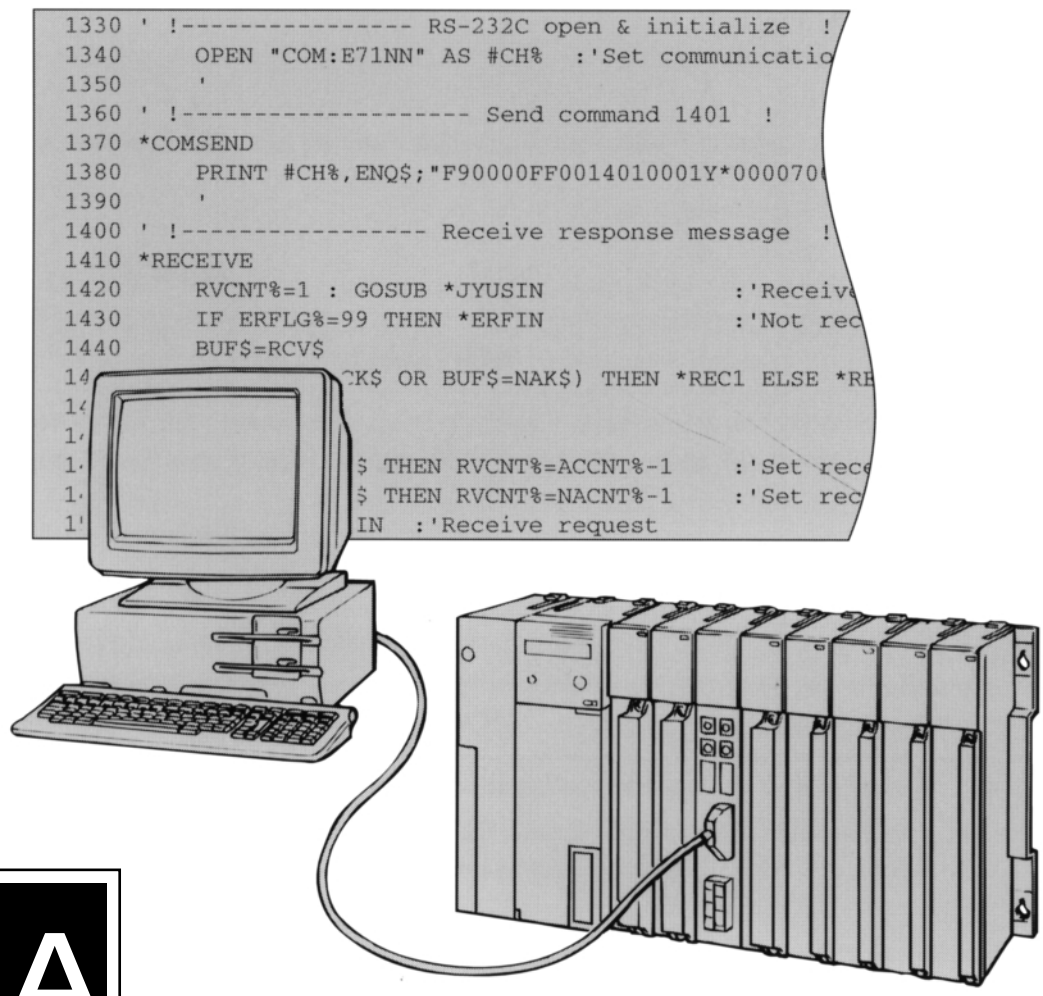
MITSUBISHI

QnA SERIES

技術がつくる高度なふれあい SOCIO-TECH

AJ71QC24 Serial Communications Module

GUIDEBOOK



QnA

MELSEC

Mitsubishi Programmable Logic Controller

● SAFETY PRECAUTIONS ●

(Please read these precautions before using you serial communication module.)

When using the Model AJ71QC24(-R2/R4), thoroughly read this manual and the associated manuals introduced in this manual.

Also pay careful attention to safety and handle the module properly. These precautions apply only to the AJ71QC24)(-R2/R4). Refer to the CPU module user's manual for a description of the PC system safety precautions.

These ● SAFETY PRECAUTIONS ● classify the safety precautions into two categories:"DANGER" and "CAUTION".

⚠ DANGER	Procedures which may lead to a dangerous condition and cause death or serious injury if not carried out properly.
⚠ CAUTION	Procedures which may lead to a dangerous condition and cause superficial to medium injury, or physical damage only, if not carried out properly.

Depending on circumstances, procedures indicated by ⚠ CAUTION may also be linked to serious results.

Store this manual in a safe place so that you can take it out and read it whenever necessary. Always forward it to the end user.

[DESIGN PRECAUTIONS]

⚠ CAUTION

- Do not bundle, on install, the control cables and communication cables with, or near, main circuit and power cables. Keep them at least 100mm away from such cables. Noise may cause erroneous operation.

[INSTALLATION PRECAUTIONS]

⚠ CAUTION

- Use the PC in the environment given in the general specifications section of this manual.
  Using the PC outside the range of the general specifications may result in electric shock, fire, or erroneous operation or may damage or degrade the product.

- Insert the tabs at the bottom of the module into the holes in the base unit before installing the module.
  Improper installation may cause erroneous operation, accidents, or the module to fall out.

[WIRING PRECAUTIONS]

⚠ CAUTION

- Before connecting the cables, check the type of interface to be connected.
  Connection, or erroneous wiring, to the wrong interface may damage the module and external devices.

- When connecting a external device to the AJ71QC24-R4 RS-422 interface, do not connect a device that must receive power from the AJ71QC24-R4.
  The module or external device may be damaged.

- Tighten the terminal screws to the specified torque.
  Loose terminal screws may cause a short circuit or erroneous operation.

- Be sure that cuttings, wire chips, or other foreign matter do not enter the module.
  Foreign matter may start a fire or cause an accident or erroneous operation.

[STARTING AND MAINTENANCE PRECAUTIONS]

⚠ DANGER

- Do not touch live terminals.
  It may cause erroneous operation.

- Turn off the power before cleaning the module or retightening the screws. Doing this work while the power is on may damage the module or cause erroneous operation.

⚠ CAUTION

- Do not disassemble or rebuild the module.
  It may cause accidents, erroneous operation, injury, or fire.

- Turn off the power before mounting and dismounting the module.
  Mounting or dismounting the module while the power is on may damage the module or cause erroneous operation.

[DISPOSAL PRECAUTIONS]

⚠ CAUTION

- When disposing of this product, handle it as industrial waste.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	*Manual Number	Revision
Mar. 1996	IB-66622-A	First edition

INTRODUCTION

Thank you for choosing a Mitsubishi MELSEC-QnA Series General Purpose Programmable Controller.

Before using your new PC, please read this manual thoroughly to gain an understanding of its functions so you can use it properly.

Please forward a copy of this manual to the end user.

TABLE OF CONTENTS

6 BIDIRECTIONAL PROTOCOL COMMUNICATIONS 6- 1~6-21

APPENDIXES APPENDIX- 1~APPENDIX- 5

1 GENERAL DESCRIPTION

This guidebook was written to provide first-time users of the MELSEC-QnA Series PC Serial Communications Module with an understanding of the following.

- What the serial communications module is
- How to use the serial communications module
- General description of communications protocols and data flow.

The manual shown below describes the contents of this guidebook in detail and describes the serial communications module data communications functions not covered in this guidebook.

Use it to gain a deeper understanding of the serial communications module.

<Detailed description manual>

- AJ71QC24(-R2/R4) Serial Communications Module User's Manual IB66612

When using this guidebook to check the data communications functions of each protocol of the serial communications module, see the following sections.

1 Checking the dedicated protocols data communications functions

(a) Selecting the system configuration ..... Section 3.1.1
(b) External wiring ..... Section 3.3
(c) Hardware setting ..... Section 4.2
(d) Data communications in ASCII mode
① Reading PC CPU device memory ..... Sections 4.5, 4.6
② Writing PC CPU device memory ..... Sections 4.7, 4.8
(e) On-demand function ..... Section 4.9

2 Checking the non procedure protocol data communications functions

(a) Selecting the system configuration ..... Section 3.1.1
(b) External wiring ..... Section 3.3
(c) Hardware setting ..... Section 5.2
(d) Data communications
① Transmitting data from PC CPU ..... Section 5.5
② Receiving data from an external device ..... Section 5.6
(e) Data communications using user frames ..... Section 5.7
(f) Data communications using ASCII↔BIN conversion ..... Section 5.8

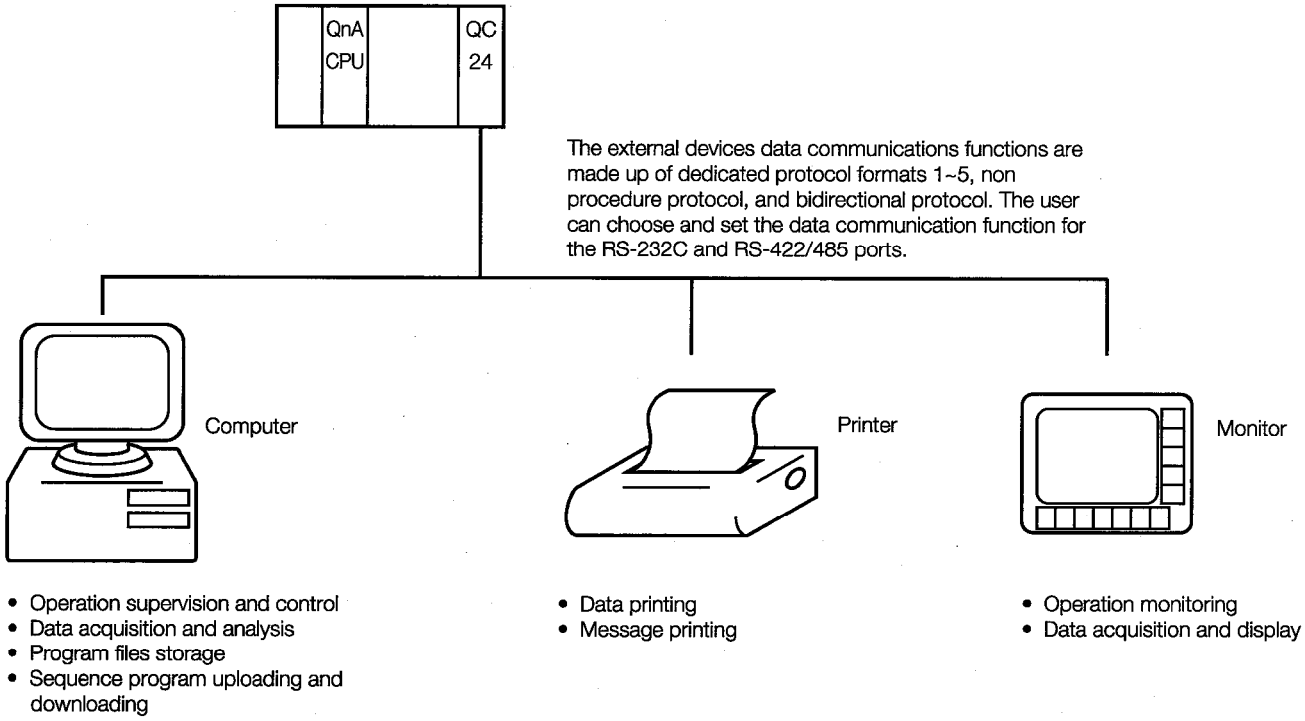
3 Checking the bidirectional protocol data communications functions

(a) Selecting the system configuration ..... Section 3.1.1
(b) External wiring ..... Section 3.3
(c) Hardware setting ..... Section 6.2
(d) Data communications
① Transmitting data from PC CPU ..... Section 6.5
② Receiving data from an external device ..... Section 6.6

1.1 What is The Serial Communications Module?

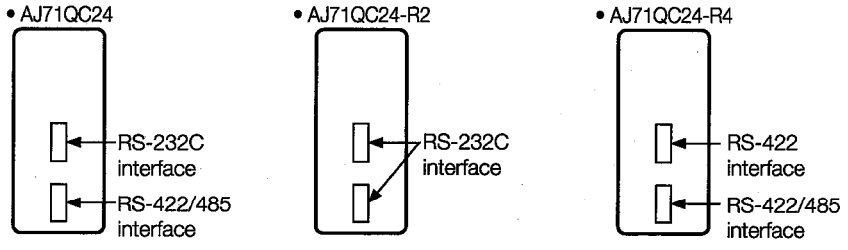
The AJ71QC24 Serial Communications Module (hereafter abbreviated QC24) uses an RS-232C and RS-422 (or RS-485) interface to send and receive data between external devices (computer, printer, monitor, sensor, measuring instrument, etc.) and a PC CPU.

The following shows the QC24 and external devices connections.



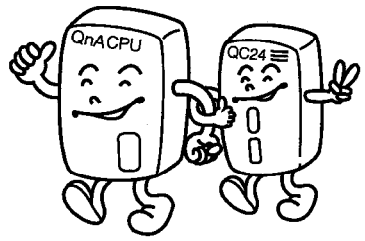
QC24 Series

Three models of QC24 are available. The user can choose the model matched to the external devices to be connected.



1.2 Features of the QC24

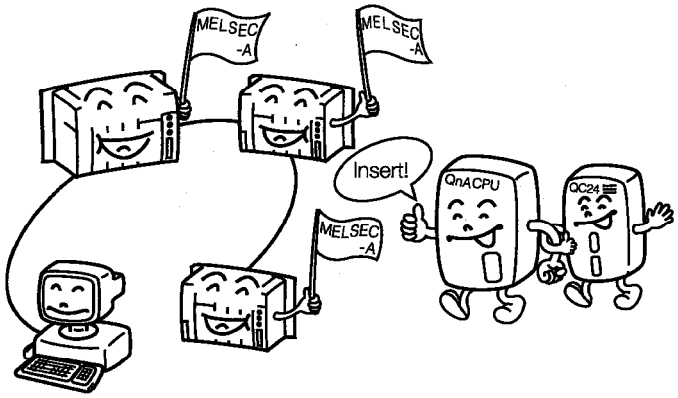
1 QnACPU dedicated special function module



The QC24 is a QnACPU dedicated special function module.

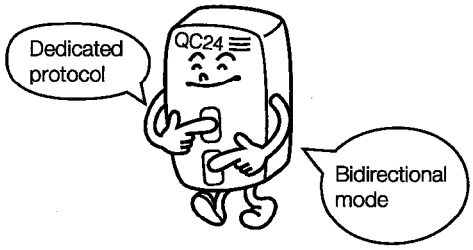
It allows data communications that use the functions of the QnACPU more effectively.

2 Interchangeable with computer link module



The QC24 has the same functions as the MELSEC-A Series Computer Link Module. Therefore, it can be installed in a system that uses a multidrop link to connect external devices and the computer link module.

3 Two interfaces can be independently set *



The communications protocol and transmission specifications of the two QC24 interfaces can be independently set.

Computer Link Module

	CH1	CH2
Mode	Dedicated protocol (format 2)	Same as CH1.
Data length	7 bits	
Transmission rate	4800BPS	
Checksum	No	

Only CH1 or CH2 can be set

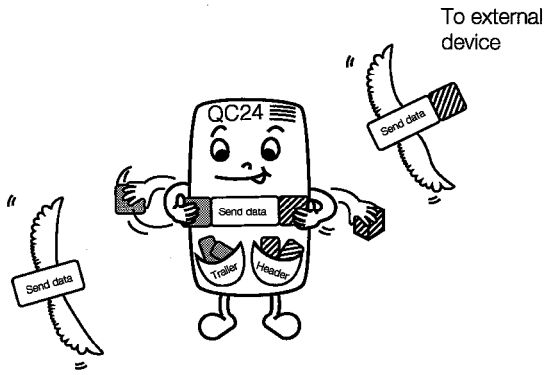
⇓

QC24

	CH1	CH2
Mode	Dedicated protocol (format 2)	Bidirectional protocol
Data length	7 bits	8 bits
Transmission rate	4800BPS	9600BPS
Checksum	No	Yes

CH1 and CH 2 can both be set.

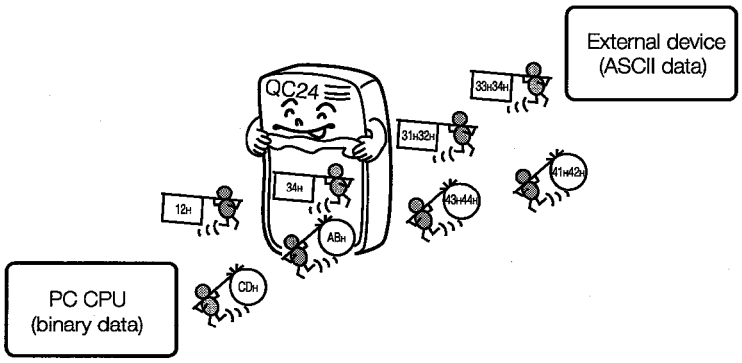
4 Data can be transmitted in the message format matched to the specifications of the external device*



The QC24 can register message frames.

Registering message frames matched to the specifications of the external devices in advance simplifies data communications.

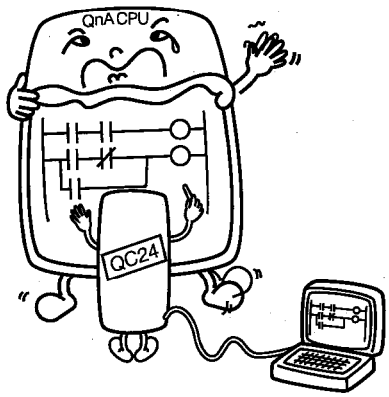
5 Data code conversion is possible (ASCII↔BIN conversion)*



The QC24 converts the data codes to match the opposite device.

For instance, when the PC CPU sends the data 1234H to an external device,

6 QnACPU can be accessed using GPP function*



A GPP function peripheral device corresponding to the QnACPU can connect to the QC24.

The same operations as when a GPP function peripheral device was connected to the QnACPU can be performed.

[CPU sequence program read/write, monitoring, remote operation, etc.]

(See Section 3.6.)

* : The computer link module does not have this function. It was newly added to the QC24.

1.3 Kinds of Data Communications Functions

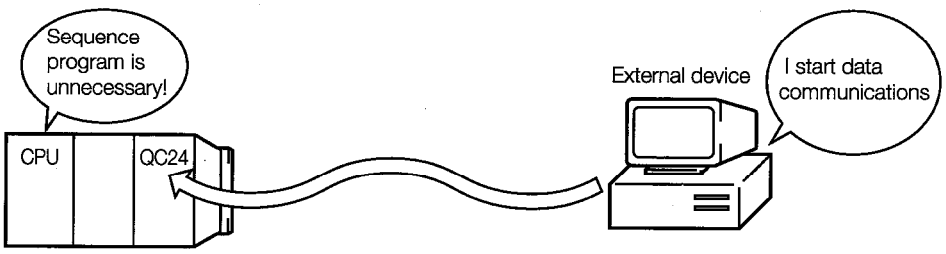
The QC24 can use dedicated protocols, non procedure protocol, and bidirectional protocol to communicate with external devices.

The following sections give a general description of each of these protocols.

1.3.1 What is dedicated protocol?

"Dedicated protocol" uses the message format specified in advance by the QC24 to communicate with external devices.

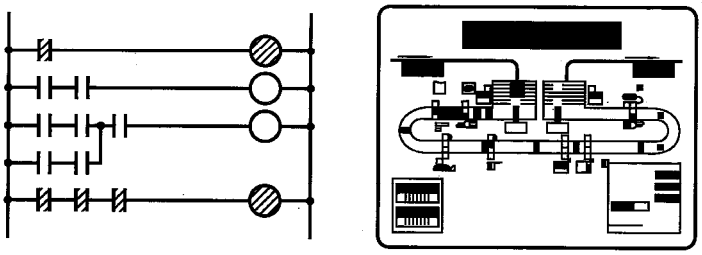
Dedicated protocols are used to read and write the data (device data, etc.) in the PC CPU. Therefore, a special data communications sequence program is unnecessary.



The following can be performed using dedicated protocol data communications.

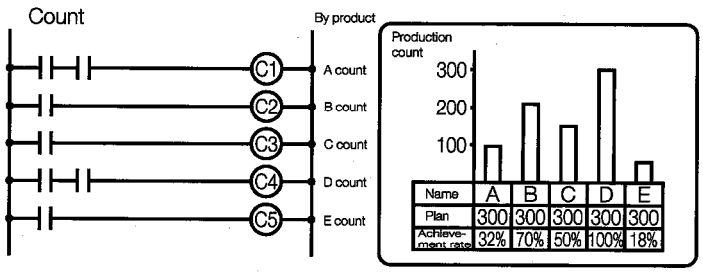
1 **Supervision of the operating status of the PC**

Operation of a production line, machine, etc. can be graphically monitored and supervised from a computer screen.

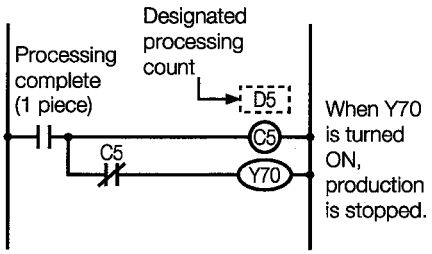


2 **Data acquisition and analysis**

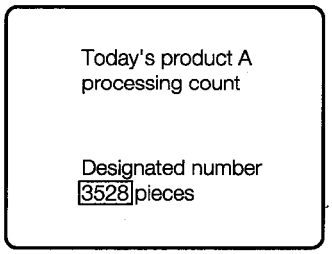
The production count, operation count, etc. of a production line, machine, etc. can be monitored using graphs and tables displayed on a computer screen.



A computer can designate the processing count for a production line, machine, etc.



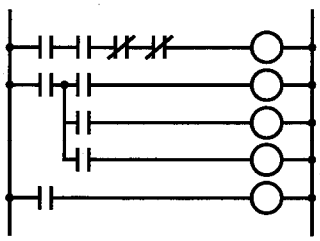
The sequence program uses a counter to count the number of pieces processed and stops production when the set count reached.



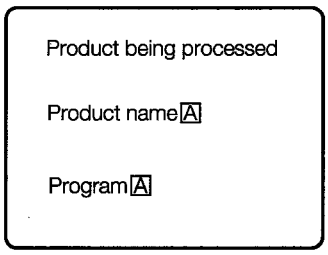
The computer writes 3528 to the counter C5 set value D5.

3 Sequence program uploading and downloading

The PC program and parameters can be read to the computer and preserved. Multiple programs can also be stored in the computer in advance, and the program matched to the control purpose can be written to the PC, as required.



(Product A processing program)
When the PC runs this program, it enables processing of product A.

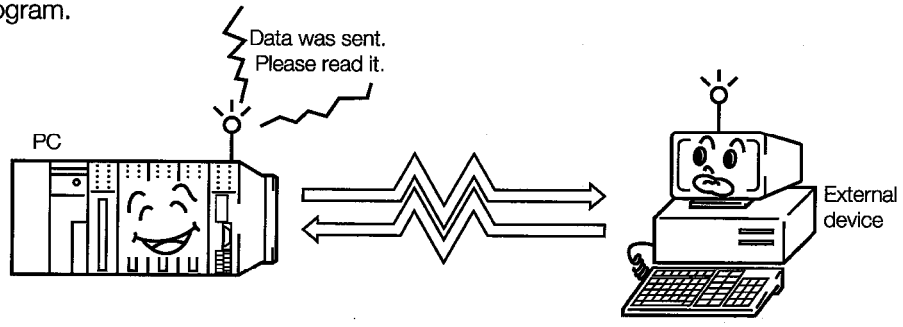


The program matched to the product name can be written from the computer to the PC and control commanded.

1.3.2 What is non procedure protocol?

"Non procedure protocol" sends and receives arbitrary data between the QC24 and external devices. The message format is not specified by the QC24 like the dedicated protocols. Therefore, the user determines the format of the messages to be sent and received. With the non procedure protocol, a sequence program to send and receive data from the QC24 is necessary. When the sequence program writes data and a send request to the QC24 buffer memory, the QC24 sends the written data code, or the written data code converted to the data code matched to the opposite device, to the external device.

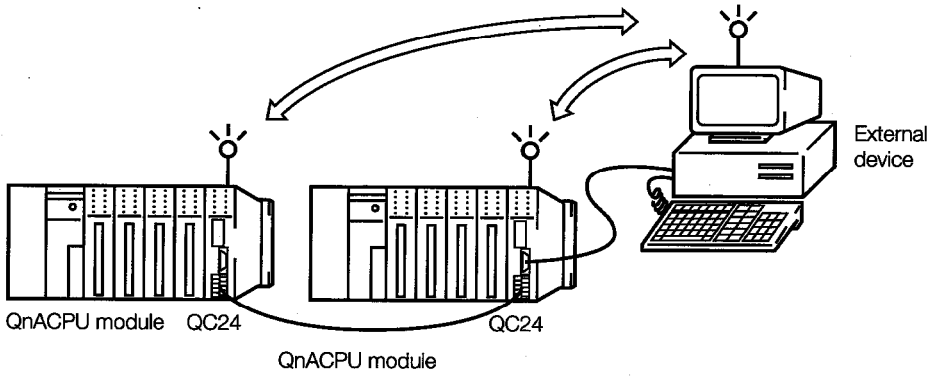
The QC24 converts the data received from the external device to a data code matched to the destination device, then stores it to the buffer memory from which it is read by the sequence program.



The following can be performed using non procedure data communications.

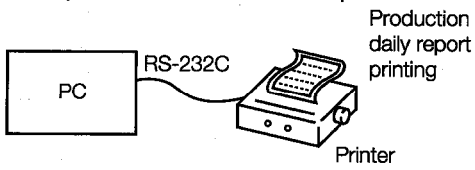
1 PC CPU and external device interaction

Arbitrary data can be exchanged between sequence program and external device.



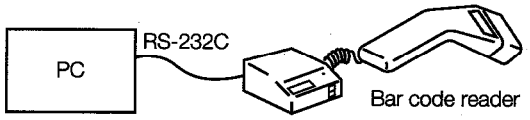
2 Data logging

The PC data can be printed on an RS-232C printer.



3 Reading of data from a terminal

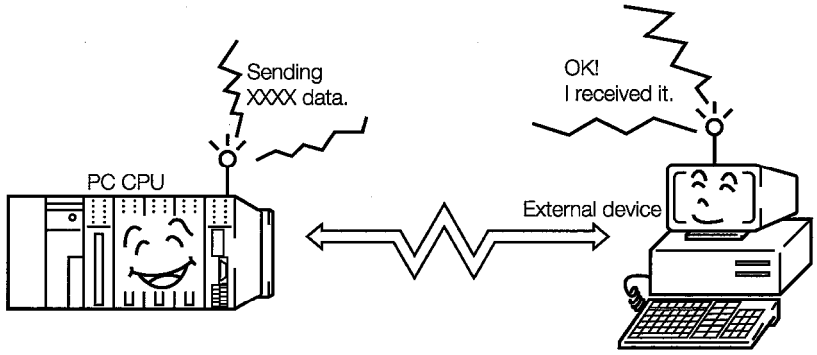
Data can be read by connecting a bar code reader to the PC.



1.3.3 What is bidirectional protocol?

"Bidirectional protocol" transmits arbitrary data agreed upon between sequence program and external device.

The difference between non procedure protocol and bidirectional protocol is that bidirectional protocol sends and receives data between PC CPU and external device while passing an End of Receive signal to the opposite device.



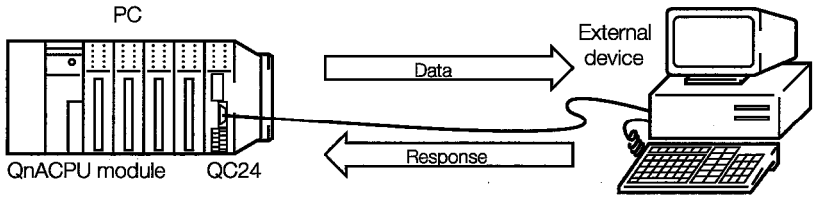
The following can be performed using the bidirectional protocol.

1 PC CPU and external device interaction

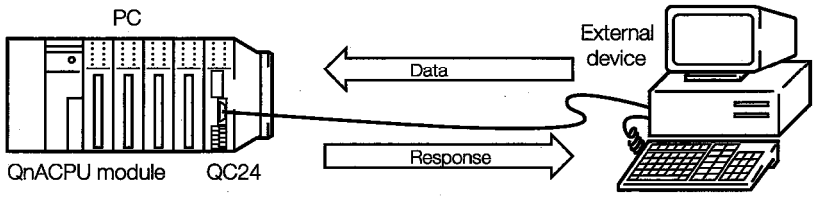
Arbitrary data can be sent and received between sequence program and external device.

The opposite device can use the response message to check whether or not the data was received correctly.

(Data transmission)



(Data transmission)



The QC24 performs all response message processing.

2 SYSTEM CONFIGURATION

This section describes the system configuration needed to use the QC24 to implement data communications.

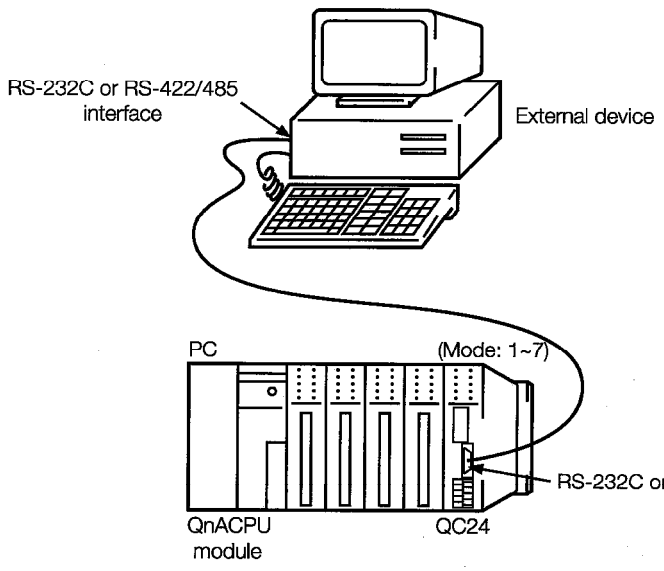
2.1 Data Communications System Configuration and Number of Link Stations

Two system external device and QC24 (PC CPU) system configurations are available: 1: 1~1 : 32 stations and 2: 1~2 : 32 stations.

The QC24 can connect to external devices through an RS-232C interface or an RS-422/485 interface.

2.1.1 1:1 ratio of external device to PC CPU

1 : 1 configuration



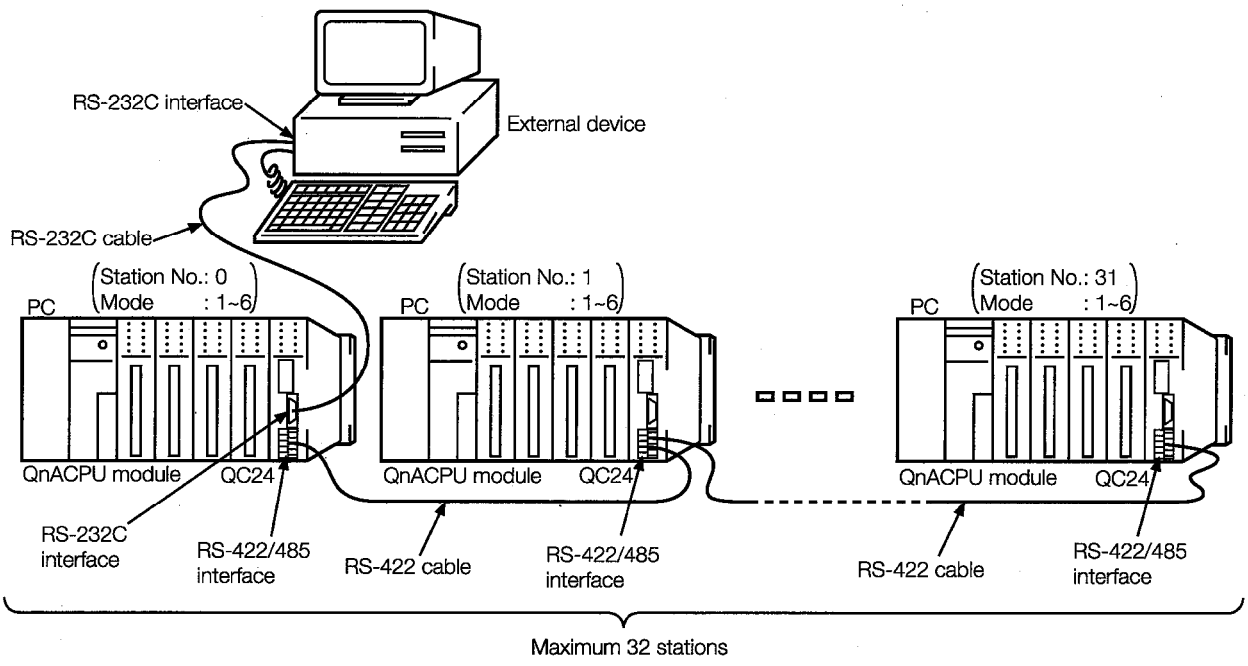
QC24 Functions That Can be Used		
Dedicated protocol	Non procedure protocol	Bidirectional protocol
○	○	○

Fig. 2.1 System Configuration (I)

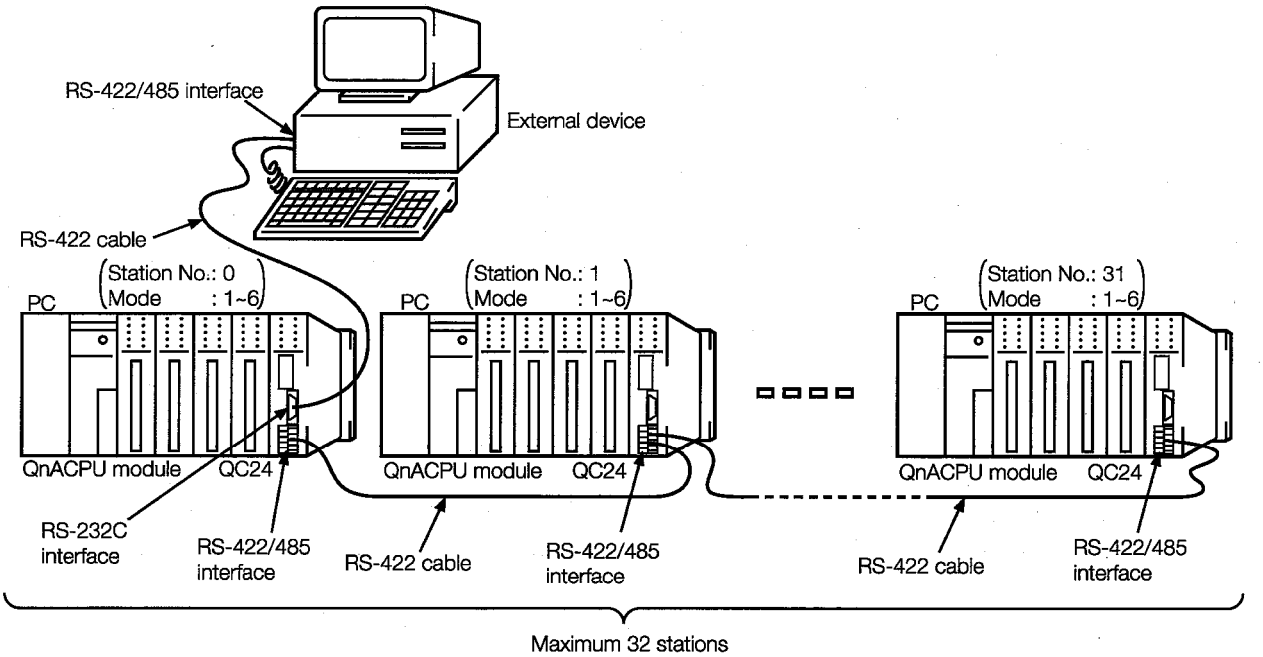
2.1.2 1:n ratio of external device to PC CPUs

1 : n configuratiox **(n: Maximum 32)**

[System configuration 1] When external device and QC24 connected by RS-232C line



[System configuration 2] When external device and QC24 connected by RS-422 line



QC24 Functions That Can be Used		
Dedicated protocol	**Non procedure protocol**	**Bidirectional protocol**
○*1	○	×

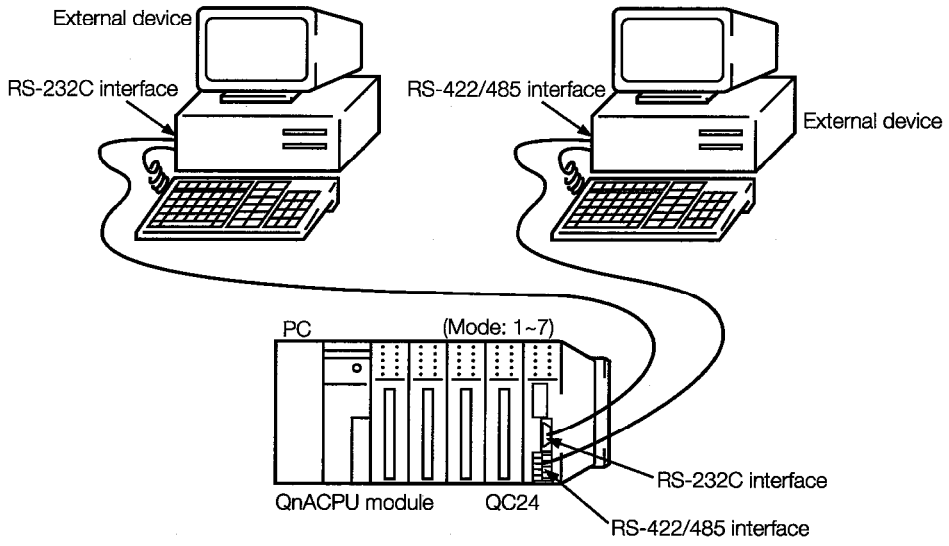
*1 On-demand function cannot be used.

Fig. 2.2 System Configuration (II)

2.1.3 2:n ratio of external devices to PC CPU

2 :1 configuration

(This configuration combines two 1;1 external device to PC CPU systems.)



QC24 Functions That Can be Used		
Dedicated protocol	Non procedure protocol	Bidirectional protocol
○	○	○

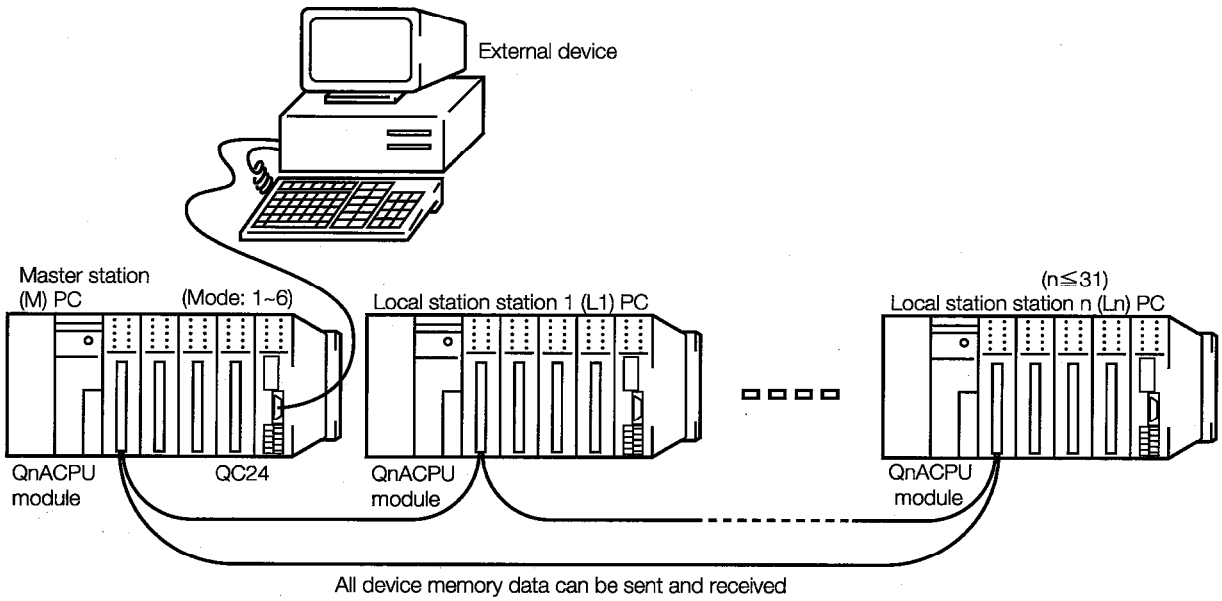
Fig. 2.3 System Configuration (III)

2.2 System Configuration to Access Another Station's PC CPU Over a MELSECNET

By using a dedicated protocol, an external device can communicate with two or more PC CPUs connected over a MELSECNET system.

If a QC24 is connected, and a PC CPU is connected to the external device, the external device can also communicate with PC CPUs not equipped with a QC24.

MELSECNET system configuration example



Range of PC CPUs that the external device can communicate with (for the system showabove)

Station connecting the external device	Station connecting the external device			
	M	L1	L2	L3
M	○	○	○	○
L1	○	○	×	×
L2	○	×	○	×
L3	○	×	×	○

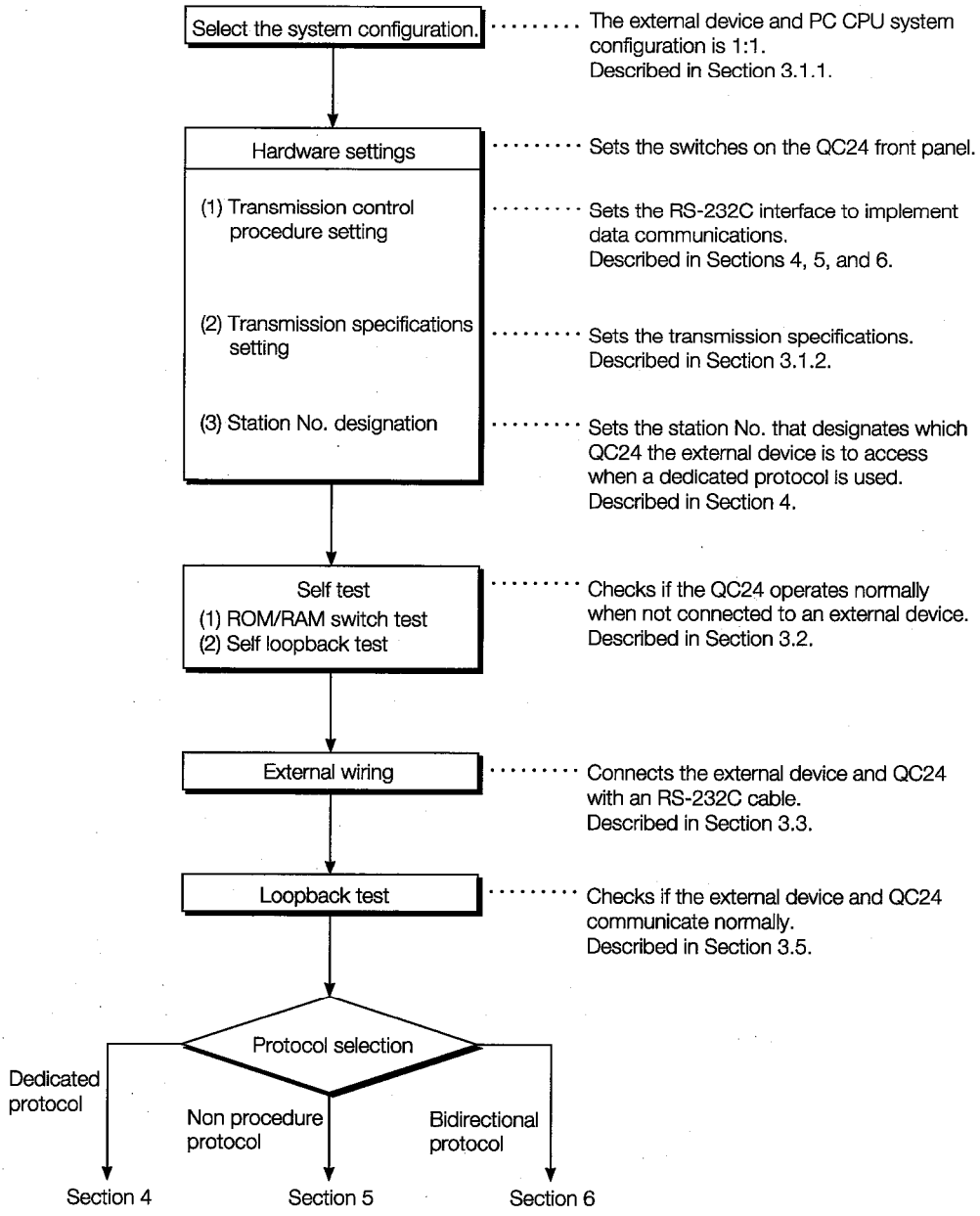
Fig. 2.4 System Configuration (IV)

3 PROCEDURE UP TO DATA COMMUNICATIONS DESCRIBED IN THIS GUIDEBOOK

3.1 Settings and Procedures up to Data Communications

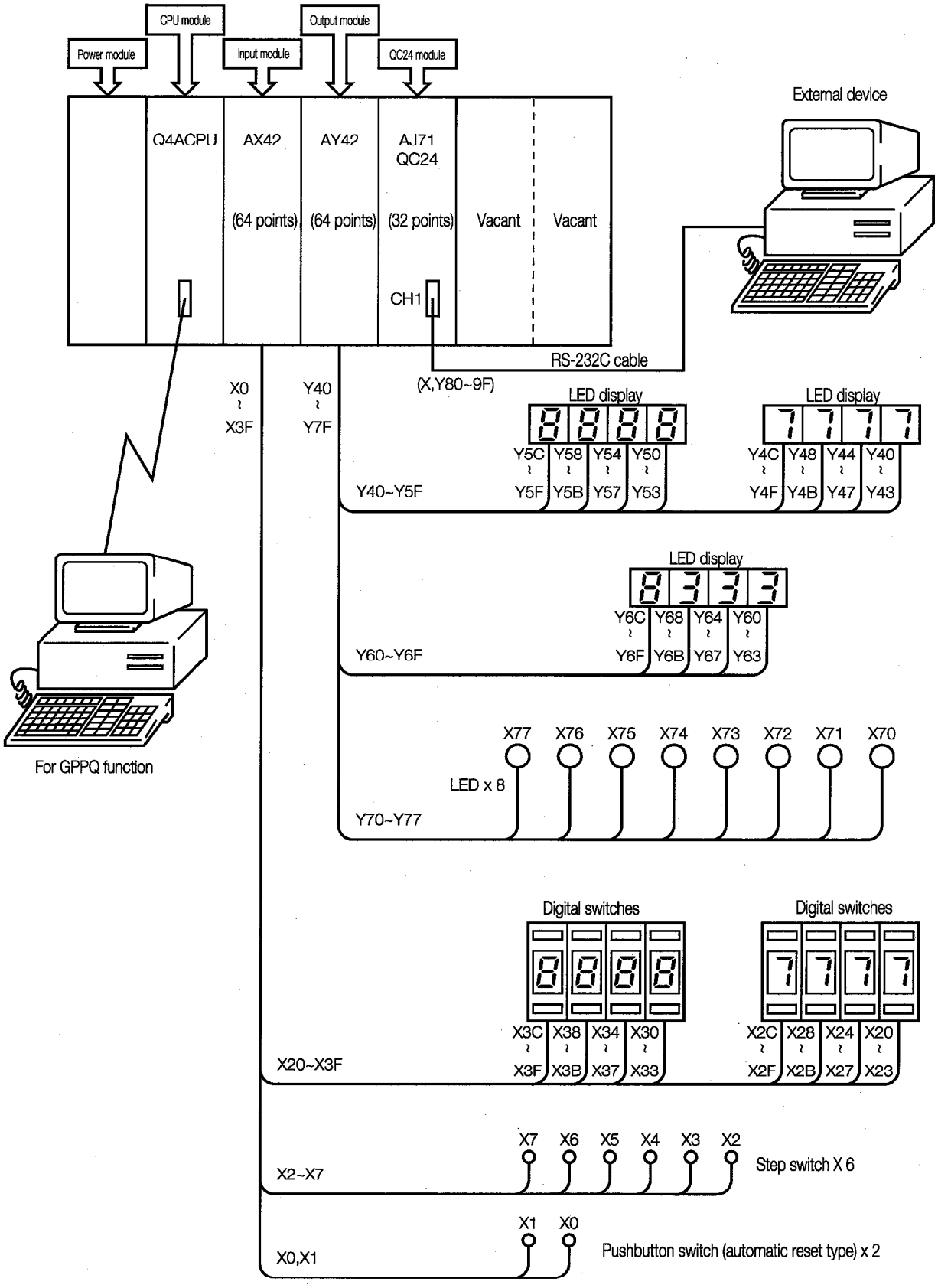
The following shows the settings and operation procedures for the data communications described in this guidebook.

