EX mode
01 (1): HALT
10 (2): RUN/RUN-F
11 (3): ERROR

b = 1: TOSLINE error
c = 1: Computer link error
d = 1: Request for diagnostic display

CE or EE is returned when a transmission error occurs.

NOTE The format for the EX control (EC) command is the same as that used for the EX status-read (ST) command.

If the EX100's RUN/HALT switch is in the HALT position when the RUN or RUN-F command is sent, a mode error will occur.

If the HALT command is sent while the EX100 is in HALT, or the RUN or RUN-F command is sent during RUN, a mode error will occur. (In the latter case, execution will continue.)

If the EX100 is in ERROR status, it will be placed in HALT status by executing ERROR RESET. If this is executed during a non-ERROR status, a mode error will occur.

If ERROR TABLE CLEAR is executed during HALT status, the error history will be cleared. (ERROR TABLE CLEAR cannot be executed during RUN status.)

Execution examples

The following three execution examples are provide ' to demonstrate the use of the "EC" code. The following data can be input to computer link Test 0 (described in Appendix A).

HALT to RUN

```
INPUT DATA  =(A00EC02)
SEND DATA   =(A00EC02&D9)
RECEIVE DATA=(A00ST0102&59)···Normal termination
```

(Actual status is RUN, although status indicated in response is HALT. This can be verified by executing a status-read as follows.)

```
INPUT DATA  =(A00ST)
SEND DATA   =(A00ST&96)
RECEIVE DATA=(A00ST0202&5A)···Verifies RUN status
```

RUN to HALT

```
INPUT DATA  =(A00EC01)
SEND DATA   =(A00EC01&D8)
RECEIVE DATA=(A00ST0202&5A)···Normal termination
```

(Actual status is HALT, although the status indicated in the response is RUN. This can be verified by executing a status read as follows.)

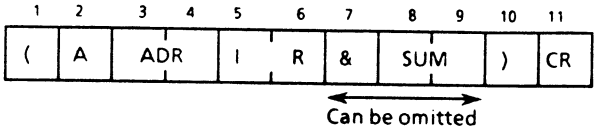
Error reset/Error table clear

```
INPUT DATA  =(A00ST)
SEND DATA   =(A00ST&96)
RECEIVE DATA=(A00ST0102&59)···Verifies HALT status

INPUT DATA  =(A00ST)
SEND DATA   =(A00ST&96)
RECEIVE DATA=(A00ST0300&59)···Hardware error

INPUT DATA  =(A00ER)
SEND DATA   =(A00ER&86)
RECEIVE DATA=(A00ER27&EF)···Read error status
                                (I/O mismatch)

INPUT DATA  =(A00EC03)
SEND DATA   =(A00EC03&DA)
RECEIVE DATA=(A00ST0100&57)···Error reset

INPUT DATA  =(A00EC08)
SEND DATA   =(A00EC08&DF)
RECEIVE DATA=(A00ST0100&57)···Error table clear
```

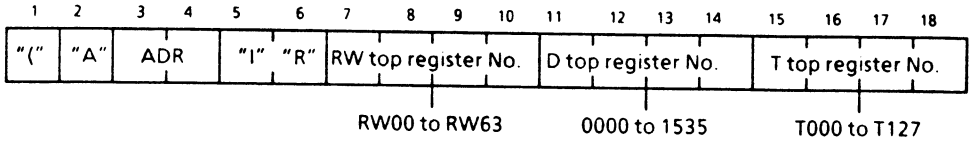
IR
Power-failure storage area read

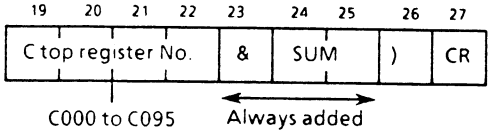
The power-failure storage area read (IR) command is used to read the first register of the power-failure storage (retention) area.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link





CE or EE is returned when a transmission error occurs.

The number of the first register that will be retained during a power failure is indicated for each of the four types of registers (RW, D, T, and C). The registers to be retained are those indicated, plus all subsequent registers of the same type. Because D registers can use up to four digits, the D indication is omitted in the response.

After the power supply has been turned off, the starting register of the register area that will retain the previous status can be checked by executing this command.

Execution examples Three execution examples are provided to demonstrate use of the "IR" code. The following data can be input to computer link Test 0 (described in Appendix A).

No power-failure area

The power-failure storage area has not been specified (no memory will be retained during a power failure).

```
INPUT DATA    =(A00IR)
SEND DATA     =(A00IR&8A)
RECEIVE DATA =(A00IRRW_______T____C____&4A)
```

···Blank codes are returned.

All registers are indicated as retained memory area

```
INPUT DATA    =(A00IR)
SEND DATA     =(A00IR&8A)
RECEIVE DATA =(A00IRRW000000T000C000&0A)
```

RW41 and D1472 indicated as first retained registers

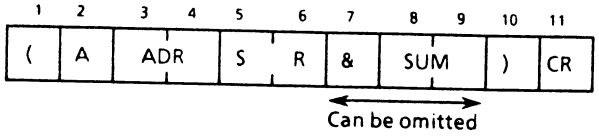
```
INPUT DATA    =(A00IR)
SEND DATA     =(A00IR&8A)
RECEIVE DATA =(A00IRRW411472T____C____&BD)
```

T and C are blank, indicating that none of these registers are retained.