The AJ71QC24 is used as the QC24 and the Q4ACPU is used as the PC CPU.



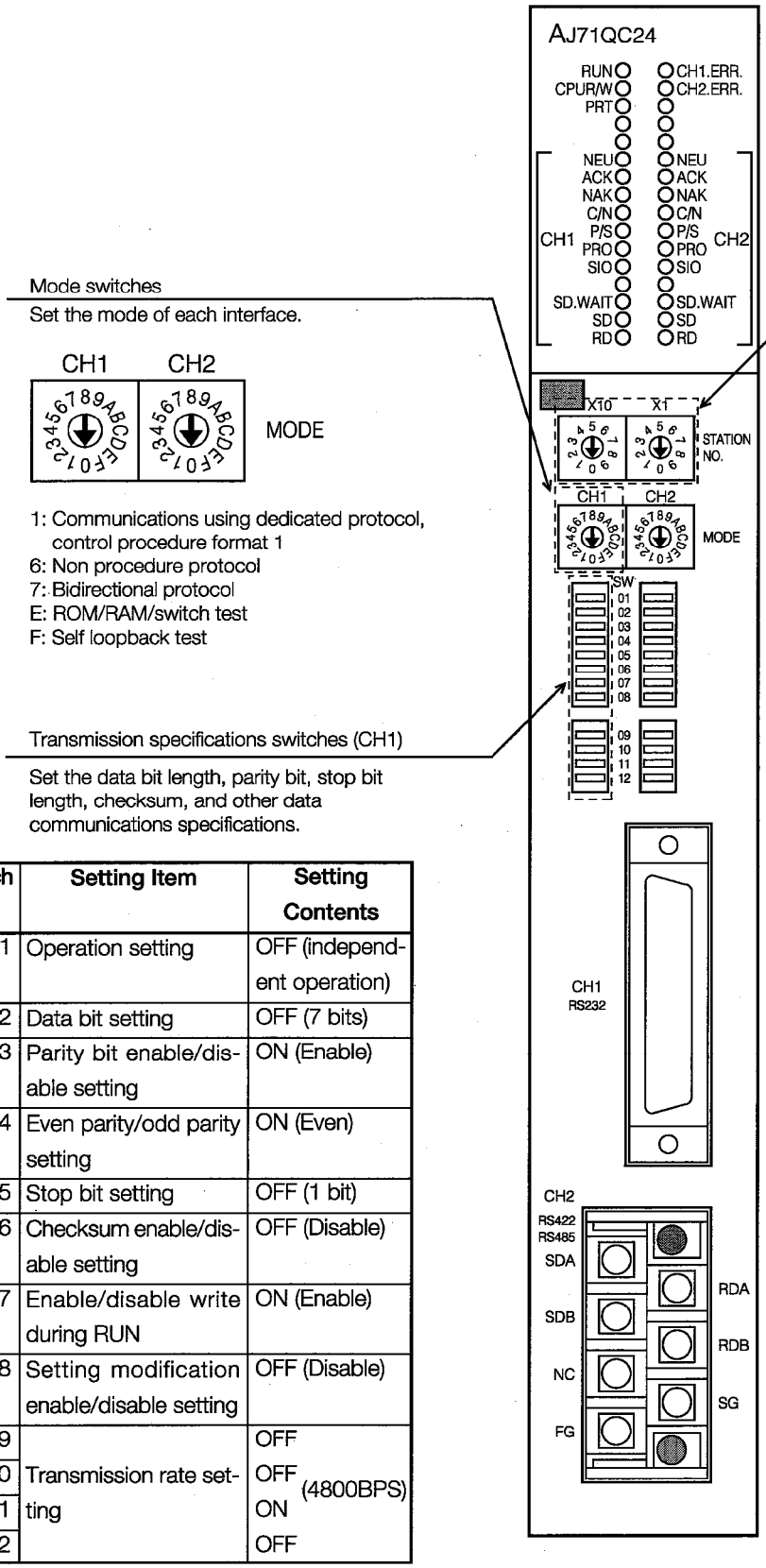
3.1.1 System configuration

The following shows the data communications system configuration used in this guidebook.



3.1.2 Functions of setting switches

The following describes the functions of the setting switches on the AJ71QC24 front panel.



Mode switches

Set the mode of each interface.

CH1 CH2

MODE

1: Communications using dedicated protocol, control procedure format 1
6: Non procedure protocol
7: Bidirectional protocol
E: ROM/RAM/switch test
F: Self loopback test

Transmission specifications switches (CH1)

Set the data bit length, parity bit, stop bit length, checksum, and other data communications specifications.

Station No. switches

Set the station No. for 1:n system configuration communications. Setting is performed with 0~31. For 1:1 communications, this switch is usually set to 0.

X10 X1

STATION NO.

"0" "0"

setting → communicates as station 0.

Switch	Setting Item	Setting Contents
SW01	Operation setting	OFF (independent operation)
SW02	Data bit setting	OFF (7 bits)
SW03	Parity bit enable/disable setting	ON (Enable)
SW04	Even parity/odd parity setting	ON (Even)
SW05	Stop bit setting	OFF (1 bit)
SW06	Checksum enable/disable setting	OFF (Disable)
SW07	Enable/disable write during RUN	ON (Enable)
SW08	Setting modification enable/disable setting	OFF (Disable)
SW09	Transmission rate setting	OFF
SW10		OFF (4800BPS)
SW11		ON
SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

3.2 Self Test

Self test is a function that checks if the QC24 operates normally when not connected to an external device. After installing the QC24 into the QnACPU base unit, first check its operation as described below. Before setting the switches, or connecting the cables, always turn OFF the power.

3.2.1 ROM/RAM/switch test

The ROM/RAM/switch test checks if the ROM/RAM operates normally and if the switch settings are correct.

1 Mode switches setting

Set the CH2 mode switch to "E" (ROM/RAM/switch test) and the CH1 mode switch to "1".

2 Transmission specifications switches setting

Set the transmission specifications switches as shown in Section 3.1.2.

3 ROM/RAM/switch test execution

① Stop the PC CPU.

Set the PC CPU RUN/STOP key switch to the STOP position.

② PC CPU power ON or CPU reset

When the PC CPU power is turned ON, or the CPU is reset, the QC24 Ready signal is turned ON and the QC24 is checked automatically.

(The Ready signal is turned ON several seconds after the power is turned ON, or the CPU is reset.)

③ Check sequence

Carry out the checks in (1) ROM check → (2) RAM check → (3) switch check order.

④ End of checks confirmation

When the CH1 and CH2 SD.WAIT LEDs are turned ON, the checks are complete.

4 Test executed result confirmation

① Check the state of the LED display shown on the following page.

ROM/RAM/switch check contents

Check Item	Check Contents	Check LED			Error Processing
		LED Name	Normal	Error	
ROM check	Checks if the ROM is normal.	CH1.ERR	OFF	ON	Check the mounting state of the module and repeat the test. If the module is correctly mounted, contact your nearest Mitsubishi Electric service center, dealer, or branch office.
RAM check	Checks if the RAM is normal.	CH2.ERR	OFF	ON	
Switch check	Checks if the switch settings are within the setting range.	CH2.ERR one of the below	OFF	ON	Station No. setting error. Check the station No. switches.
		CH1-C/N CH2-C/N	OFF	ON	Mode setting error. Check the mode switches.
		CH1-P/S CH2-P/S	OFF	ON	Transmission specifications setting error. Check the transmission specifications switches.
End of check	All the checks are complete.	CH1-SD.WAIT CH2-SD.WAIT	ON		———

Normal ..... To end the test, perform item ⑤ below.

Error ..... Turn OFF the power, then check the station switches, mode switches, transmission specifications switches, etc. and set the switches correctly and repeat the test.

⑤ Ending the tests

(1) Turn OFF the power.

(2) Since the self loopback test is performed, set the CH1 and CH2 mode switches to "F".

3.2.2 Self loopback test

The self loopback test checks the functions that communicate with the PC CPU and the CH1 (RS-232C) or CH2 (RS-422/485) interface send and receive functions.

1 Mode switches setting

Set the CH1 and CH2 mode switches to "F" (self loopback test).

2 Transmission specifications switches setting

Set the CH1 and CH2 transmission specifications switches as shown in Section 3.1.2.

3 Cable connection

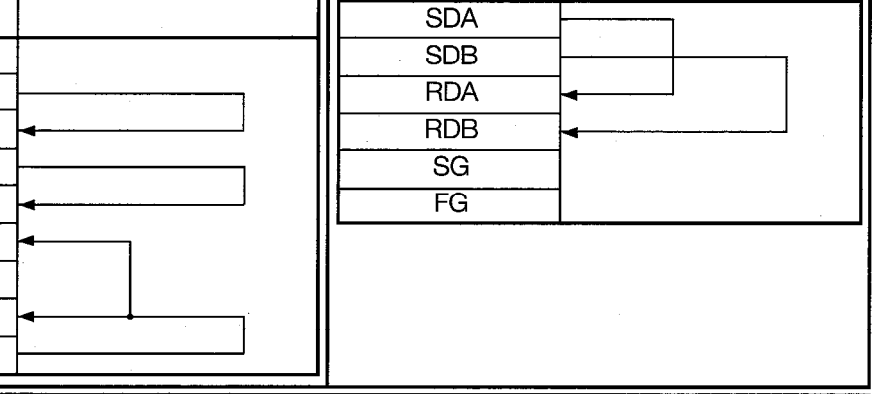
Connect the CH1 (RS-232C) connector and CH2 (RS-422/485) terminal board with a cable as shown below.

CH1 (RS-232C) Cable Connection

AJ71QC24 Signal Name	Pin No.	Cable Connection
FG	1	
SD	2	
RD	3	
RS	4	
CS	5	
DSR	6	
SG	7	
CD	8	
DTR	20	

CH2 (RS-422/485) Cable Connection

AJ71QC24 Signal Name	Cable Connection
SDA	
SDB	
RDA	
RDB	
SG	
FG	



① Connect the connector connecting the test cable shown above to the CH1 RS-232C connector.
② At the CH2 RS-422/485, jumper the pins as shown above.

4 Self loopback test execution

① Set the PC CPU RUN/STOP key switch to the STOP position.

② When the PC CPU power is turned ON, or the CPU is reset, the QC24 Ready signal is turned ON and the checks start automatically.

(The Ready signal is turned ON several seconds after the power is turned ON, or the CPU is reset.)

③ Check sequence

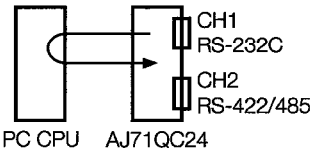
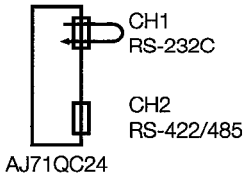
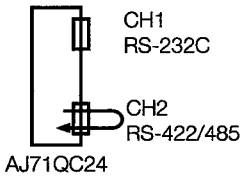
Repeat the checks in (1) PC CPU communications check → (2) CH1 RS-232C check → (3) CH2 RS-422/485 check order. (The QC24 does this automatically.)

④ Check the state of the LED displays shown on the next page

Normal..... End the tet by performing item ⑤ below.

Error ..... Turn OFF the power, then check the modes switches, transmission specifications switches, cable connections,etc. and set the switches correctly and repeat the test.

Self loopback test check contents

Check Item	Check Contents	Nomal Display		Error Display		Notes
PC CPU communications check	Checks the functions that communicate with the PC CPU.	CH1-C/N	OFF	CH1-C/N	ON	
		CPUR/W	Light dimly	——	——	
		CH1-NEU, ACK, NAK	Blink sequentially			
CH1(RS-232C) communications check	Checks the CH1 RS-232C interface send and receive functions.	CH1.ERR	OFF	CH1.ERR	ON	
		CH1-SD	Blink	——	——	
		CH1-RD				
CH2(RS-422/485) communications check	Checks the CH2 RS-422/485 interface send and receive functions.	CH2.ERR	OFF	CH2.ERR	ON	
		CH2-SD	Blink	——	——	
		CH2-RD				

*The tests continue even if an error occurs in any of the check items.

⑤ Ending the tests

(1) Turn OFF the power.

(2) Disconnect the cable and connect the cable to communicate with the external device.

(3) Change the setting of the mode switches.

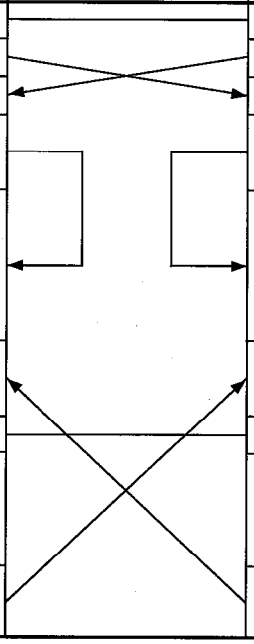
CH1: "1" (dedicated protocol format 1) / "6" (non procedure protocol) / "7" bidirectional protocol)

CH2: "1"

3.3 External Wiring

The RS-232C cable wiring diagram is shown below.

(When connected to a device that does not turn ON the CD signal.)

AJ71QC24 Signal Name	AJ71QC24 Pin No.	Cable Connection and Signal Direction	External Device Signal Name	Name	Contents (Based on External Device)
FG	1		FG	Frame Ground	Cable shield pin
SD	2		SD	Send Data	Signal pin that sends data.
RD	3		RD	Receive Data	Signal pin that receives data.
RS	4		RS	Request to Send	Signal pins that turn ON and send a signal to the local station and simultaneously inform the opposite device that the local station has data to send when the local station is ready to send.
CS	5		CS	Clear to Send	
DSR	6		DSR	Data Set Ready	Pin that receives the ready signal from the opposite device.
SG	7		SG	Signal Ground	Signal ground pin
CD	8		CD	Carrier Detection	Pin that receives the ON signal when there is send data from the opposite deivice.
DTR	20		DTR	Data Terminal Ready	Pin used by the local station to show that it is ready to operate.

Conect the cable to match the specifications of the external device.

The RS-232C connector used is shown below. Connect the connector of the opposite device matched to this connector to the cable.
20 pin D sub (female) screw type
DDK Electronics Ltd. 17LE-13250-22-D2AC

3.4 External Device Setting

Set the external device as shown below.

RS-232C-0 ..... 4800 bauds 7 bits* even parity stop bit 1 No X parameters

Printer ..... 24-bit system

Memory size (KB) 640

Screen display attribute ..... White

Numeric coprocessor ..... None

BOOT device ..... Standard

Numeric coprocessor 2 ..... None

*In the bidirectional mode, 8 bits is set.

3.5 Loopback Test

The loopback test function uses dedicated protocol format dedicated commands to check the external device and QC24 communication functions.

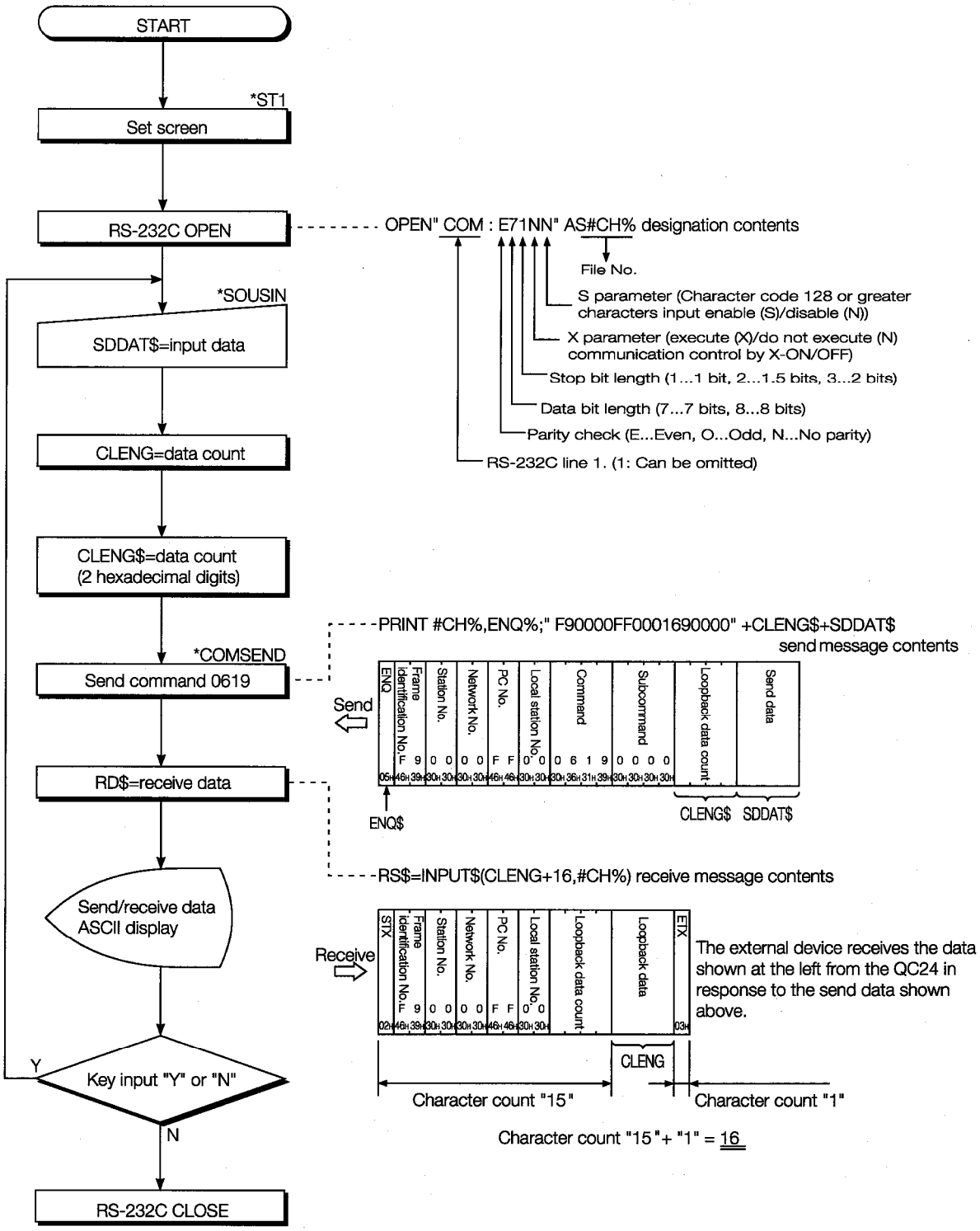
The external device sends data to the QC24 and the QC24 returns the received data to the external device as is.

The loopback test checks if the data sent from the external device and the data returned from the QC24 are the same.

3.5.1 Loopback test program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                      AJ71QC24 Loopback Test Sample Program
1020 ' !                                (Command 0619)
1030 ' ! ------------------------------------------------------------------------------!
1040 *ST1
1050    CLS                                 :'Clear screen
1060    CH%=1                               :'Channel No.
1070    ENQ$=CHR$(&H5)                      :'ENQ code
1080 ' ! ------------------------- RS-232C open & initialize----------------------------!
1090    OPEN"COM:E71NN" AS #CH%             :'Set communications mode, etc.
1100    '
1110 ' ! ------------------------------ Send data input---------------------------------!
1120 *SOUSIN
1130    CLS
1140    LOCATE 1, 1 : INPUT "Send data";SDDAT$
1150    CLENG=LEN(SDDAT$)                   :'Character length
1160    CLENG$=RIGHT$("000"+HEX$(CLENG),4):'Character length→hexadecimal conversion
1170    '
1180 ' ! ------------------------------ Send command 0619-------------------------------!
1190 *COMSEND
1200    PRINT #CH%,ENQ$;"F90000FF0006190000"+CLENG$+SDDAT$
1210    RD$=INPUT$(CLENG+ 16,#CH%) :MAXR%=LEN(RD$) :'Loopback data reception
1220 ' ! ------------------------Send data ASCII code display -------------------------!
1230    LOCATE 48, 0 : PRINT "Send data"
1240    LOCATE 48, 1 : PRINT "ASCII code"
1250    SD$=ENQ$+"F90000FF0006190000"+CLENG$+SDDAT$ :MAXS%=LAN(SD$)
1260    FOR I%=1 TO MAXS%
1270      IF I%>19 THEN Y%=I%-17 : X%=5 ELSE Y%=I%+2 : X%=0
1280      LOCATE 48+X%,Y%
1290      PLINT RIGHT$("0"+HEX$(ASC(MID$(SD$,I%,1))),2)
1300    NEXT I%
1310 ' ! ---------------------- Receive data ASCII code display-------------------------!
1320    LOCATE 65, 0 : PRINT "Receive data"
1330    LOCATE 65, 1 : PRINT "ASCII code"
1340    FOR I%+1 TO MAXR%
1350      IF I%>11 THEN Y%=I%-9 : X%=5 ELSE Y%=I%+2 : X%=0
1360      LOCATE 65+X%,Y%
1370      PLINT RIGHT$("0"+HEX$(ASC(MID$(RD$,I%,1))),2)
1380    NEXT I%
1390 '
1400    LOCATE 1,20 : INPUT "Retransmit (Y/N) ?"; Y$
1410    IF Y$="Y" THEN *SOUSIN
1420 CLOSE
1430 END
```

* BASIC commands must be changed according to BASIC software.



*[ ]: Label name used with the program

3.5.2 Loopback test operation procedure

① Connect the external device and QC24 in the regular configuration with a cable as described in Section 3.3.

② Set the QC24 front panel switches as shown in Section 3.1.2.

Set the mode switch to "1".

③ Start the PC CPU.

When the QC24 NEU LED is turned on, the loopback test can be executed.

(The NEU LED is turned ON several seconds after the PC CPU power is turned ON, or the CPU is reset.)

④ Write the program shown in Section 3.5.1 to the external device. After writing, set the external device to "RUN".

If there are no errors, the following prompt will appear on the external device screen.

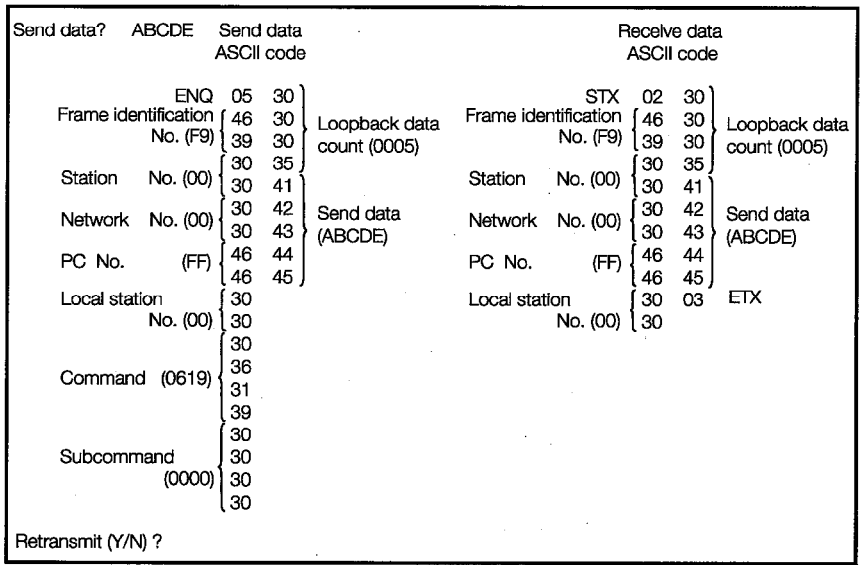
```
Send data? ■
```

⑤ Input arbitrary data from the external device keyboard.

(For example, A B C D E ↵)

⑥ The external device checks if the data it send and the data returned from the QC24 are the same.

If the two data are the same, communications between the external device and QC24 are normal.



If you have any situations as follows, please read the explanations

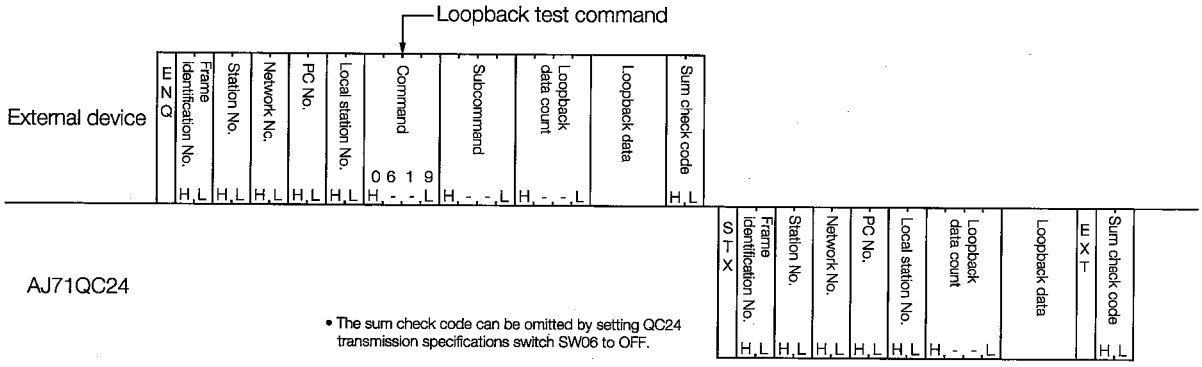
• Cannot communicate	Are the hardware settings (Section 3.1.2) and cable connections (Section 3.3) correct ? Check the QC24 LEDs (Section 4.3), then check the settings.
• The data sent from the computer and the data send from the QC24 are not the same.	Are the transmission specifications settings (Section 3. 1.2), cable connections (Section 3.3), and external device settings (Section 3.4) correct ? Check the QC24 LEDs (section 4.3), then check the settings.

After resetting the settings, reset the PC CPU and repeat the loopback test.

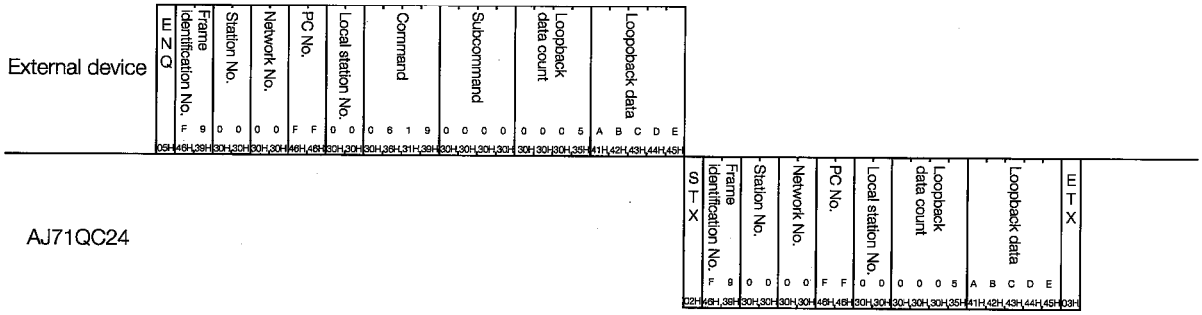
3.5.3 Loopback test message structure

The following shows the loopback test control procedure and message structure. (QnA frame format 1, command 0619)

[Control Procedure] The control procedure (QnA frame: format 1) message structure is shown below.



[Designation Example] The following examples designates "ABCDE" in the data area of the send data and executes the loopback test. (Sum check code omitted.)



See Section 4.1.1 for a detailed description of the control procedure and message structure.

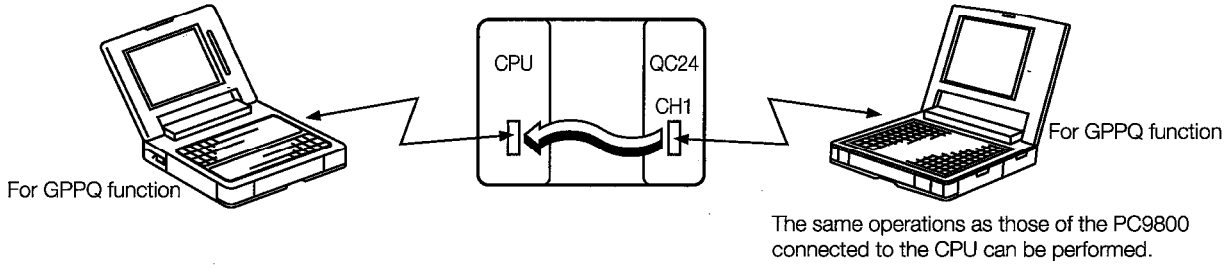
After the end of the loopback test, use a dedicated protocol to send and receive data.

Go to Section 4.

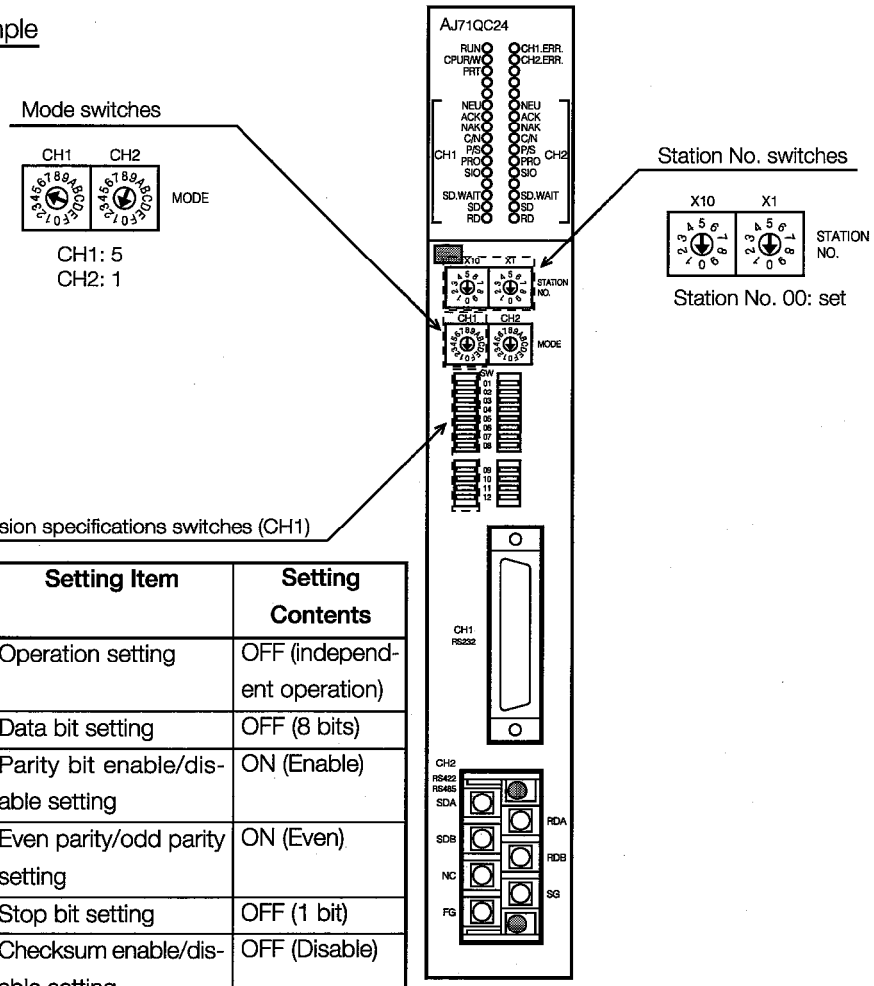
When using the non procedure protocol, or bidirectional protocol, to send and receive data, go to Section 5 or Section 6.

3.6 Communicationg With QnACPU Using the GPP Function

The QC24 CH1 RS-232C interface can be set to QnA extension frame format 5 (binary mode) and a GPP functoin peripheral device corresponding to the QnACPU can be connected.



QC24 setting example



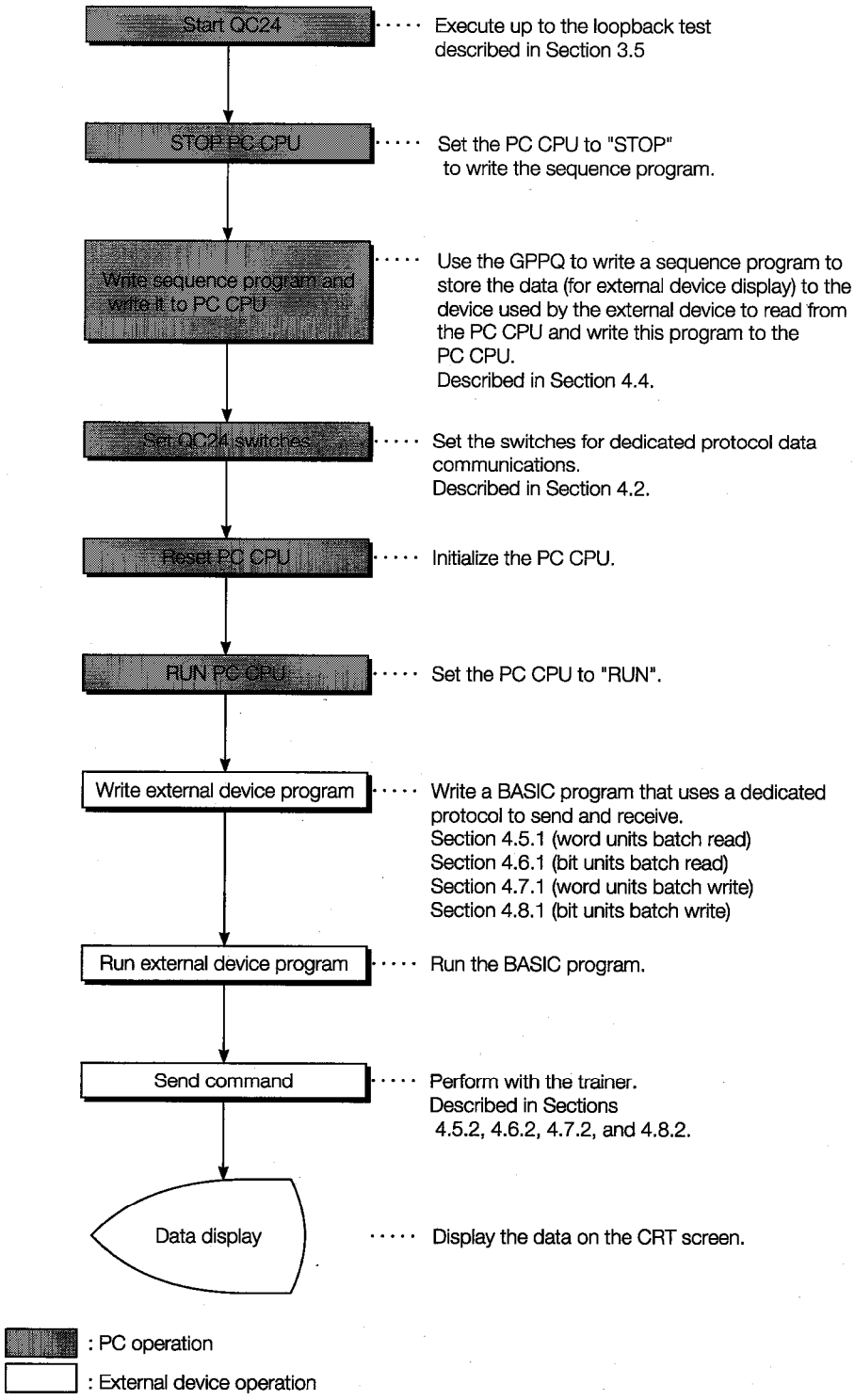
Switch Setting	Switch	Setting Item	Setting Contents
SW 01 02 03 04 05 06 07 08	SW01	Operation setting	OFF (independent operation)
	SW02	Data bit setting	OFF (8 bits)
	SW03	Parity bit enable/disable setting	ON (Enable)
	SW04	Even parity/odd parity setting	ON (Even)
	SW05	Stop bit setting	OFF (1 bit)
	SW06	Checksum enable/disable setting	OFF (Disable)
	SW07	Enable/disable write during RUN	ON (Enable)
	SW08	Setting modification enable/disable setting	OFF (Disable)
09 10 11 12	SW09	Transmission rate setting	OFF
	SW10		OFF (9600BPS)
	SW11		ON
	SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

*Set to match the transmission rate of the GPPQ function. (Default value 9600 BPS)

4 COMMUNICATIONS USING A DEDICATED PROTOCOL

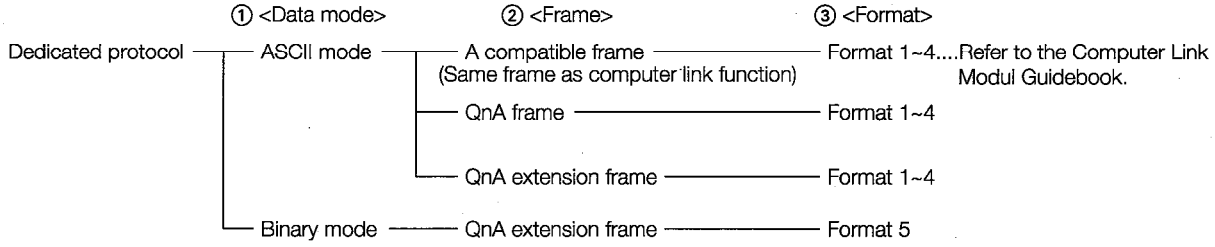
The following shows the dedicated protocol communications procedure used in this guidebook.



4.1 Dedicated Protocol Formats

The following shows the dedicated protocols that send and receive data between external device and QC24.

Each frame can be chosen to match the data communications purpose and the specifications of the external device.



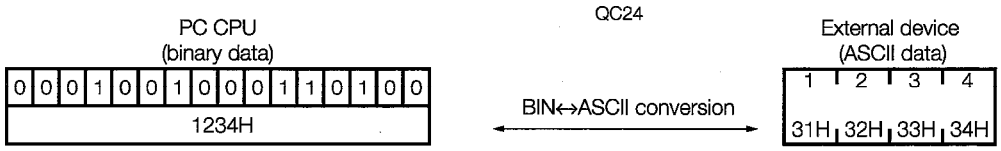
① The QC24 can select the ASCII mode or binary mode.

The following shows ASCII mode and binary mode data communications examples.

- ASCII mode

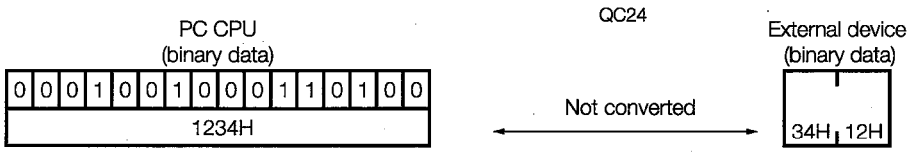
  The QC24 converts the binary data sent from the PC CPU to ASCII data and stores the ASCII data to the device memory.

  The QC24 converts the ASCII data received from the external device to binary data and stores the binary data to the device memory.



- Binary mode

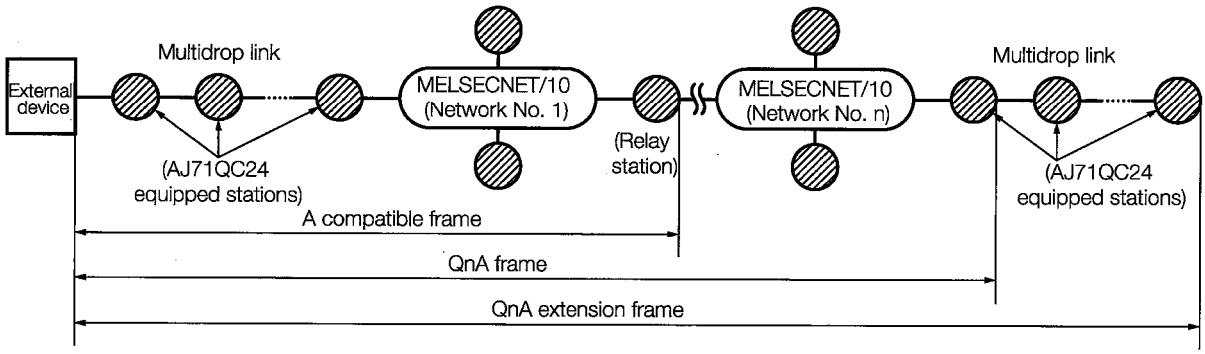
  The QC24 stores the data sent from the PC CPU to the device memory as is. It also stores the binary data received from the external device to the device memory as is.



② Frames can be used according to the application.

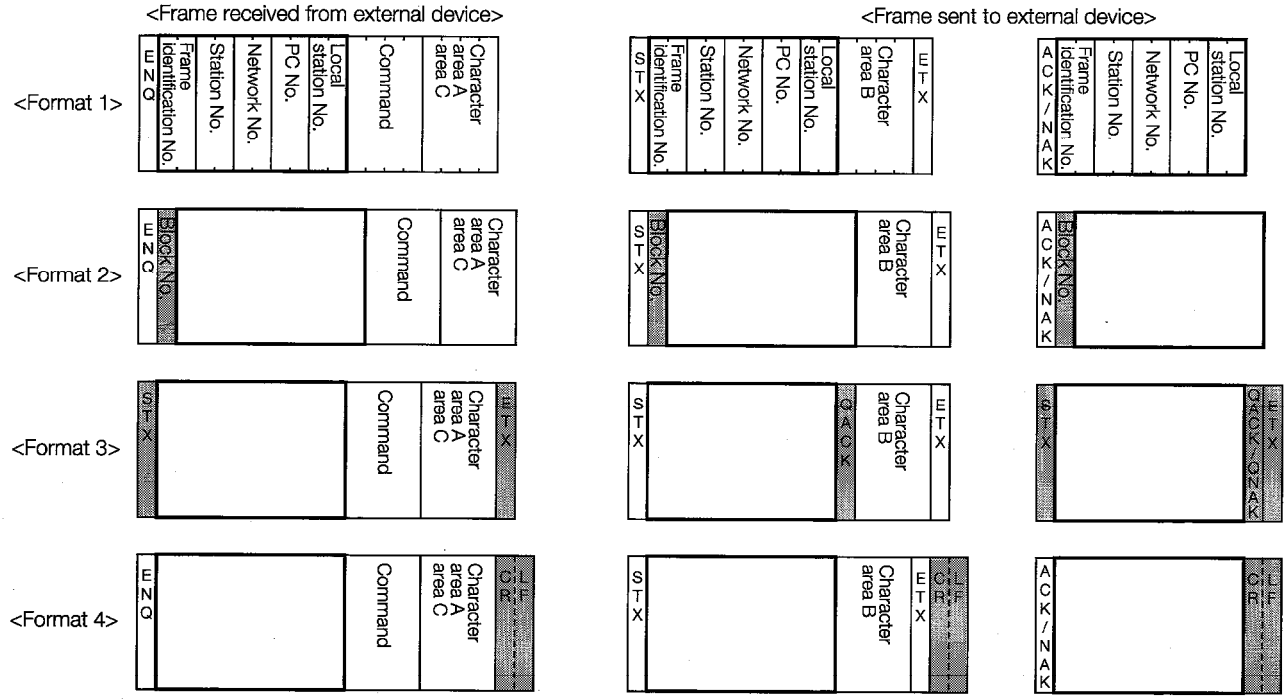
Frame	A Compatible Frame	QnA Frame	QnA Extension Frame
Access to QnACPU	Reading and writing over the same device range as the AnUCPU is possible.	All devices can be accessed.	All devices can be accessed.
Features	Same message structure as MELSEC-A Series Computer Link Module dedicated protocol. (The commands that can be used are limited.)	All the QC24 functions can be used.	All the QC24 functions can be used. Extends the range of access to other stations to beyond that of the QnA frame.

The following shows the range of access to other stations over a MELSECNET/10.



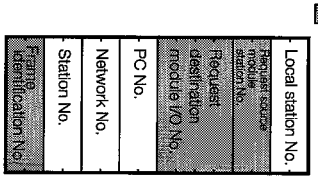
③ The following shows the differences between QnA frame and QnA extension frame formats 1 to 4.

QnA frame



QnA extension frame

QnA extension frame formats 1 to 4 are the same, except for the parts enclosed in ☐. The parts enclosed in ☐ are replaced by the following.



items are different from those of the QnA frame.

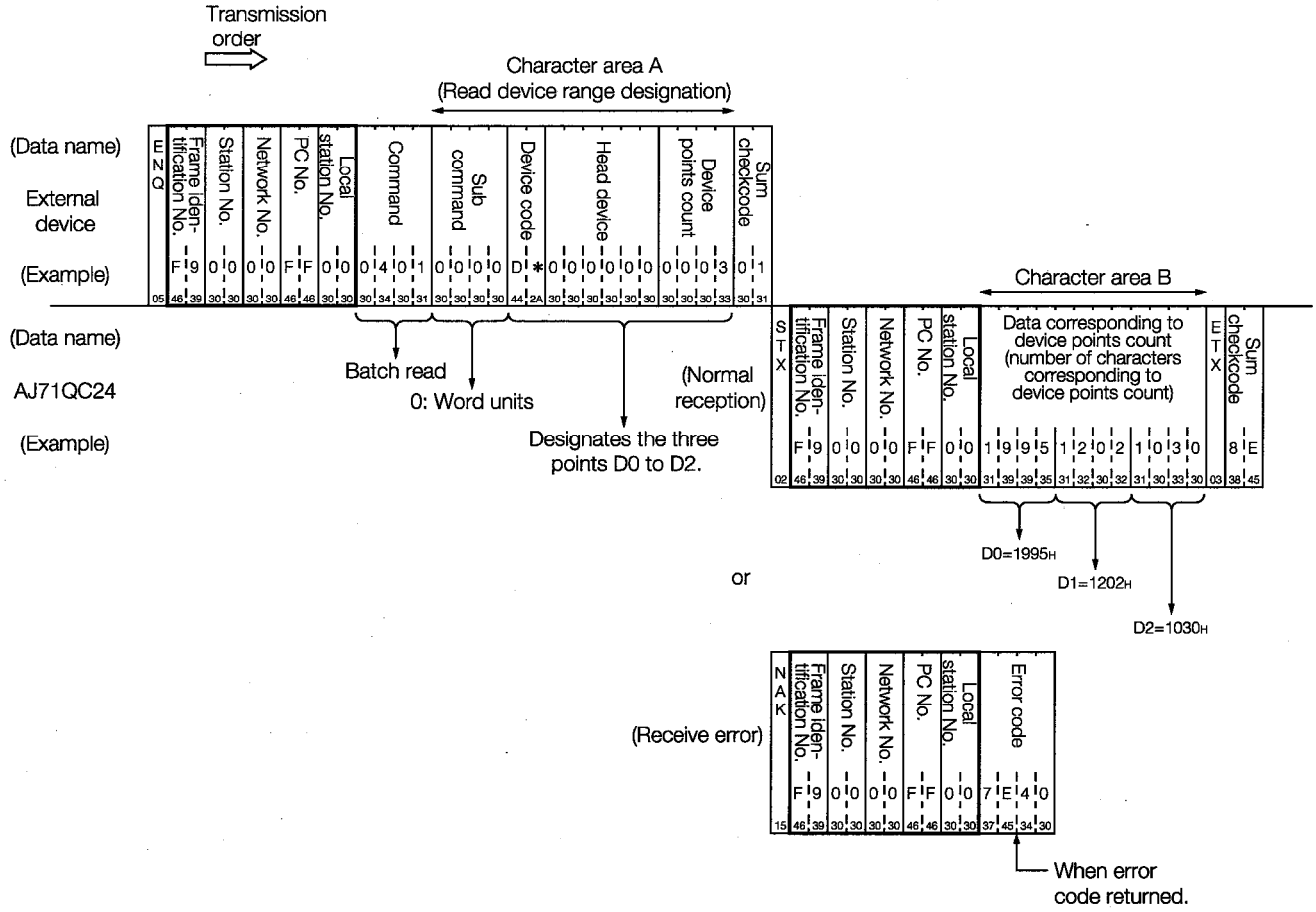
This guidebook describes data communications in QnA frame format 1 message format.

4.1.1 QnA frame format 1 control procedure and message structure

1 Control procedure an external device reads PC data

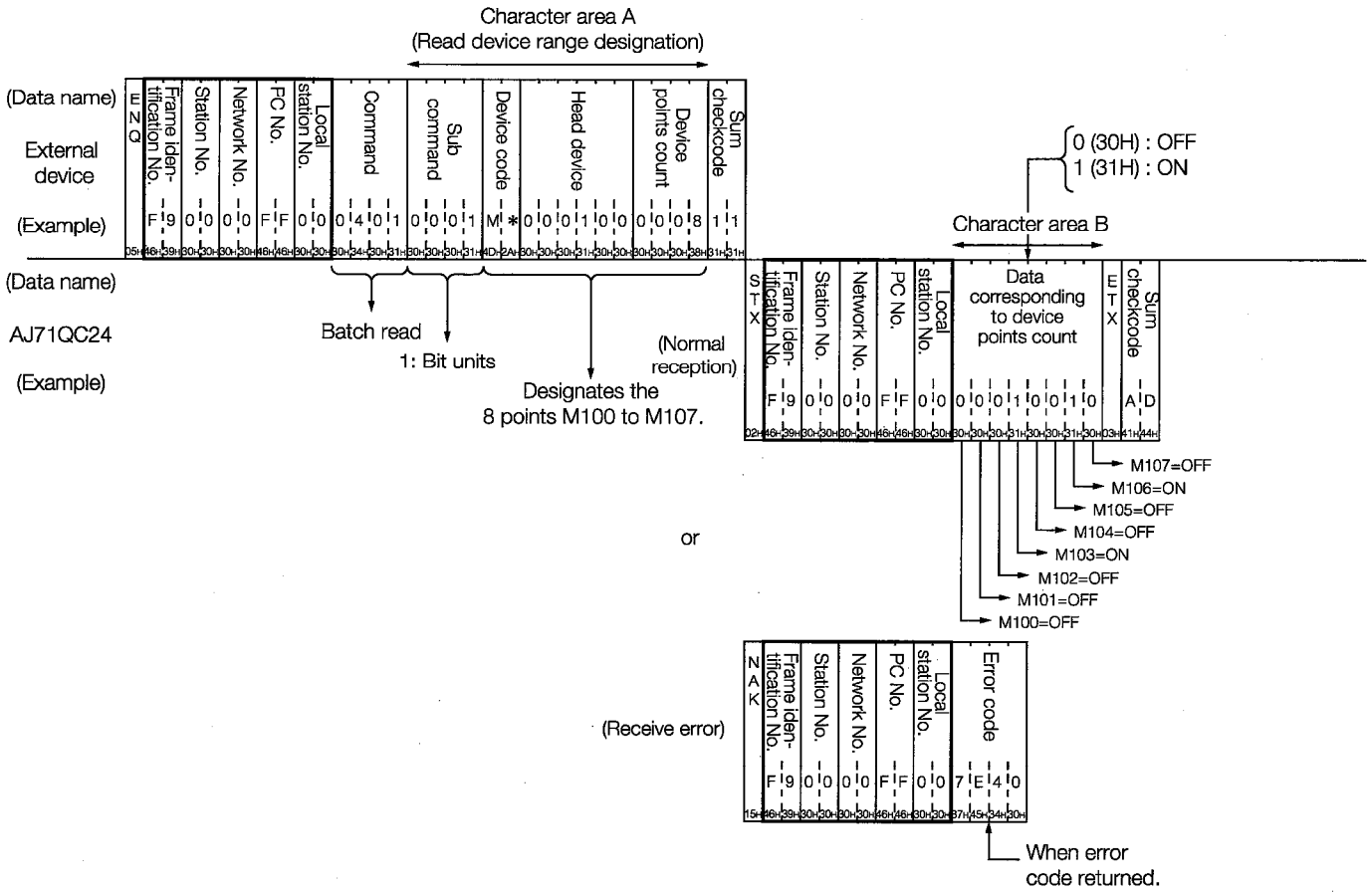
(a) Word units read

(Example) Use command 0401 (device memory batch read) to read the contents of PC CPU data registers D0 to D2.



(b) Bit units read

(Example) Use command 0401 (device memory batch read) to read PC CPU internal relays M100 to M107.



Memo

Relationship between the device points count the external device is requested to read and the data byte count of character area B in message

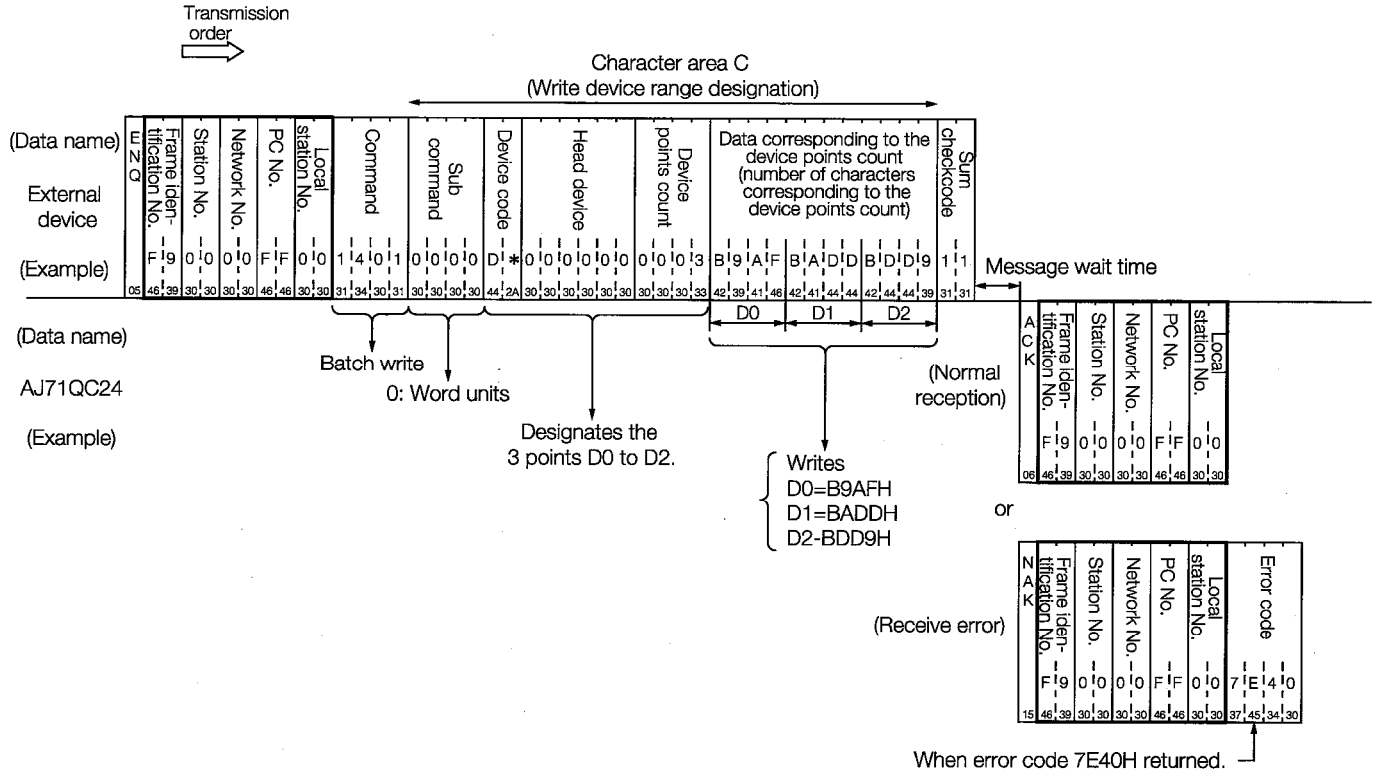
		Device points count		Character area B data byte count
Word units	: 1 word	= 1 point	=	4 bytes of data
Bit units	: 1 bit	= 1 point	=	1 byte of data

This relationship is also the same when writing data.

2 Control procedure when writing data from external device to PC

(a) Word units write

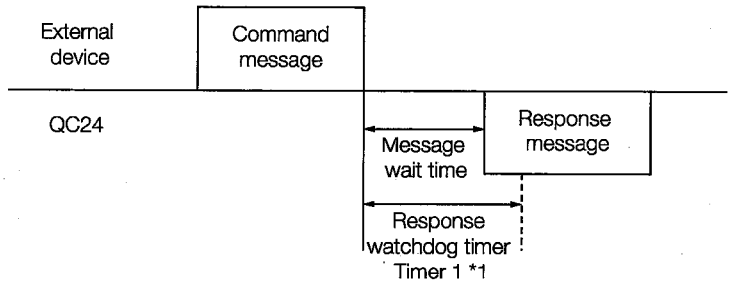
(Example) Use command 1401 (device memory batch write) to write data to PC CPU data registers D0 to D2.



Memo

Message Wait Time Designation

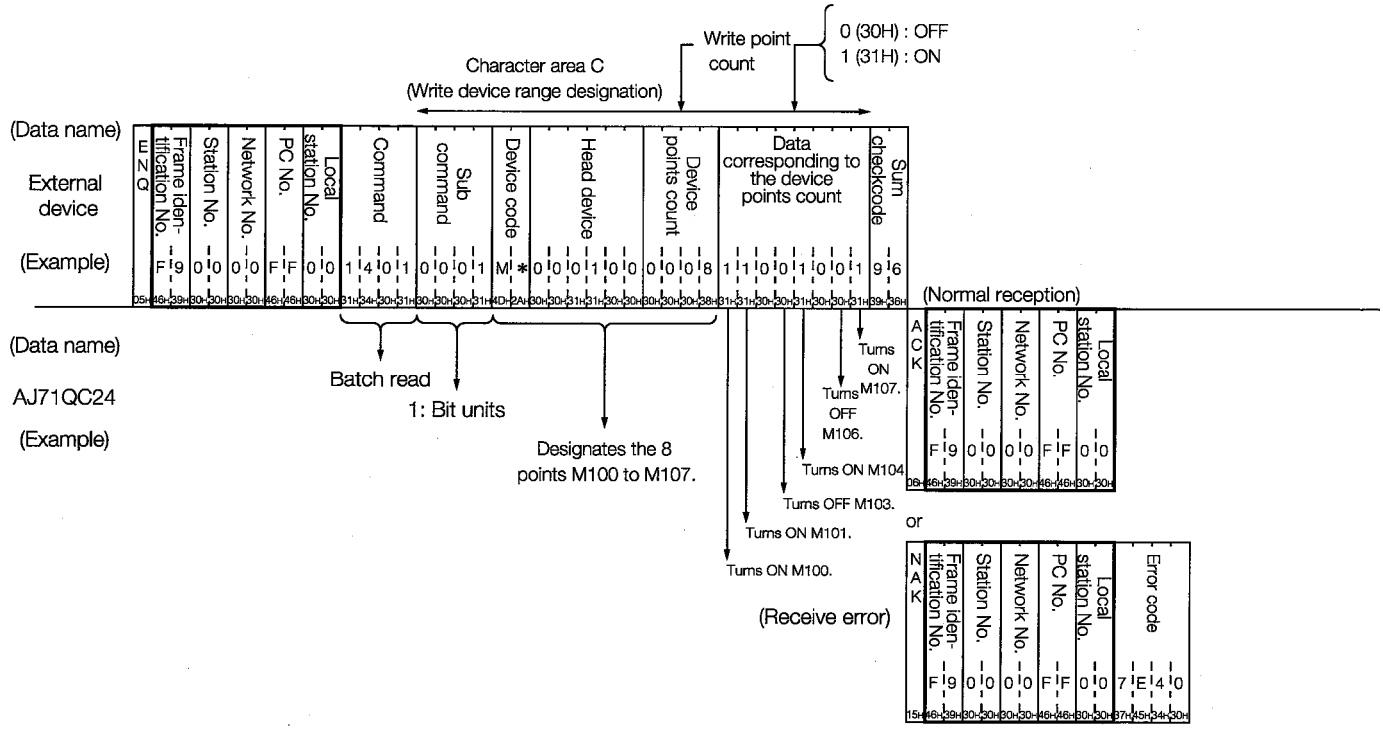
With the A compatible frame, the message wait time is designated in the message, but with the QnA (extension) frame, QC24 buffer memory address 11EH (message wait time designation) designates the message wait time.



*1 A response watchdog timer (timer 1) is available for message wait time time-out check.

(b) Bit units write

(Example) Use command 1401 (device memory batch write) to write to PC CPU internal relays M100 to M107.



Note

(1) The contents of 'character area A', 'character area B', and 'character area C' in the figure above depend on the contents of the command. Refer to the command descriptions in the User's Manual for more information.

(2) Transmission and reception of the sum check code can be omitted with QC24 transmission specifications switch SW06.

In this guidebook, transmission and reception of the sum check code are omitted by setting transmission specifications switch SW06 to OFF.

4.1.2 Contents of data setting items

The following describes the contents of each data name designated in the QnA frame control procedure.

1 Control codes

Control Code	Code (hexadecimal)	Contents	Application
STX	02H	Start of Text	Indicates the start of the send data.
ETX	03H	End of Text	Indicates the end of the send data.
ENQ	05H	Enquiry	Indicates the start of the send data.
ACK	06H	Acknowledge	Response to the opposite device when data communications ended normally.
NAK	15H	Negative Acknowledge	Response to the opposite device when data communications ended abnormally.
EOT	04H	End of Transmission	Initializes the data communications transmission sequence and makes the QC24 ready to receive commands from an external device.
CL	0CH	Clear	
CR	0DH	Carriage Return	Indicates the end of the send data. (It is used with format 4.)
LF	0AH	Line Feed	

2 QnA frame No. ("F9")

The external device uses QnA frame No. to discriminate between QnA frames and QnA extension frames.
When QnA frames are used, "F9" is set and when QnA extension frames is used, "F8" is set.

3 Station No. ("00")

The external device uses the station No. to identify the QC24 that is to be accessed.
In this guidebook, "00" is set in the station No. switches on the front of the QC24.

4 Network No. ("00")

Network No. is the MELSECNET/10 network No. It is set when the QC24 is accessed over a MELSECNET/10 or MELSECNET II (MELSECNET/B) system.
In this guidebook, "00" is set to communicated with the connected stations.

5 PC No. ("FF")

PC No. identifies which PC CPU on the MELSECNET is to be accessed through the PC CPU connected to the external device.
In this guidebook, "FF" is set because data is sent and received with PC CPUs connected to an external device.

6 Local station No. ("00")

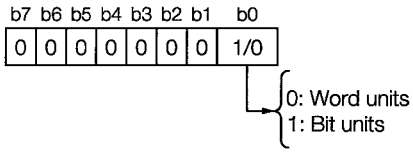
The local station No. is fixed at "00".

7 Command

Command designates which contents are to be accessed when an external device read data from, or write data to, a PC CPU.
In this guidebook, QnA frame format 1 is used to batch read/write device memory. Therefore, commands "0401/1401" are set.

8 Subcommand

Subcommand designates the read/write units, kind of device designated, and the data read conditions.
In this guidebook, the read and write operations are performed in word units or bit units. Therefore, the subcommand is set as shown below.

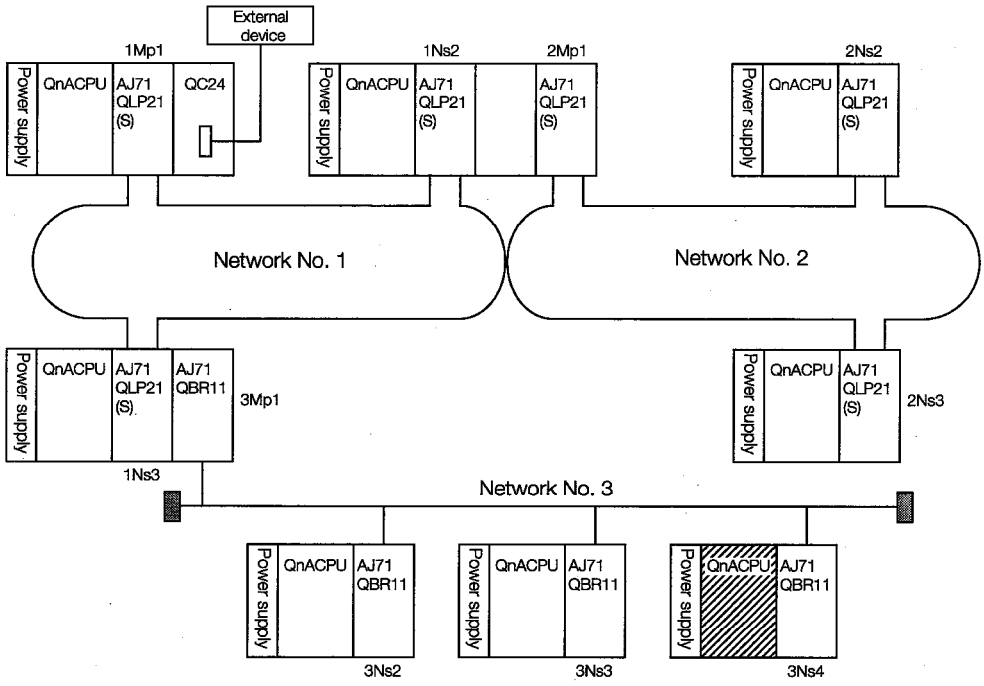


9 Error No.

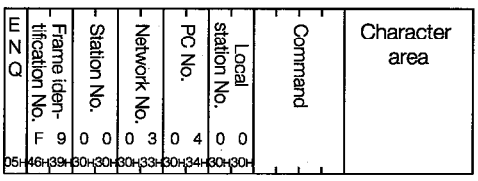
The error No. shows the error contents when the opposite device sent a NAK response.

If two or more errors are generated at the same time, the error No. detected first is sent.

For instance, the following shows an example of connection of three networks.



The following shows the settings when an external device uses QnA frame format 1 to access the QnACPU indicated by ▨.



Note

Sum check code

When a dedicated protocol format is used to send and receive data, the reliability of the send/receive data can be increased by sending and receiving the sum check code shown below by setting switch SW06 (sum check code enable/disable setting) on the AJ71QC24 to ON.

If SW06 is set to ON, the AJ71QC24,

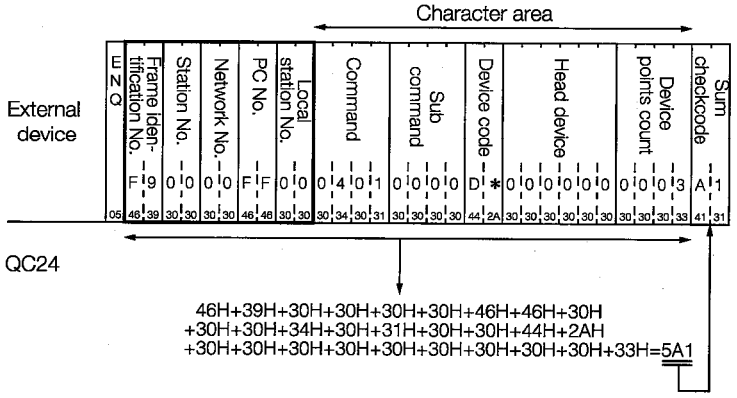
① adds a sum check code to the send data and sends the data,

② checks the sum check code of the receive data.

In this guidebook, data is transmitted by setting SW06 to OFF. Therefore, the sum check code is not handled.

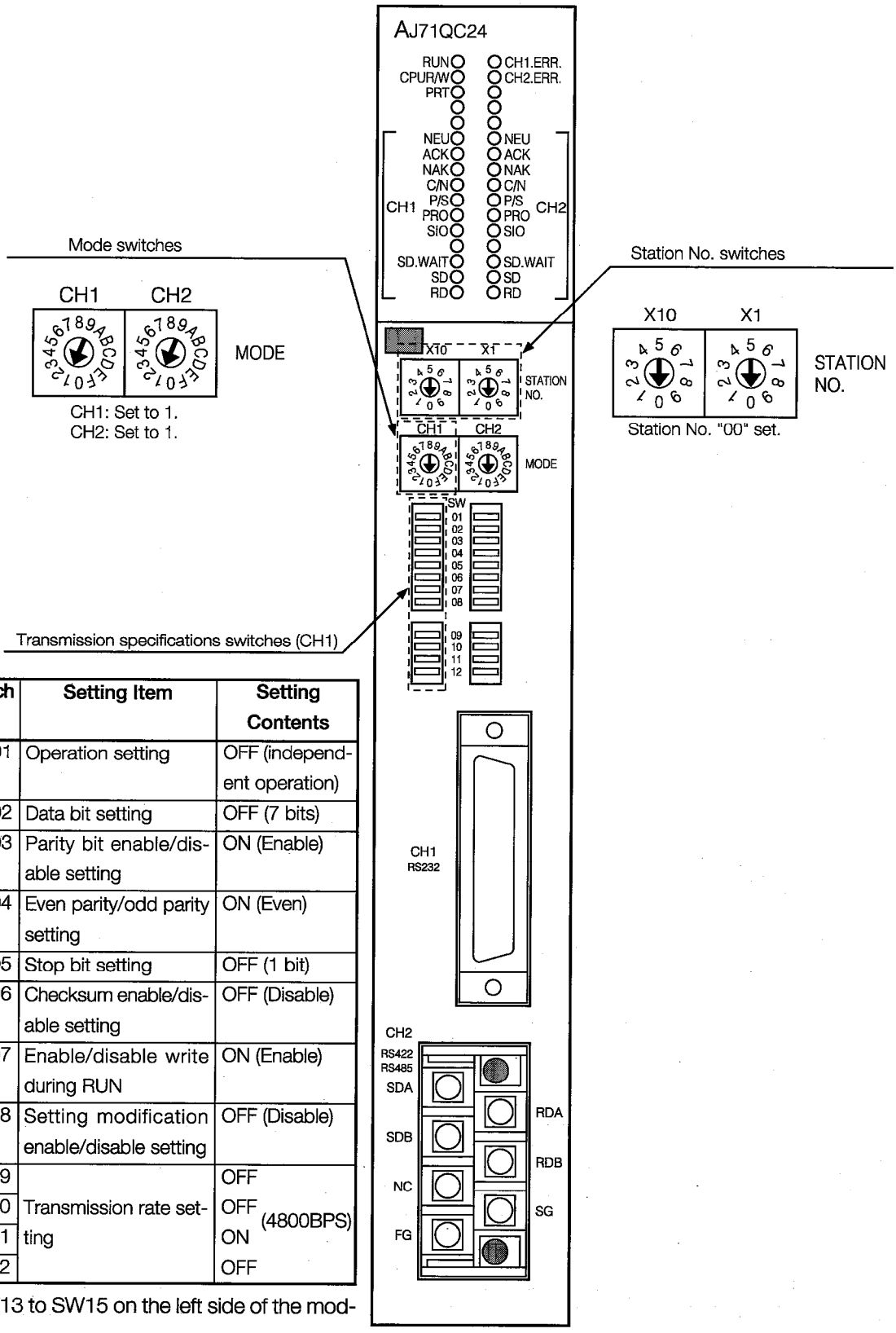
The following uses an example to describe the contents of the sum check code for your reference.

[Ex.] The following shows the sum check code value when the data of three points, beginning from D0, is read for QnA frame format 1 station No. 0, network No. 00, PC No. FF, local station No. 00 command 0401 (device memory word units batch read).



4.2 AJ71QC24 Serial Communications Module Settings

The following shows the setting of each AJ71QC24 switch.



Switch Setting	Switch	Setting Item	Setting Contents
	SW01	Operation setting	OFF (independent operation)
	SW02	Data bit setting	OFF (7 bits)
	SW03	Parity bit enable/disable setting	ON (Enable)
	SW04	Even parity/odd parity setting	ON (Even)
	SW05	Stop bit setting	OFF (1 bit)
	SW06	Checksum enable/disable setting	OFF (Disable)
	SW07	Enable/disable write during RUN	ON (Enable)
	SW08	Setting modification enable/disable setting	OFF (Disable)
	SW09	Transmission rate setting	OFF
	SW10		OFF
	SW11		ON (4800BPS)
	SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

4.3 Dedicated Protocol LED Display Contents

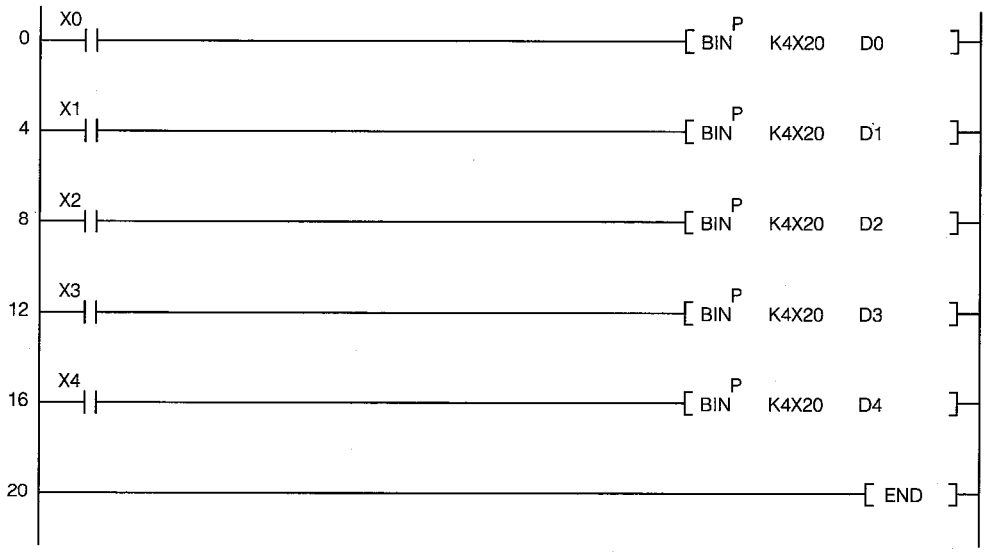
The following shows the display contents of the AJ71QC24 LEDs related to dedicated protocol data communications with the system configuration described in Section 3.1.1.

LED Name	LED Applications	ON	OFF	Error Processing
RUN	QC24 operation status display	Module normal	Module abnormal	• Check SW09~SW12. • Check the number set at the mode switches.
CPU R/W	QC24 and PC CPU communications status	Blinks while communicating (Lights steadily while communications stopped)	Abnormal	• Check the status of the PC CPU and external device. • Check the number set at the mode switches.
1-NEU	Displays the QC24 processing status for a request from the external device.	Waiting for request from external device	Processing a request from external device	If the display does not change even when the external device sends a command, check the following. • Cable connections • Number set at mode switches
1-ACK		Preceding request ended normally	Preceding request ended abnormally	
1-NAK		Preceding request ended abnormally	Preceding request ended normally	
1-C/N	QC24 and PC CPU communications status display	Communications abnormal	Communications normal	• Check SW07 (Enable/disable write during RUN)
1-P/S	Parity and checksum error display	Error	Normal	Receive data and transmission specifications settings are not the same. • Make the QC24 transmission specifications and the external device transmission specifications the same. • Check the transmission specifications switches.
1-PRO	Protocol status display	Error	Normal	Mode setting and receive data format are not the same. • Check the number set at the mode switches. • Check the external device send message.
1-SIO	Data reception status display	Error	Normal	Transmission specifications settings and receive data are not the same. • Make the QC24 transmission specifications and the external device transmission specifications the same. • Check the transmission specifications switches. • Decrease the transmission rate.
1-SD · WAIT	Shows the status of the data sent to the external device from the QC24	Waiting to send	Sending No send data	———
1-SD	QC24 to external device send status	Blinks while data is being sent	Not sending data	• If this LED does not blink even when the external device is sending a command, check the cable connections.
1-RD	QC24 from external device receiving status	Blinks while data is being received	Data not being received	
CH1. ERR	CH1 communications status display	CH1 error	Normal	———

☐ : Normal state. If an error is generated, an error LED will come ON. Thereafter, the error LED will remain ON even if the error recovers. Use the buffer memory (201H) to turn OFF the error LED.

4.4 Sequence Program

The following uses the GPPQ to writes a sequence program to store the device data (for external device monitoring) external device that is read by an external device from the PC CPU and writes this program to the PC CPU.



The program shown above is the control program for the system configuration shown in Section 3.1.1. It is not for communications with external devices.

4.5 Reading From PC CPU Word Device Memory (Word Units Batch Read)

This section reads the contents of PC CPU registers D0 to D4 (5 words) to the external device in word units and displays them on the CRT screen.

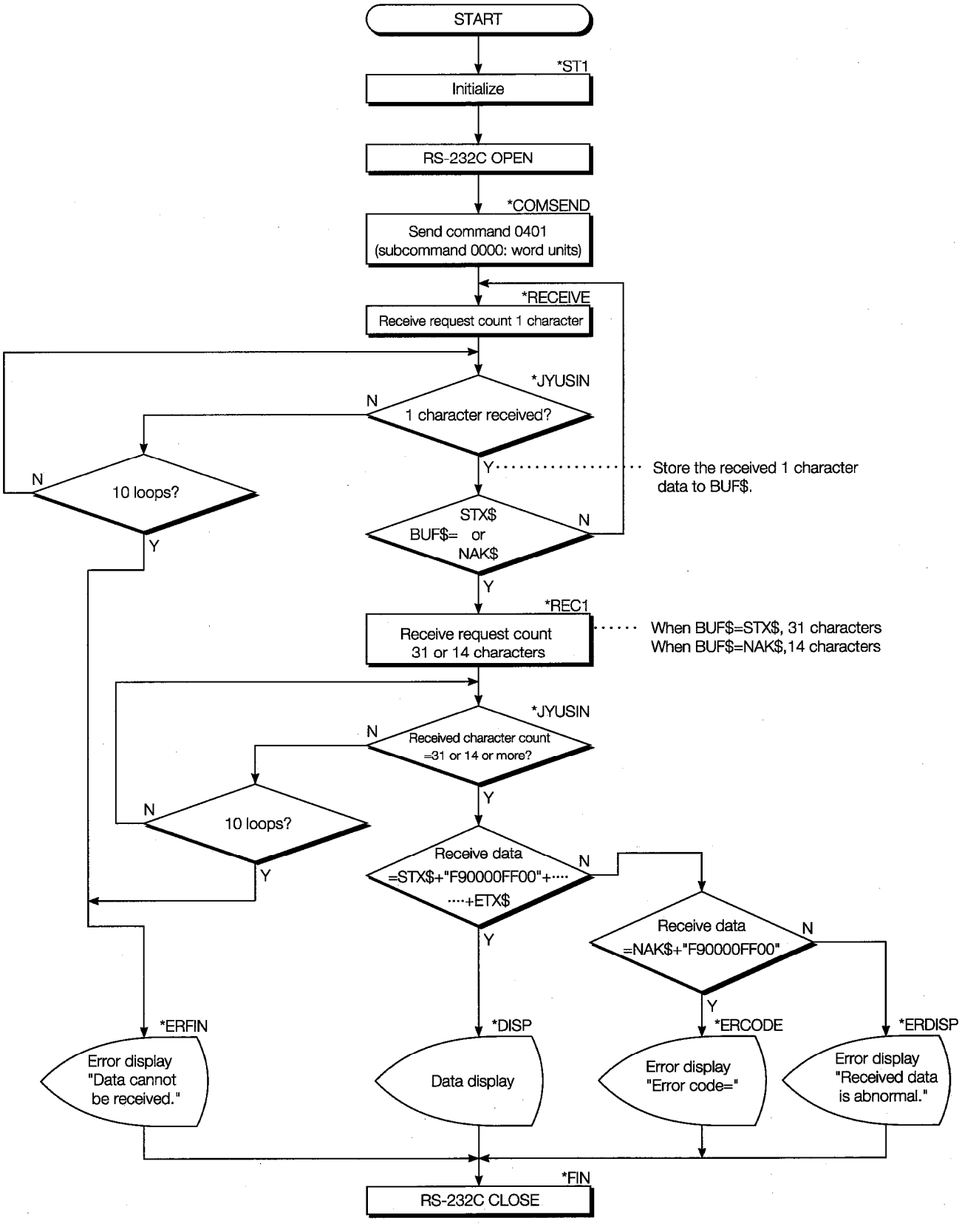
4.5.1 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                      AJ71QC24 Command 0401 Sample Program                     !
1020 ' !                                ASCII Mode                                     !
1030 ' !                  (Registers D0-D4 (current value) batch read)                 !
1040 ' ! ------------------------------------------------------------------------------!
1050 *ST1
1060     CLS                                :'Clear screen
1070     WTCNT%=10                          :'Data receive wait retry counter
1080     DLCNT%=1000                        :'Counter for data receive time adjustment
1090     STCNT%=32                          :'Receive data length when STX received
1100     NACNT%=15                          :'Receive data length when NAK received
1110     ERFLG%=0                           :'Error flag storage at end of reception
1120     RVCNT%=0                           :'Receive request data count storage
1130     CH%   =1                           :'Channel No.
1140     STX$  =CHR$(&H2)                   :'STX code
1150     ETX$  =CHR$(&H3)                   :'ETX code
1160     ENQ$  =CHR$(&H5)                   :'ENQ code
1170     NAK$  =CHR$(&H15)                  :'NAK code
1180     '
1190 ' ! ------------------------- RS-232C open & initialize ---------------------------!
1200     OPEN "COM:E71NN" AS #CH%           :'Set communications mode, etc.
1210     '
1220 ' ! ------------------------------ Send command 0401 -------------------------------!
1230 *COMSEND
1240     PRINT #CH%,ENQ$;"F90000FF0004010000D*0000000005"
1250     '
1260 ' ! ----------------------Receive current value of D0-D4 -------------------------!
1270 *RECEIVE
1280     RVCNT%=1 : GOSUB *JYUSIN           :'1 character receive request
1290     IF ERFLG%=99 THEN *ERFIN           :'Not received
1300     BUF$=RCV$
1310     IF (BUF$=STX$ OR BUF$= NAK$)THEN *REC1 ELSE *RECEIVE
1320 *REC1
1330     IF BUF$=STX$ THEN RVCNT%=STCNT%-1 :'Set receive request character count to 31.
1340     IF BUF$=NAK$ THEN RVCNT%=NACNT%-1 :'Set receive request character count to 14.
1350     GOSUB *JYUSIN                      :'Receive request
1360     IF ERFLG%=99 THEN *ERFIN           :'Not received
1370     BUF$=BUF$+RCV$                     :'Storage all receive data to BUF$.
1380     '
1390     IF LEFT$(BUF$,11)=STX$+"F90000F00" AND RIGHT$(BUF$,1)=ETX$ THEN *DISP
1400     IF LEFT$(BUF$,11)=NAK$+"F90000F00" THEN *ERCODE ELSE *ERDISP
1410     '
1420 ' ! --------------------------- Display receive data. ----------------------------!
1430 *DISP
1440     LOCATE 27,4 : PRINT "Word units batch read normal end"
1450     D$=MID$(BUF$,12,20)
1460     LOCATE 27, 8 : PRINT "D$=";D$
1470     LOCATE  7,12 : PRINT "  D0-D4 current value list"
1480     FOR I%=1 TO 5
1490       DAT%=VAL("&H"+MID$(D$,(I%-1)*4+1,4)) :'Data (4 hexadecimal digits)
                                                  → decimal conversion
1500         DNO%=I%-1                           :'Data register No.
1510         LOCATE 15,14+I%
1520         PRINT USING"D#=####";DNO%,DAT%
```

```
1530      NEXT I%
1540      GOTO *FIN
1550 ' ! ------------------- Data display when error data received----------------------!
1560 *ERFIN
1570      LOCATE 25,10 :PRINT "Data cannot be received. Stop."
1580      GOTO *FIN
1590      '
1600 *ERCODE
1610      LOCATE 27,4 : PRINT "Batch read abnormal end"
1620      ERCD$=MID$(BUF$,12,4)
1630      LOCATE 27,8 : PRINT "Error code=";ERCD$
1640      GOTO *FIN
1650      '
1660 *ERDISP
1670      LOCATE 25,10 :PRINT "Received data is abnormal. Stop."
1680      GOTO *FIN
1690      '
1700 ' ! ------------------------------- RS-232C close---------------------------------!
1710 *FIN
1720      CLOSE #CH%
1730      '
1740 END
1750 '
1760 ' ! ------------------------- Data reception subroutine---------------------------!
1770 *JYUSIN
1780      FOR I%=1 TO WTCNT%
1790        FOR J%=1 TO DLCNT%              :'Data receive wait
1800        NEXT J%
1810        IF LOC(CH%) → RVCNT% THEN *BUFIN :'Jump if designated number of characters,
                                                                    or more, received.
1820      NEXT I%
1830        ERFLG%=99 :RETURN               :'Data not received error
1840      '
1850 *BUFIN
1860      RCV$=INPUT$(RVCNT%,CH%)           :'Read receive data
1870      RETURN
1880      '
```

* BASIC commands must be changed according to BASIC software.

Batch read from register D0 to register D4



*□□□: Label name used with the program

4.5.2 Trainer operation

Write the sequence program shown in Section 4.4 to the PC CPU and the BASIC program shown in Section 4.5.1 to the external device.

After writing, set the PC CPU to "RUN".

If there are no errors, send and receive data using the following procedure.

① Set an arbitrary value in digital switches X20 to X2F of the trainer.

② When X0 is turned ON, the value set in the digital switches is stored to data register D0.

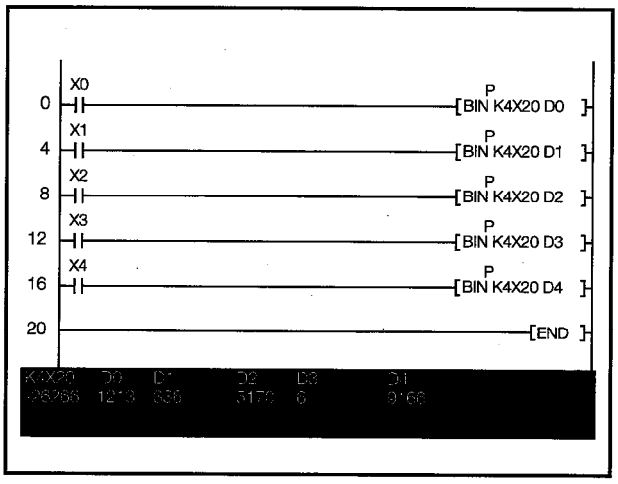
Similarly, X1 stores data to D1, X2 stores data to D2, X3 stores data to D3, and X4 stores data to D4.

(The sequence program shown in Section 4.4 controls this storage.)

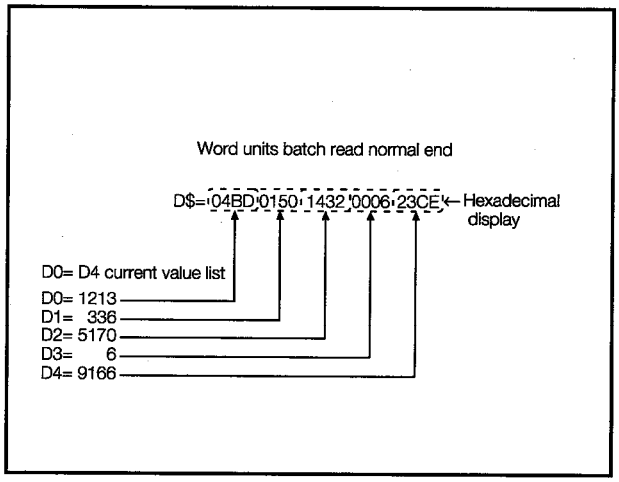
③ Check while monitoring D0 to D4 at the GPPQ circuit monitor screen.

④ When the BASIC program is "RUN", the contents of PC data registers D0 to D4 are normally read and displayed on the CRT.

Check this.



GPPQ circuit monitor screen



External device CRT screen

If you have any situations as follows, please read the explanations

- A batch read request from an external device is executed, but

Data cannot be received. Stop.

- Was the sequence program (Section 4.4) written to the PC CPU?
- Are the cables correctly connected (Section 3.3)?
- Are the external device settings (Section 3.4) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.

- Batch read ended, but

Batch read abnormal end. Error code=□

- NAK was received from the QC24.

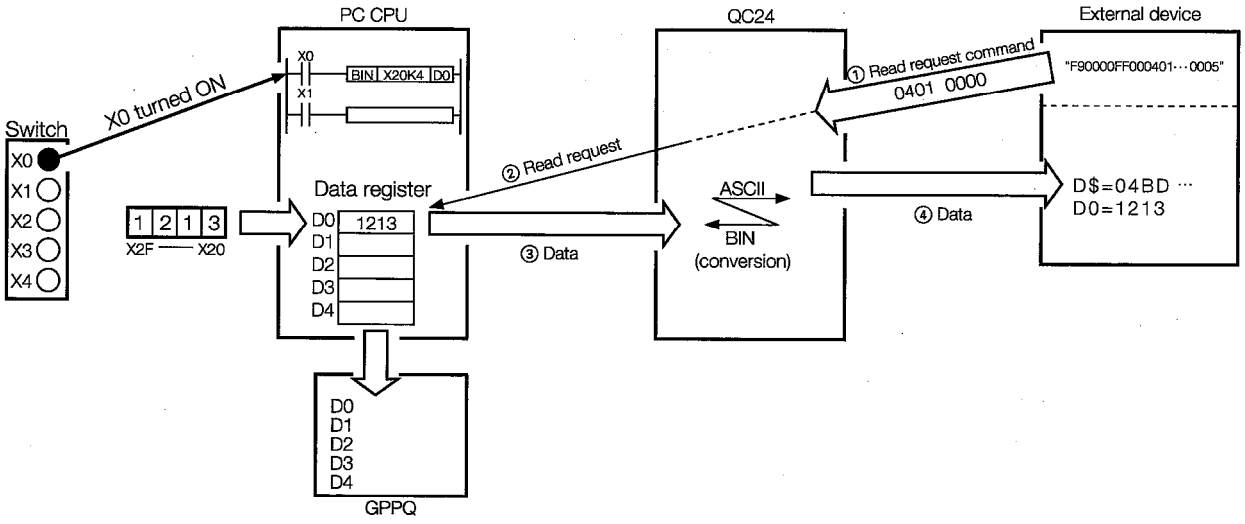
  Are the external program (RS-232C setting, read request command) settings (Section 4.5.1) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3). then check the settings.
- Refer to the Serial Communications Module User's Manual for the contents of the displayed error code and take appropriate action.