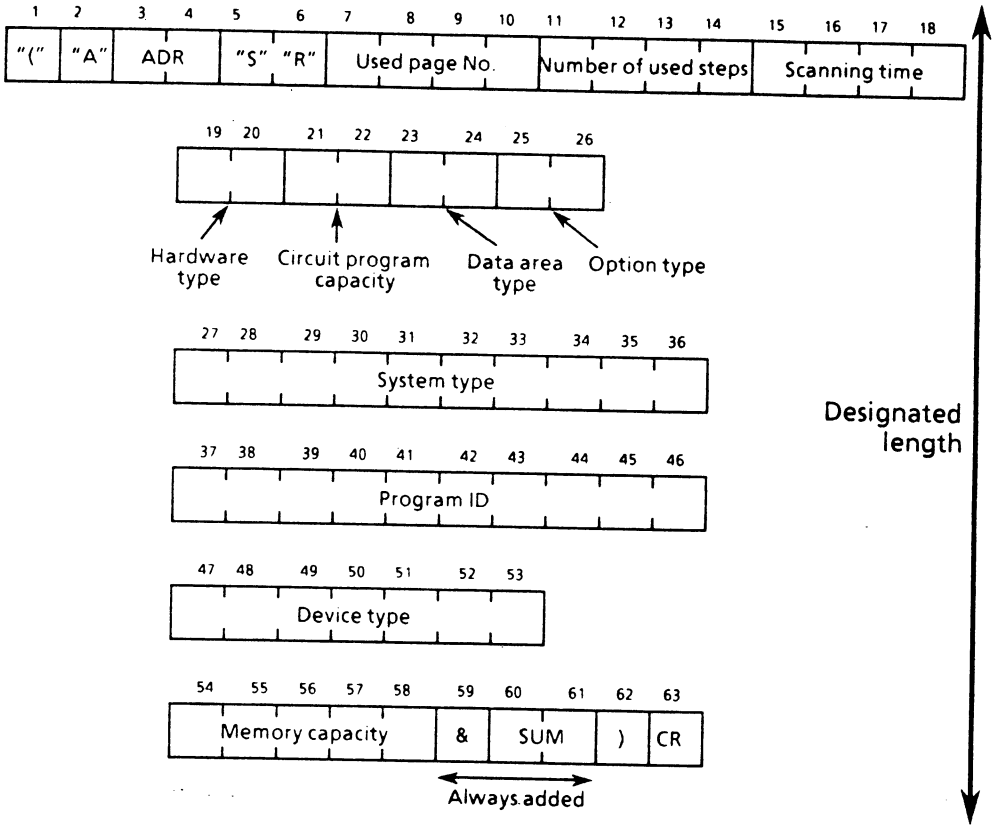
SR System parameter read

The system parameter read (SR) command is used to read EX100 system information.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



Format	Contents	Explanation
□ □ □ □ 4 bytes 0 0 0 0 ≀ 0 9 9 9	Number of last page used	Last page used
□ □ □ □ 4 bytes 0 0 0 0 ≀ 9 9 9 9	Number of steps used	Steps of memory used

Format	Contents	Explanation
□ □ □ □ 4 bytes 0 0 0 0 ↕ 9 9 9 9	Scan time	Program scan time
□ □ 2 bytes 0 1	Hardware type	See NOTE below
□ □ 2 bytes 0 0 ↕ 9 9	Capacity of circuit program	Program area memory capacity (in 1-K byte increments)
□ □ 2 bytes 0 0 0 1	Data area type	Data register range 00·····D0000 to D0511 01·····D0000 to D1535
□ □ 2 bytes 0 0 0 1	Option type	See NOTE below
□□□□□□□□□□ 10 ASCII code characters	System type	EX version number
□□□□□□□□□□ 10 ASCII code characters	Program ID	Registered name of program (program name)
□□□□□□□ 7 ASCII code characters	Device type	EX device type
□□□□□ 5 ASCII code characters	Memory capacity	Maximum memory available

NOTE Each type of EX programmable controller is represented by a combination of its hardware and option types.

EX model	Hardware type	Option type
EX100	01	01
EX250	00	00
EX500	01	00

Execution example

The following execution example is provided to demonstrate use of the "SR" code. The following data can be input to computer link Test 0 (described in Appendix A).

```
INPUT DATA  =(A00SR)
SEND DATA   =(A00SR&94)
RECEIVE DATA=(A00SR 0088 3954 0076 01 04 01 01
             EX-V2.0__________EX100__4.0K_&F2)
```

Receive data consists of the information listed above.

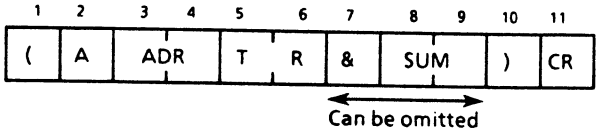
◦Number of pages used	88 pages
◦Number of steps used	3954 steps
◦Scan time	76ms
◦Head type	01 ←
◦Circuit program capacity	4k steps
◦Data area type	01 (D0000 ~ D1535)
◦Option type	01 ←
◦System type	EX-V2.0
◦Program ID	No ID
◦Model	EX100
◦Memory capacity	4.0k