After resetting the settings, reset the PC CPU and communicate again.

4.5.3 Summary of device memory word units batch read

The following uses an image diagram to outline the communications procedure for the device memory word units batch read described in Section 4.5.2.

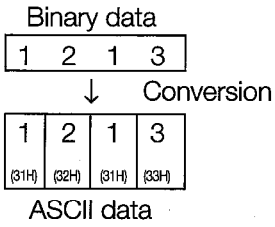


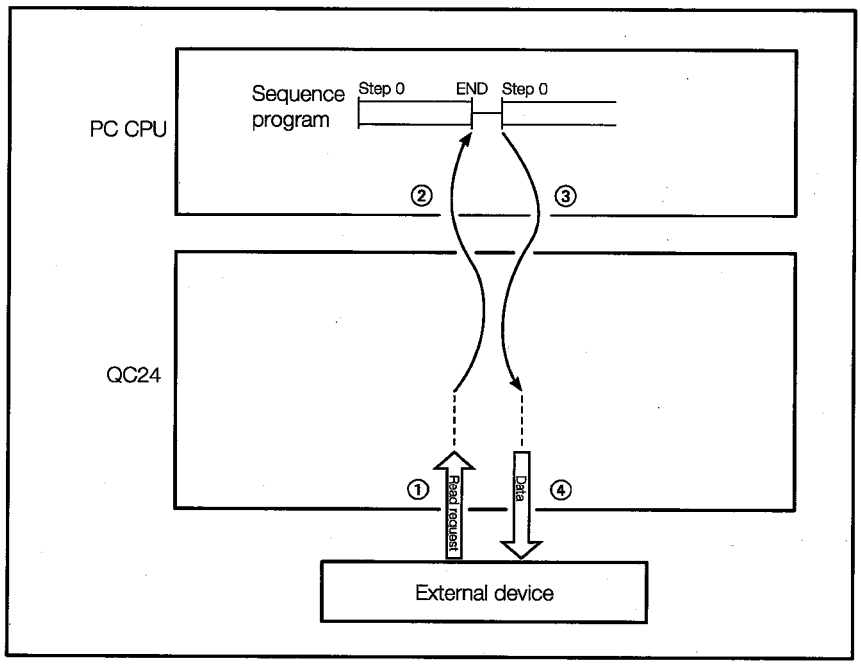
① The external device sends a read request (command: 0401, subcommand: 0000) to the QC24.

② When the QC24 receives this request, if the PC CPU is performing END processing, the QC24 sends a read request to the PC CPU.

③ The PC CPU passes the data to the QC24.

④ The QC24 converts the read data to ASCII data and sends the ASCII data to the external device.





4.6 Reading From PC CPU Bit Device Memory (Bit Units Batch Read)

This sections reads the ON/OFF state of PC CPU bit devices X0 to X7 (8 points) to the external device in bits units and displays the read state on the CRT screen.

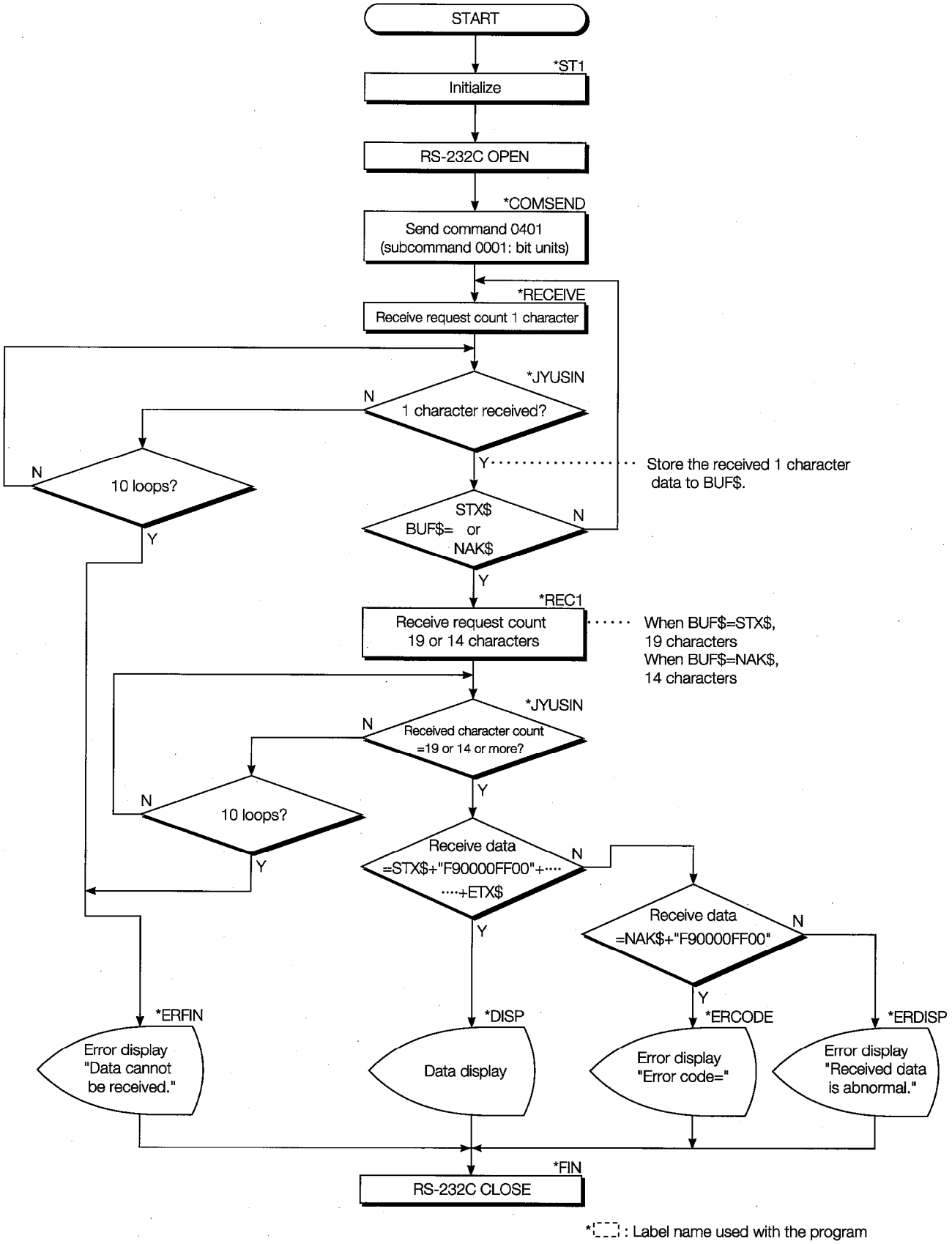
4.6.1 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                      AJ71QC24 Command 0401 Sample Program                     !
1020 ' !                                ASCII Mode                                     !
1030 ' !             (Batch read of ON/OFF state of bit devices X0 to X7)             !
1040 ' ! ------------------------------------------------------------------------------!
1050 *ST1
1060     CLS                                :'Clear screen
1070     WTCNT%=10                          :'Data receive wait retry counter
1080     DLCNT%=1000                        :'Counter for data receive wait time adjustment
1090     STCNT%=20                          :'Receive data length when STX received
1100     NACNT%=15                          :'Receive data length when NAK received
1110     ERFLG%=0                           :'Error flag storage at end of reception
1120     RVCNT%=0                           :'Receive request data count storage
1130     CH%   =1                           :'Channel No.
1140     STX$  =CHR$(&H2)                   :'STX code
1150     ETX$  =CHR$(&H3)                   :'ETX code
1160     ENQ$  =CHR$(&H5)                   :'ENQ code
1170     NAK$  =CHR$(&H15)                  :'NAK code
1180     '
1190 ' ! ------------------------ RS-232C open & initialize---------------------------!
1200     OPEN "COM:E71NN" AS #CH%           :'Set communications mode, etc.
1210     '
1220 ' ! ----------------------------- Send command 0401 -------------------------------!
1230 *COMSEND
1240     PRINT #CH%,ENQ$;"F90000FF0004010001X*0000000008"
1250     '
1260 ' ! ------------------------ Receive D0-D4 current value---------------------------!
1270 *RECEIVE
1280     RVCNT%=1 : GOSUB *JYUSIN           :'1 character receive request
1290     IF ERFLG%=99 THEN *ERFIN           :'Not received
1300     BUF$=RCV$
1310     IF (BUF$=STX$ OR BUF$=NAK$) THEN *REC1 ELSE *RECEIVE
1320 *REC1
1330     IF BUF$=STX$ THEN RVCNT%=STCNT%-1  :'Set receive request character count to 19.
1340     IF BUF$=NAK$ THEN RVCNT%=NACNT%-1  :"Set receive request character count to 14.
1350     GOSUB *JYUSIN                      :'Receive request
1360     IF ERFLG%=99 THEN *ERFIN           :'Not received
1370     BUF$=BUF$+RCV$                     :'Storage all receive data to BUF$.
1380     '
1390     IF LEFT$(BUF$,11)=STX$+"F90000FF00" AND RIGHT$(BUF$,1)=ETX$ THEN *DISP
1400     IF LEFT$(BUF$,11)=NAK$+"F90000FF00" THEN *ERCODE ELSE *ERDISP
1410     '
1420 ' ! ----------------------------Display receive data ----------------------------!
1430 *DISP
1440     LOCATE 27,4 : PRINT "Bit units batch read normal end"
1450     X$=MID$(BUF$,12,8)
1460     LOCATE 27, 8 : PRINT :X$=";X$
1470     LOCATE  7,12 : PRINT "X0-X7 ON/OFF state"
1480     FOR I%=1 TO 8
1490       IF MID$(X$,I%,1)="1" THEN DAT$= "ON"   :'When 1, ON
1500       IF MID$(X$,I%,1)="0" THEN DAT$="OFF"   :'When 0, OFF
1510       XNO%=I%-1                              :'X No.
1520       LOCATE 15,14+I%
1530       PRINT USING"X#=& &";XNO%,DAT$
```

```
1540      NEXT I%
1550      GOTO *FIN
1560 "! -------------------- Data display when error data received----------------------!
1570 *ERFIN
1580      LOCATE 25,10 : PRINT "Data cannot be received. Stop."
1590      GOTO *FIN
1600      '
1610 *ERCODE
1620      LOCATE 27,4 : PRINT "Batch read abnormal end"
1630      ERCD$=MID$(BUF$,12,4)
1640      LOCATE 27,8 : PRINT "Error code=";ERCD$
1650      GOTO *FIN
1660      '
1670 *ERDISP
1680      LOCATE 25,10 :PRINT "Receive data is abnormal. Stop."
1690      GOTO *FIN
1700      '
1710 '! ------------------------------- RS-232C close---------------------------------!
1720 *FIN
1730      CLOSE #CH%
1740      '
1750 END
1760 '
1770 ' ! ------------------------- Data reception subroutine----------------------------;
1780 *JYUSIN
1790      FOR I%=1 TO WTCNT%
1800        FOR J%=1 TO DLCNT%              :'Data receive wait
1810        NEXT J%
1820        IF LOC(CH%) → RVCNT% THEN *BUFIN  :'Jump if designated number of characters,
                                                                    or more, received.
1830      NEXT I%
1840      ERFLG%=99 :RETURN                 :'Data not received error
1850      '
1860 *BUFIN
1870      RCV$=INPUT$(RVCNT%,CH%)           :'Read receive data
1880      RETURN
1890      '
```

* BASIC commands must be changed according to BASIC software.

X0 to X7 batch read



4.6.2 Trainer operation

Write the BASIC program shown in Section 4.6.1 to the external device.

Set the PC CPU to "RUN".

If there are no errors, send and receive data using the following procedure.

① Set trainer switches X0 to X7 to ON (or OFF).

② When the BASIC program is "RUN", the PC X0 to X7 ON/OFF information is read in 8-point groups in bit units and displayed on the CRT screen for each device.

Check this.

```
Bit units batch read normal end
X$=00101100
X0-X7 ON/OFF state
X0=OFF
X1=OFF
X2=ON
X3=OFF
X4=ON
X5=ON
X6=OFF
X7=OFF
```

External device CRT screen

If you have any situations as follows, please read the explanations

- A batch read request from the external device is executed, but

```
Data cannot be received. Stop.
```

- Are the cables correctly connected (Section 3.3.)?
- Are the external device settings (Section 3.4) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.

- Batch read ended, but

```
Batch read abnormal end.
Error code=□
```

- NAK was sent from the QC24.

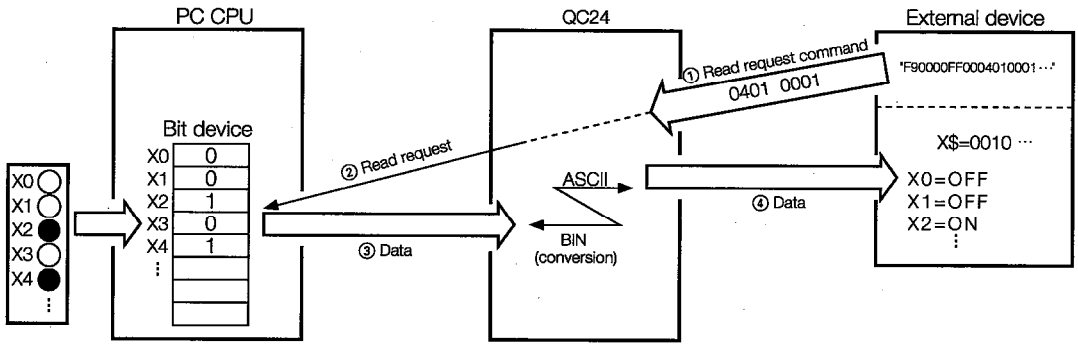
  Are the external program (RS-232C setting, read request command) settings (Section 4.6.1) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.

- Refer to the Serial Communications Module User's Manual for the contents of the displayed error code and take appropriate action.

4.6.3 Summary of device memory bit units batch read

The following uses an image diagram to outline the communications procedure for the device memory bit units batch read described in Section 4.6.2.

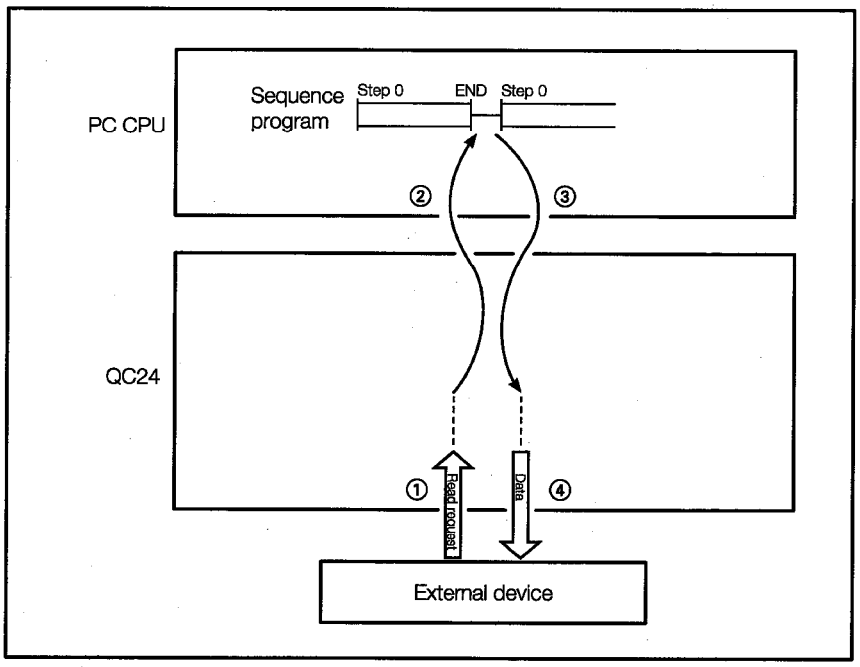


① The external device sends a read request (command: 0401, subcommand: 0001) to the QC24.

② When the QC24 receives this request, it sends a read request to the PC CPU while the PC CPU is performing END processing.

③ The PC CPU passes the data to the QC24.

④ The QC24 converts the read data to ASCII data and sends the ASCII data to the external device.



4.7 Writing to PC CPU Word Device Memory (Word Units Batch Write)

This section inputs numeric data from an external device and writes it to PC CPU data registers D0 to D4.

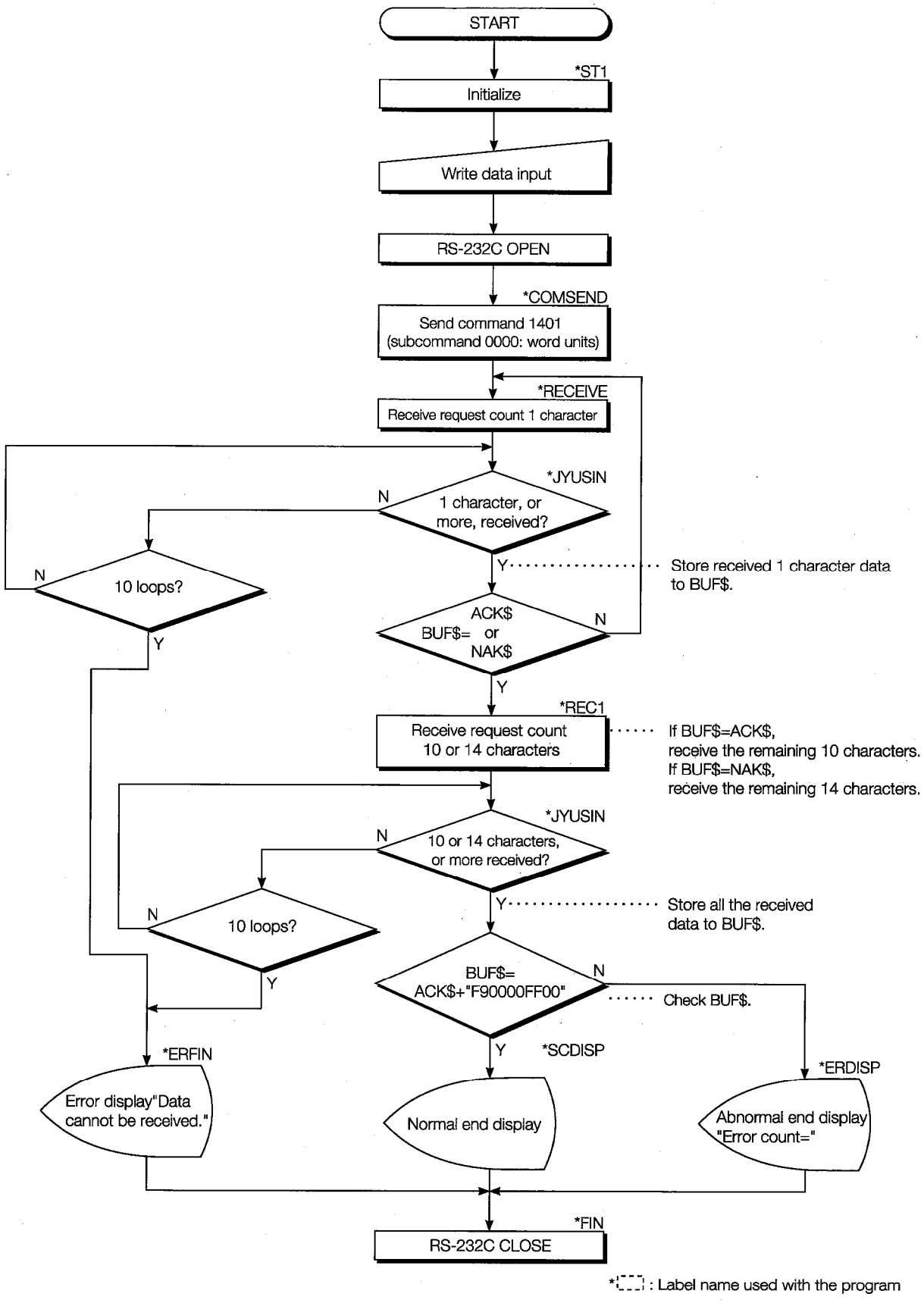
4.7.1 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                      AJ71QC24 Command 1401 Sample Program                     !
1020 ' !                                 ASCII Mode                                    !
1030 ' !                        (Batch write to registers D0-D4)                       !
1040 ' ! ------------------------------------------------------------------------------!
1050 *ST1
1060     CLS                                :'Clear screen
1070     WTCNT%=10                          :'Data receive wait retry counter
1080     DLCNT%=1000                        :'Counter for data receive wait time adjustment
1090     ACCNT%=11                          :'Receive data length when ACK received
1100     NACNT%=15                          :'Receive data length when NAK received
1110     ERFLG%=0                           :'Error flag storage at the end of reception
1120     RVCNT%=0                           :'Receive request data count storage
1130     CH%   =1                           :'Channel No.
1140     ENQ$  =CHR$(&H5)                   :'ENQ code
1150     ACK$  =CHR$(&H6)                   :'ACK code
1160     NAK$  =CHR$(&H15)                  :'NAK code
1170     '
1180 ' ! ------------------------------Write data input ---------------------------------!
1190     CLS
1200     LOCATE 10,10 : PRINT "Input in decimal."
1210     LOCATE 10,12 : INPUT "D0= " ; D(0)
1220     LOCATE 10,13 : INPUT "D1= " ; D(1)
1230     LOCATE 10,14 : INPUT "D2= " ; D(2)
1240     LOCATE 10,15 : INPUT "D3= " ; D(3)
1250     LOCATE 10,16 : INPUT "D4= " ; D(4)
1260     '
1270 ' ! -------------------- Write data > hexadecimal conversion-----------------------!
1280     D$=""
1290     FOR I%=0 TO 4
1300       D$=D$+RIGHT$("000"+HEX$(D(I%)),4)
1310     NEXT I%
1320     '
1330 ' ! ------------------------- RS-232C open & initialize-----------------------------!
1340     OPEN "COM:E71NN" AS #CH%           :'Set communications mode, etc.
1350     '
1360 ' ! ----------------------------- Send command 1401--------------------------------!
1370 *COMSEND
1380     PRINT #CH%,ENQ$;"F90000FF0014010000D*0000000005"+D$
1390     '
1400 ' ! --------------------------Receive response message ----------------------------!
1410 *RECEIVE
1420     RVCNT%=1 : GOSUB *JUYSIN           :'1 character receive request
1430     IF ERRFLG%=99 THEN *ERFIN          :'Not received
1440     BUF$=RCV$
1450     IF (BUF$=ACK$ OR BUF$=NAK$) THEN *REC1 ELSE *RECEIVE
1460     '
1470 *REC1
1480     IF BUF$=ACK$ THEN RVCNT%=ACCNT%-1  :'Set receive request character count to 10.
1490     IF BUF$=NAK$ THEN RVCNT%=NACNT%-1  :'Set receive request character count to 14.
1500     GOSUB *JYUSIN                      :'Receive request
1510     IF ERFLG%=99 THEN *ERFIN           :'Not received
1520     BUF$=BUF$+RCV$                     :'Storage all receive data to BUF$.
1530     '
1540     CHK$=ACK$+"F90000F00"              :'Normal end response message structure
```

```
1550     IF BUF$=CHK$ THEN *SCDISP ELSE *ERDISP          :'Response message judgment
1560     '
1570 ' ! ----------------------------Normal end display ------------------------------!
1580 *SCDISP
1590     LOCATE 20,20 : PRINT     "Batch write normal end"   : GOTO *FIN
1600     '
1610 ' ! ----------------------------Abnormal end display ----------------------------!
1620 *ERDISP
1630     LOCATE 20,20 : PRINT     "Batch write abnormal end"
1640     ERCD$=MID$(BUF$,12,4)
1650     LOCATE 27,22 : PRINT     "Error code=";ERCD$
1660     GOTO *FIN
1670     '
1680 *ERFIN
1690     LOCATE 20,20 : PRINT     "Response message cannot be received. Stop."
1700 GOTO *FIN
1710     '
1720 ' ! ------------------------------ RS-232C close---------------------------------!
1730 *FIN
1740     CLOSE #CH%
1750     '
1760 END
1770 '
1780 ' ! ------------------------- Data reception subroutine---------------------------!
1790 *JYUSIN
1800     FOR I%=1 TO WTCNT%
1810       FOR J%= 1 TO DLCNT%            :'Data receive wait
1820       NEXT J%
1830       IF LOC(CH%) → RVCNT% THEN *BUFIN  :'Jump if designated number of characters,
                                                                    or more, received.
1840     NEXT I%
1850     ERFLG%=99 : RETURN               :'Data not received error
1860 *BUFIN
1870     RCV$=INPUT$(RVCNT%,CH%)          :'Read receive data
1880     RETURN
1890     '
```

* BASIC commands must be changed according to BASIC software.

Batch write to register D0 to D4



4.7.2 Trainer operation

Write the sequence program shown in Section 4.4 to the PC CPU and the BASIC program shown in Section 4.7.1 to the external device.

After writing, set the PC CPU and external device to "RUN".

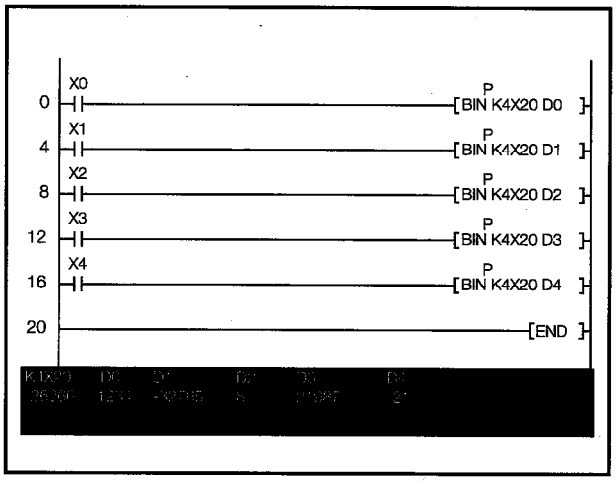
If there are no errors, send and receive data using the following procedure.

① Input numeric data (-32768 ~ 32767) to data registers D0 to D4 for each device from the external device keyboard.

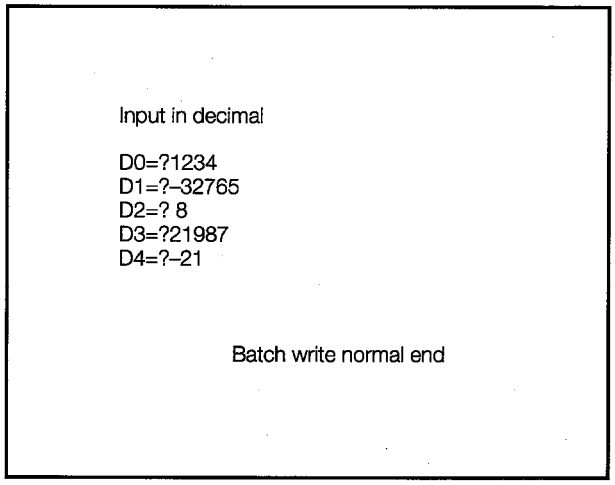
(Example: 1 2 3 4 ↵)

② At the end of input of all the numeric data to be written to D0 to D4, batch write the data to the PC.

③ While monitoring D0 to D4 with the GPPQ, check the written values.



GPPQ circuit monitor screen



External device screen

If you have any situations as follows, please read the explanations

- When data is written from an external device to the PC CPU

Response message cannot be received. Stop.

- Was a sequence program (Section 4.4) written to the PC CPU?
- Are the cables connected (Section 3.3) correctly?
- Are the external device settings (Section 3.4) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.

Batch read abnormal end Error code=

- NAK was sent from the QC24.

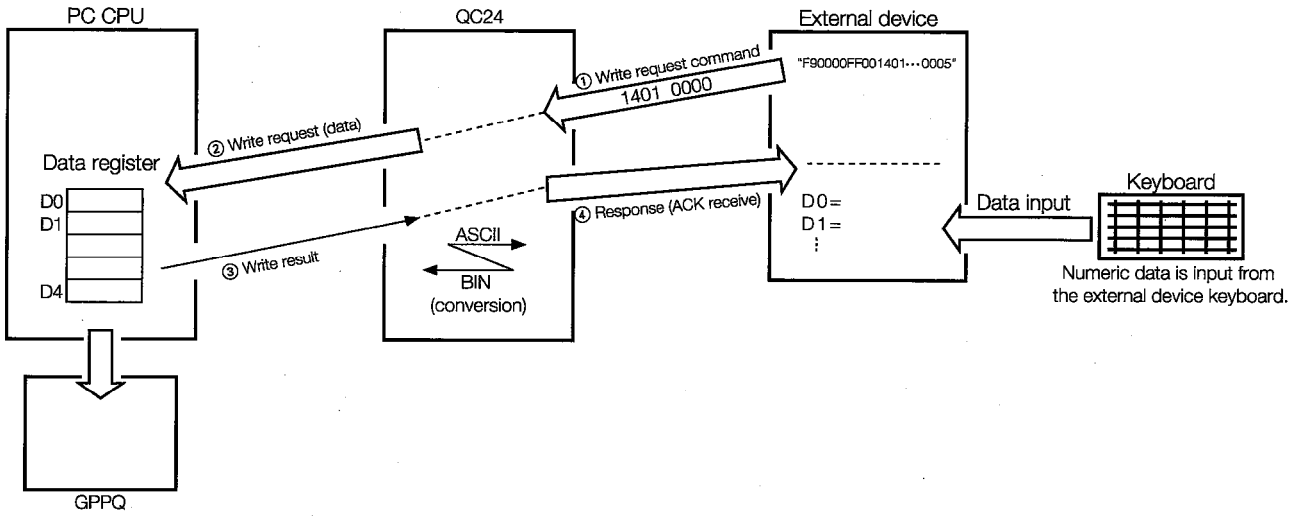
  Are the external device program (RS-232C setting, write request command) settings (Section 4.7.1) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.
- Refer to the Serial Communications Module User's Manual for the contents of the displayed error code and take appropriate action.

After resetting the settings, reset the PC CPU and communicate again.

4.7.3 Summary of word units batch write to device memory

The following uses an image diagram to outline the communications procedure for the word device batch write described in Section 4.7.2.

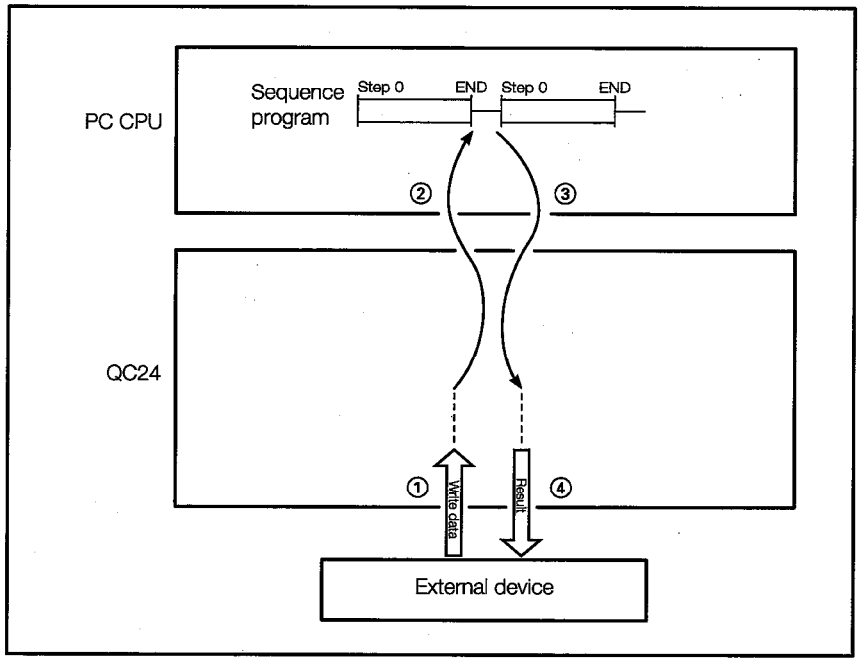


① The external device sends a write request (command: 1401, subcommand: 0000) to the QC24.

② When the QC24 receives this request, it sends a data write request to the PC CPU while the PC CPU is performing END processing.

③ After the PC CPU writes the data, it sends the result to the QC24.

④ The QC24 sends the write result to the external device.



4.8 Writing to PC CPU Bit Device Memory (Bit Units Batch Write)

This section inputs 0(OFF)/1(ON) from an external device and writes them to PC CPU bit devices Y70 to Y77 (8 points).

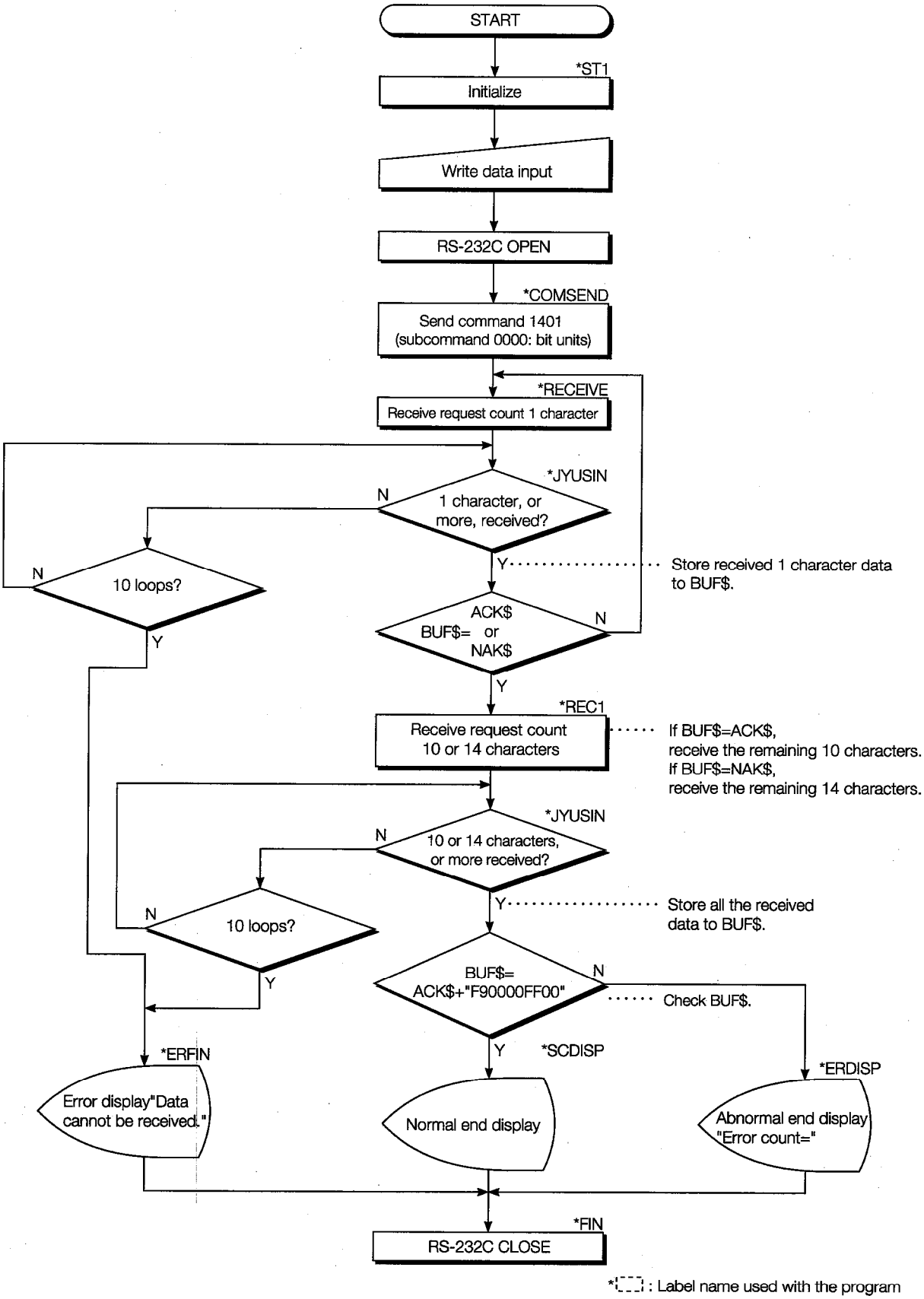
4.8.1 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                      AJ71QC24 Command 1401 Sample Program                     !
1020 ' !                                ASCII Mode                                     !
1030 ' !                    (Batch write to bit devices Y70 - Y77)                     !
1040 ' ! ------------------------------------------------------------------------------!
1050 *ST1
1060     CLS                                :'Clear screen
1070     WTCNT%=10                          :'Data receive wait retry counter
1080     DLCNT%=1000                        :'Counter for data receive wait time adjustment
1090     ACCNT%=11                          :'Receive data length when ACK received
1100     NACNT%=15                          :'Receive data length when NAK received
1110     ERFLG%=0                           :'Error flag storage at end of reception
1120     RVCNT%=0                           :'Receive request data count storage
1130     CH%   =1                           :'Channel No.
1140     ENQ$  =CHR$(&H5)                   :'ENQ code
1150     ACK$  =CHR$(&H6)                   :'ACK code
1160     NAK$  =CHR$(&H15)                  :'NAK code
1170     '
1180 ' ! ------------------------------Write data input ------------------------------!
1190     CLS
1200     LOCATE 10,10 : PRINT "0(OFF)/1(ON) data input"
1210     LOCATE 10,12 : INPUT "Y70= " ; Y70$
1220     LOCATE 10,13 : INPUT "Y71= " ; Y71$
1230     LOCATE 10,14 : INPUT "Y72= " ; Y72$
1240     LOCATE 10,15 : INPUT "Y73= " ; Y73$
1250     LOCATE 10,16 : INPUT "Y74= " ; Y73$
1260     LOCATE 10,17 : INPUT "Y75= " ; Y74$
1270     LOCATE 10,18 : INPUT "Y76= " ; Y76$
1280     LOCATE 10,19 : INPUT "Y77= " ; Y77$
1290     '
1300 ' ! ---------------------------------Write data ------------------------------------!
1310     Y$=Y70$+Y71$+Y72$+Y73$+Y74$+Y75$+Y76$+Y77$
1320     '
1330 ' ! ------------------------- RS-232C open & initialize----------------------------!
1340     OPEN "COM:E71NN" AS #CH%           :'Set communications mode, etc.
1350     '
1360 ' ! ----------------------------- Send command 1401 ------------------------------!
1370 *COMSEND
1380     PRINT #CH%,ENQ$;"F90000FF0014010001Y*0000700008"+Y$
1390     '
1400 ' ! -------------------------Receive response message ---------------------------!
1410 *RECEIVE
1420     RVCNT%=1 : GOSUB *JYUSIN           :'Receive 1 character request
1430     IF ERFLG%=99 THEN *ERFIN           :'Not received
1440     BUF$=RCV$
1450     IF (BUF$=ACK$ OR BUF$=NAK$) THEN *REC1 ELSE *RECEIVE
1460     '
1470 *REC1
1480     IF BUF$=ACK$ THEN RVCNT%=ACCNT%-1    :'Set receive request character count to 10.
1490     IF BUF$=NAK$ THEN RVCNT%=NACNT%-1    :'Set receive request character count to 14.
1500     GOSUB *JYUSIN                      :'Receive request
1510     IF ERFLG%=99 THEN *ERFIN           :'Not received
1520     BUF$=BUF$+RCV$                     :'Storage all receive data to BUF$.
1530     CHK$=ACK$+"F90000FF00"             :'Normal end response message structure
```

```
1540     IF BUF$=CHK$ THEN *SCDISP ELSE *ERDISP          :'Response message judgment
1550     '
1560 ' ! ------------------------------Normal end display ------------------------------!
1570 *SCDISP
1580     LOCATE 20,20 : PRINT "Batch write normal end"    : GOTO *FIN
1590     '
1600 ' ! ------------------------------Abnormal end display ----------------------------!
1610 *ERDISP
1620     LOCATE 20,20 : PRINT "Batch write abnormal end"
1630     ERCD$=MID$(BUF$,12,4)
1640     LOCATE 27,22 : PRINT "Error code=";ERCD$
1650     GOTO *FIN
1660     '
1670 *ERFIN
1680     LOCATE 20,20 : PRINT "Response message cannot be received. Stop."
1690     GOTO *FIN
1700     '
1710 ' ! ------------------------------- RS-232C close-----------------------------------!
1720 *FIN
1730     CLOSE #CH%
1740     '
1750 END
1760 '
1770 ' ! ------------------------- Data reception subroutine----------------------------!
1780 *JYUSIN
1790     FOR I%= 1 TO WTCNT%
1800       FOR J%=1 TO DLCNT%                    :'Data receive wait
1810       NEXT J%
1820       IF LOC(CH%) → RVCNT% THEN *BUFIN     :'Jump if designated number of
                                                          characters, or more, received.
1830     NEXT I%
1840     ERFLG%=99 : RETURN                      :'Data not received error
1850 *BUFIN
1860     RCV$=INPUT$(RVCNT%,CH%)                 :'Read receive data
1880     '
```

* BASIC commands must be changed according to BASIC software.

Batch write to Y70-Y77



4.8.2 Trainer operation

Write the BASIC program shown in Section 4.8.1 to the external device.

After writing, set the PC CPU to "RUN".

If there are no errors, send and receive data using the following procedure.

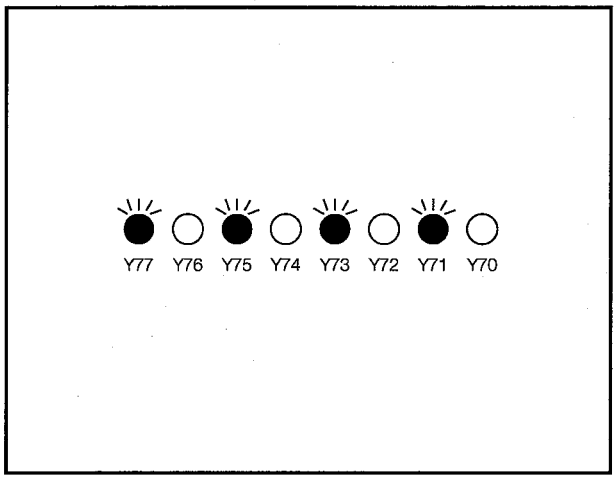
① Input all 8 points for each device from the external device keyboard.

ON : 1 (1 ↵)

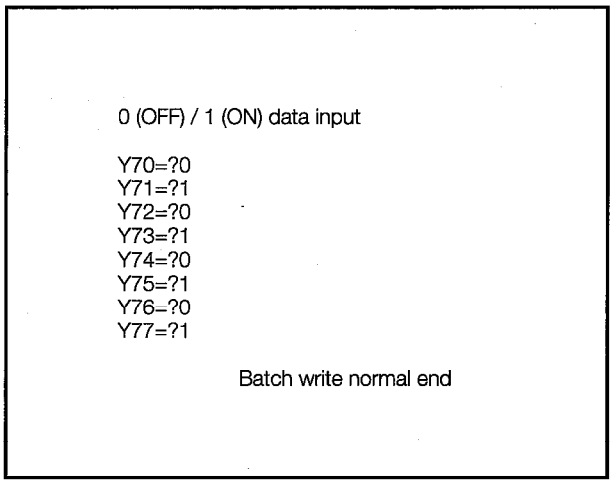
OFF : 0 (0 ↵)

② At the end of input, batch write the data to the PC.

③ Check if trainer output LEDs Y70 to Y77 are ON or OFF.



Trainer I/O panel



External device screen

If you have any situations as follows, please read the explanations

- When the write data is sent from the external device to the PC CPU

Response message cannot be received. STOP.

- Are the cables connected (Section 3.3) correctly?
- Are the external device settings (Section 3.4) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.

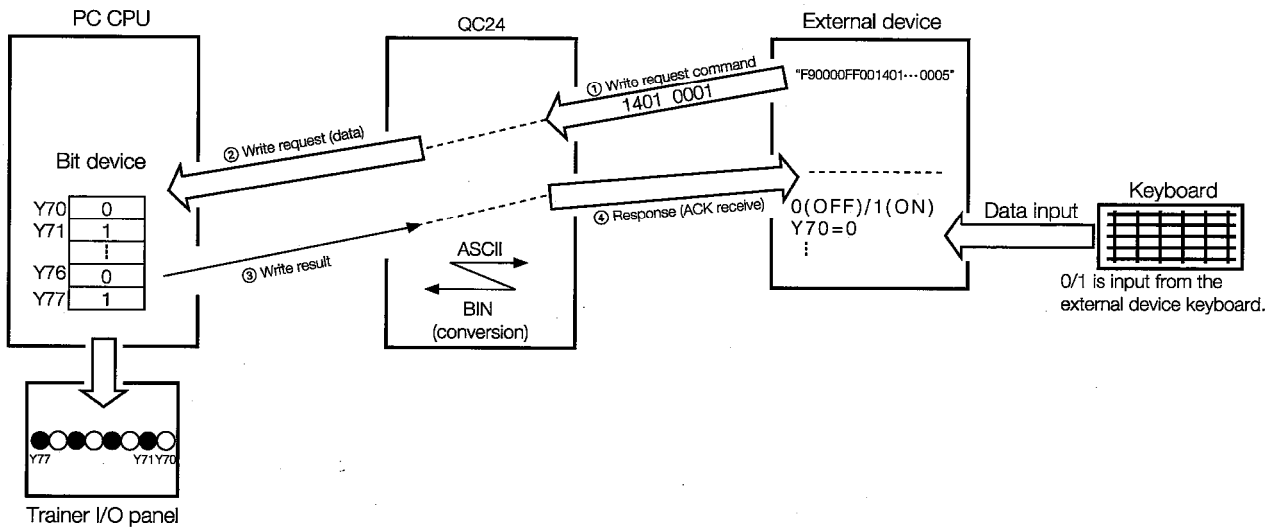
Batch write abnormal end Error code=

- NAK was sent from the QC24.
- Was data other than 0/1 input?
- Are the external device program (RS-232C setting, write request command) settings (Section 4.8.1) and QC24 switch settings (Section 4.2) correct?

  Check the QC24 LEDs (Section 4.3), then check the settings.
- Refer to the Serial Communications Module User's Manual for the contents of the displayed error code and take appropriate action.

4.8.3 Summary of bit units batch write to device memory

The following uses an image diagram to outline the communications procedure for the bit device batch write described in Section 4.8.2.

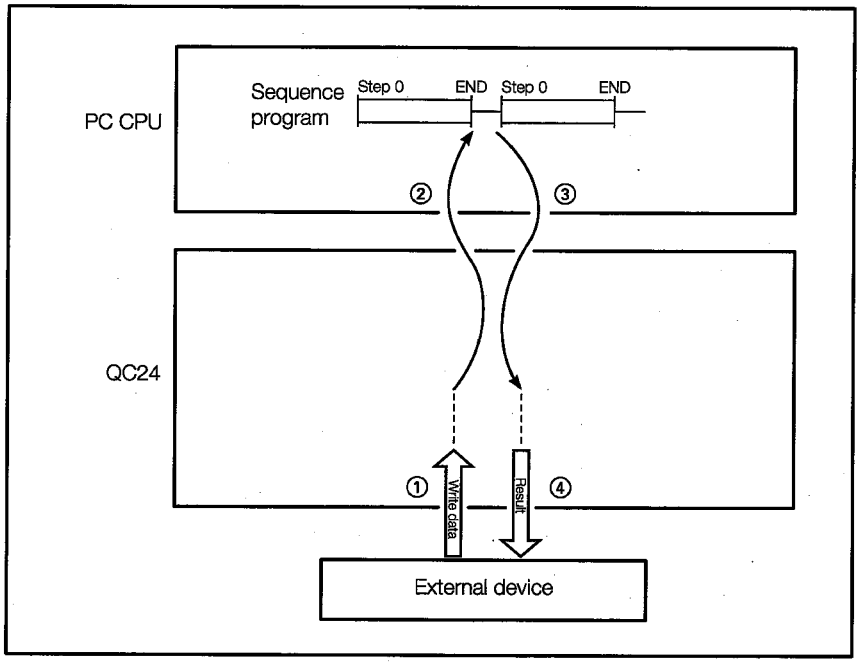


① The external device sends a write request (command: 1401, subcommand: 0001) to the QC24.

② When the QC24 receives this request, it sends a data write request to the PC CPU while the PC CPU is performing END processing.

③ After the PC CPU writes the data, it sends the result to the QC24.

④ The QC24 sends the write result to the external device.



4.9 On-Demand Function

4.9.1 What is the on-demand function?

During dedicated protocol data communications, usually only the external device starts data transmission.

However, when the PC CPU wants to send emergency data, etc. to the external device, it can start transmission. This function is called the "on-demand function".

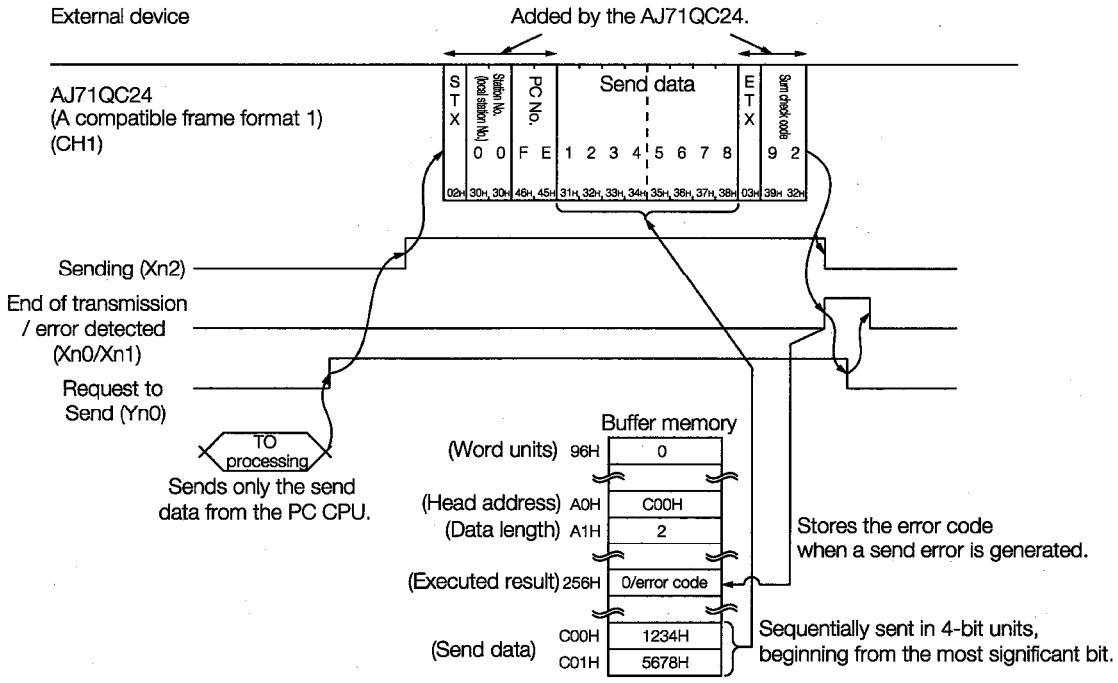
The on-demand function can be used when the external device and PC CPU configuration is 1:1.

4.9.2 On-demand function control procedure and message structure

In the ASCII mode, the on-demand function uses A compatible frames (formats 1~4) and in the binary mode, the on-demand function uses an QnA extension frame (format 5) as the data send frame.

The A compatible frame format can be selected with the QC24 mode switch.

The following describes the control procedure that uses the CH1 on-demand function in the ASCII mode (A compatible frame format 1).



Note

When A compatible frame transmission control procedure format 2 is used, the block No. becomes "00".

The header and trailer added by the QC24 can be changed to data registered in a user frame. (Buffer memory addresses A9H to ACH designate the user frame No.)

4.9.3 On-demand function setting items

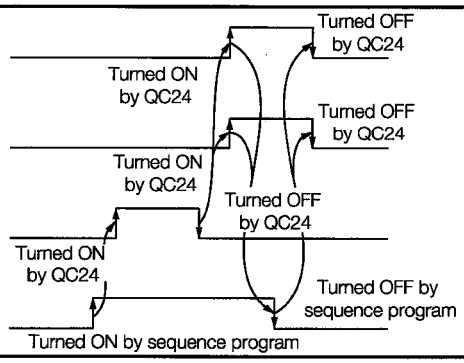
The following describes the items which are set when the on- demand function is used in data communications.

• Buffer memories used by on-demand function

Address	Name	Description
96H	Word/byte designation	Designates the send data length units.
A0H	On-demand buffer memory head address designation area	Uses the sequence program TO instruction to designate the head address of the buffer memory storing the data to be sent by the on-demand function.
A1H	On-demand data length designation area	Uses the sequence program TO instruction to designate the data length to be sent by the on-demand function.
256H	On-demand error storage area	When the on-demand function generates a data send error, the AC71QC24 writes the error code to this area. 0: No error Nonzero: Error
400H to 12FFH	User area	The user data sent to the external device is written to this area.

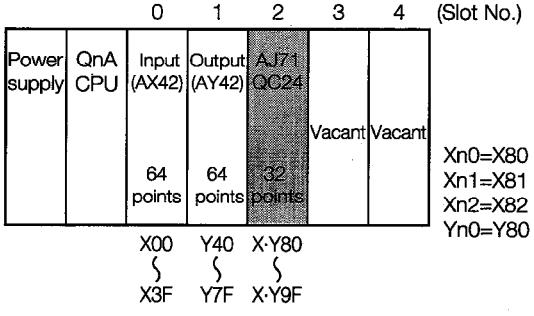
• On-demand handshake signals

The on-demand handshake signals are signals that are turned ON when the AJ71QC24 sends a Request to Send data from the PC CPU to an external device and are turned OFF at the end of transmission of the designated data by the AJ71QC24. The on-demand handshake signals are used to interlock the devices so that multiple on- demand requests are not issued at the same time.

Signal	Contents	Timing
Xn0	On-demand send normal end	
Xn1	On-demand send error detected	
Xn2	On-demand send underway	
Yn0	On-demand request to send	

(Note) "n" of Xn0, etc. is the I/O No. determined by the slot into which the AJ71QC24 is inserted.

With the system configuration shown below, the AC71QC24 is allocated to X.Y80~9F.



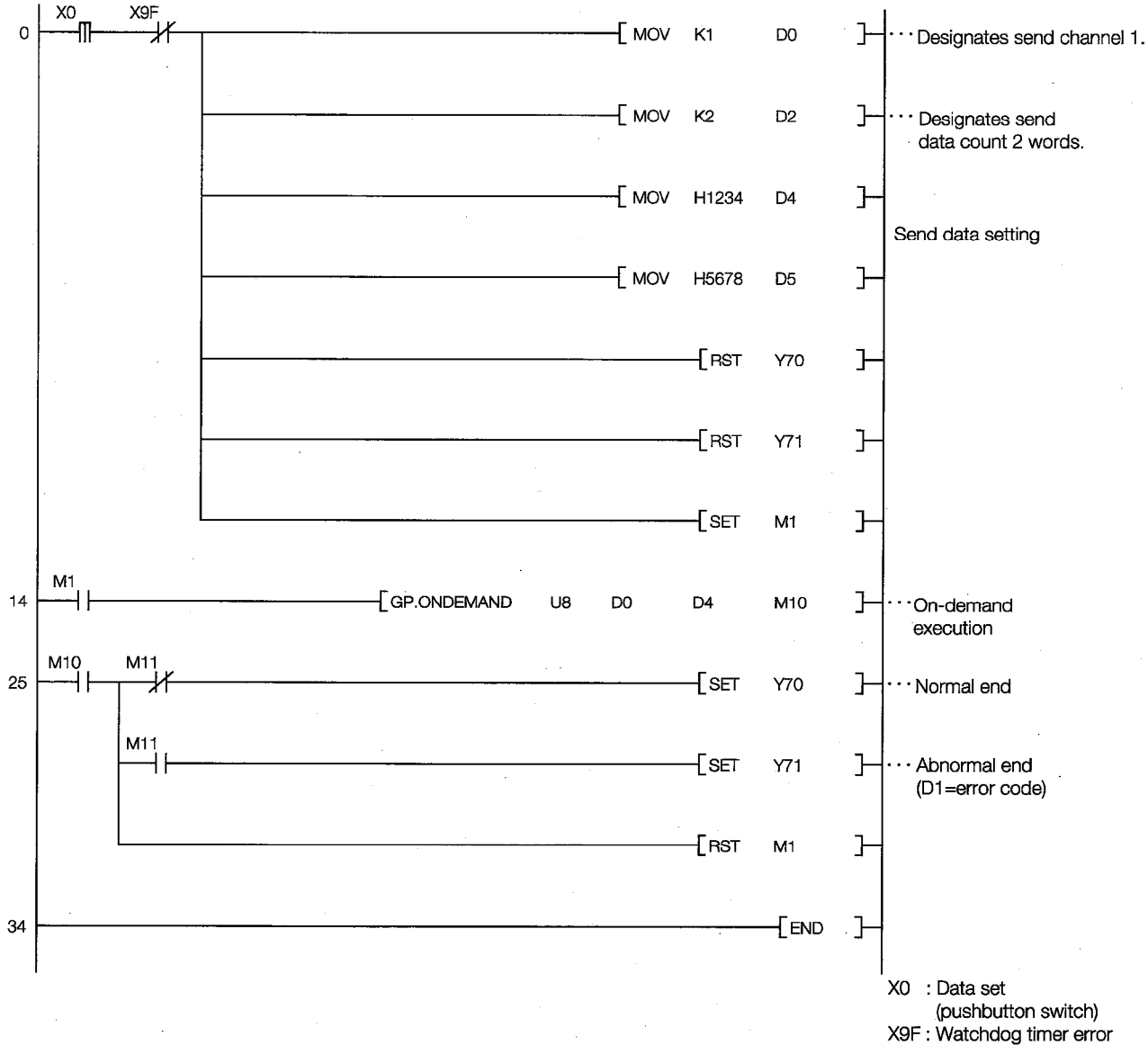
When the AJ71QC24 dedicated command (ONDEMAND) is used in data communications, data can be sent and received without being aware of the buffer memories and on-demand handshake signals described above.

4.9.4 Transmission from PC CPU by on-demand function

(Practice Contents)

This section sends data from the PC CPU to an external device when X0 is turned ON during dedicated protocol data communications.

1 Sequence program

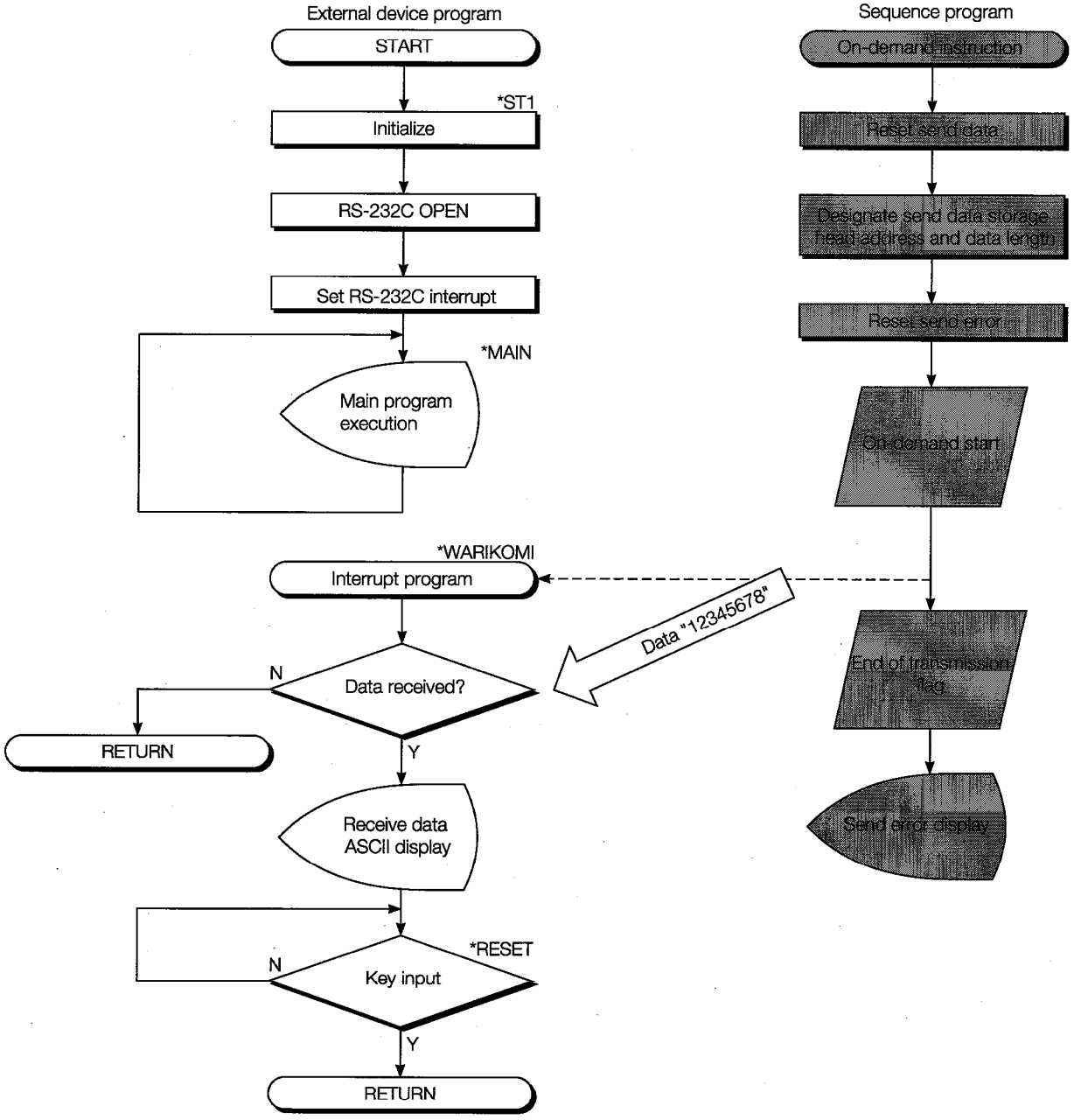


2 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                  AJ71QC24 On-Demand Function Sample Program                   !
1020 ' !                                                                              !
1030 ' ! ------------------------------------------------------------------------------!
1040 *ST1
1050     CLS 3                              :'Clear screen
1060     CH% =1                             :'Channel No.
1070     '
1080 ' ! ------------------------ RS-232C open & initialize---------------------------!
1090     OPEN "COM:E71NN" AS #CH%           :"Set communication mode, etc.
1100     ON COM  GOSUB *WARIKOMI : COM ON :'RS-232C interrupt setting
1120 ' ! --------------------------------Main program ---------------------------------!
1130 *MAIN
1140     CLS
1150     LOCATE 10,1 : PRINT "Main program running."
1160     LOCATE 10,3 : PRINT "When the on-demand function is executed,"
1170     LOCATE 10,4 : PRINT "execute the interrupt program."
1180     LOCATE 1,10
1190     FOR I%= 1 TO 100 :PRINT USING "### "; I% ;
1200       FOR J%= 1 TO 100
1210       NEXT J%
1220     NEXT I%
1230     LOCATE  1,10
1240     FOR I%= 1 TO 100 :PRINT       "    ";
1250       FOR J%= 1 TO 100
1260       NEXT J%
1270     NEXT I%
1280     GOTO *MAIN
1290     '
1300 ' ! ---------------------------- Interrupt program-------------------------------!
1310 *WARIKOMI
1320     CLS
1330     IF LOC(CH%)=0 THEN RETURN
1340     LOCATE 25,1 : PRINT "Data receive from AJ71QC24"
1350     LOCATE 25,2 : PRINT "Receive data ASCII code (hexadecimal)"
1360     FOR WT%=1 TO 1000 : NEXT WT%
1370     RD$=INPUT$(LOC(CH%),#CH%)       :MAX%=LEN(RD$)
1380     FOR L%=1 TO MAX%
1390       A$=MID$(RD$,L%,1)
1400       PRINT SPC(35)RIGHT$("0"+HEX$(ASC(A$)),2)
1410     NEXT L%
1420     LOCATE 25,23 : PRINT "Hit any key to return to main program."
1430 *REST
1440     IK$=INKEY$ : IF IK$="" THEN *REST
1450     RETURN
```

* BASIC commands must be changed according to BASIC software.

On-demand function



*[ ] : Label name used with the program

3 Trainer operation

Write the sequence program shown in item 1 to the PC CPU and the BASIC program shown in item 2 to the external device.

After writing, set the PC CPU and external device to RUN.

If there are no errors, send and receive data using the following procedure.

① The external device CRT screen continuously displays 1 to 100 numeric characters.

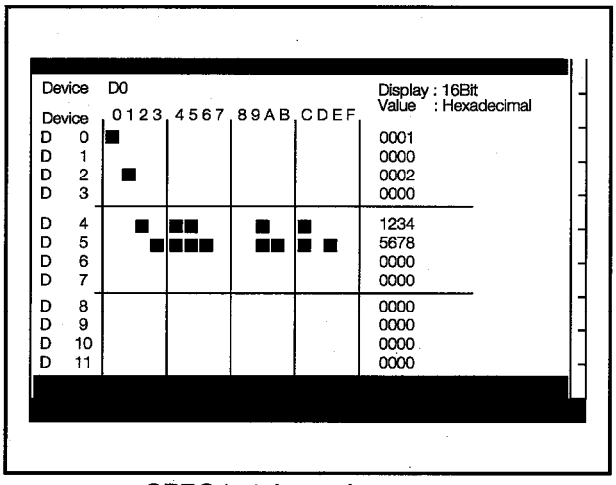
(On-demand function interrupt wait state)

```
        Main program running
        When the on-demand function is executed,
        the interrupt program.
1 2 3 4 5·····
·····
·····
····· (Continuous numeric characters)·····99 100
```

External device screen ①

② When X0 is turned ON, the sequence program sends an on-demand command to the QC24 and sends the data "12345678H" to the external device.

The external device executes the interrupt processing routine and displays the receive data.



GPPQ batch monitor screen

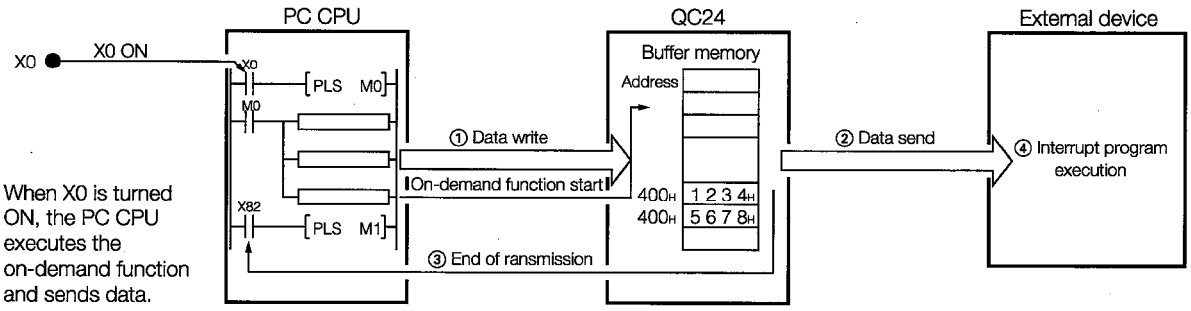
```
Data received from AJ71QC24
Receive data ASCII code (hexadecimal)
    02 ← STX
    30 } 00
    30 }
    46 } FE
    45 }
    31 }
    32 }
    33 }
    34 } "12345678"
    35 }
    36 }
    37 }
    38 }
    03 ← ETX
Hit any key to return to main program
```

External device screen 2

If you have any situations as follows, please read the explanations

CRT does not switch from external display screen ① to external display screen ② even though X0 is turned ON	• Was sequence program 1 written to the PC CPU? • Are the cables connected (Section 3.3) correctly?
Data set in the PC CPU and the data received by the external device are different	• Are the external device settings (Section 3.4) and QC24 switch settings (Section 4.2) correct?

4 Summary of on-demand function data communications

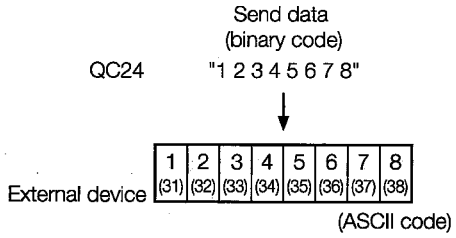


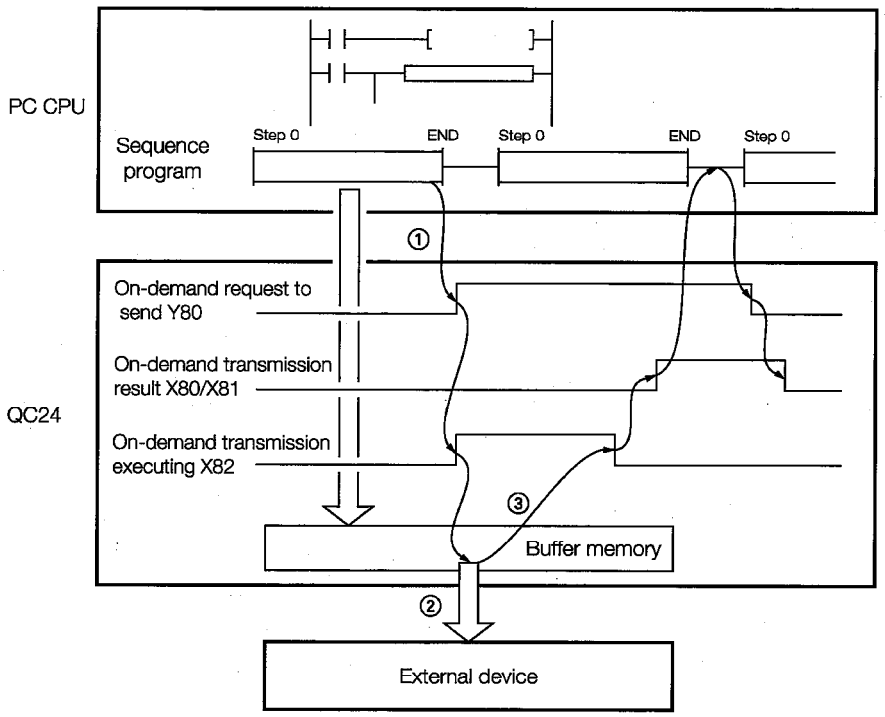
① When the PC CPU writes data to the QC24 buffer memory with the TO instruction and turns ON the on-demand request to send signal (Y80), the on-demand transmission executing signal (X82) is turned ON.

② The QC24 converts the written data to ASCII code and sends it to the external device.

③ When the QC24 finishes sending the data, it turns OFF the on-demand transmission executing signal (X82).

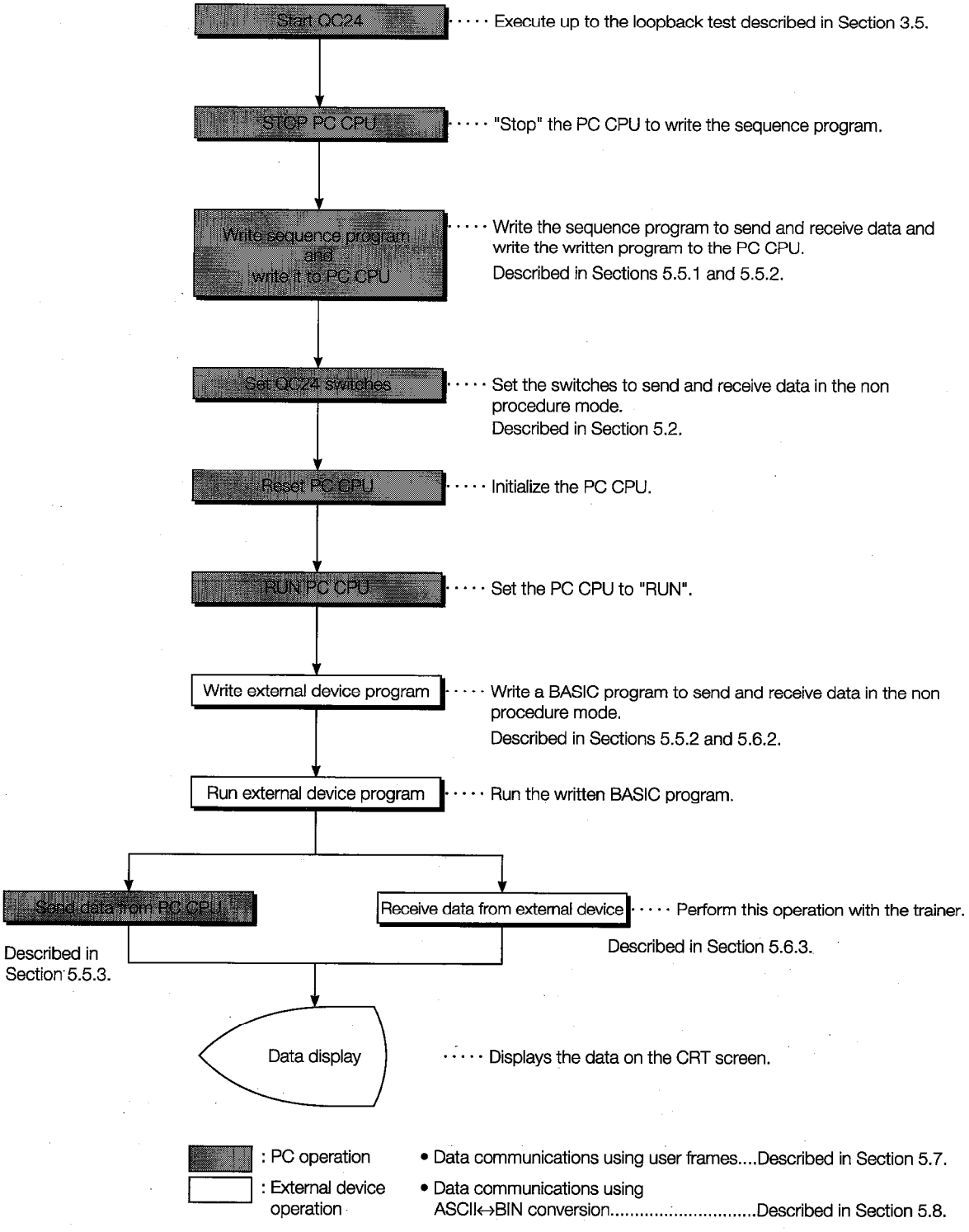
④ When the external device receives the data, it executes the interrupt processing routine gram and displays the receive data.





5 NON PROCEDURE PROTOCOL COMMUNICATIONS

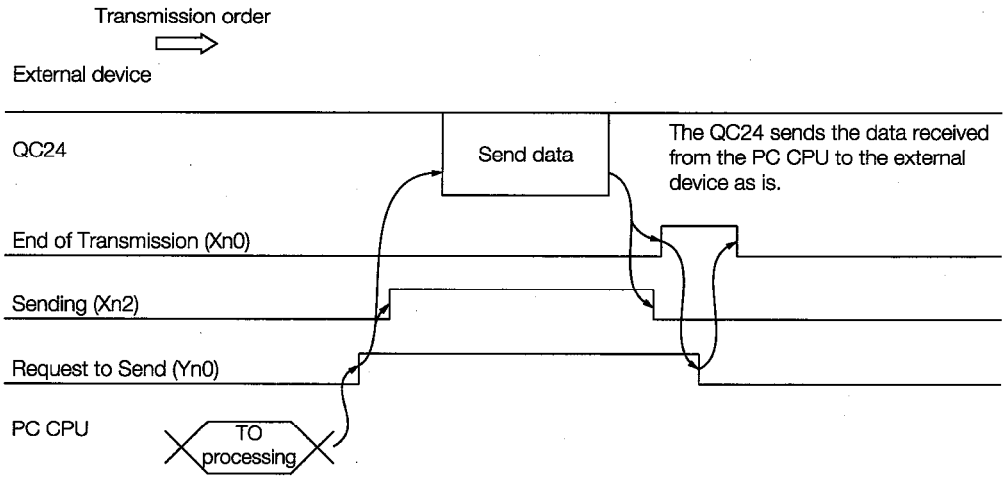
The following shows the non procedure protocol communications procedure used in this guide-book.



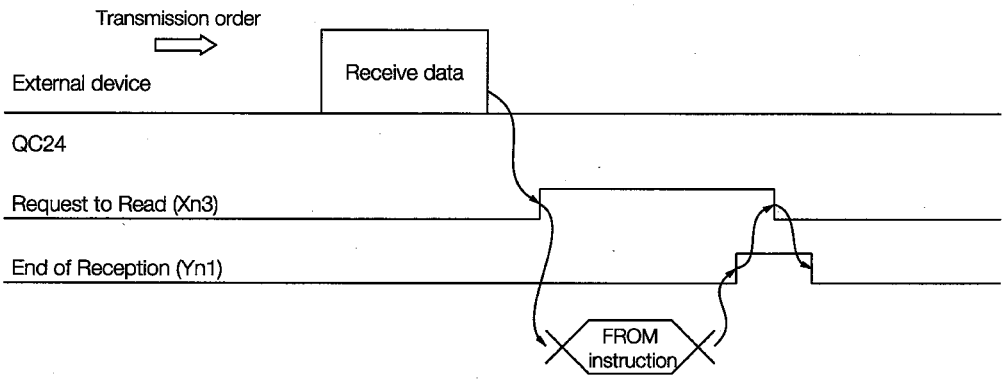
5.1 Non Procedure Protocol Control Procedure

The following describes the control procedure contents when CH1 uses the non procedure protocol to send and receive data.

- Data transmission from PC CPU to external device



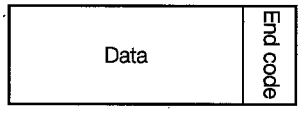
- Reception of data from external device at the PC CPU



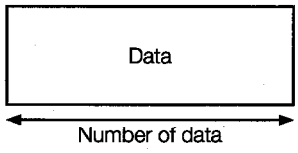
To turn ON the Request to Read signal (Xn3)...

The QC24 turns ON the Request to Read signal when it receives any of data ① to ③ below from the external device.

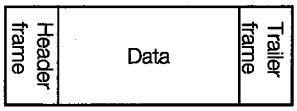
① When end code written to buffer memory (A5H) received



② When the number of data written to the buffer memory (A4H) received

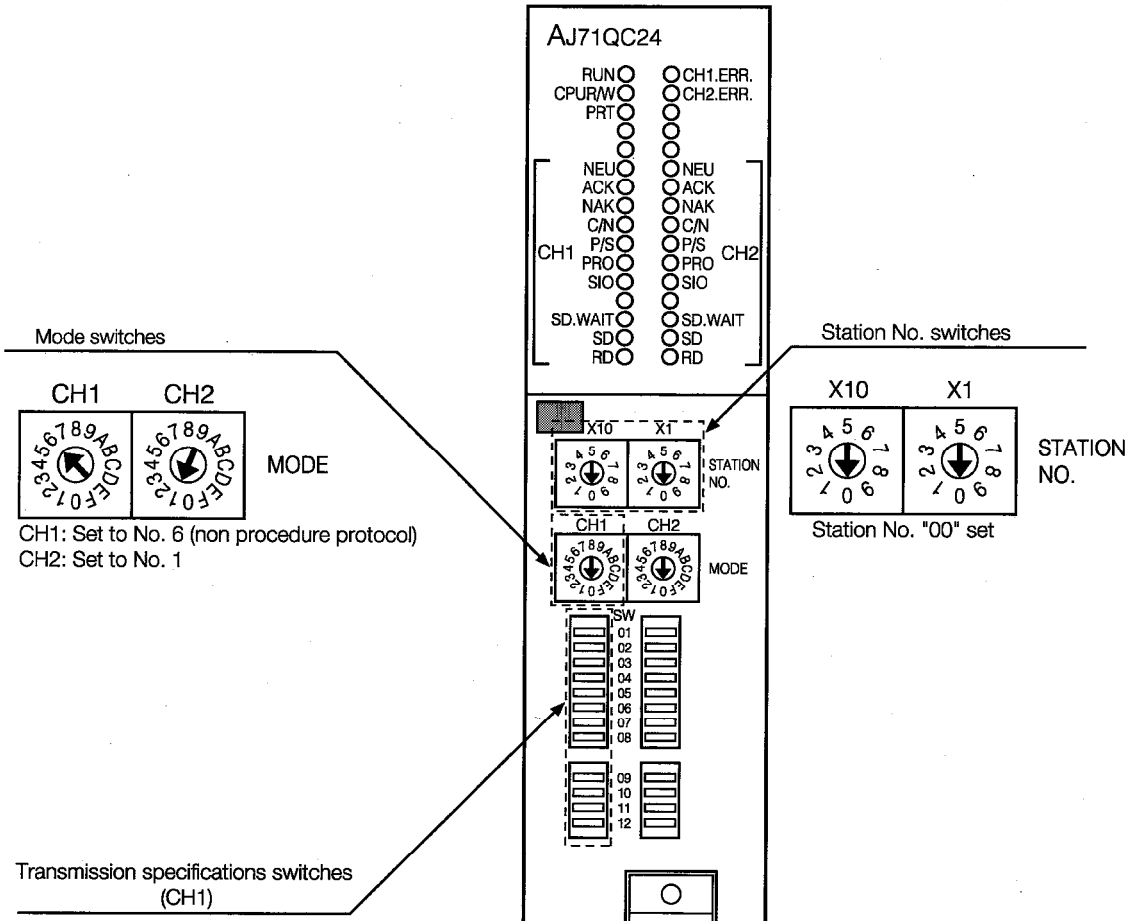


③ When user frame data received (See Section 5.6 for a description of user frame.)



5.2 AJ71QC24 Serial Communications Module Settings

Set the AJ71QC24 switches as shown below.



Switch Setting	Switch	Setting Item	Setting Contents
	SW01	Operation setting	OFF (independent operation)
	SW02	Data bit setting	OFF (7 bits)
	SW03	Parity bit enable/disable setting	ON (Enable)
	SW04	Even parity/odd parity setting	ON (Even)
	SW05	Stop bit setting	OFF (1 bit)
	SW06	Checksum enable/disable setting	OFF (Disable)
	SW07	Enable/disable write during RUN	ON (Enable)
	SW08	Setting modification enable/disable setting	OFF (Disable)
	SW09	Transmission rate setting	OFF
	SW10		OFF
	SW11		ON (4800BPS)
	SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

5.3 Non Procedure Protocol LED Display Contents

The following shows the display contents of the QC24 LEDs related to non procedure protocol data communications using the system configuration shown in Section 3.1.1.

LED Name	LED Application	ON	OFF	Error Processing
RUN	QC24 operation status display	Module normal	Module abnormal	• Check SW09~SW12. • Check the mode switch setting.
CPU R/W	QC24 and PC CPU communications status	Blinks while communicating (Lights steadily while stopped)	Abnormal	• Check the status of the PC CPU and the external device. • Check the mode switch setting.
1-NEU	———	Abnormal	Normal	• Check the mode switch setting.
1-C/N	QC24 and PC CPU communications status display	Communications abnormal	Communications normal	• Check SW07 (enable/disable write during (RUN)
1-P/S	Parity error display	Error	Normal	Transmission specifications setting and receive data are not the same. • Make the QC24 transmission specifications and the external device transmission specifications the same. • Check the transmission specifications switches.
1-SIO	Displays the data send/receive status.	Error	Normal	Transmission specifications and receive data are not the same. • Make the QC24 transmission specifications and the external device specifications the same. • Check the transmission specifications switches. • Decrease the transmission rate. • Reduce the number of data sent from the external device.
1-SD · WAIT	Displays the status of the data sent to the external device by the QC24.	Send wait	Sending No send data	———
1-SD	Displays the status of transmission from QC24 to external device.	Blinks while data is being sent.	Data not being sent	• If this LED does not blink even though the external device is sending a command, check the cable connections.
1-RD	Displays the status of reception from external device to QC24.	Blinks while data is being received.	Data not being received	
CH1. ERR	CH1 communications status display	CH1 error	Normal	———

□ : Normal state.

If an error is generated, the error LED will come on.

Thereafter, the error LED will remain ON even if the error recovers.

Use the buffer memory (201H) to turn OFF the error LED.

5.4 Non Procedure Protocol Setting Items

The following describes the buffer memory and I/O signal items which are set during the non procedure protocol data communications described in this guidebook. (CH1 only)

(See Appendixes 1 and 2 for the QC24 buffer memories table and I/O signals table.)

5.4.1 Buffer memory setting

The following table lists the buffer memories used with the non procedure protocol.

Address Hexadecimal (Decimal)	Buffer Memory Address Name				Default Value
2H (2)	System setting	EEPROM		Write/delete/read directive	0
3H (3)				Frame No. directive	0
4H (4)				Write/read result storage	0
5H (5)				Frame byte count designation	0
6H (6) to 2DH (49)				Write/read data storage	0
A2H (162)		Non procedure	Send	Buffer memory head address designation	400H
A3H (163)				Buffer memory length designation	200H
A4H (164)			Receive	End data count designation	1FFH
A5H (165)				End code designation	0D0AH(CR,LF)
A6H (166)				Buffer memory head address designation	600H
A7H (167)				Buffer memory length designation	200H
ADH (173)		Non procedure receive user frame		User frame enable/disable designation	0
AEH (174)				Header frame No. designation (1st)	0
AFH (175)				(2nd)	0
B2H (178)				Trailer frame No. designation (1st)	0DH(CR)
B3H (179)				(2nd)	0AH(LF)
B6H (182)		Non procedure send user frame		Sending user frame No.	0
B7H (183)				End code (CR/LF) designation	0
B8H (184)				Send pointer designation	0
B9H (185)				Send frame count designation	0
BAH (186)				Send frame No. designation (1st)	0
BBH (187)				(2nd)	0
121H (289)		ASCII/binary conversion designation			0
204H (516)	System setting	EEPROM access		User frame register count	0
205H (517) to 21DH (541)				User frame registered status (for catalogue No. check)	0
21EH (542)		ROM registered count			0
220H (544)		EEPROM system settings write result			0
257H (599)		Data transmission result			0 (No errors)
258H (600)		Data reception result			0 (No errors)
25BH (603)		Receive user frame No.			0 (Not received)
400H (1024)	CH1 send/ receive	Send area (user frame No. 8000H)		Send data count	0
401H (1025) to 5FFH (1535)				Send data (storage) 511 words	0
600H (1536)		Receive area		Receive data count (read data count)	0
601H (1537) to 7FFH (2047)				Receive data (storage) 511 words	0

5.4.2 PC CPU input/output signals

The following shows the input/output signals used with the non procedure protocol.

1 Input signals (AJ71QC24→PC CPU)

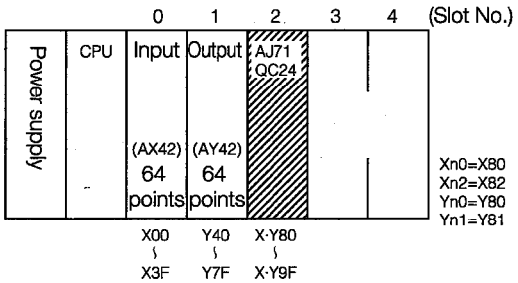
Input Signal	Signal Name	Contents
Xn0	Send normal end	Turned ON at the end of transmission in response to a Request to Send signal from the PC CPU.
Xn1	Send error detected	Turned ON when an error was detected while data was being sent.
Xn3	Receive data read request	Turned ON when the QC24 receives data from an external device after the QC24 was started.
Xn4	Receive error detected	Turned ON when an error was detected while receiving data.
Xn+1E	QC24 Ready signal	Turned ON when the QC24 becomes operative.

2 Output signals (PC CPU→QC24)

Output Signal	Signal Name	Contents
Yn0	Request to Send	Turned ON when the buffer memory data sent to the QC24 from the PC CPU is sent to the external device after the QC24 is started.
Yn1	End of Receive Data Read	Turned ON when the PC CPU finishes reading the data received at the QC24 from the external device.

(Note) Xn0 and Yn0 are input and output signals determined by the slot into which the QC24 was inserted.

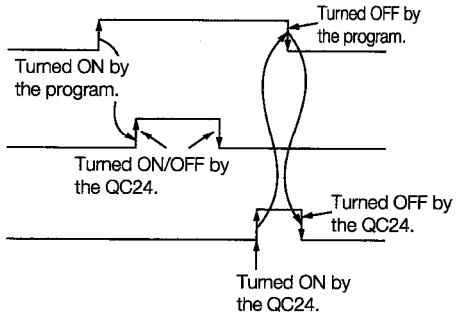
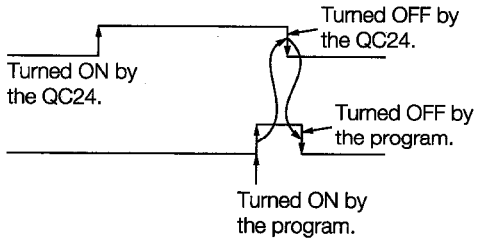
With the system configuration shown below, the QC24 is allocated to X·Y80~9F.



3 Non procedure protocol handshake input/output signals

The signals that send the data output from the sequence program to the external device and the signals that detect the arrival of data from the external device and enable the sequence program to read the data when the non procedure protocol is used to send and receive data are called "handshake signals". These signals are necessary with the non procedure protocol.

The following shows the handshake input/output signals.

	Signal	Timing
PC CPU ↓ External device	Yn0 (Request to Send) Xn2 (Sending) Xn0 (End of Transmission)	
External device ↓ PC CPU	Xn3 (Request to Read Receive Data) Yn1 (End of Reading of Receive Data)	

In this guidebook, AJ71QC24 dedicated instructions are used to send and receive data in the non procedure protocol mode.

Data transmission from PC CPU : OUTPUT instruction (Section 5.5)
PRR instruction (Section 5.7.2)

Data reception from external device : INPUT instruction (Section 5.6)

Registering of user frame to EEPROM : PUTE instruction (Section 5.7.3)

By using dedicated instructions, data can be sent and received without being aware of the handshake signals and input/output signals above and the buffer memory address.

The data send/receive programs are also simple.

5.5 Data Transmission From PC CPU

This section sends data from the PC CPU to an external device.

The send data count units are word units.