EX100

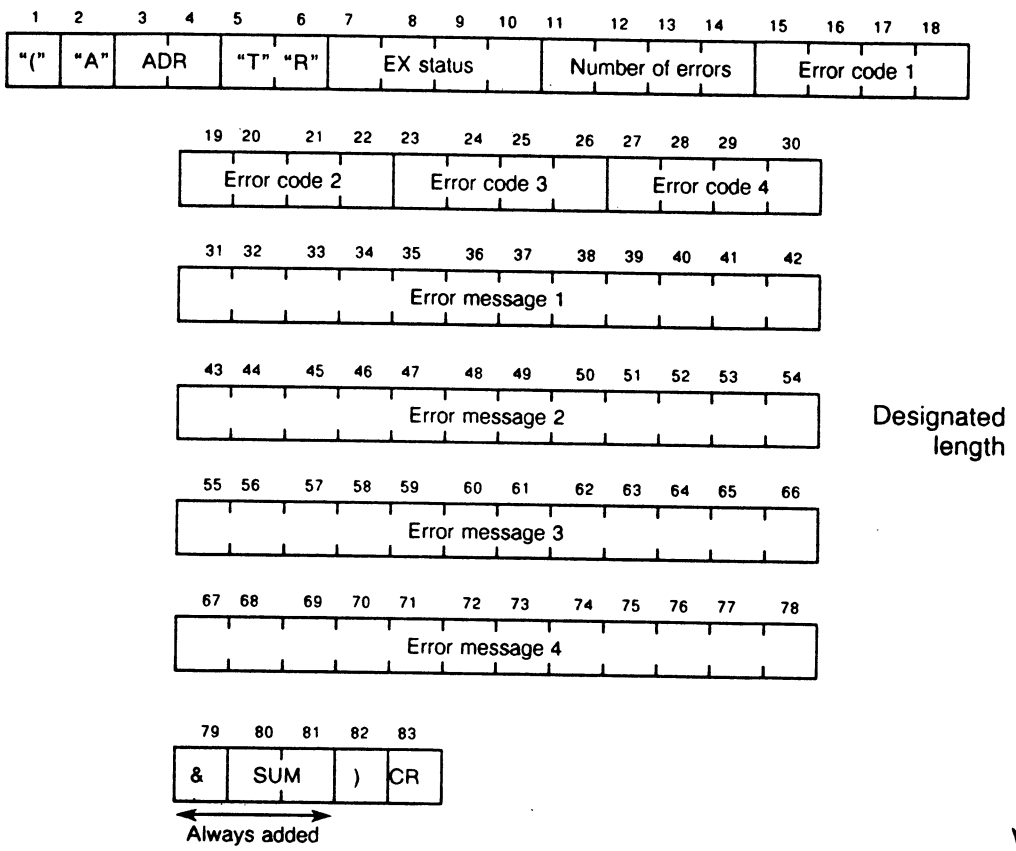
TR
Diagnostic table read

The diagnostic table read (TR) command is used to read the EX100 unit's diagnostic table.

Transmit data Host computer → Computer link



Receive data Host computer ← Computer link



CE or EE is returned when a transmission error occurs.

When the failure condition input is turned ON by a diagnostic display instruction (DDSP or DDSM) in the program, the diagnostic display request bit in the EX status is set to 1. When the diagnostic table is read at this time, the error code and message defined by the user are displayed.

When displaying the diagnostic display in the host computer using this function, it is necessary to execute periodically the EX status-read command and check the diagnostic display request bit. When this bit is found to be 1, execute the diagnostic table-read command to read the message. The diagnostic display request bit is reset by executing the diagnostic table-read command, but the error codes and messages in the table are not altered.

The diagnostic display request bit is reset when the programmer reads the diagnostic information.

Execution examples

The following two execution examples are provided to demonstrate use of the "SR" code. The following data can be input to computer link Test 0 (described in Appendix A).

No user-defined error

```
INPUT DATA  =(A00TR)
SEND DATA   =(A00TR&95)
RECEIVE DATA=(A00TR 0100 0000 0000 0000 0000 0000 ____
                ____________________________________
                ________________ &B8)
```

User-defined error

```
INPUT DATA  =(A00TR)
SEND DATA   =(A00TR&95)
RECEIVE DATA=(A00TR0200 0004 0004 0003 0002 0001
              987654321098 LMNOPQRSTUVW
              ABCDEFGIHKLM0 12345678901 &34)
```

NOTE

- Detailed information of both receive data is provided on the previous page.
- Error codes with up to four numeric characters and error messages with up to 12 alphanumeric characters can be defined by the user to represent special error statuses.

Computer link tests

Five test programs are provided to test communications between a host computer and the EX100 programmable controller.

Test 0 **Loopback test:** Confirms data reception. Test 0 is also used for maintenance purposes.

Test 1 **Send/receive test:** Sends the input text, and displays the received text.

Test 2 **Block-read test:** Saves the block to the floppy disk (FD).

Test 3 **Block-write test:** Loads the block from the FD. (The write-enable switch must be ON, and the program-block write switch must be set to enable.)

Test 4 **Block-compare** (verify) **test:** Compares the block in the host computer with the one in the EX100 unit.

- The five test programs have been written in IBM-PC™ BASIC language.
- The set values in the test program are 9600 baud, parity disable, 8-bit length, and 1 stop bit.

Test 0 Loopback test

Computer link Test 0 is used to check send/receive data as a loopback test. Test 0 can be used for start-up, and it is also useful for maintenance purposes.

Programming Test 0

Input the following program into the host computer. This program is written in IBM-PC™ BASIC language.

```
100 '************************************************
110 '*          COMPUTER LINK TEST 0                *
120 '*          (SEND AND RECEIVE TEST)             *
130 '*        1989-4-1   FILENAME=TEST0             *
140 '************************************************
150 '***************** INITIALIZE *********************
160 SCREEN 0:KEY OFF:CLS
170 DEFINT A-Z
180 '************* COMMUNICATION INITIALIZE **********
190 OPEN "COM1:9600,N,8,1" AS #1
200 '***************** KEY INPUT **********************
210 TEXT$="":DAT$=""
220 PRINT "INPUT DATA   =";
230 IF DAT$=CHR$(&HD) THEN 290
240   DAT$=INKEY$:PRINT DAT$;
250   TEXT$=TEXT$+DAT$
260 GOTO 230
270 '
280 '****************** DATA SEND ********************
290 PRINT "SEND DATA    = ";
300 PRINT TEXT$;
310 PRINT #1,TEXT$;
320 '***************** RECEIVE DATA *****************
330 RD$="":A$="":B$="":TIME$="00:00:00"
340 PRINT "RECEIVE DATA = ";
350 IF B$=CHR$(&HD) THEN 430
360   IF TIME$="00:00:03" THEN PRINT "TIMEOUT ERROR":END
370   IF LOC(1)=0 THEN 420
380     A$=INPUT$(LOC(1),#1)
390     RD$=RD$+A$
400     B$=RIGHT$(A$,1)
410   GOTO 370
420 GOTO 350
430 PRINT RD$;
440 PRINT
450 GOTO 210
```

Running Test 0

Run the program. Next, input (A00TS0123456789) CR.