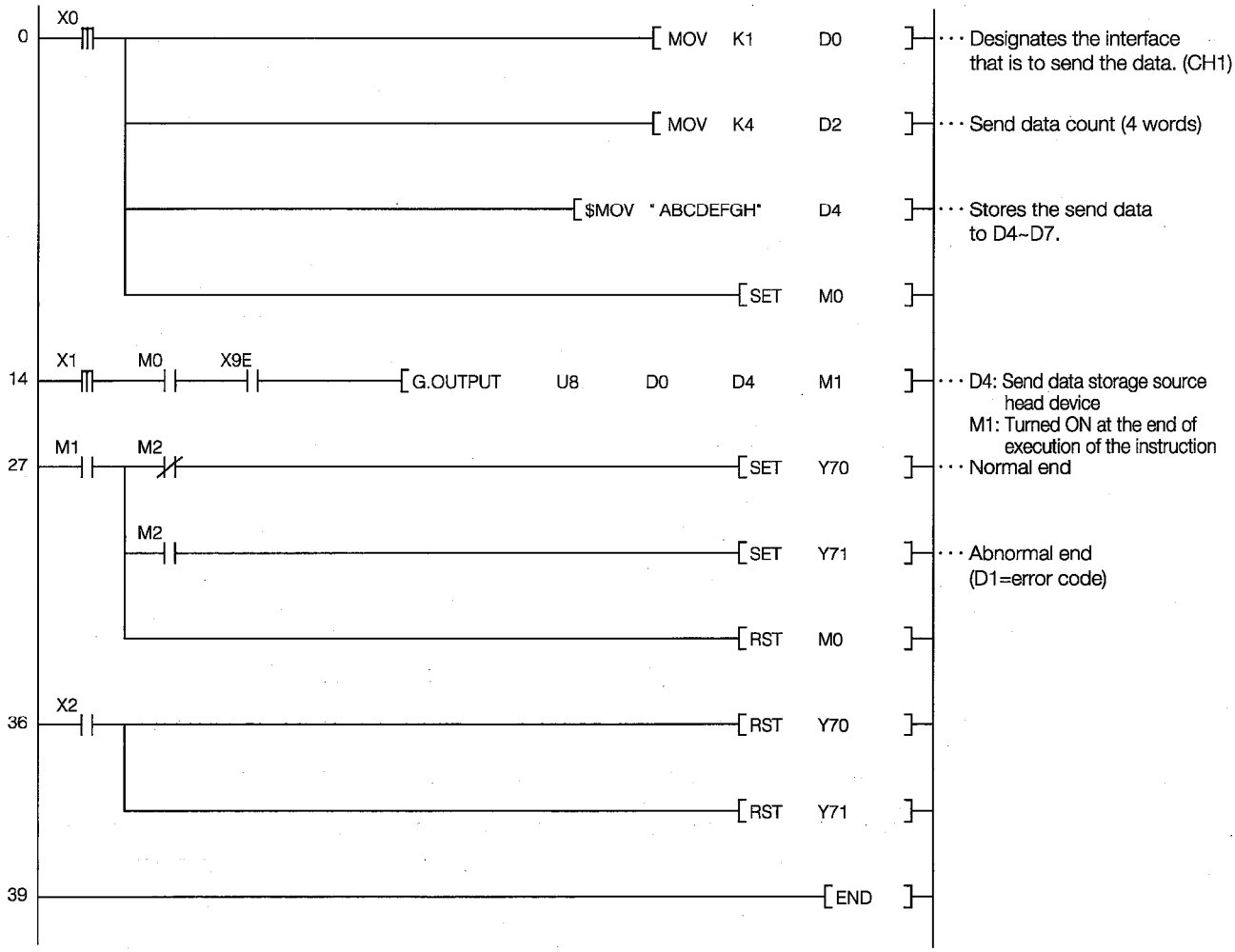
(Practice Contents)

Use the sequence program OUTPUT instruction to write the character data (ASCII code) [ABCDEFGH] to the QC24 buffer memory when X0 is turned ON.

When X1 is turned ON, the written data is sent to the external device using the non procedure protocol.

The external device displays the received data on its CRT screen.

5.5.1 Sequence program

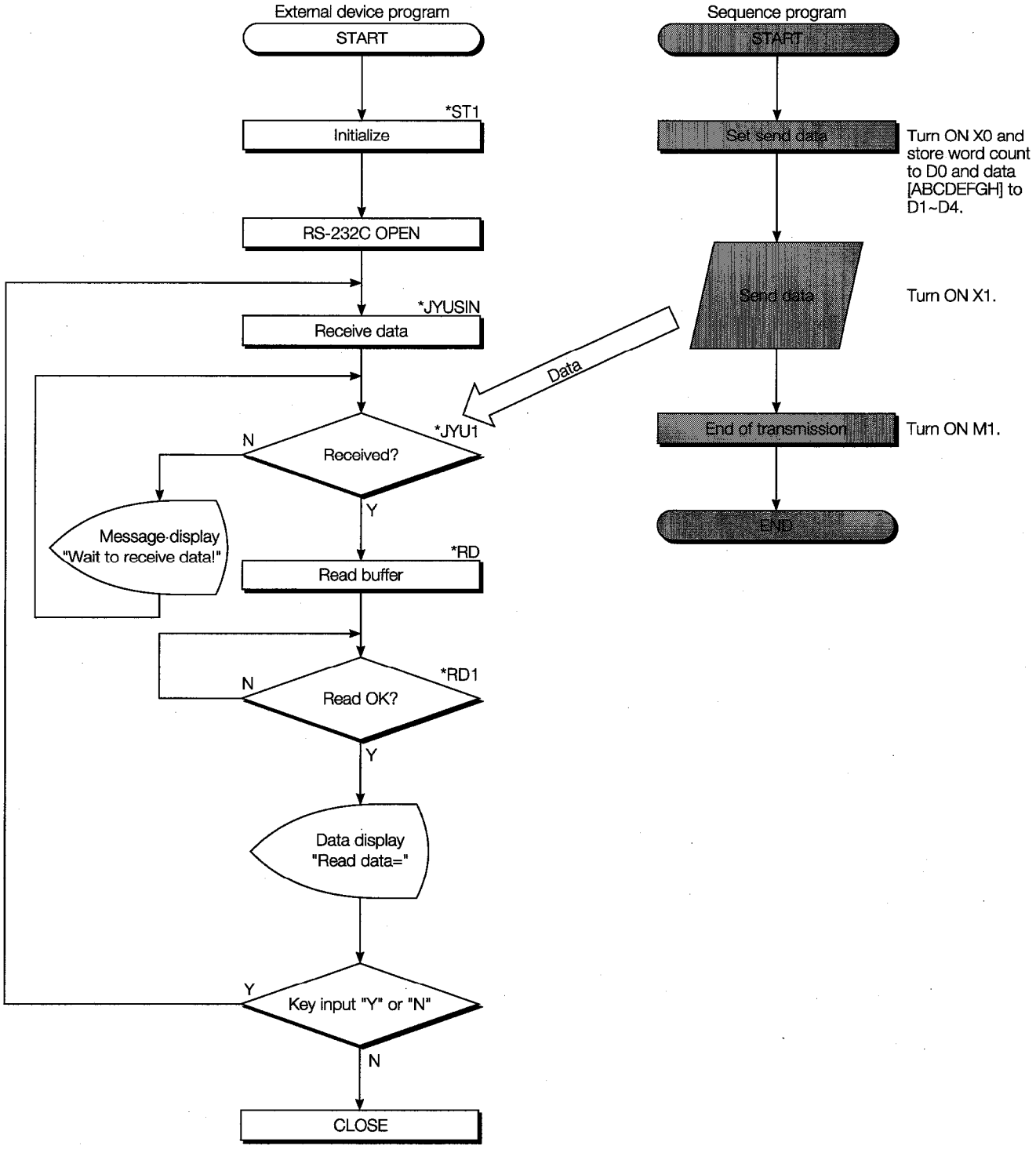


5.5.2 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                  AJ71QC24 Non Procedure Mode Sample Program                   !
1020 ' !                       (Data transmission from PC CPU)                         !
1030 ' ! ------------------------------------------------------------------------------!
1040 *ST1
1050     CLS                                :'Clear screen
1060     CH% =1                             :'Channel No.
1070     '
1080 ' ! ------------------------ RS-232C open & initialize----------------------------!
1090     OPEN "COM:E71NN" AS #CH%           :'Set communications mode, etc.
1100 ' ! -------------------------------Data reception ---------------------------------!
1110 *JYUSIN
1120     CLS
1130     LOCATE 6,1:PRINT "**Reception from PC in non procedure mode **"
1140 *JYU1
1150     IF LOC(CH%)<>0 THEN *RD
1160     LOCATE 10,5:PRINT "Wait to receive data!"
1170     LOCATE 10,8:PRINT "Send data from PC.":GOTO *JYU1
1180     '
1190 ' ! ------------------------------- Buffer read-------------------------------------!
1200 *RD
1210     B$=""
1220 *RD1
1230     FOR I%=0 TO 1000  :NEXT I%
1240     IF LOC(CH%)<>0 THEN B$=B$+INPUT$(LOC(CH%),#CH%):GOTO *RD1
1250     '
1260 ' ! ----------------------------Receive data display ----------------------------!
1270     LOCATE 10,5:PRINT "Read data=";B$
1280     LOCATE 10,8:INPUT "     Receive again? (Y/N) ";Y$
1290     IF Y$="Y" THEN *JYUSIN
1300     '
1310 CLOSE
1320 END
```

* BASIC commands must be changed according to BASIC software.

Data transmission from PC CPU



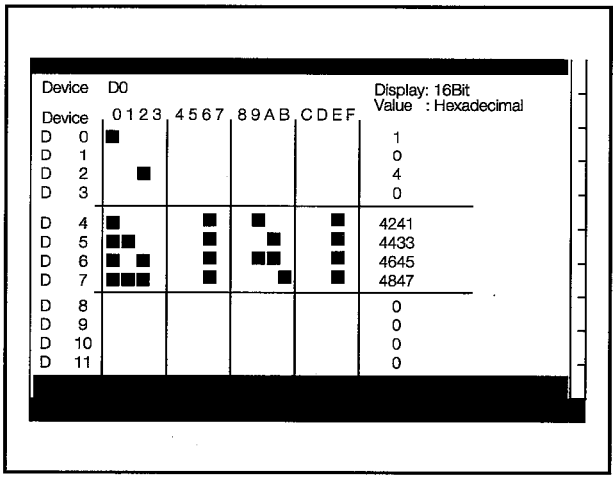
5.5.3 Trainer operation

Write the sequence program shown in Section 5.5.1 to the PC CPU and the BASIC program shown in Section 5.5.2 to the external device.

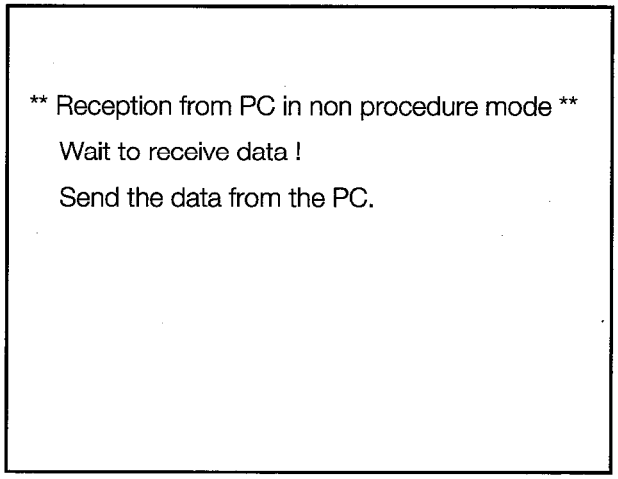
After writing, set the PC CPU and external device to "RUN".

If there are no errors, send and receive data using the following procedure.

① Turn ON X0. The number of words of data to be sent to the external device is stored to D2 and the send data is stored to D4 to D7. Use the GPPQ batch monitor screen to check if the data was stored.

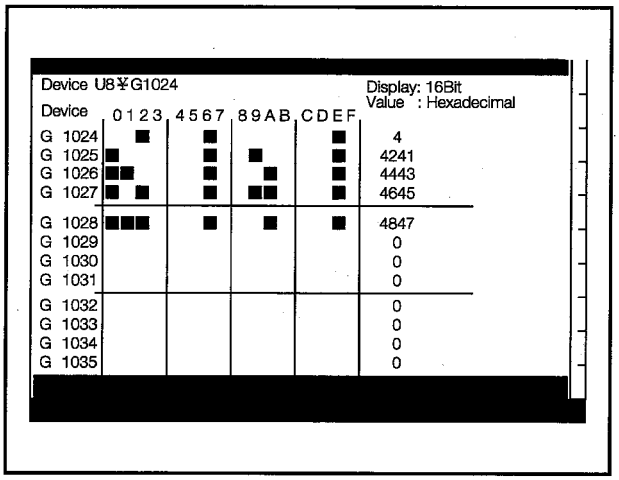


GPPQ batch monitor screen

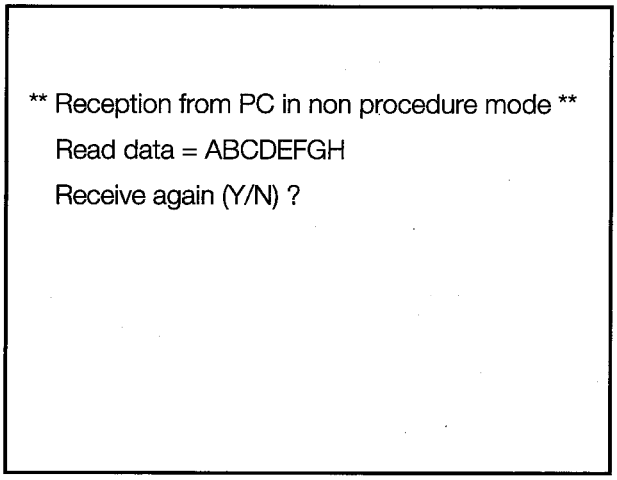


External device screen

② Turn ON X1. Send data D0 to D4 are written to QC24 buffer memory addresses 400H to 404H (1024 to 1028) and sent to the external device.



GPPQ buffer batch monitor screen



External device screen

If you have any situations as follows, please read the explanations

- After the data was set to the PC CPU, the Request to Send signal was turned ON and the data was sent to the external device.

  The following was displayed even though the AJ71QC24 turned ON the End of Transmission signal:

```
** Reception from PC in non procedure mode **

    Wait to receive data!
    Send data from PC.
```

- Was the sequence program (Section 5.5.1) written to the PC CPU?
- Are the QC24 mode switch (Section 5.2), cable connections (Section 3.3), and other settings correct?
- After checking the QC24 LEDs (Section 5.3), check the settings.

- The data sent from the PC CPU and the data received at the external device are not the same.

```
** Reception from PC in non procedure mode **

    Read data = □
    Receive again (Y/N) ?
```

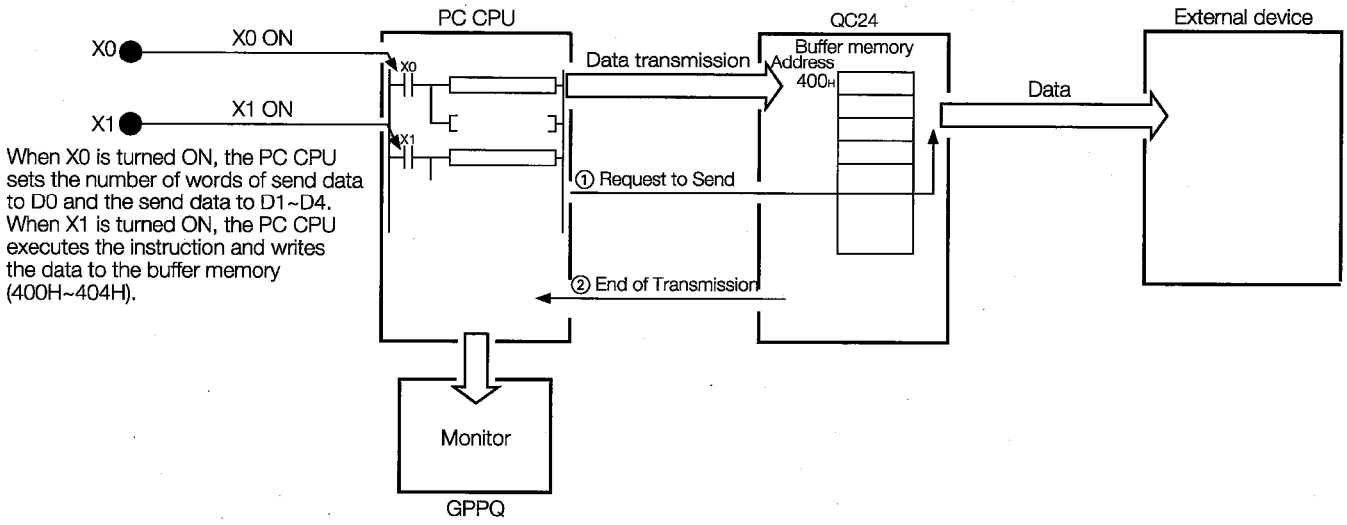
- Are the external device settings (Section 3.4), QC24 switch settings (Section 5.2), and external device program (RS-232C setting) settings (Section 5.5.2) correct?

  After checking the QC24 LEDs (Section 5.3), check the settings.

· After resetting the settings, reset the PC CPU and communicate again.

5.5.4 Summary of data transmission from PC CPU using non procedure protocol

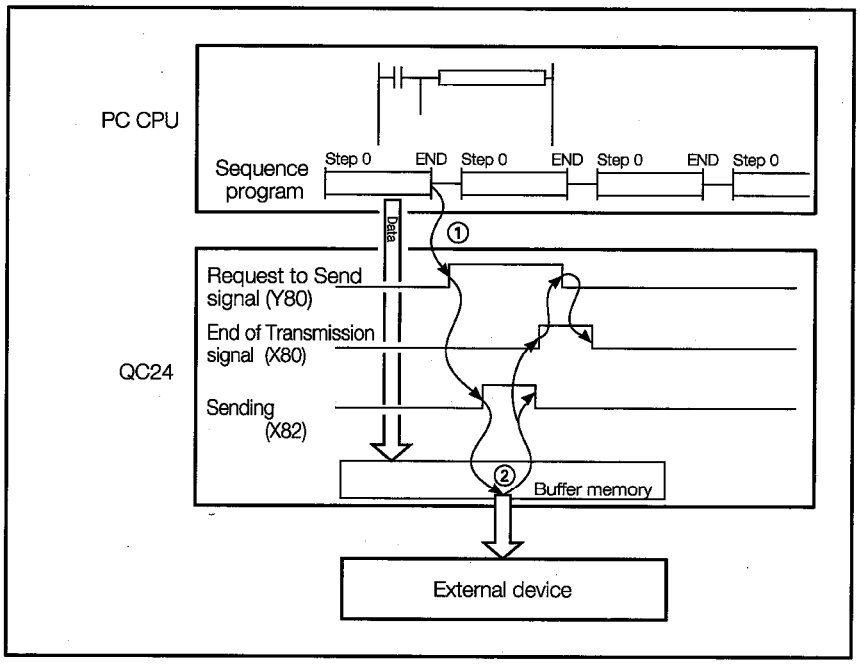
The following uses an image diagram to outline the PC CPU data transmission procedure using the non procedure protocol described in Section 5.5.3.



① *When the Request to Send signal is turned ON, the QC24 sends the data to the external device.

② When the QC24 finishes sending the data, it turns ON the *End of Transmission signal (X80) to the PC CPU and completes one data transmission.

* : The OUTPUT instruction automatically turns the QC24 Request to Send and End of Transmission signals ON and OFF by internal processing.



5.6 Data Reception From External Device

In this section, the PC CPU receives the send data from an external device.

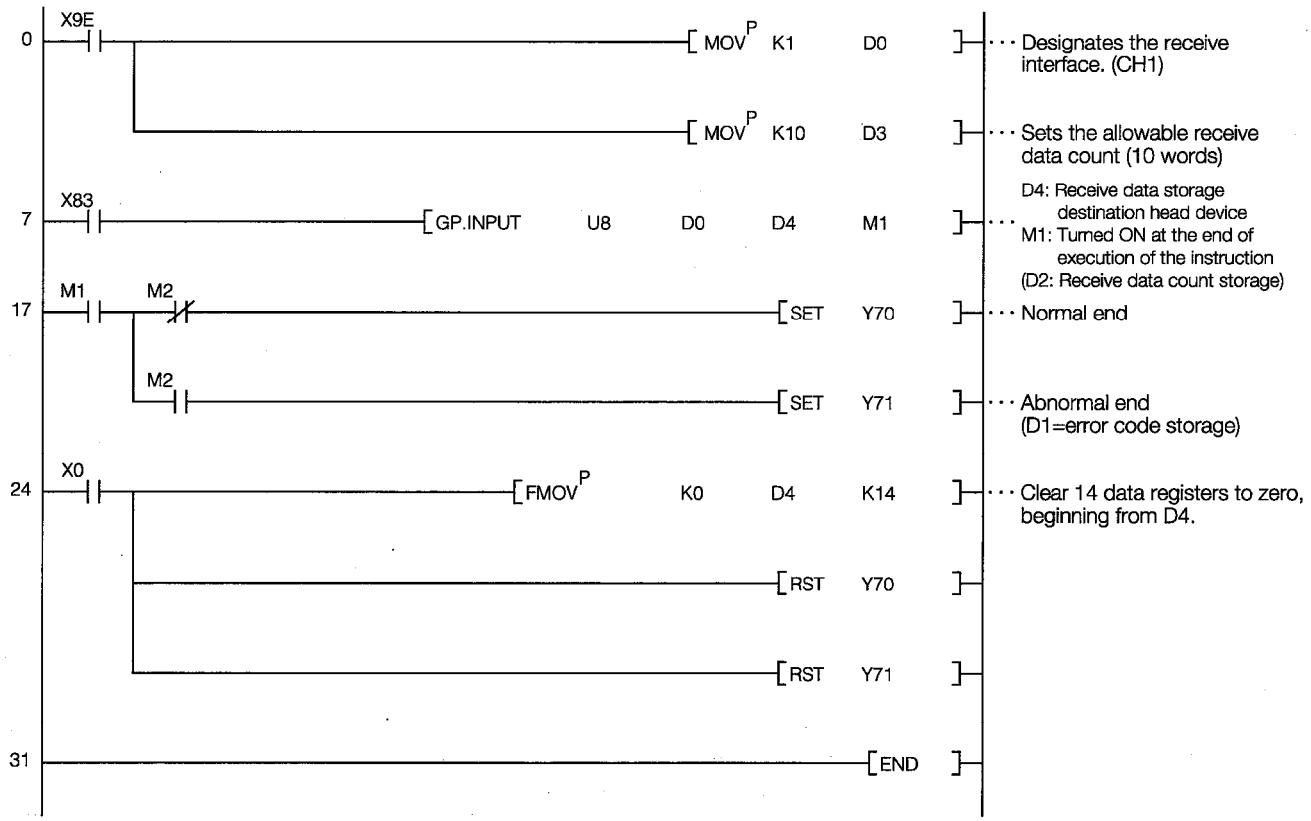
The receive data count units are word units.

(Practice Contents)

Send the data input from the external device keyboard to the PC CPU.

Use the sequence program INPUT instruction to read the receive data from the QC24 buffer memory to the PC CPU.

5.6.1 Sequence program

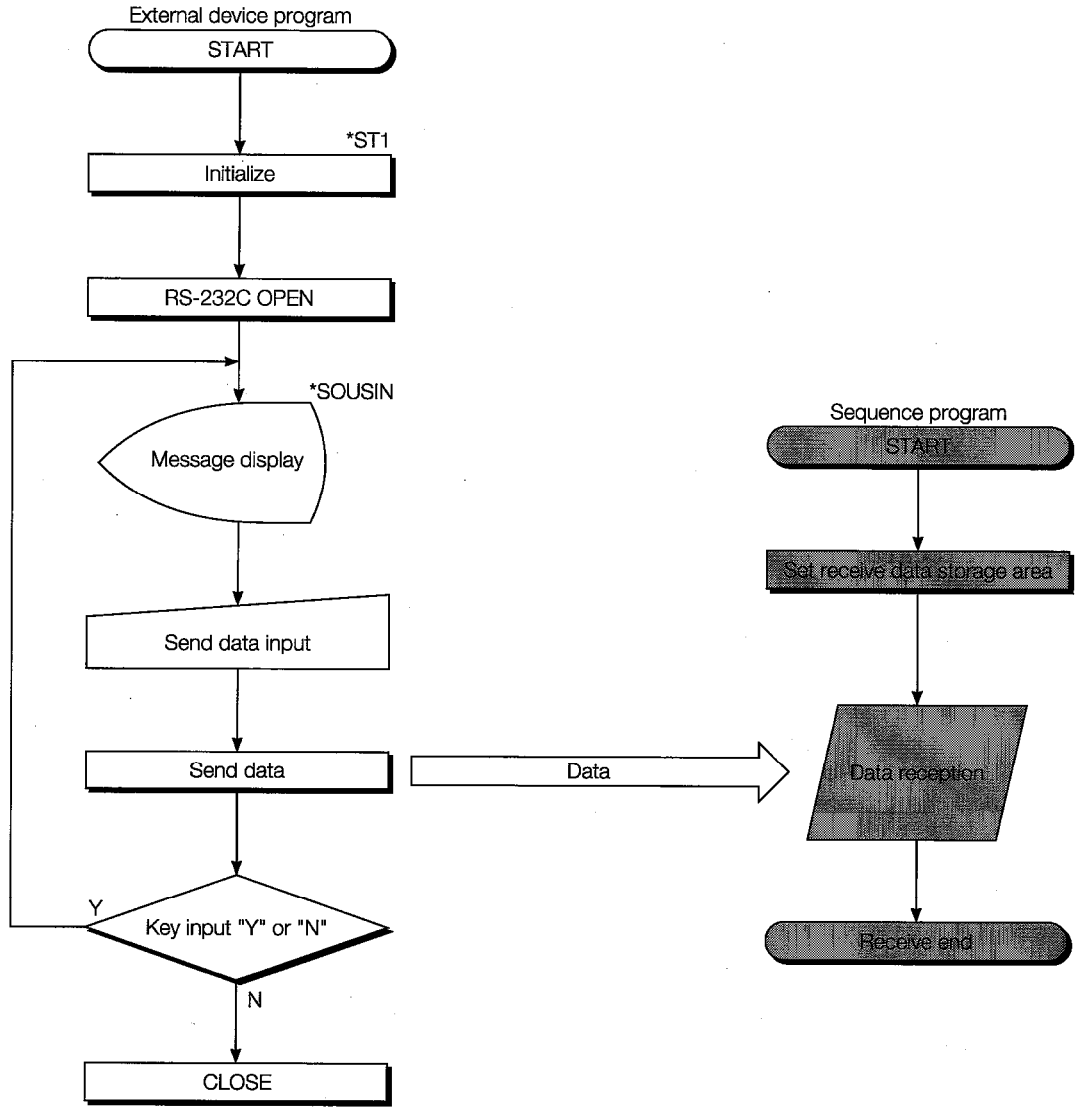


5.6.2 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                    AJ71QC24 Non Procedure Mode Sample Program                 !
1020 ' !                          (Data reception from computer)                       !
1030 ' ! ------------------------------------------------------------------------------!
1040 *ST1
1050     CLS                                :'Clear screen
1060     CH% =1                             :'Channel No.
1070     CR$ =CHR$(&HD)                     :'CR code
1080     LF$ =CHR$(&HA)                     :'LF code
1090     '
1100 ' ! ------------------------- RS-232C open & initialize --------------------------!
1110     OPEN "COM:E71NN" AS #CH%           :' Set communications mode, etc.
1120     '
1130 ' ! ------------------------------ Data transmission-------------------------------!
1140 *SOUSIN
1150     CLS
1160     LOCATE  6,1:PRINT "**Transmission to PC in non procedure mode**"
1170     LOCATE 10,5:INPUT " Send data ";SD$
1180     PRINT #CH%,SD$;CR$;LF$;
1190     LOCATE 10,8;INPUT "End of data transmission. Reset transmission? (Y/N)";Y$
1200     IF Y$="Y" THEN *SOUSIN
1210     '
1220 CLOSE
1230 END
```

* BASIC commands must be changed according to BASIC software.

Data reception from external device



*[ ] : Label name used with the program.

5.6.3 Trainer operation

Write the sequence program shown in Section 5.6.1 to the PC CPU and the BASIC program shown in Section 5.6.2 to the external device.

After writing, set the PC CPU and external device to "RUN".

If there are no errors, send and receive data using the following procedure.

① Input arbitrary send data from the external device keyboard and send this data to the PC CPU.

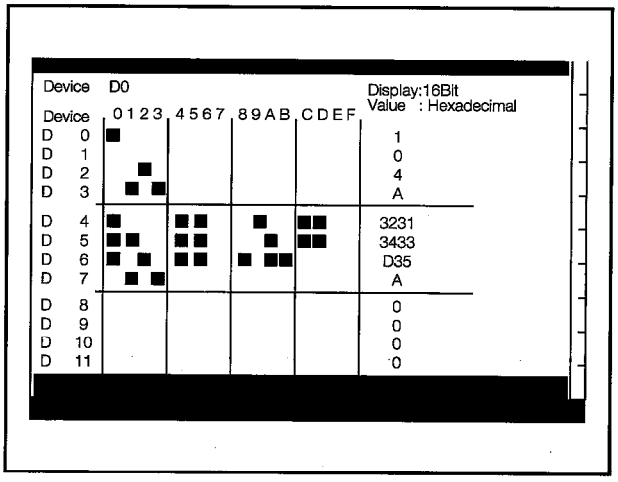
(For example, 1 2 3 4 5 ↵)

```
** Transmission to PC in non procedure mode **

Send data?
```

External device screen

② The data sent from the external device is written to QC24 buffer memory addresses 601H~(1537~).

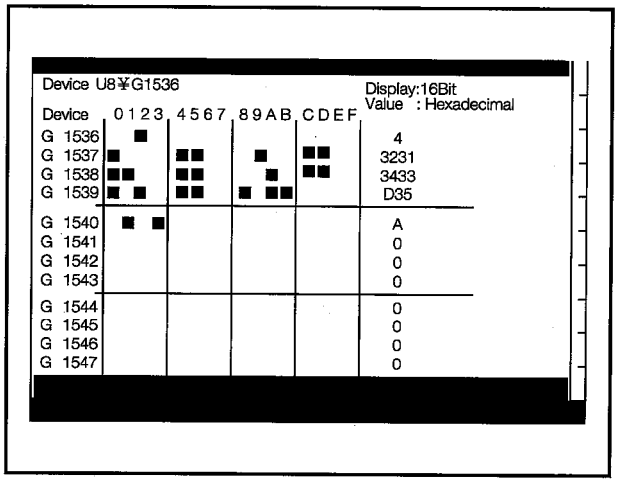


GPPQ batch monitor screen

```
** Transmission to PC in non procedure mode **

Send data?12345
End of data transmission.
Reset transmission (Y/N) ?
```

External device screen



GPPQ buffer batch monitor screen

If you have any situations as follows, please read the explanations

- Data was sent from the external device and the End of Transmission screen was displayed, but the PC CPU did not receive the data.

```
** Transmission to PC in non procedure mode **
    Send data ? □
    End of transmission.
    Reset transmission (Y/N) ?
```

- Was the sequence program (Section 5.6.1) written to the PC CPU?
- Are the QC24 switch settings (Section 5.2) correct?
- Check the QC24 LEDs (Section 5.3), then check the settings.

- Data cannot be sent from the external device.

```
** Transmission to PC in non procedure mode **
    Send data ? □
```

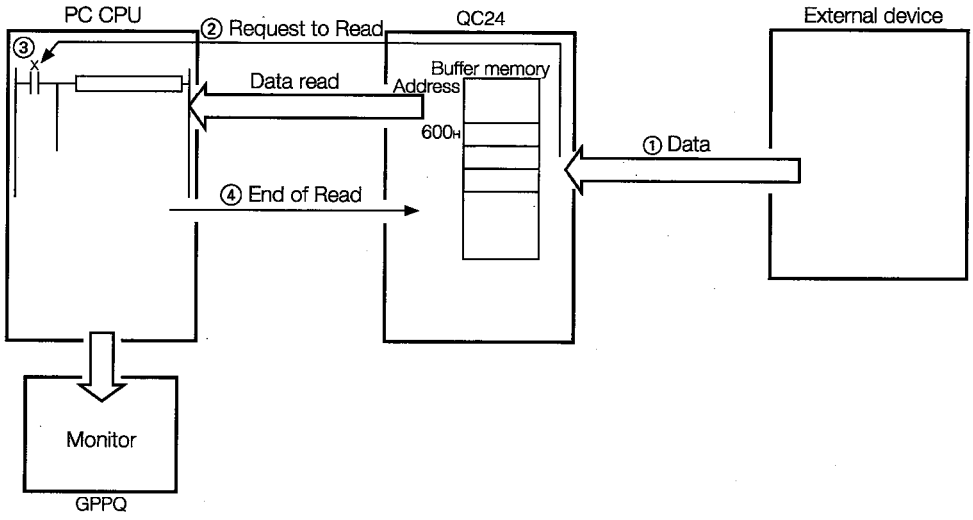
- Are the external device settings (Section 3.4) correct?

  Check the QC24 LEDs (Section 5.3), then check the settings.

After resetting, reset the PC CPU and communicate again.

5.6.4 Summary of data reception from external device using non procedure protocol

The following uses an image diagram to outline data reception from an external device using the non procedure protocol described in Section 5.6.3.



① When data is sent from the external device, the QC24 stores the receive data to its buffer memory (receive data storage area).

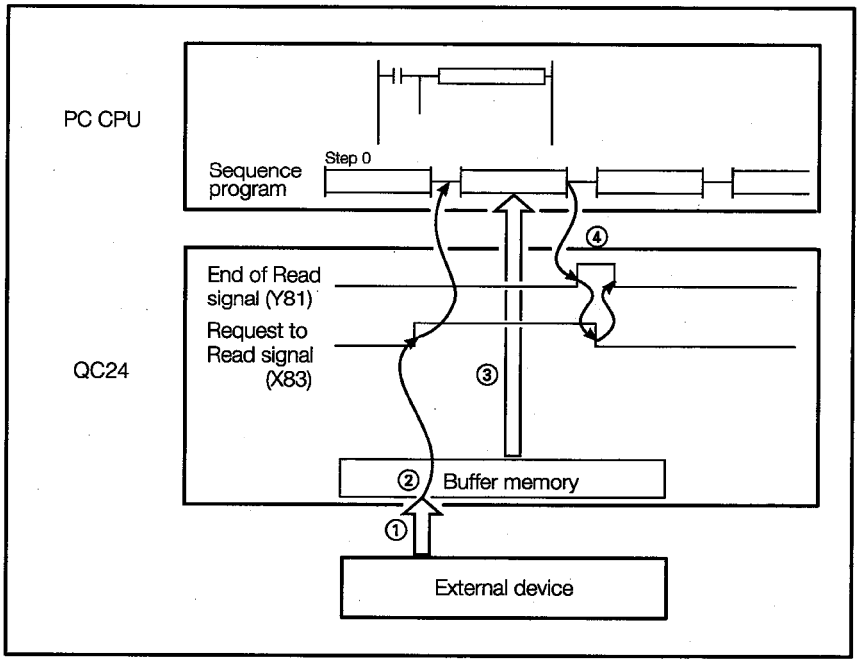
② When the QC24 receives the set end code CR, LF (0D0AH), it turns ON the *Request to Read signal (X83) to the PC CPU.

③ The PC CPU uses the sequence program to read the receive data from the QC24.

④ At the end of data read processing, the PC CPU turns ON the End of Read signal (Y81).

When the End of Read signal (Y81) is turned ON, the QC24 turns OFF the Request to Read signal (X83) and completes one data communications.

* : The INPUT instruction automatically turns the QC24 Request to Read and End of Read signals ON and OFF by internal processing. A sequence program that turns these signals ON and OFF is unnecessary.

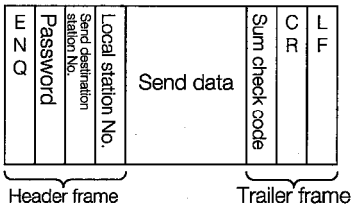


5.7 User Frame

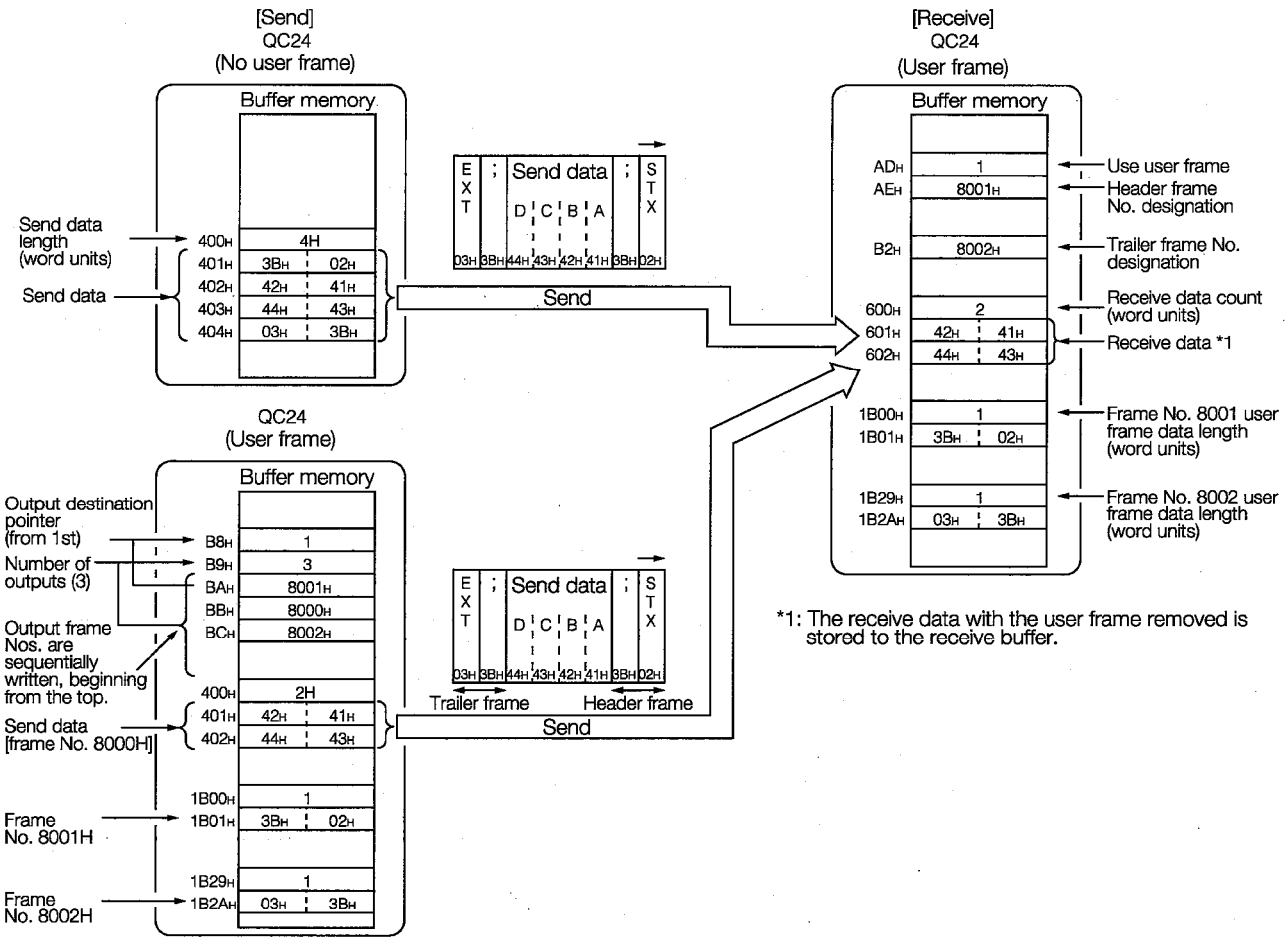
5.7.1 What is a user frame?

A user frame is the data list part the message exchanged with the opposite device. It is written to the QC24 buffer memory (or QC24 internal EEPROM) by the user beforehand and used to send and receive data.

The user can communicate using a message format matched to the specifications of the external device by adding a header frame and trailer frame to the data sent and received using the dedicated protocol on-demand function and the non procedure protocol.



1 User frame transmission control procedure



2 Kinds of user frames and cataloguing destination

The following table shows the kinds of user frame and their registering destinations.

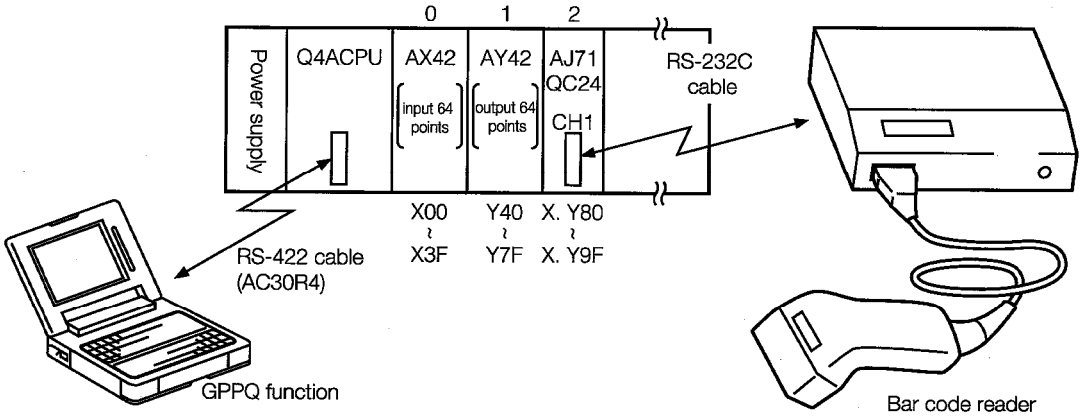
Kind	User frame No.	Registering destination	Features
Default frame (fixed data)	1H~3E7H	QC24 OS ROM	Since the frame is written to the ROM in advance, the frame No. is designated.
User frame (arbitrary data)	3E8H~4AFH	QC24 EEPROM	User frames written to the EEPROM are not lost even when the power is turned OFF.
	8001H~801FH	QC24 buffer memory (addresses 1B00H~1FF6H)	The frame must be written to buffer memory each time data is sent and received.

5.7.2 Practice using a user frame (default frame)

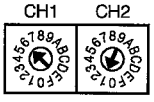
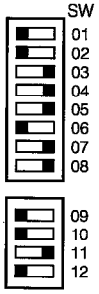
Use a user frame to read data with a bar code reader.

1 System configuration

The data to be read with the bar code reader is written to the QC24 buffer memory.
The PC reads this data from the buffer memory to the data memory.
The A7HGP monitors the data.



2 AJ71QC24 settings

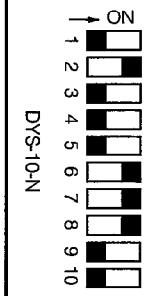
Mode Setting	Transmission Specifications			
	Switch Setting	Switch	Setting Item	Setting Contents
CH1 CH2  CH1: Set to No. 6 (non procedure protocol) CH2: Set to No. 1	 SW 01 02 03 04 05 06 07 08 09 10 11 12	SW01	Operation setting	OFF (independent operation)
		SW02	Data bit setting	OFF (7 bits)
		SW03	Parity bit enable/disable setting	ON (Enable)
		SW04	Even parity/odd parity setting	ON (Even)
		SW05	Stop bit setting	ON (2 bits)
		SW06	Checksum enable/disable setting	OFF (Disable)
		SW07	Enable/disable write during RUN	ON (Enable)
		SW08	Setting modification enable/disable setting	ON (Enable) *
		SW09	Transmission rate setting	OFF
		SW10		OFF
		SW11		ON (4800BPS)
		SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

* When using a user frame, set SW08 (setting modification enable/disable setting) to ON.