- If (A00TS0123456789&32) CR is returned, operation is functioning normally.
- If (A00EE—) or (A00CE—) is returned, an error has occurred. In this case, check whether the settings for station number, baud rate, parity, bit length, stop bit, etc., match.
- If "TIMEOUT ERROR" is displayed on the host computer's screen, the response from the computer link has not been returned. In this case, check whether the cable connections are correct, and whether the power supply to the EX100 unit has been turned on. Also check whether the settings for baud rate, parity, bit length, stop bit etc., match.
- If "DEVICE I/O ERROR" is displayed on the programmer's screen, a transmission error has occurred. In this case, check whether the terminal connections for DSR, DTR, RTS, CTS, CD, and SG have been properly wired. Also check whether the settings are correct for baud rate, parity, bit length, stop bit, etc.
- When multiple computer link modules are used, make sure that each has been assigned a different station number.

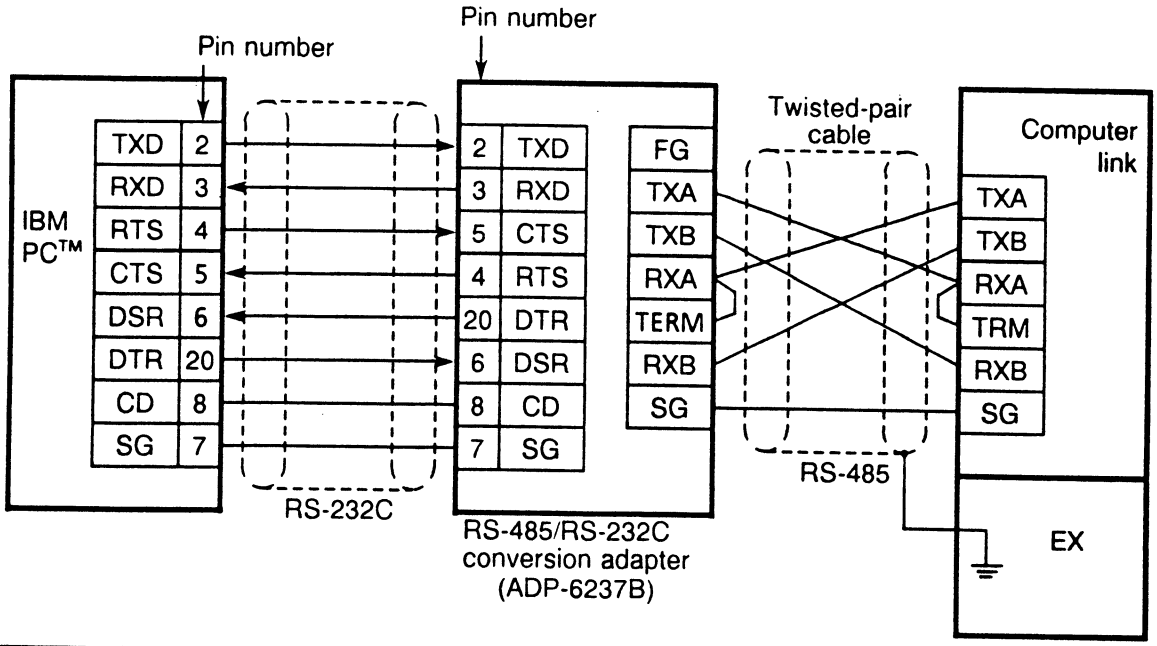
Execution example

Test of station No. 0

```
INPUT DATA  =(A00TS01234)···For normal conditions
SEND DATA   =(A00TS01234)
RECEIVE DATA=(A00TS01234&60)
```

Connections of Test 0

If the computer link malfunctions, execute the loopback test using the following terminal connections.



Test 1 Send/receive

```
100 '************************************************
110 '*          COMPUTER LINK TEST 1                *
120 '*          (SEND AND RECEIVE TEST)             *
130 '*        1989-4-1   FILENAME=TEST1             *
140 '************************************************
150 '***************** INITIALIZE *********************
160 SCREEN 0:KEY OFF:CLS
170 DEFINT A-Z
180 ON ERROR GOTO 700
190 '************* COMMUNICATION INITIALIZE ******,****
200 OPEN "COM1:9600,N,8,1" AS #1
210 '***************** KEY INPUT **********************
220 TEXT$="":DAT$="":PRINT "INPUT DATA   =";
230 WHILE DAT$<>CHR$(&HD)
240   DAT$=INKEY$:PRINT DAT$;
250   TEXT$=TEXT$+DAT$
260 WEND
270 '*************** CHECK SUM MAKE ******************
280 SUM=0:I=1
290 L=LEN(TEXT$):IF L<=5 THEN PRINT
    "---- INPUT ERROR ----":GOTO 220
300 SD$=LEFT$(TEXT$,(L-2)):SD$=SD$+"&"
310 WHILE I<=L-1
320   A$=MID$(SD$,I,1)
330   SUM=SUM+ASC(A$)
340   I=I+1
350 WEND
360 SUM$=HEX$(SUM)
370 SUM$=RIGHT$(("O"+SUM$),2)
380 SD$=SD$+SUM$+")"+CHR$(&HD)
390 '****************** DATA SEND *********************
400 PRINT "SEND DATA    = ";
410 PRINT SD$;
420 PRINT #1,SD$;
430 '***************** RECEIVE DATA *******************
440 RD$="":A$="":B$="":TIME$="00:00:00"
450 PRINT "RECEIVE DATA = ";
460 WHILE B$<> CHR$(&HD)
470   IF TIME$="00:00:03" THEN PRINT "TIMEOUT ERROR":END
480   WHILE LOC(1)<>0
490     A$=INPUT$(LOC(1),#1)
500     RD$=RD$+A$
510     B$=RIGHT$(A$,1)
```

(Next page)

```
520   WEND
530 WEND
540 PRINT RD$;
550 PRINT
560 '***************CHECK SUM CHECK ****************
570 SUM=0:I=1:A$=""
580 L=LEN (RD$)
590 WHILE A$<>"&"
600   A$=MID$(RD$,I,1)
610   SUM=SUM+ASC(A$)
620   I=I+1
630 WEND
640 SUM$=HEX(SUM)
650 SUM$=RIGHT$(("O"+SUM$),2)
660 TESTSUM$=LEFT$(RIGHT$(RD$,4),2)
670 IF SUM$<>TESTSUM$ THEN PRINT
    "---- SUM ERROR ----":END
680 GOTO 220
690 '************ ERROR TASK ************************
700 PRINT "ERROR ":CLOSE #1:RESUME 100
```

Test 2 Block read

```
1000 '***********************************
1010 '*     COMPUTER LINK TEST 2        *
1020 '*       (BLOCK READ TEST)         *
1030 '*    1989-4-1  FILENAME=TEST2     *
1040 '***********************************
1050 '
1060 '********* INITIALIZE ************
1070 CLS: DEFINT A-Z
1080 BLK=0:B$="":RD$=""
1090 INPUT "SAVE FILE =  ";NA$
1100 OPEN NA$ FOR OUTPUT AS #2
1110 '
1120 '**** COMMUNICATION INITIALIZE ***
1130 OPEN "COM1:9600,N,8,1" AS #1
1140 '
1150 '******  BLOCK READ START ********
1160 B$="":RD$="":BL=BLK+1000
1170 BLK$=RIGHT$(STR$(BL),3)
1180 SD$="(0BR"+BLK$+")"+CHR$(&HD)
1190 '
1200 '*****     DATA   SEND      ******
1210 COLOR 7,,0
1220 PRINT "SEND DATA    =  ";SD$;
1230 PRINT #1,SD$;
1240 '
1250 '******    DATA RECEIVE     ******
1260 COLOR 7,,0
1270 PRINT "RECEIVE DATA  =   ";
1280 IF RIGHT$(B$,1)=CHR$(&HD)  THEN 1350
1290   IF LOC (1)=0 THEN 1280
1300    B$=INPUT$(LOC(1),#1)
1310    RD$=RD$+B$
1320    GOTO 1280
1330 '
1340 '*******  DATA SAVE **************
1350 COLOR 3,,0:PRINT RD$
1360 IF LEN(RD$)<>140 THEN GOTO 1410
1370 SV$=MID$(RD$,8,128)
1380 WRITE #2,SV$
1390 BLK=BLK+1
1400 GOTO 1160
1410 COLOR 7,,0:CLOSE:PRINT "COMPLETE"
1420 END
```

Test 3
Block write

```
1000 '***********************************
1010 '*      COMPUTER LINK TEST 3        *
1020 '*       (BLOCK WRITE TEST)         *
1030 '*    1989-4-1  FILENAME=TEST3      *
1040 '***********************************
1050 '
1060 '********* INITIALIZE ************
1070 CLS: DEFINT A-Z :B$="":BLK=0
1080 INPUT "LOAD FILE =  ";NA$
1090 OPEN NA$ FOR INPUT AS #2
1100 '**** COMMUNICATION INITIALIZE ***
1110 OPEN "COM1:9600,N,8,1" AS#1
1120 '
1130 '*******   DATA LOAD   ************
1140 PRINT
1150 IF EOF (2)  THEN 1370
1160 INPUT #2,LD$
1170 '
1180 '*********** DATA SEND ************
1190 COLOR 7,,0:BL=BLK+1000:BLK=BLK+1
1200 BLK$=RIGHT$(STR$(BL),3)
1210 SD$="(0BW"+BLK$+LD$+")"+CHR$(&HD)
1220 PRINT "SEND DATA    = ";
1230 PRINT SD$;
1240 PRINT #1,SD$;
1250 '
1260 '********** DATA RECEIVE **********
1270 B$="":COLOR 3,,0
1280 PRINT "RECEIVE DATA  = ";
1290 B$=RIGHT$(B$,1)
1300 IF B$=CHR$(&HD) THEN 1140
1310  IF LOC(1)=0 THEN 1290
1320  B$=INPUT$(LOC(1),#1)
1330  PRINT B$;
1340  GOTO 1290
1350 '
1360 '*********** COMPLETE *************
1370 A$="(0ST)"+CHR$(&HD):PRINT #1,A$
1380 CLOSE
1390 COLOR 7,,0:PRINT "---- COMPLETE ----"
1400 END
```