3 Bar code reader setting

(TOKEN CO., LTD. TCD-4000/TBR-4000)

Switch			Switch State		
	1	OFF	Baud rate 4800bps		
	2	ON			
	3	OFF			
	4	OFF	Parity (even)		
	5	OFF	Terminator STX/ETX		
	6	ON	Bar code type	ON	USS-39 (code 39)
	7	ON		OFF	
	8	ON		OFF	
	9	OFF	JAN	OFF	
	10	OFF		OFF	

Transmission specifications

a) RS-232C interface Asynchronous

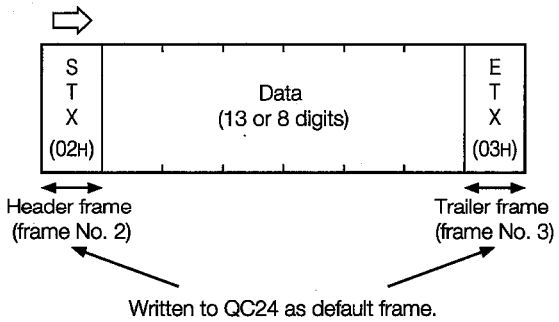
b) 7-bit ASCII code

c) Data specifications

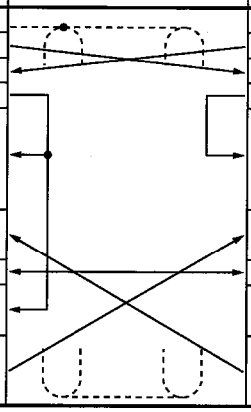
- Start bit ..... 1 bit
- Data ..... 7 bits
- Parity (even) ..... 1 bit
- Stop bit ..... 2 bits

d) Baud rate 300~19200(bps) selectable

Data format sent from bar code reader



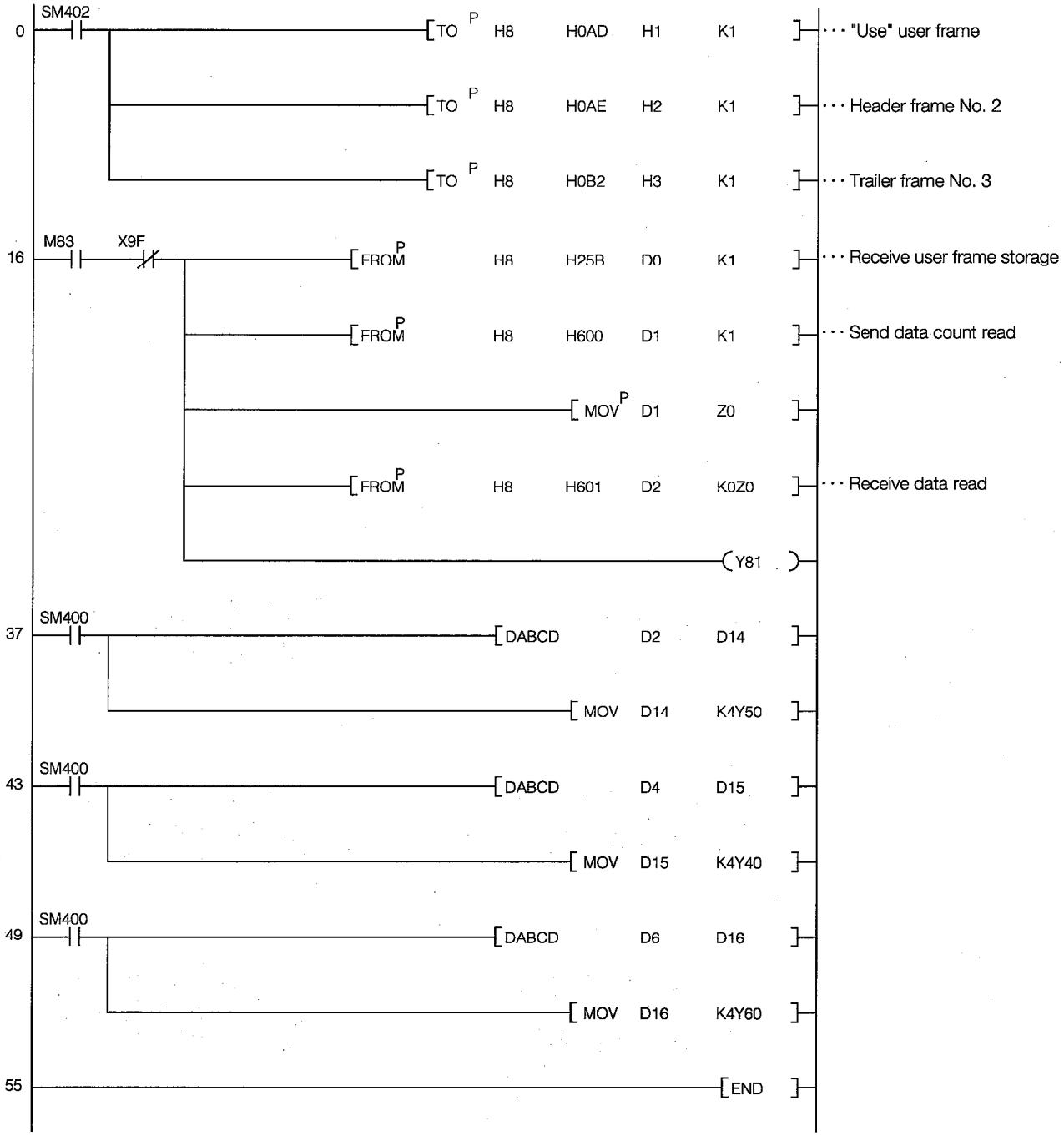
4 External wiring

AC71QC24		Connection and Signal Direction	Bar Code Reader		Name	Contents (Based on bar code reader)
Signal Name	Pin No.		Pin No.	Signal Name		
FG	1		1	FG	Frame Ground	Cable shield pin.
SD	2		2	SD	Send Data	Signal pin that sends the actual send data.
RD	3		3	RD	Receive Data	Signal pin that receives the actual receive data.
RS	4		4	RS	Request to Send	Signals that are turned ON when the local station is
CS	5		5	CS	Clear to Send	ready to send and simultaneously sends (simplified) a signal to the CS pin of the local station and informs the opposite device that there is data to be sent.
DR	6		6	DR	Data set Ready	Pin that receives the Ready signal from the opposite device.
SG	7		7	SG	Signal Ground	Signal ground pin.
CD	8		8	CD	Carrier Detect	Pin that receives the ON signal when there is send data from the opposite device.
ER	20		20	ER	Equipment Ready	Pin that informs the opposite device that the local station is ready to operate.

5 Bar code examples

USS-39 (code 39)

6 Sequence program



7 Trainer operation

① Set the AJ71QC24 mode switches and transmission specifications switches and the bar code reader switches to the specified values.

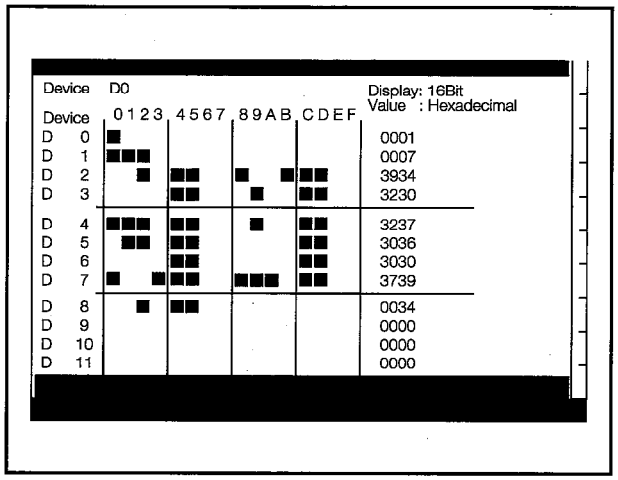
(See item 2 for the QC24 switch settings and item 3 for the bar code reader switch settings.)

② Write the sequence program shown in item 6 to the PC CPU.

After writing, set the PC CPU to "RUN".

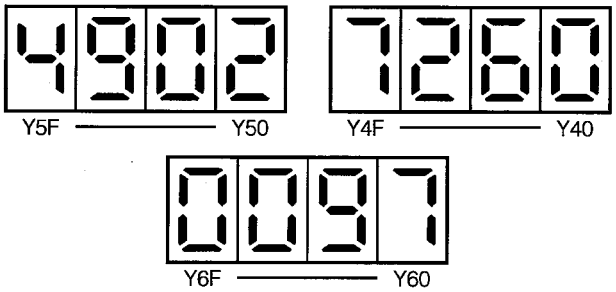
③ Read the bar code value with the bar code reader.

To check the read data, monitor it with the GPPQ.



The receive data is stored to D2 to D8. Check if the header frame (STX:02H) and trailer frame (ETX:03H) were removed from the received data.

④ Check if the bar code No. was stored to the Y40 to Y6F I/O panels.



5.7.3 Registering user frames to EEPROM

This section Registers user frames to the EEPROM.

Use the registered data with Section 5.8.

Set the system configuration and AJ71QC24 as described in Section 5.7.2.

1 Contents of user frames

User frame

No.3E8H(1000) registered data

R	E	S	U	L	T	S	
52H	45H	53H	55H	4CH	54H	53H	(7 bytes)

User frame

No.3E9H(1001) registered data

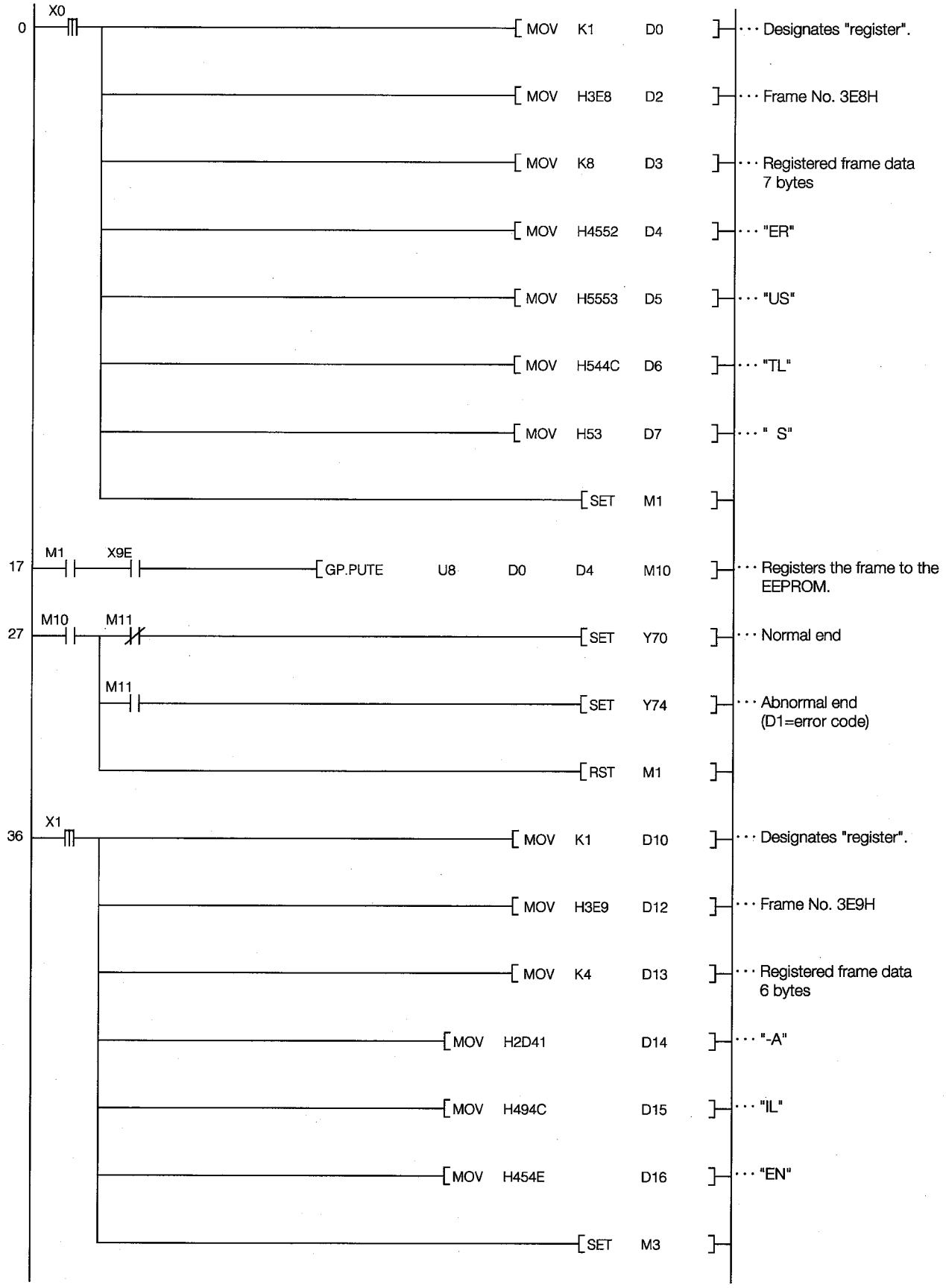
A	—	L	I	N	E	
41H	2DH	4CH	49H	4EH	45H	(6 bytes)

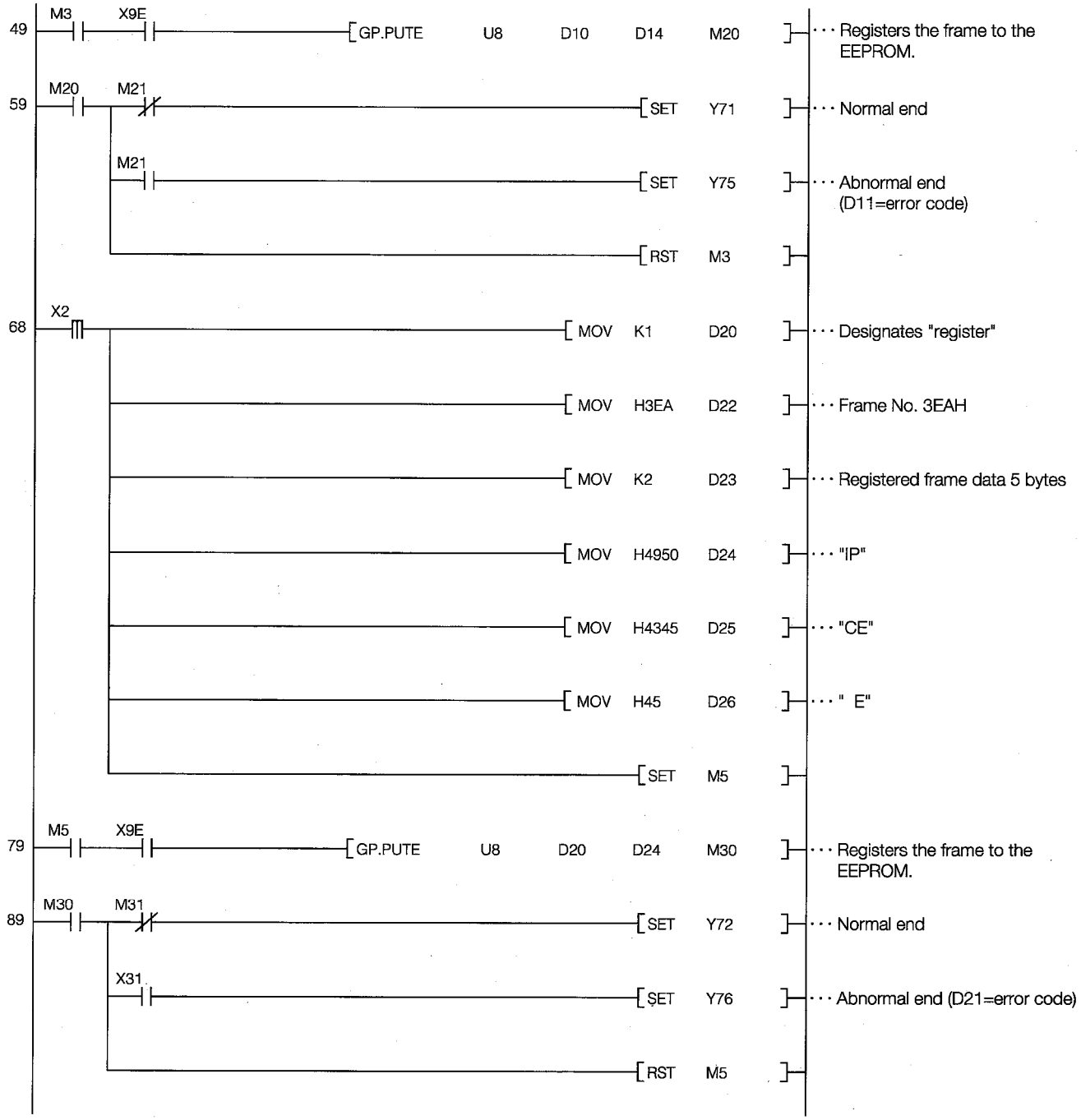
User frame

No.3EAH(1002) registered data

P	I	E	C	E	
50H	49H	45H	43H	45H	(5 bytes)

2 Sequence program




49
M3
X9E
GP.PUTE U8 D10 D14 M20
Registers the frame to the EEPROM.
59
M20
M21
SET Y71
Normal end
M21
SET Y75
Abnormal end (D11=error code)
RST M3
68
X2
MOV K1 D20
Designates "register"
MOV H3EA D22
Frame No. 3EAH
MOV K2 D23
Registered frame data 5 bytes
MOV H4950 D24
"IP"
MOV H4345 D25
"CE"
MOV H45 D26
" E"
SET M5
79
M5
X9E
GP.PUTE U8 D20 D24 M30
Registers the frame to the EEPROM.
89
M30
M31
SET Y72
Normal end
X31
SET Y76
Abnormal end (D21=error code)
RST M5

3 Trainer operation

① Set the AC71QC24 mode switches and transmission specifications switches as described in Section 5.7.2 [2].

Always set SW08 (setting modification enable/disable switch) of both CH1 and CH2 to ON.

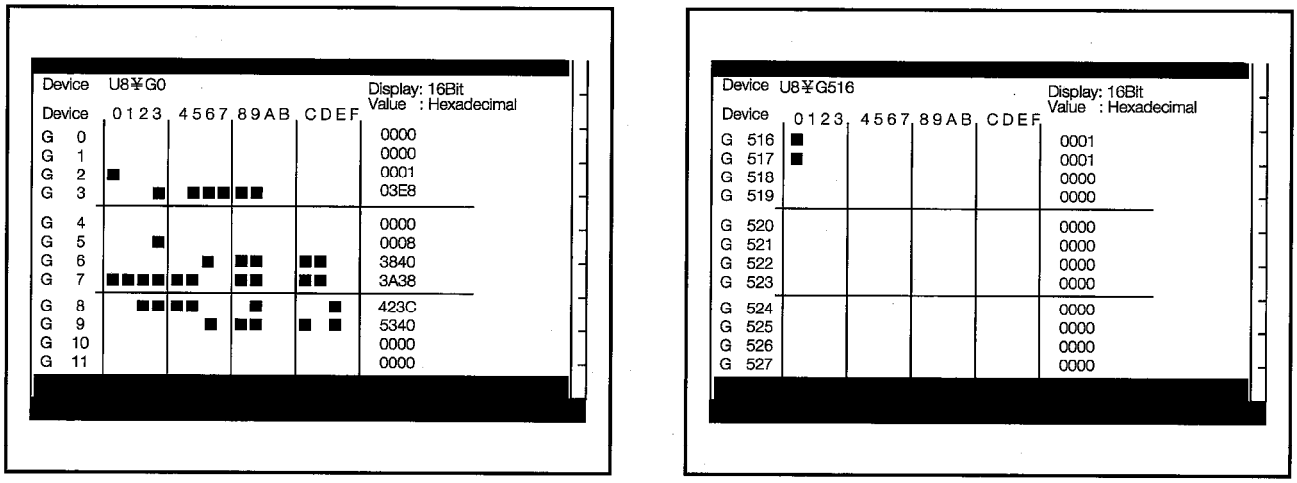
(If a switch is set to OFF, the QC24 will recognize an EEPROM write error.)

② Write the sequence program shown in item [2] to the PC CPU.

After writing, set the PC CPU to "RUN".

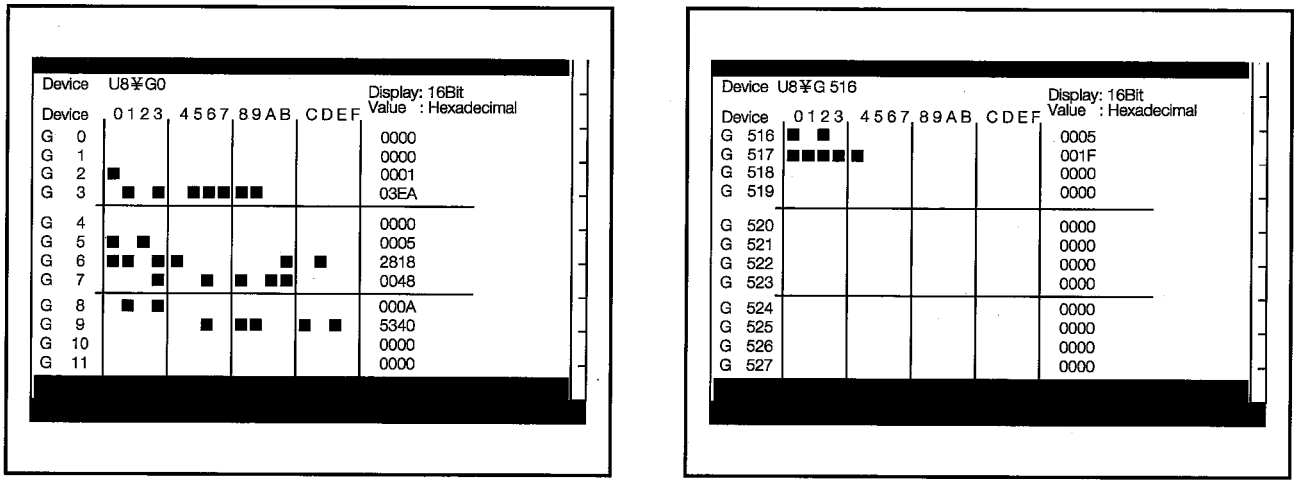
③ When X0 is turned ON, frame No.3E8H is written to the EEPROM.

Monitor the data with the GPPQ.



GPPQ buffer batch monitor screen

④ When X1 and X2 are turned ON, frame Nos. 3E9H and 3EAH are written to the EEPROM, respectively.



GPPQ buffer batch monitor screen

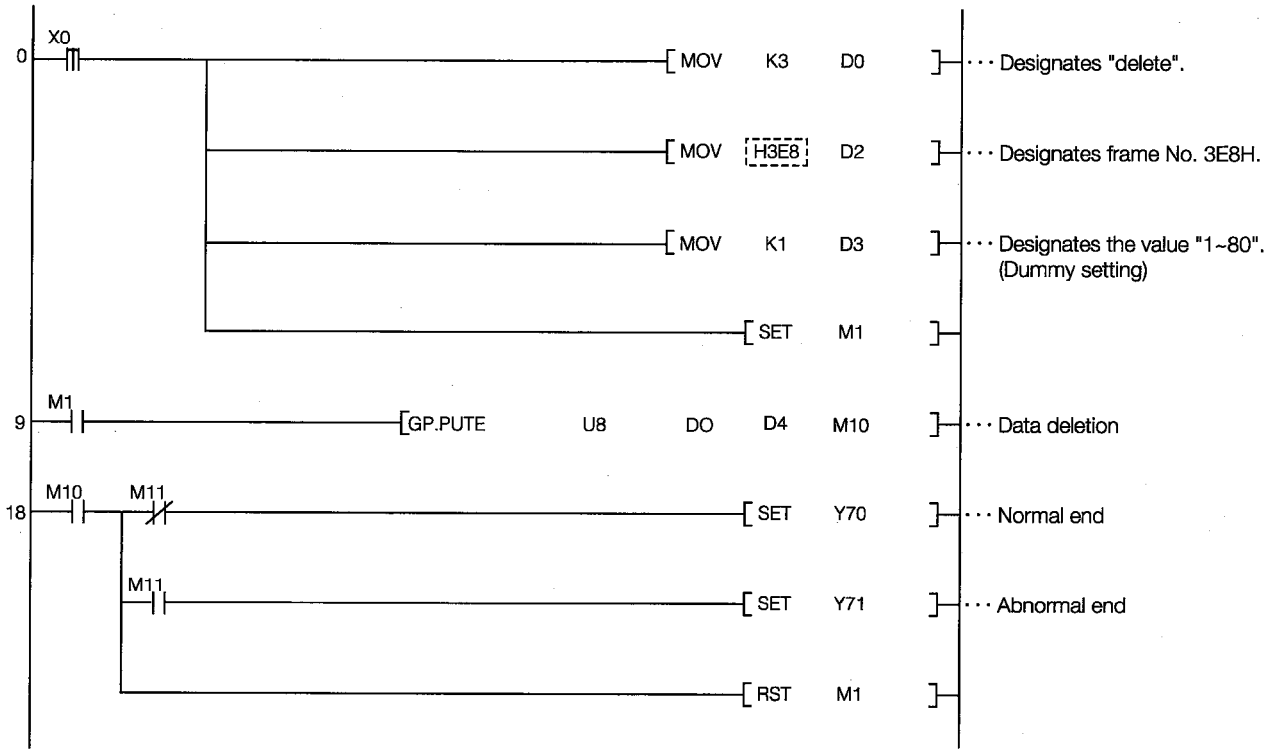
⑤ After all the frames have been written, perform the operation described in Section 5.8.

If you have any situations as follows, please read the explanations

• Frame is not written to EEPROM	• Was the sequence program (Section 5.7.3 [2]) written to the PC CPU? • Are the QC24 switches settings (Section 5.7.2. [2]) correct ? • An error code is stored to a data register (D1, D11, D21, D31, D41). Check the error code, then check the settings.
• Error data was written to EEPROM	• EEPROM data cannot be overwritten. When you want to write new data, the data of the given frame No. must be deleted first. Refer to the delete program and delete the data, then write the new data.

<Delete program>

The following shows a sample program that deletes the frame No. 3E8H data.



<Operation>

When X0 is turned ON, the frame No. 3E8H data is deleted.
When you want to change fram No. , above program ☐ must be changed.

5.8 ASCII↔BIN Conversion

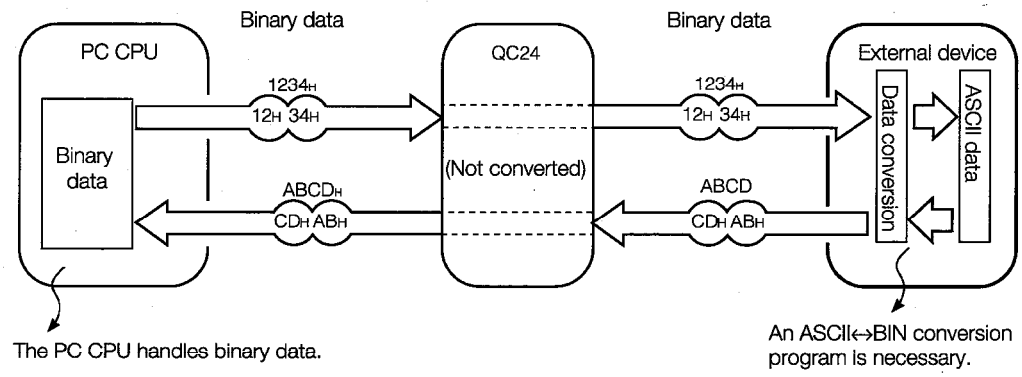
5.8.1 What is ASCII↔BIN conversion?

The QC24 converts the ASCII data received from an external device to binary data and stores the binary data to its buffer memory. The PC CPU converts the designated binary data to ASCII data and sends the ASCII data to the external device.

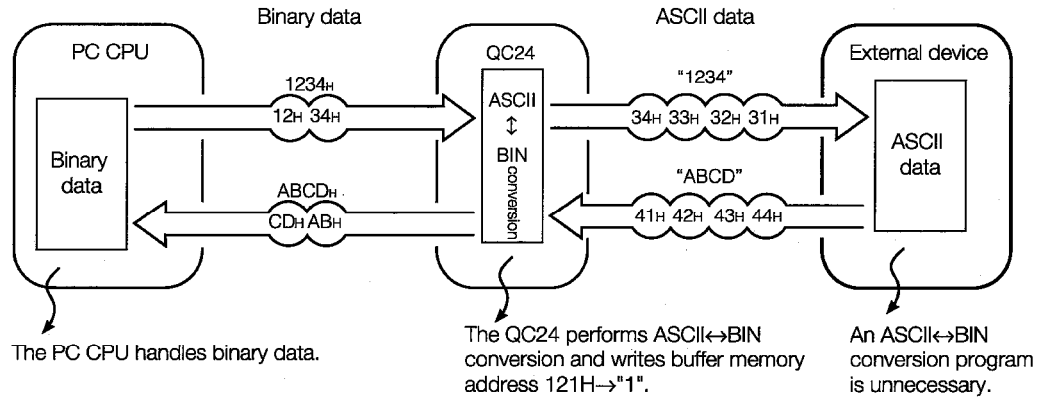
This function is called "ASCII↔BIN conversion".

Data communications matched to the specifications of the external device can be easily implemented by designating QC24 buffer memory address 121H. A data exchange program is unnecessary.

Binary data communications



ASCII data communications



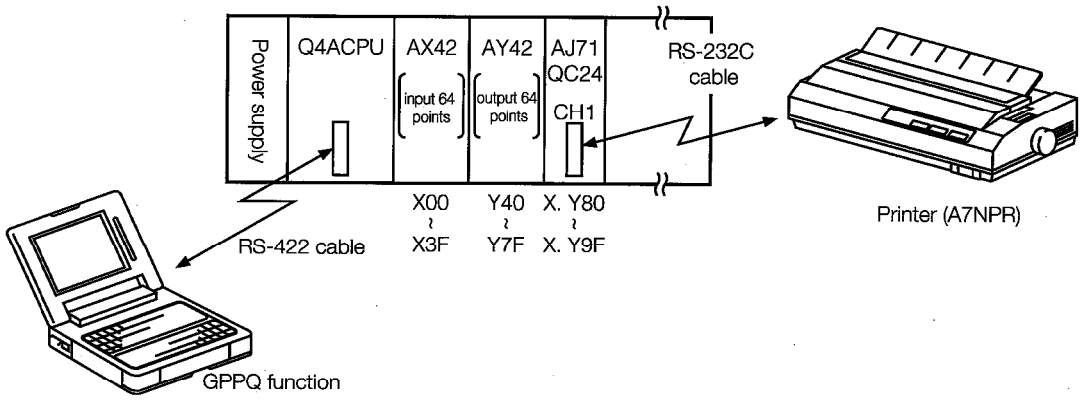
5.8.2 Practice using ASCII↔BIN conversion

Use ASCII↔BIN conversion and print the PC data to a printer.

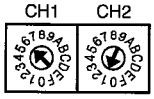
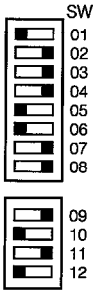
1 System configuration

This system writes the data from the PC to the QC24 buffer memory (non procedure transmission area).

The printer receives and prints the data from the QC24.



2 AJ71QC24 settings

Mode Setting	Transmission Specifications			
	Switch Setting	Switch	Setting Item	Setting Contents
CH1 CH2  CH1: Set to No. 6 (non procedure protocol) CH2: Set to No. 1		SW01	Operation setting	OFF (independent operation)
		SW02	Data bit setting	ON (8 bits)
		SW03	Parity bit enable/disable setting	ON (Enable)
		SW04	Even parity/odd parity setting	ON (Even)
		SW05	Stop bit setting	OFF (1 bits)
		SW06	Checksum enable/disable setting	OFF (Disable)
		SW07	Enable/disable write during RUN	ON (Enable)
		SW08	Setting modification enable/disable setting	ON (Enable) *
		SW09	Transmission rate setting	ON
		SW10		OFF
		SW11		ON (9600BPS)
		SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

* Since user frames are used, set SW08 to ON.

3 A7NPR transmission specifications

(a) RS-232C interface Asynchronous

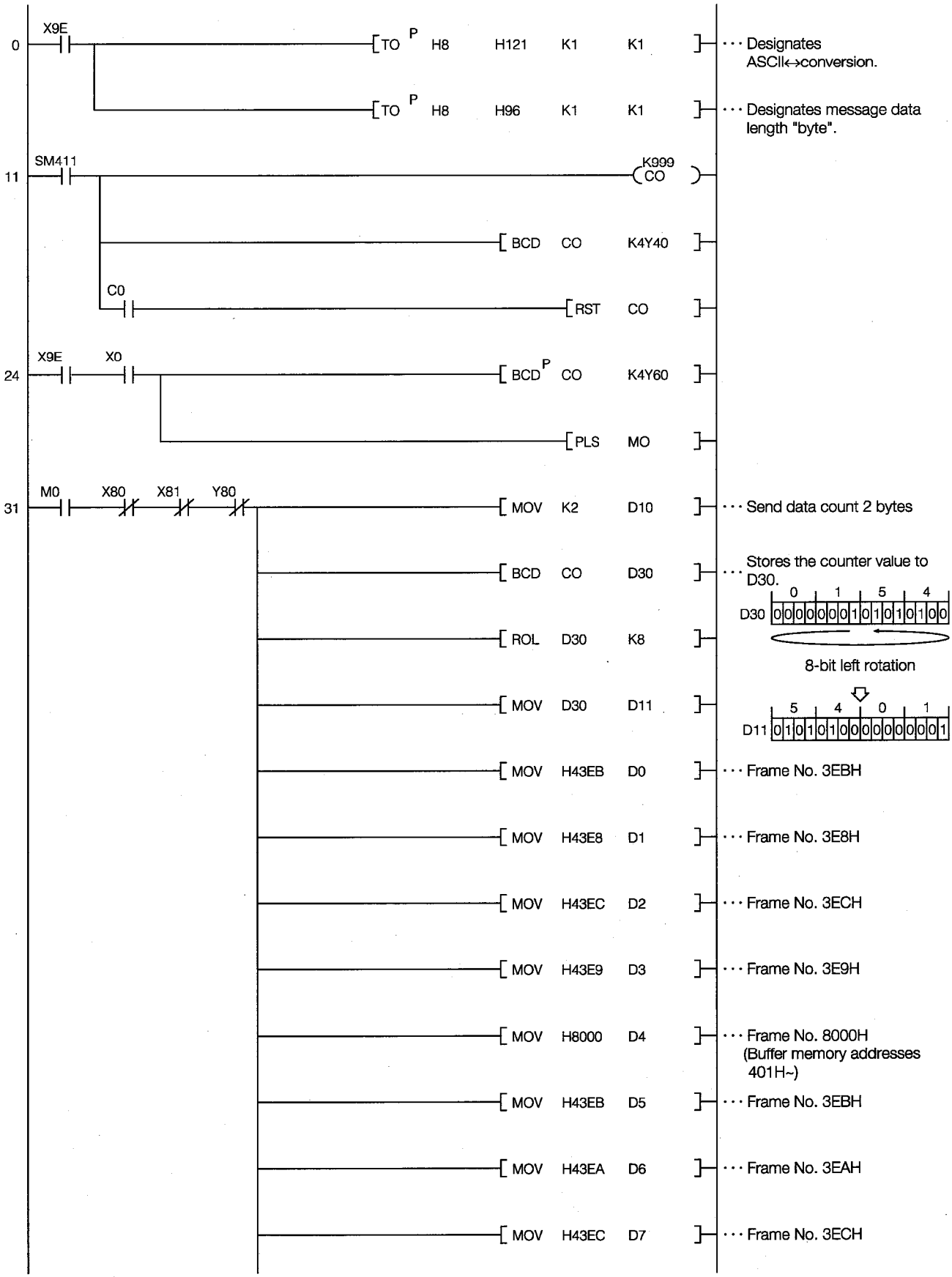
(b) Mode Half-duplex

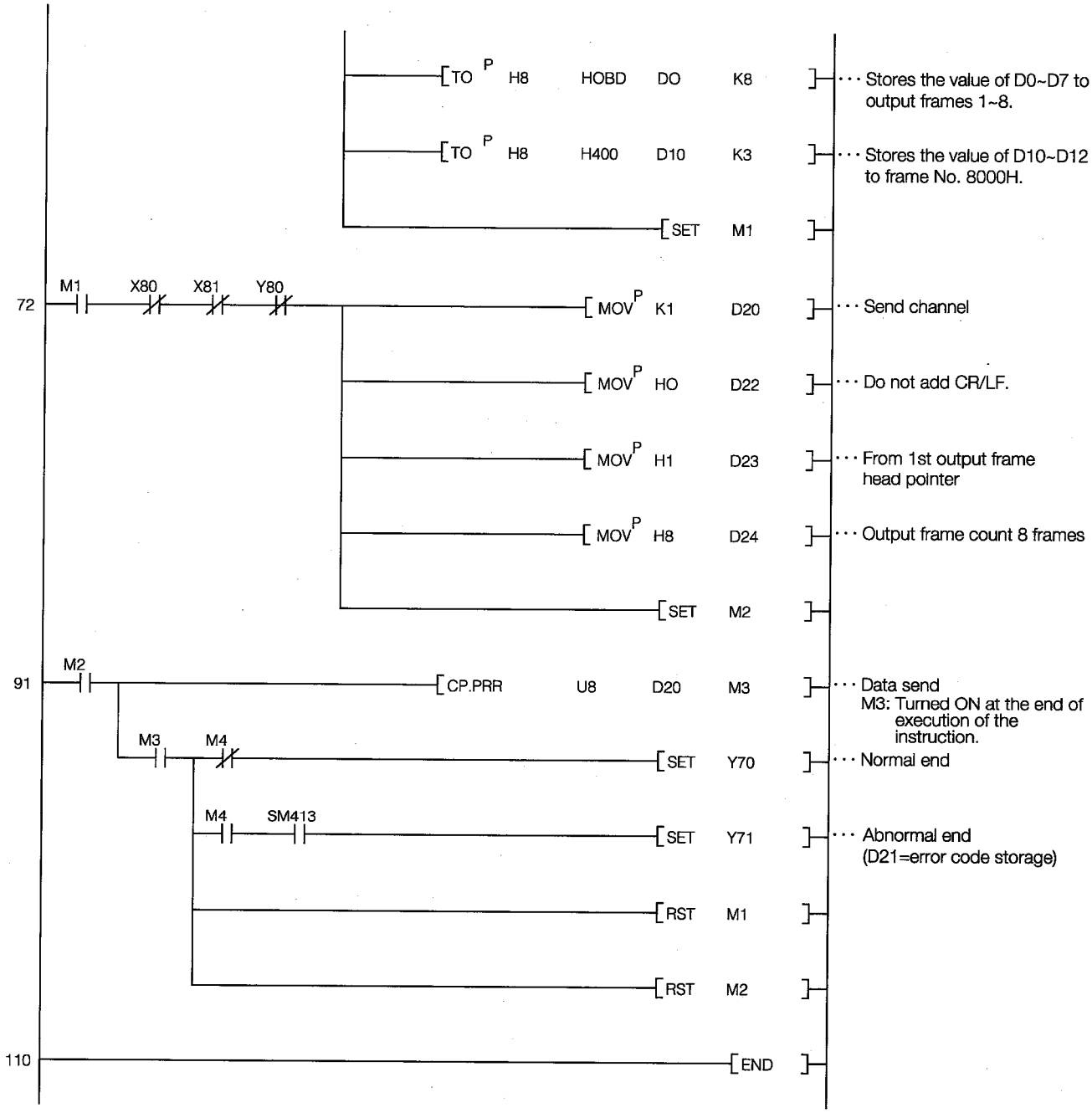
(c) Baud rate 9600bps

(d) Data format 11 bits/character

- Start bit ..... 1 bit
- Data ..... 8 bits
- Parity (even) .... 1 bit
- Stop bit ..... 1 bit

4 Sequence program




TO P H8 HOBD DO K8
··· Stores the value of D0~D7 to output frames 1~8.
TO P H8 H400 D10 K3
··· Stores the value of D10~D12 to frame No. 8000H.
SET M1
72
M1 X80 X81 Y80
MOV P K1 D20
··· Send channel
MOV P HO D22
··· Do not add CR/LF.
MOV P H1 D23
··· From 1st output frame head pointer
MOV P H8 D24
··· Output frame count 8 frames
SET M2
91
M2
CP.PRR U8 D20 M3
··· Data send
M3: Turned ON at the end of execution of the instruction.
M3 M4
SET Y70
··· Normal end
M4 SM413
SET Y71
··· Abnormal end
(D21=error code storage)
RST M1
RST M2
110
END

5 Trainer operation (Perform the operations described in Section 5.7.3 beforehand.)

① Set the AJ71QC24 mode switches and transmission specifications switches to the values given in item 2 above.

② Write the sequence program shown in item 4 above to the PC CPU.

After writing, set the PC CPU to RUN.

③ When X0 is turned ON, the frame written in Section 5.7.3 and the value indicated by Y60 to Y6F are printed on the printer.

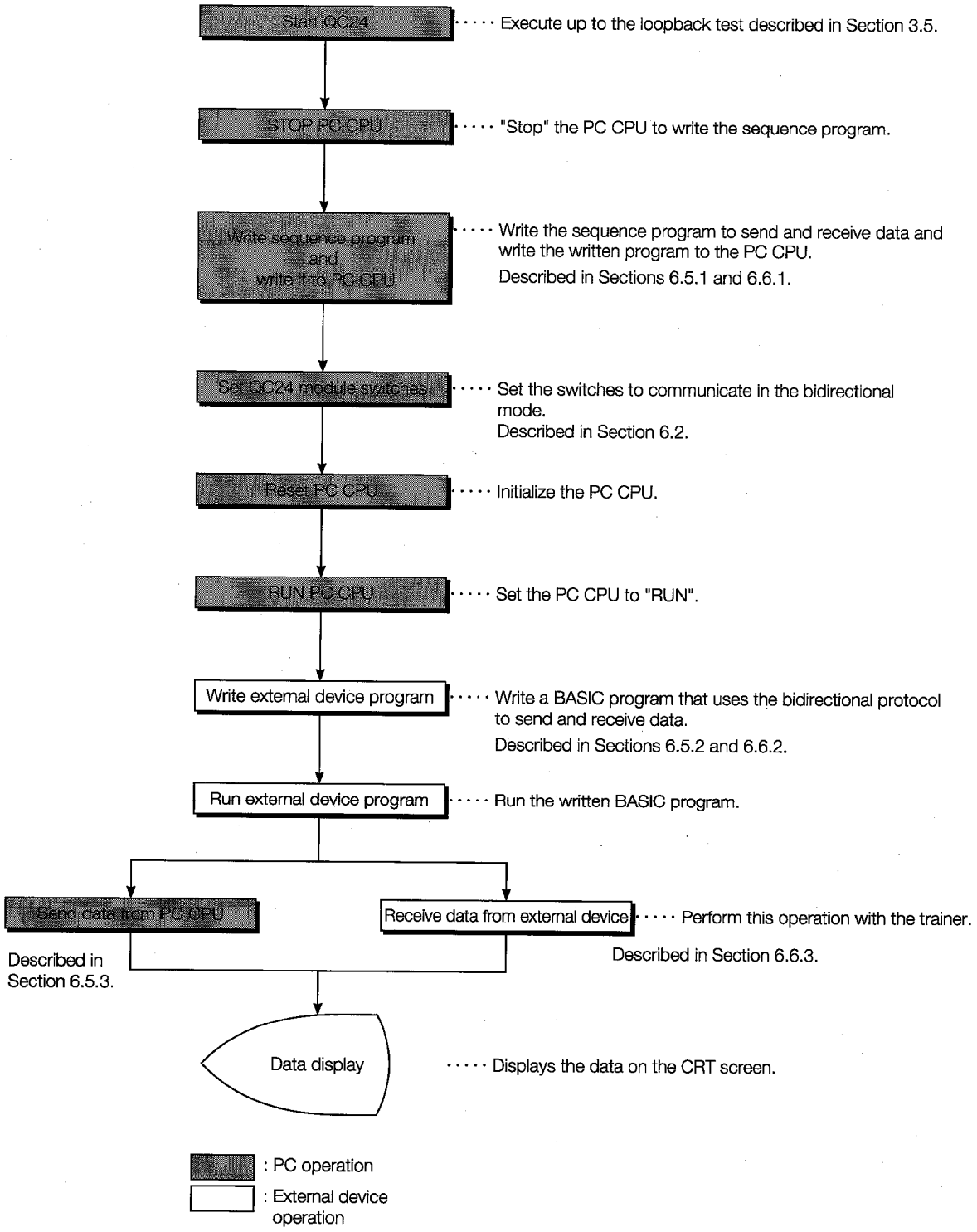
PRODUCTION RECORD

A LINE □PCS

↑——— Value indicated by Y60~Y6F.

6 BIDIRECTIONAL PROTOCOL COMMUNICATIONS

The following shows the bidirectional protocol communications procedure used in this guidebook.

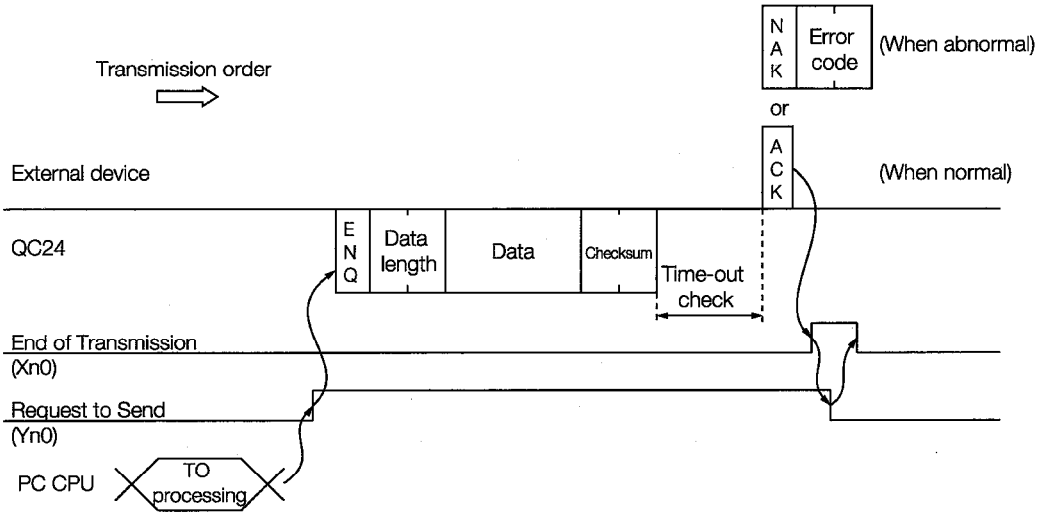


6.1 Bidirectional Protocol Format and Data Setting Items

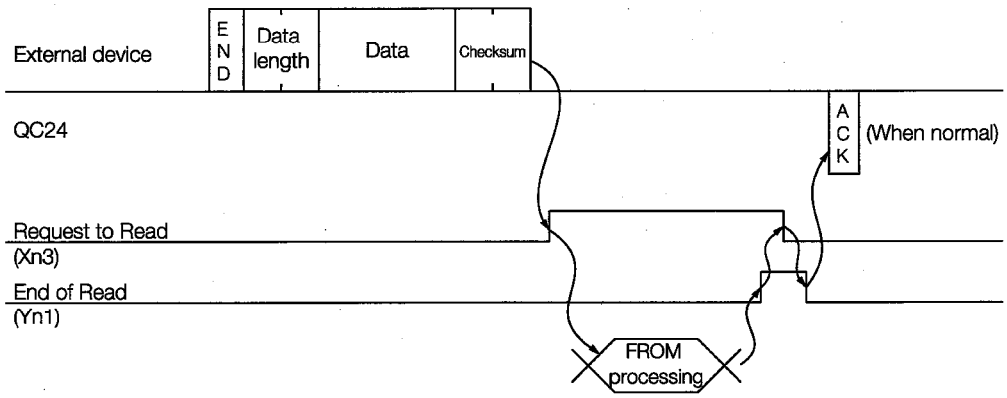
The following describes the control procedure and the contents of the setting items designated with the control procedure when using the bidirectional protocol basic format to send and receive data.

6.1.1 Control procedure and message structure

- Data transmission from PC CPU to external device



- Data reception from external device at PC CPU



6.1.2 Data setting items

The following describes the contents of the data that are designated in the bidirectional mode control procedure.

1 Control code

The following table shows the control codes.

Control Code	Code (Hexadecimal)	Contents	Application
ENQ	05H	Enquily	Indicates the start of the send data.
ACK	06H	Acknowledge	Response to opposite device when data communications ended normally.
NAK	15H	Negative Acknowledge	Response to opposite device when data communications ended abnormally.