Test 4
Block compare

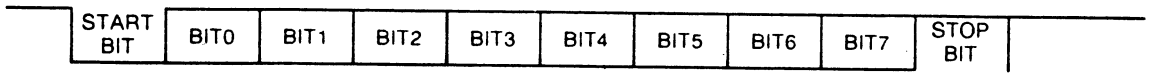
```
1000 '***********************************
1010 '*     COMPUTER LINK TEST 4        *
1020 '*       (BLOCK VERIFY TEST)       *
1030 '*    1989-4-1  FILENAME=TEST4     *
1040 '***********************************
1050 '
1060 '********** INITIALIZE ************
1070 CLS: DEFINT A-Z
1080 BLK=0:B$="":RD$=""
1090 INPUT "VERIFY FILE =  ";NA$
1100 OPEN NA$ FOR INPUT AS #2
1110 '
1120 '**** COMMUNICATION INITIALIZE ***
1130 OPEN "COM1:9600,N,8,1" AS #1
1140 '
1150 '******  BLOCK READ START ********
1160 B$="":RD$="":BL=BLK+1000
1170 BLK$=RIGHT$(STR$(BL),3)
1180 SD$="(0BR"+BLK$+")"+CHR$(&HD)
1190 '
1200 '*****     DATA   SEND      ******
1210 COLOR 7,,0
1220 PRINT "SEND DATA    =  ";SD$;
1230 PRINT #1,SD$;
1240 '
1250 '******    DATA RECEIVE     ******
1260 COLOR 7,,0
1270 PRINT "RECEIVE DATA  =   ";
1280 IF RIGHT$(B$,1)=CHR$(&HD)  THEN 1350
1290   IF LOC (1)=0 THEN 1280
1300    B$=INPUT$(LOC(1),#1)
1310    RD$=RD$+B$
1320    GOTO 1280
1330 '
1340 '*******  DATA VERIFY **************
1350 COLOR 3,,0:PRINT RD$
1360 IF LEN(RD$)<>140 THEN GOTO 1410
1370 SV$=MID$(RD$,8,128)
1380 INPUT #2,SV$
1390 BLK=BLK+1
1400 GOTO 1160
1410 COLOR 7,,0:CLOSE:PRINT "COMPLETE"
1420 END
```

Transmission specifications

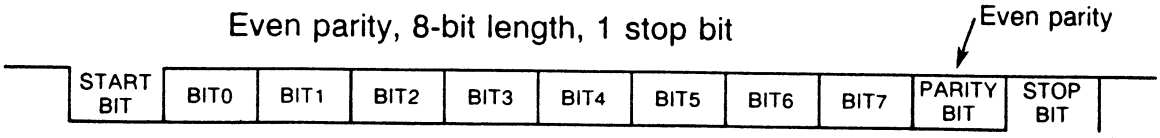
Item	Specifications
Interface	Conforms to RS-485 (RS-232C is used with RS-485/RS-232C converter)
Transmission mode	Half-duplex, 4-wire system
Synchronizing	System start-stop (asynchronous)
Transmission line configuration	Multidrop data highway
Transmission speed	1,200/2,400/4,800/9,600 bps (switch settings)
Transmission distance	1 km (3,281 ft) maximum
Transmission code	8-bit ASCII
Data length	8 bits (fixed)
Stop bit	1 stop bit (fixed)
Parity	Even/odd/no parity (switch setting)
Number of attached stations	16 stations maximum
Error check	Parity, checksum
Number of channels	1 channel

Frame format

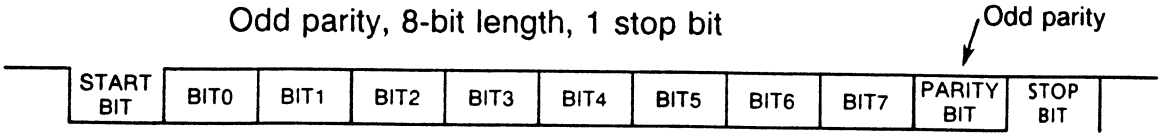
No parity, 8-bit length, 1 stop bit



Even parity, 8-bit length, 1 stop bit



Odd parity, 8-bit length, 1 stop bit



Appendix C: RS-485/RS-232C converter

Specifications

The specifications of the RS-232C adapter (RS-485/RS-232C converter) are as follows:

Model: EX25PADP6237B (code name ADP-6237B)

General specifications

Item		Specifications
Supply voltage		85 to 132 Vac / 170 to 264 Vac (50/60 Hz)
Power consumption		15 VA or less
Environmental conditions	Operating temperature	0° to 55°C (32° to 131°F)
	Storage temperature	–20° to 75°C (– 4° to 167°F)
Humidity		20% to 90% RH (no condensation)
Dielectric strength		1500 Vac for 1 minute

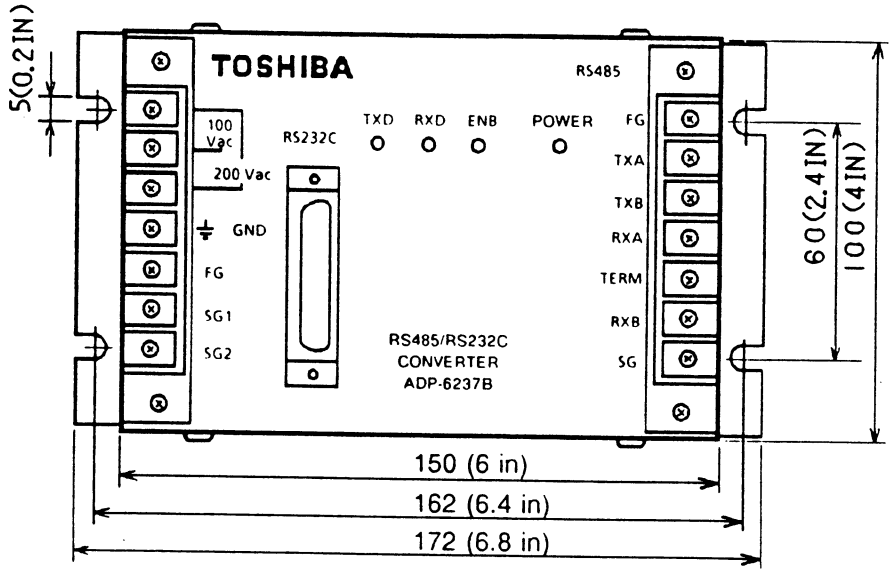
Transmission specifications

	Item	Specifications
RS-232C	Interface	Conforms to RS-232C
	Transmission distance	15 meters (45 ft) maximum
RS-485	Interface	Conforms to RS-485
	Transmission distance	1 km (3,281 ft) maximum
	Number of stations	Up to 32 stations (see NOTE)

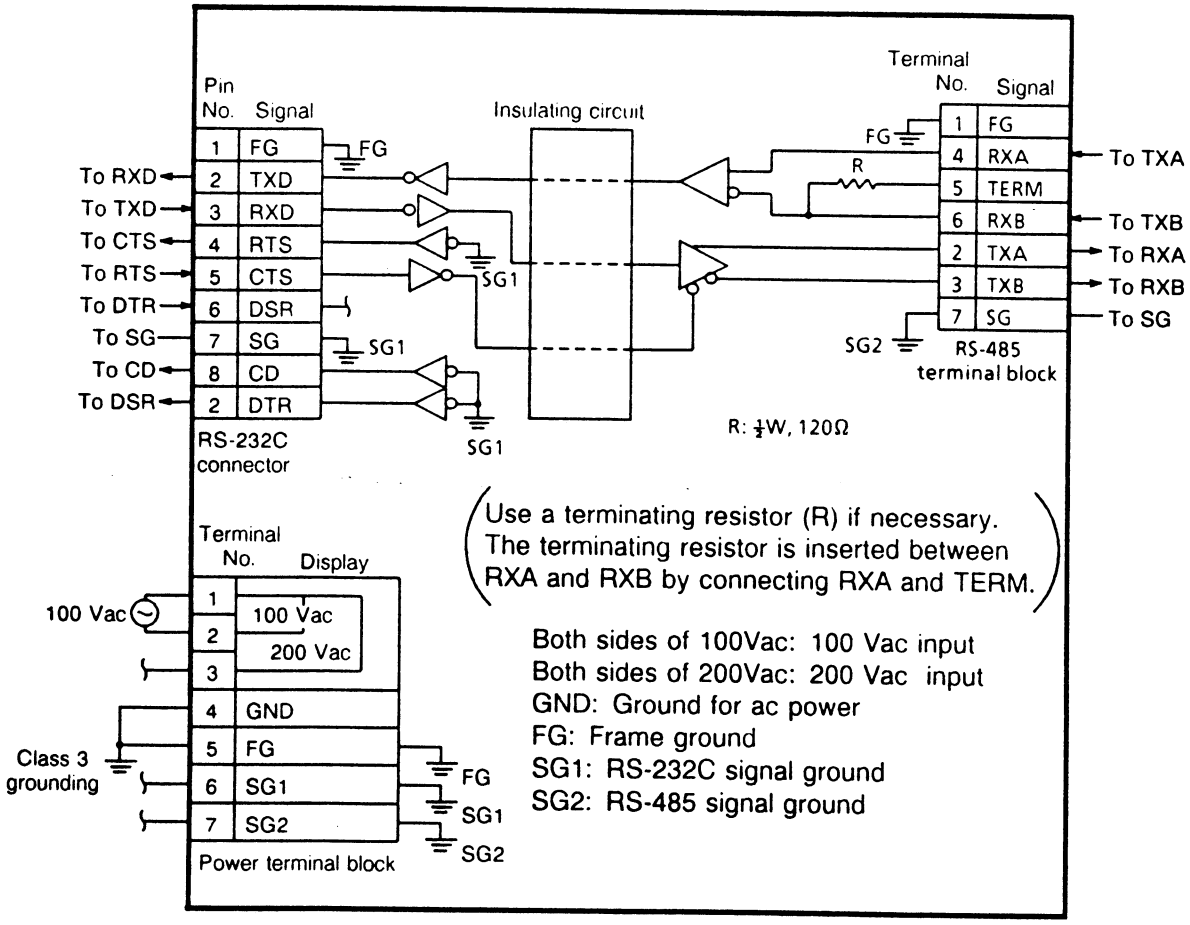
NOTE When only EX100 programmable controllers are used, the maximum number of connectable stations is limited to 16 because of the 0-through-F limitation of the EX100's station selection switch (see page 7).

External view

Units: mm (in)

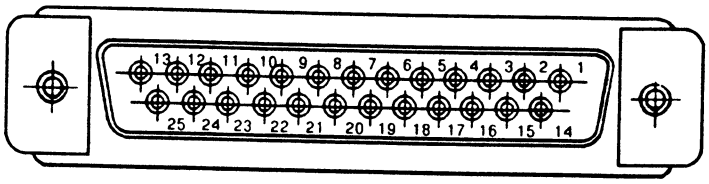


Circuit configuration



RS-232C connector

RS-232C connector pin allocation



NOTE The RS-232C connector on the adapter is female. Use a male connector on the cable.

RS-232C interface pin functions

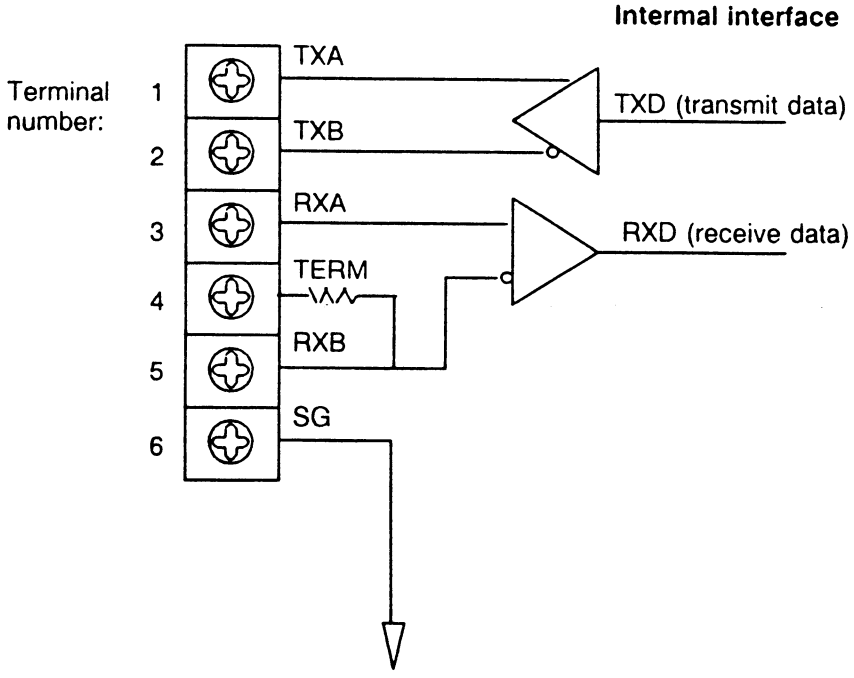
Pin No.	Signal	CCITT* equivalent	Interchange circuit	Description
1	FG	101	AA	Protective ground
2	TXD (SD)	103	BA	Transmitted data
3	RXD (RD)	104	BB	Received data
4	RTS (RS)	105	CA	Request to send
5	CTS (CS)	106	CB	Clear to send
6	DSR (DR)	107	CC	Data set ready
7	SG	102	AB	Signal ground
8	CD (DCD)	109	CF	Receive line signal detector
20	DTR (ER)	108.2	CD	Data terminal ready

* Consultative Committee in International Telegraphy and Telephony

RS-485 connector signal names

Signal	Name	Specifications
TXA	Transmit data A	In-phase signal of transmit data (output)
TXB	Transmit data B	Reverse phase of transmit data (output)
RXA	Receive data A	In-phase signal of transmit data (input)
RXB	Receive data B	Reverse phase of transmit data (input)
SG	Signal ground	Ground of signal line
FG	Frame ground	Ground of shielding wire, etc.

RS-485 internal interface



Appendix E: 8-bit ASCII table

8-bit ASCII code

The left half of the transmission table (below) used by the EX100 programmable controller is the same as the standard American ASCII code. The right side (beginning with hex value 80) is used for special diagram symbols (hex 80 to 9F, and E0 to FF), and for Japanese *katanana* figures (hex A0 to DF).

4 higher-order bits

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E		0	@	P		p				ー	タ	ミ		
1	S_H	D_1	!	1	A	Q	a	q			。	ア	チ	ム		
2	S_X	D_2	"	2	B	R	b	r			「	イ	ツ	メ		
3	E_X	D_3	#	3	C	S	c	s			」	ウ	テ	モ		
4	E_T	D_4	$	4	D	T	d	t			、	エ	ト	ヤ		
5	E_Q	N_K	%	5	E	U	e	u			・	オ	ナ	ユ		
6	A_K	S_N	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	B_L	E_B	'	7	G	W	g	w			ァ	キ	ヌ	ラ		
8	B_S	C_N	(	8	H	X	h	s			ィ	ク	ネ	リ		
9	H_T	E_M	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A	L_F	S_B	*	:	J	Z	j	z			ェ	コ	ハ	レ		
B	H_M	E_C	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C	C_L	→	,	<	L	¥	l	|			ャ	シ	フ	ワ		
D	C_R	←	—	=	M	]	m	}			ュ	ス	ヘ	ン		
E	S_O	↑	.	>	N	^	n	~			ョ	セ	ホ	"		
F	S_I	↓	/	?	O	_	o				ッ	ソ	マ	°		

SOH·····H01

ETX·····H03

CR·····H0D

Hexadecimal/decimal

Hexa-decimal	Decimal
1000	4096
2000	8192
3000	12288
4000	16384
5000	20480
6000	24576
7000	28672
8000	32768
9000	36864
A000	40960
B000	45056
C000	49152
D000	53248
E000	57344
F000	61440

Hexa-decimal	Decimal
100	256
200	512
300	768
400	1024
500	1280
600	1536
700	1792
800	2048
900	2304
A00	2560
B00	2816
C00	3072
D00	3328
E00	3584
F00	3840

Hexa-decimal	Decimal
10	16
20	32
30	48
40	64
50	80
60	96
70	112
80	128
90	144
A0	160
B0	176
C0	192
D0	208
E0	224
F0	240

Hexa-decimal	Decimal
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Hexa-decimal	Decimal
2710	10000
4E20	20000
7530	30000
9C40	40000
C350	50000
EA60	60000
11170	70000
13880	80000
15F90	90000

Hexa-decimal	Decimal
3E8	1000
7D0	2000
BB8	3000
FA0	4000
1388	5000
1770	6000
1B58	7000
1F40	8000
2328	9000

Hexa-decimal	Decimal
64	100
C8	200
12C	300
190	400
1F4	500
258	600
2BC	700
320	800
384	900

Hexa-decimal	Decimal
A	10
14	20
1E	30
28	40
32	50
3C	60
46	70
50	80
5A	90

Troubleshooting

If the computer link module does not work properly, confirm the following items.

- Is the power supply to the EX100 switched on?
- Are all switches (DIP, station selection) set correctly? For new switch settings to take effect, the power supply switch must be cycled OFF and then ON.
- Are the terminal resistors properly installed?
- Are the cables connected correctly?
- Is the ADP-6237B installed for RS-485/RS-232C conversion? (If a converter of another make is used, check for interface problems, such as improper connections.)
- Has a LINK ERROR occurred?
  (Use a programmer (MP100 or GP110/GP110AP1) to confirm whether a LINK ERROR occurred, and to read the status of the EX100 unit. If an error is shown, the computer link may cause a system shutdown.)
- Are the results of the loopback diagnostic test normal?