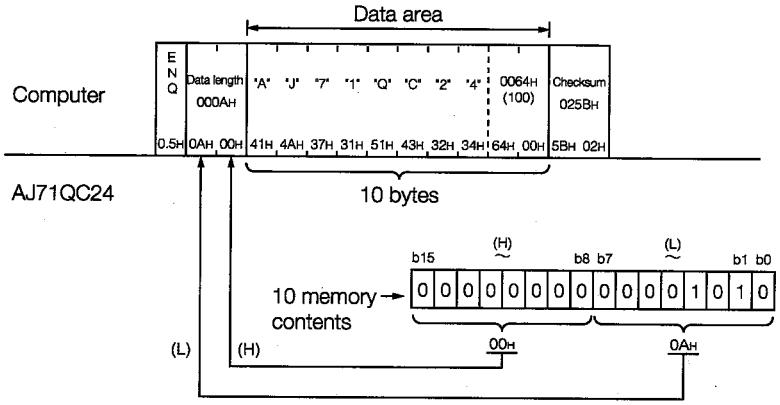
2 Data length

Two-byte binary data represents the number of bytes or number of words of the data area in a message.

The contents set to QC24 buffer memory address 96H (bidirectional mode word/byte designation area) determines the data length units.

In this guidebook, data is sent and received in word units.

[Example] The data length when "AJ71QC24" and the value 100 are sent as data is shown below. (Units: Bytes)



3 Data area

This is the ordinary 1-byte data list that is to be sent to the opposite device. Data codes 00H~FFH are handled.

Note

Checksum

Checksum is the value of the lower 2 bytes (16 bits) of the result (sum) addition of the data length and data area in the message as binary data.

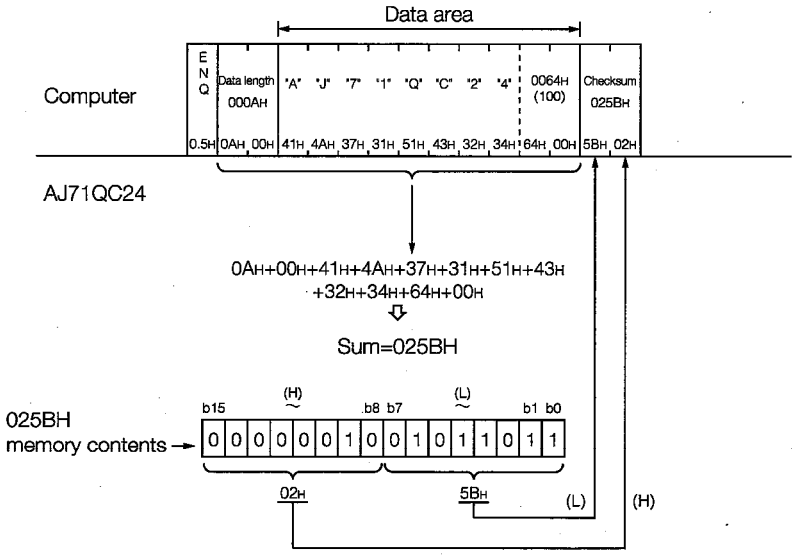
When the AJ71QC24 transmission specifications checksum enable/disable switch (SW06) is set to OFF, checksum is not handled. Therefore, the message does not have to include a sum check code.

When checksum is not handled, when a data area of the data length in the message is received, the receive data up to the next control code (see item [1] above) is ignored.

In this guidebook, the transmission specifications checksum enable/disable switch is set to OFF. Therefore, checksum is not handled.

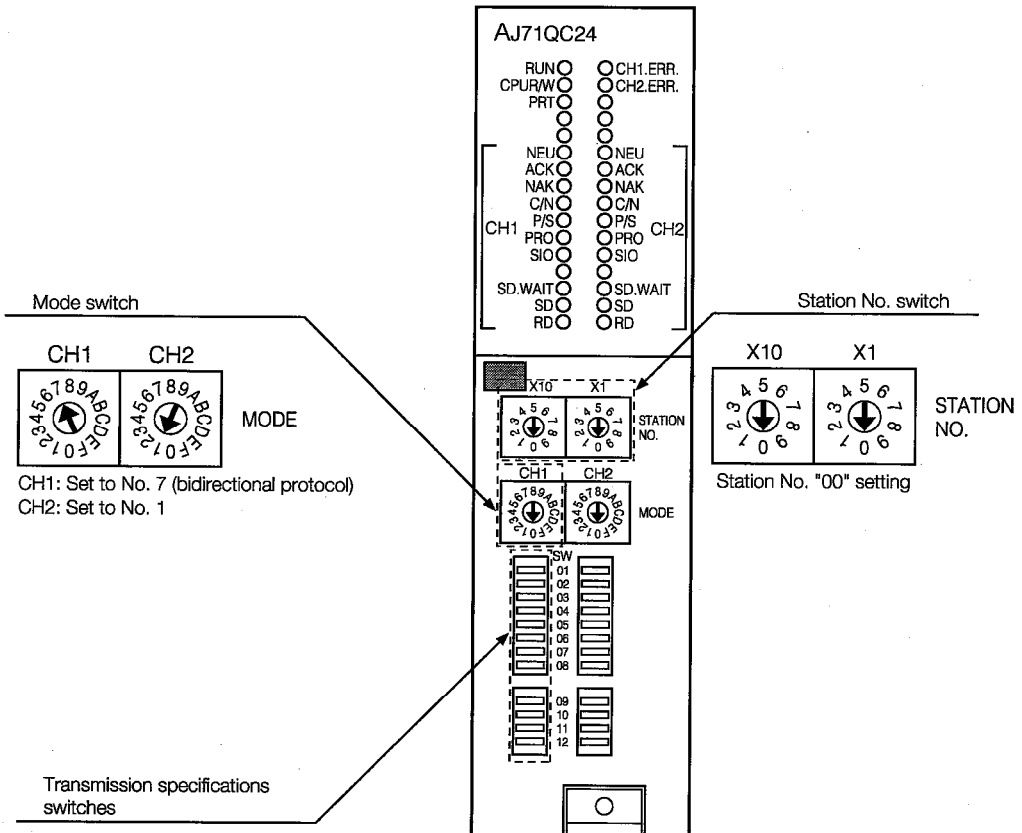
The following shows an example of the contents of the checksum in a message for your reference.

[Example] Checksum when "AJ71QC24" and the value 100 are send as data is shown below. (Units: Bytes)



6.2 AJ71QC24 Serial Communications Module Settings

Set the AJ71QC24 switches as shown below.



Switch Setting	Switch	Setting Item	Setting Contents
	SW01	Operation setting	OFF (independent operation)
	SW02	Data bit setting	OFF (8 bits)
	SW03	Parity bit enable/disable setting	ON (Enable)
	SW04	Even parity/odd parity setting	ON (Even)
	SW05	Stop bit setting	OFF (1 bit)
	SW06	Checksum enable/disable setting	OFF (Disable)
	SW07	Enable/disable write during RUN	ON (Enable)
	SW08	Setting modification enable/disable setting	OFF (Disable)
	SW09	Transmission rate setting	OFF
	SW10		OFF (4800BPS)
	SW11		ON
	SW12		OFF

Set switches SW13 to SW15 on the left side of the module to OFF.

6.3 Bidirectional Protocol LED Display Contents

The following shows the QC24 LED display contents related to bidirectional protocol data communications using the system configuration shown in Section 3.1.1.

LED Name	LED Application	ON	OFF	Error Processing
RUN	QC24 operation status display	Module normal	Module abnormal	• Check SW09~SW12. • Check the number set at the mode switches.
CPU R/W	QC24 and PC CPU communications status	Blinks while communicating (Lights steadily while stopped)	Abnormal	• Check the status of the PC CPU and external device. • Check the number set at the mode switches.
1-NEU	———	Abnomal	Nomal	• Check the number set at the mode switches.
1-C/N	QC24 and PC CPU communications status display	Communications abnormal	Communications normal	• Check SW07 (enable/disable write during RUN).
1-P/S	Parity error display	Error	Normal	Transmission specifications settings and receive data are not the same. • Make the QC24 transmission specifications and the external device specifications the same. • Check the setting of the transmission specifications switches.
1-SIO	Displays the data reception status	Error	Normal	Transmission specifications settings and receive data are not the same. • Make the QC24 transmission specifications and the external device specifications the same. • Check the setting of the transmission specifications switches. • Decrease the transmission rate. • Decrease the number of data sent from the external device.
1-SD · WAIT	Displays the status of the data the QC24 sends to the external device	Waiting to send	Sending No send data	———
1-SD	Displays QC24 to external device send status	Blinks while data is being sent	Data not being sent	• If this LED does not blink even even though external device is sending a command, check the cable connections.
1-RD	Displays the external device to QC24 receive status	Blinks while data is being received	Data not being received	
CH1. ERR	CH1 communications status display	CH1 error	Normal	———

□ : Normal state.

If an error is generated, the error LED come ON.

Thereafter, the error LED will remain ON even if the error recovers.

Use the buffer memory (201H) to turn OFF the error LED.

6.4 Bidirectional Protocol Setting Items

The following describes the buffer memory and input/output signals that are set when using the bidirectional protocol to send and receive data in this guidebook. (See Appendixes 1 and 2 for the QC24 buffer memory table and input/output signals table.)

6.4.1 Buffer memory setting

With the bidirectional protocol data communications described in this guidebook, the sequence program reads and writes only the areas shown below.

Address Hexadecimal (Decimal)	Buffer Memory Address Name			Default Value
0H(0)	LED OFF request (right side : RUN~RD)			0
9DH(157)	Response watchdog (timer 1) designation			32H (5 seconds)
257H(599)	Data transmission result			0 (No errors)
258H(600)	Data reception result			0 (No errors)
400H(1024)	CH1 send/ receive	Send area	Send data count	0
401H(1025) to 5FFH(1535)			Send data (storage) 511 words	0
600H(1536)		Receive area	Receive data count (read data count)	0
601H(1537) to 7FFH(2047)			Receive data (storage) 511 words	0

6.4.2 PC CPU input/output signals

The following shows the input/output signals used with the bidirectional protocol.

1 Input signals (AJ71QC24→PC CPU)

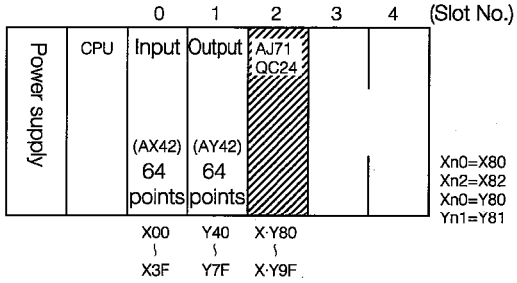
Input Signal	Signal Name	Contents
Xn0	Transmission normal end	Turned ON at the end of transmission in response to a Request to Send signal from the PC CPU.
Xn1	Transmission abnormal end	Turned ON when an error was detected during transmission.
Xn2	Sending	Turned ON while the QC24 is sending data to the PC CPU.
Xn3	Receive data read request	Turned ON when data is received from the external device after the QC24 is started.
Xn4	Receive error detected	Turned ON when an error was detected while receiving data.
Xn1E	QC24 Ready signal	Turned ON when the QC24 is ready to operate.
Xn1F	Watchdog timer error	Turned ON when the QC24 cannot operate normally.

2 Output signals (PC CPU→AJ71QC24)

Output Signal	Signal Name	Contents
Yn0	Request to Send	Turned ON when the data sent to the QC24 buffer memory from the PC CPU is sent to the externa device after the QC24 is started.
Yn1	End of Receive Data Read	Turned ON by the QC24 when the PC CPU has finished reading the data received from the external device.

(Note) Xn0 and Yn0 are determined by the slot into which the QC24 is inserted.

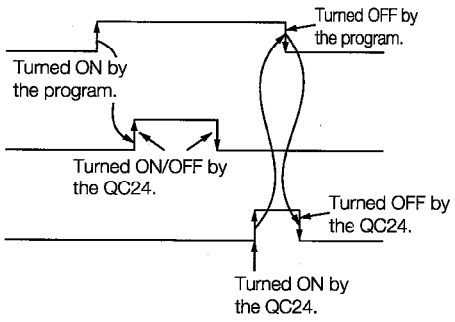
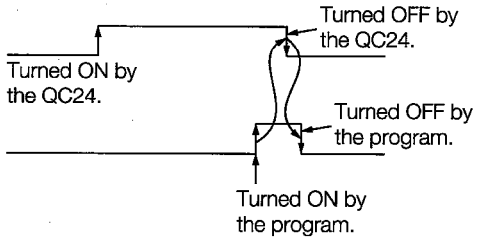
With the system configuration shown below, the QC24 is allocated to X·Y80~9F.



3 Bidirectional protocol handshake signals

The signal that sends the data output from the sequence program to the external device, the signal that detects that data has arrived from the external device and enables the sequence program to read the received data, and similar signals used when sending and receiving data with the bidirectional protocol are called handshake input/output signals. These signals are necessary in the bidirectional mode.

The following shows the handshake input/output signals.

	Signal	Timing
PC CPU ↓ External device	Yn0 (Request to Send) Xn2 (Sending) Xn0 (End of Transmission)	
External device ↓ PC CPU	Xn3 (Request to Read Receive Data) Yn1 (End of Reading of Receive Data)	

In this guidebook, bidirectional protocol data communications is implemented using the AJ71QC24 dedicated instructions.

Data transmission from PC CPU: BIDOUT instruction (Section 6.5)

Data reception from external device: BIDIN instruction (Section 6.6)

When the dedicated instructions are used, data can be sent and received without being aware of the handshake signals and input/output signals above and buffer memory address. The data send and receive programs also become simple.

6.5 Data Transmission From PC CPU

This section sends data from the PC CPU to an external device.

The send data count units are word units.

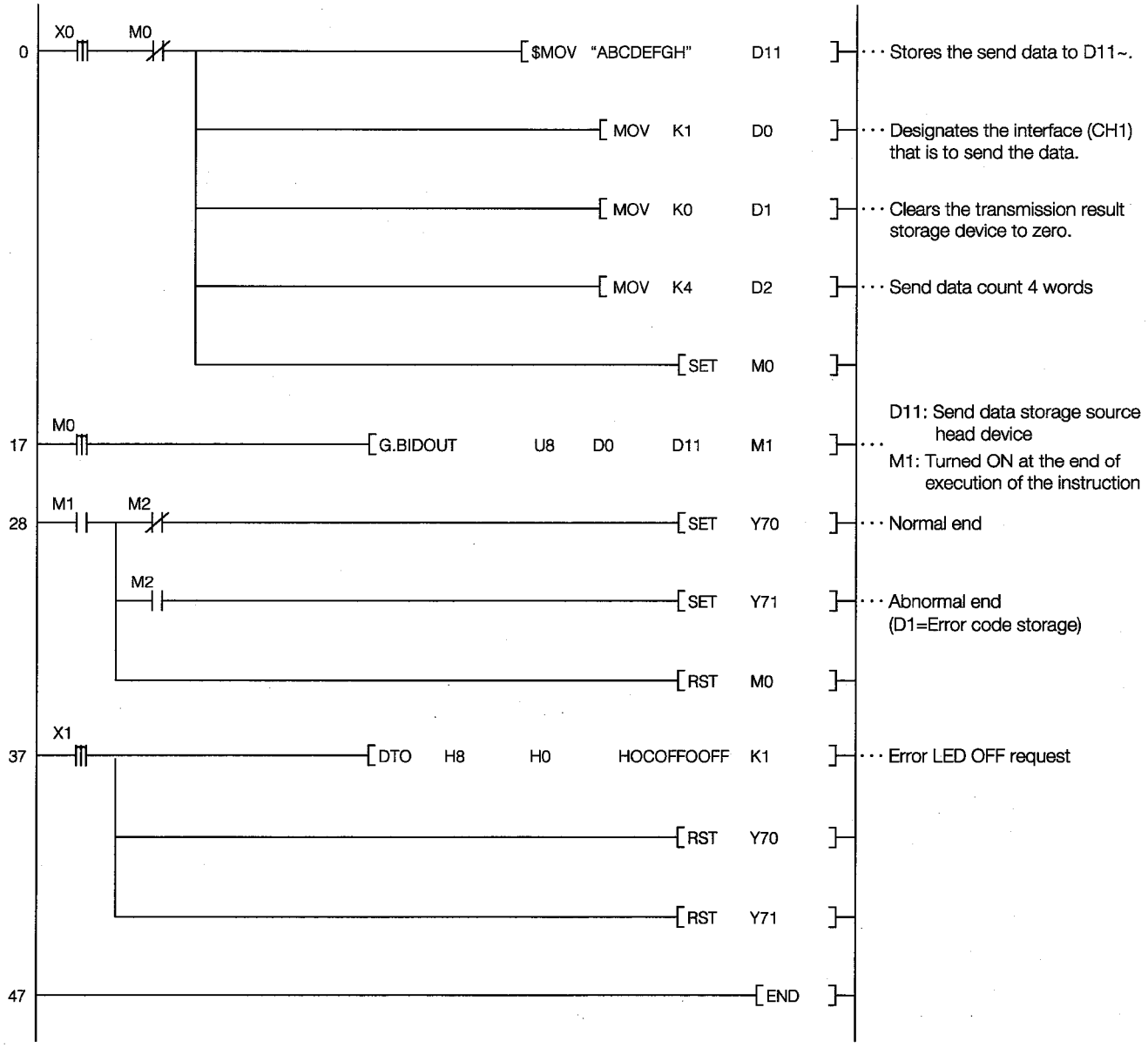
[Practice Contents]

When X0 is turned ON, the character data (ASCII code) [ABCDEFGH] is written to the QC24 buffer memory (addresses 400H~404H).

When X1 is turned ON, the character data [ABCDEFGH] is sent to the external device using the bidirectional protocol.

When the PC CPU receives a response from the external device, it ends transmission.

6.5.1 Sequence program

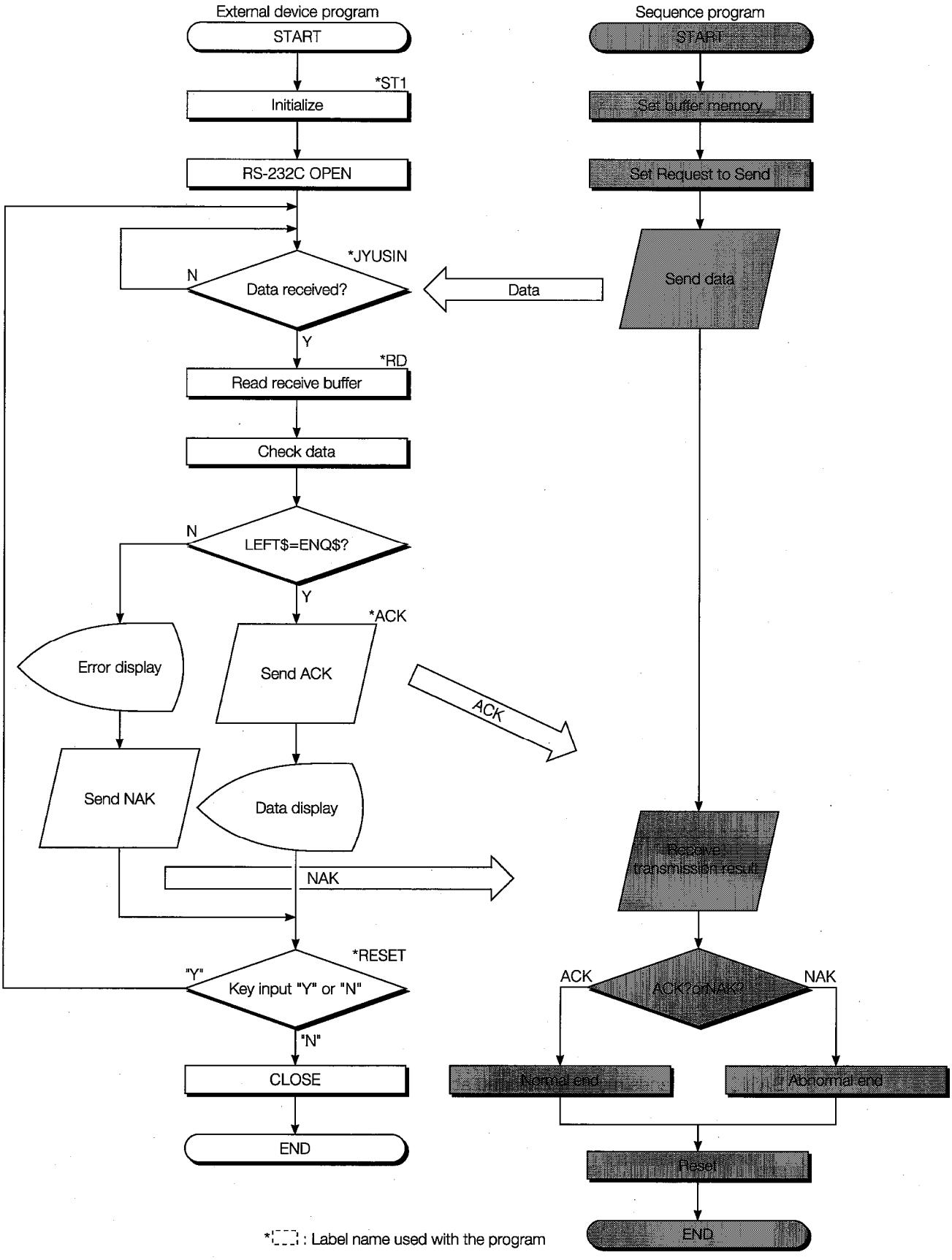


6.5.2 External device program

```
1000 ' !  ------------------------------------------------------------------------!
1010 ' !                  AJ71QC24 Bidirectional Mode Sample Program                !
1020 ' !                       (Data transmission from PC CPU)                      !
1030 ' !  ------------------------------------------------------------------------!
1040 *ST1
1050     CLS                                :'Clear screen
1060     CH% =1                             :'Channel No.
1070     ENQ$=CHR$(&H5)                     :'ENQ code
1080     ACK$=CHR$(&H6)                     :'ACK code
1090     NAK$=CHR$(&H15)                    :'NAK code
1100     '
1110 ' ! ------------------------- RS-232C open & initialize---------------------------!
1120     OPEN "COM:E81NN" AS #CH%           :'Set communications mode, etc.
1130 ' ! -------------------------------Data reception ---------------------------------!
1140 *JYUSIN
1150     CLS
1160     LOCATE 6,1:PRINT "**Reception from PC in bidirectional mode**"
1170 *JYU1
1180     FOR I%=0 TO 1000 :NEXT I%
1190     IF LOC(CH%)<>0 THEN *RD
1200     LOCATE 10,5:PRINT "Wait to receive data !"
1210     LOCATE 10,8:PRINT "Send data from PC.":GOTO *JYU1
1220     '
1230 ' ! ---------------------------------- Buffer read-------------------------------!
1240 *RD
1250     B$=""
1260 *RD1
1270     FOR I%=0 TO 1000 :NEXT I%
1280     IF LOC(CH%)<>0 THEN B$=B$+INPUT$(LOC(CH%),#CH%) ELSE *RD1
1290     '
1300 ' ! ------------------------------Receive data check ----------------------------!
1310     IF LEFT$(B$,1)=ENQ$ THEN *ACK
1320     E1$=CHR$(&H22)+CHR$(&H0):E2$="0022" :'&H0022=user definition error
1330     COLOR 2: LOCATE 10,5
1340     PRINT ""Communications error!! Error code=" ;E2$:COLOR 0
1350     PRINT #CH%,NAK$;E1$:GOTO *REST    :'Send NAK
1360     '
1370 ' ! ------------------------------ACK transmission ------------------------------!
1380 *ACK
1390     PRINT #CH%,ACK$
1400     '
1410 ' ! ------------------------------Receive data display --------------------------!
1420     DAT$=MID$(B$,4,LEN(B$)-3)
1430     LOCATE 10,5:PRINT "Receive data=";DAT$
1440     '
1450 *REST
1460     LOCATE 10,8:INPUT "Receive again? (Y/N)";Y$
1470     IF Y$="Y" THEN *JYUSIN
1480     '
1490 CLOSE
1500 END
```

* BASIC commands must be changed according to BASIC software.

Data transmission from PC CPU



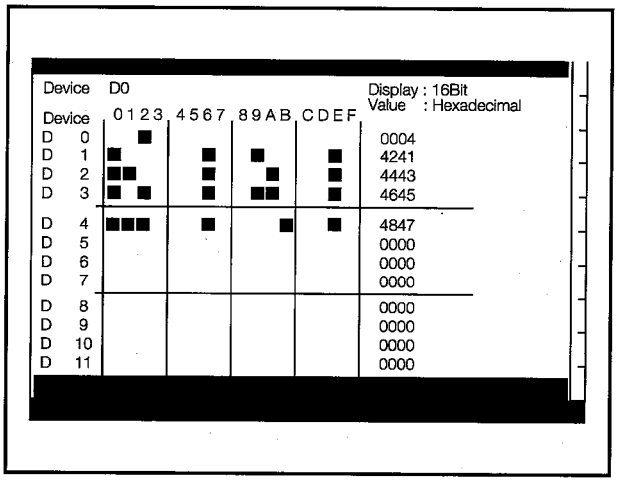
6.5.3 Trainer operation

Write the sequence program shown in Section 6.5.1 to the PC CPU and the BASIC program shown in Section 6.5.2 to the external device.

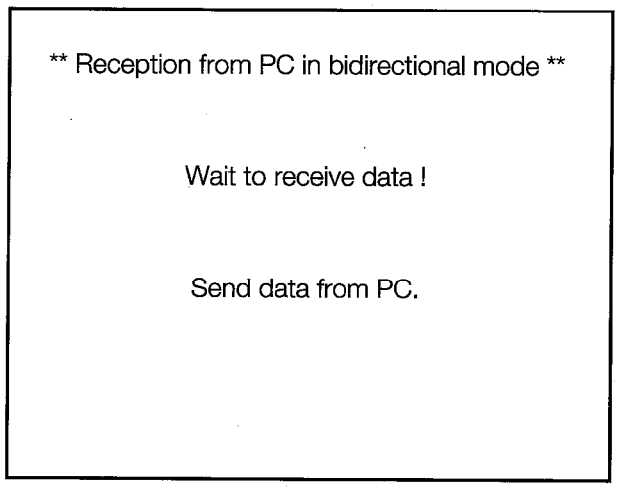
After writing, set the PC CPU and external device to "RUN".

If there are no errors, send and receive data using the following procedure.

① When X0 is turned ON, word count "4" is stored to data register D0 and data "ABCDEFGH" is stored to data registers D1 to D4 and written to QC24 buffer memory addresses 400H to 404H. Check this by monitoring the value of data registers D0 to D4 with GPPQ.

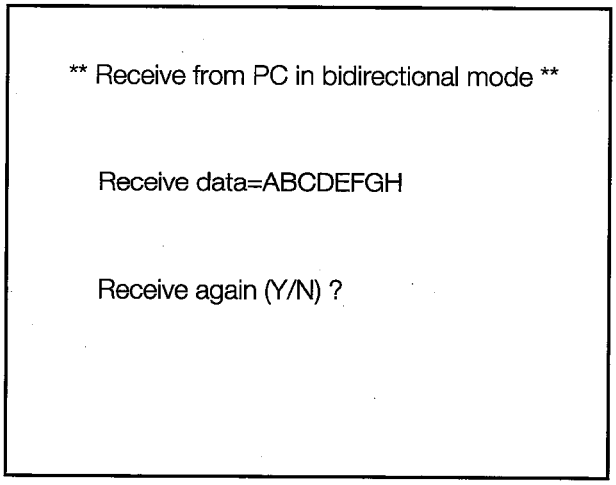


GPPQ batch monitor screen



External device screen

② When X1 is turned ON, the data is sent to the external device.



External device screen

If you have any situations as follows, please read the explanations

- Even though send data is set to the PC CPU and the Request to Send signal is turned ON

```
** Reception from PC in bidirectional mode **

Wait to receive data !

Send data from PC.
```

- Was the sequence program (Section 6.5.1) written to the PC CPU ?
- Are the QC24 mode switch setting (Section 6.2) and cable connections (Section 3.3) and other set tings correct ?

  Check the QC24 LEDs (Section 6.3), then check the settings.

- When sending data from the PC CPU

```
** Reception from PC in bidirectional mode **

Communications error!! Error code=0022

Receive again (Y/N) ?
```

- NAK was sent from the external device.

  Are the external device settings (Section 3.4)and QC24 switch settings (Section 6.2) correct ? Check the QC24 LEDs (Section 6.3), then check the settings.

- When send data and external device receive data are different

```
** Reception from PC in bidirectional mode **

Receive data= □

Receive again (Y/N) ?
```

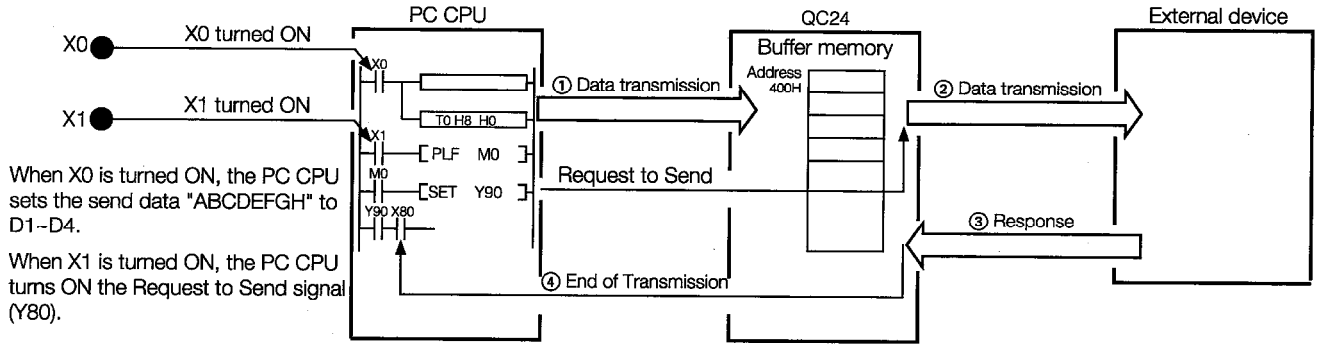
- Are the QC24 switch settings (Section 6.2) and computer progaram (RS-232C setting) settings (Section 6.5.2) correct ?

  Check the QC24 LEDs (Section 6.3), then check the settings.

After resetting the settings, reset the PC CPU and communicate again.

6.5.4 Summary of data transmission from PC CPU using bidirectional protocol

The following uses an image diagram to outline data transmission from the PC CPU using the bidirectional protocol described in Section 6.5.3.



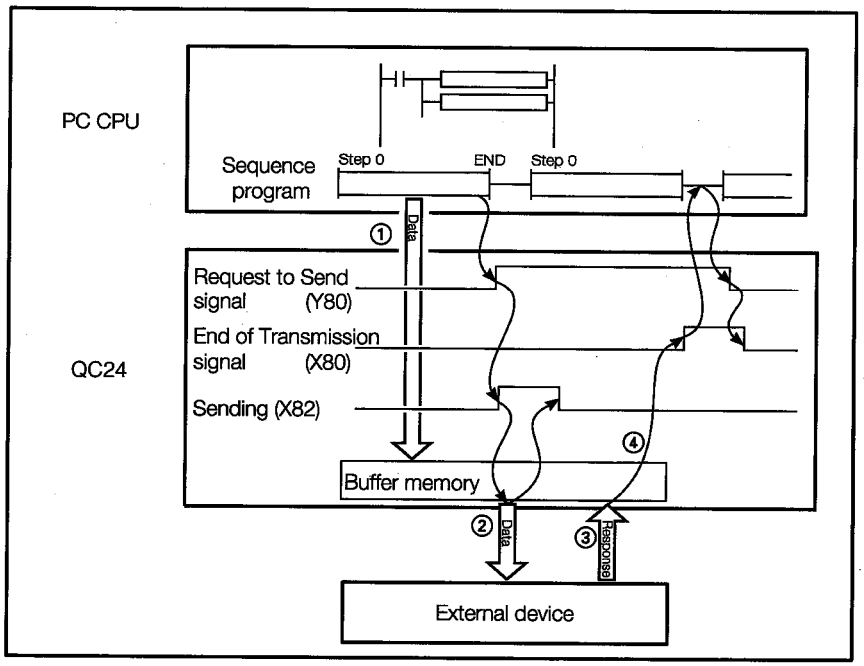
① The PC CPU uses the sequence program TO instruction (*BIDOUT instruction) to write the data to the QC24 buffer memory.

② When the Request to Send signal (Y80) is turned ON, the QC24 sends the data code to the external device as is.

③ The external device checks the receive data and sends a response (result of reception) to the QC24.

④ When the QC24 receives the response, it turns ON the End of Transmission signal (X80) and ends one data transmission.

* : The BIDOUT instruction automatically turns the QC24 Request to Send and End of Transmission signals ON and OFF by internal processing. The sequence program does not have to turn these signals ON and OFF.



6.6 Data Reception From External Device

In this section, the PC CPU receives the send data from the external device.

The receive data count units are word units.

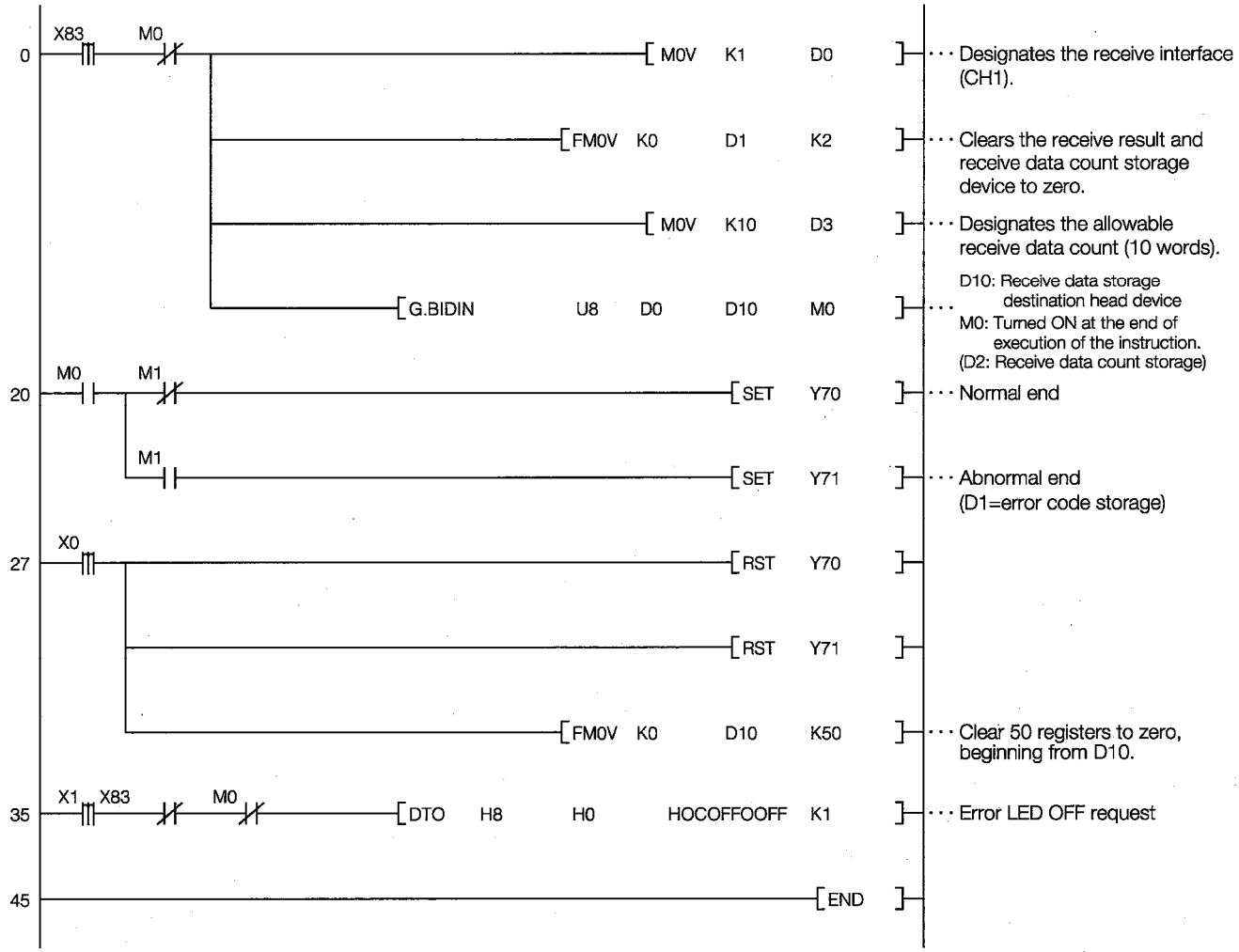
[Practice Contents]

The external device sends the data input from its keyboard to the PC CPU.

The PC CPU uses the sequence program FROM instruction to read the receive data from the QC24 buffer memory.

When the external device receives the transmitted result, it ends communications.

6.6.1 Sequence program

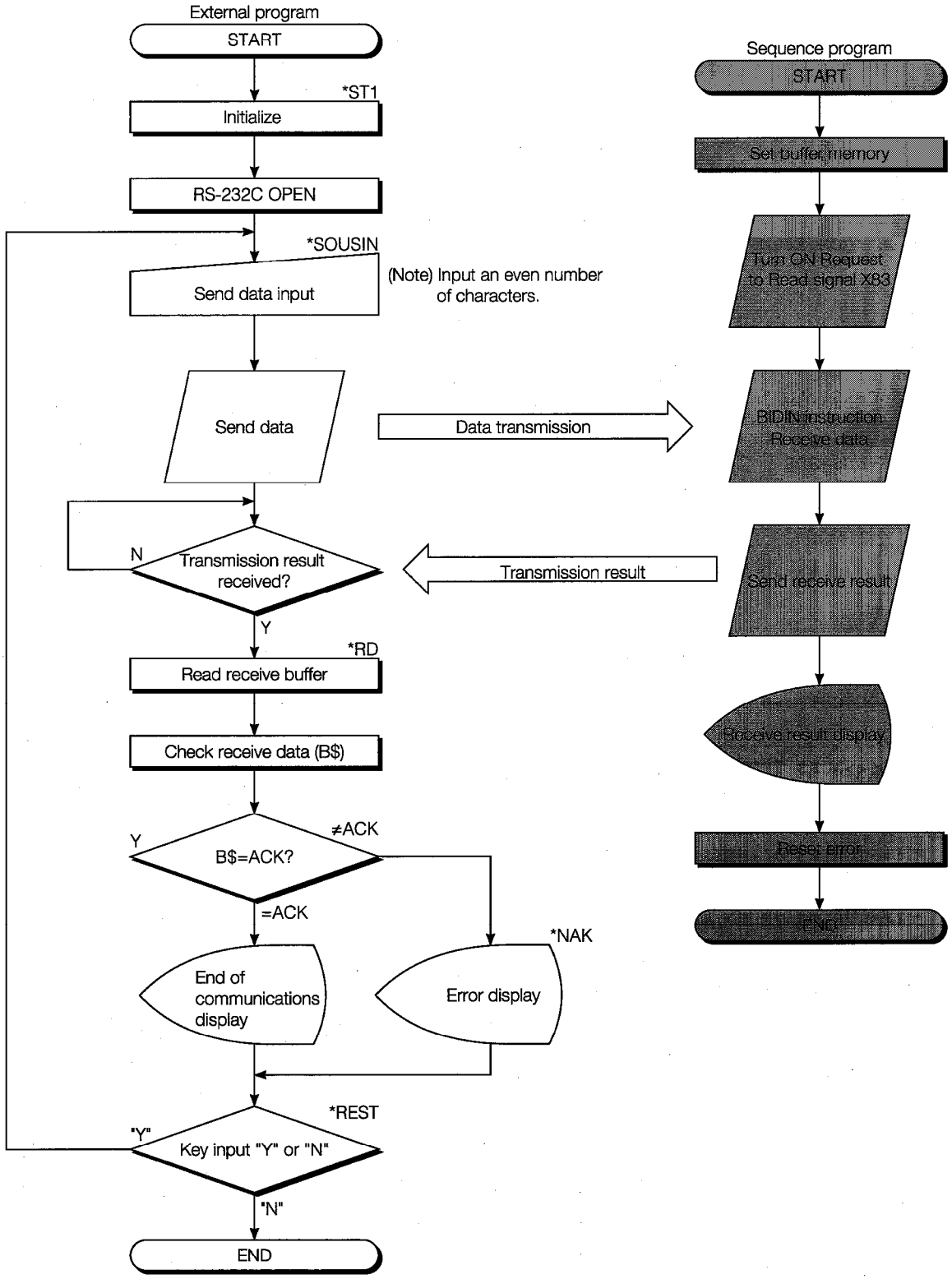


6.6.2 External device program

```
1000 ' ! ------------------------------------------------------------------------------!
1010 ' !                  AJ71QC24 Bidirectional Mode Sample Program                   !
1020 ' !                       (Data reception from computer)                          !
1030 ' ! ------------------------------------------------------------------------------!
1040 *ST1
1050     CLS                                :'Clear screen
1060     CH% =1                             :'Channel No.
1070     ENQ$=CHR$(&H5)                     :'ENQ code
1080     ACK$=CHR$(&H6)                     :'ACK code
1090     NAK$=CHA$(&H15)                    :'NAK code
1100     '
1110 ' ! ------------------------ RS-232C open & initialize---------------------------!
1120     OPEN "COM:E81NN" AS #CH%           :'Set communications mode, etc.
1130 ' ! ----------------------------- Data transmission-------------------------------!
1140 *SOUSIN
1150     CLS
1160     LOCATE  6,1:PRINT " **Transmission to PC in bidirectional mode**"
1170     LOCATE 10,5:INPUT "Send data" ;SD$
1180     SN$=RIGHT$("000+HEX$(LEN(SD$)/2),4):'Compute data count.
1190     N1$=CHR$(VAL("&H"+RIGHT$(SN$,2)))
1200     N2$=CHR$(VAL("&H"+LEFT$(SN$,2)))
1210     NS$=N1$+N2$
1220     DAT$=NS$+SD$
1230     PRINT #CH% ,ENQ$;DAT$;
1240     '
1250 ' ! --------------------------ACK/NAK data reception ------------------------------!
1260 *RECEIVE
1270     IF LOC(CH%)<>0 THEN *RD ELSE *RECEIVE
1280 ' ! ---------------------------------- Buffer read-------------------------------!
1290 *RD
1300     B$=""
1310 *RD1
1320     FOR I%= 0 TO 1000 :NEXT I%
1330     IF LOC(CH%)<>0 THEN B$=B$+INPUT$(LOC(CH%),#CH%) ELSE *RD1
1340     '
1350 ' ! ----------------------------Receive data check ------------------------------!
1360     IF B$<>ACK$ THEN *NAK
1370     LOCATE 10,8;PRINT "End of communications.": GOTO *REST
1380     '
1390 *NAK
1400     E1$=RIGHT$("0'+HEX$(ASC(RIGHT$(B$,1))),2)
1410     E2$=RIGHT$("0"+HEX$(ASC(MID$(B$,2,1))),2)
1420     IF LEFT$(B$,1)=NAK$ THEN E$=E1$+E2$ ELSE E$="####"
1430     COLOR 2:LOCATE 10,8
1440     PRINT "Communications error!! Error code=";E$:COLOR 0
1450 *REST
1460     LOCATE 10,11:INPUT "Retransmit (Y/N) ?";Y$
1470     IF Y$="Y" THEN *SOUSIN
1480     '
1490 CLOSE
1500 END
```

* BASIC commands must be changed according to BASIC software.

Data reception from external device



*[ ] : Label name used with the program

6.6.3 Trainer operation

Write the sequence program shown in Section 6.6.1 to the PC CPU and the BASIC program shown in Section 6.6.2 to the external device.

After writing, set the PC CPU and external device to "RUN".

If there are no errors, send and receive data using the following procedure.

① Input an even number of arbitrary send data from the external device keyboard and send the data to the PC CPU.

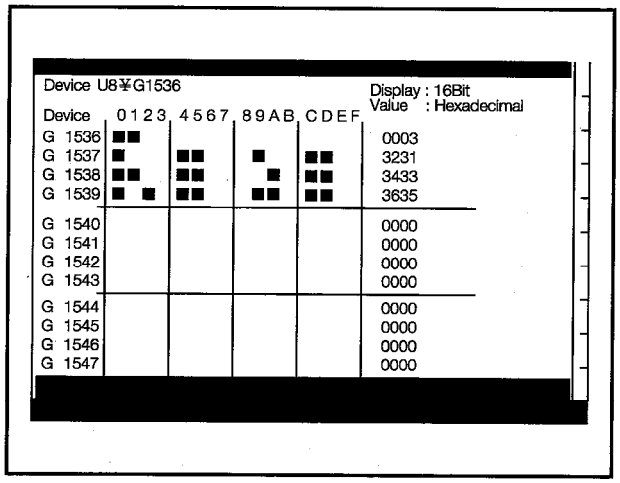
(For example, 1 2 3 4 5 6 ↵)

```
** Transmission to PC in bidirectional mode **

Send data?
```

(Note) Make the input data an even number.

② The data sent from the external device is written to the QC24 buffer memory (addresses 600H~).



GPPQ buffer batch monitor screen

```
** Transmission to PC in bidirectional mode **

Send data?123456

End of communications.

Reset transmission (Y/N) ?
```

External device screen

If you have any situations as follows, please read the explanations

- Data cannot be sent from external device

```
**Transmission to PC in bidirectional mode**

  Send data? □
```

- Was the sequence program (Section 6.6.1) written to the PC CPU ?
- Are the QC24 mode switch settings (Section 6.2), cable connections (Section 3.3), and other settings correct ?
- Check the QC24 LEDs (Section 6.3), then check the settings.

- Error code was received from the PC CPU ?

```
**Transmission to PC in bidirectional mode**

  Send data? □

  Communications error!! Error code= □

  Reset transmission (Y/N) ?
```

- NAK was sent from the QC24.

  Are the QC24 awitch settings (Section 6.2), external device settings (Section3.4), and external device program m (RS-232C) settings (Section 6.6.2) correct ?

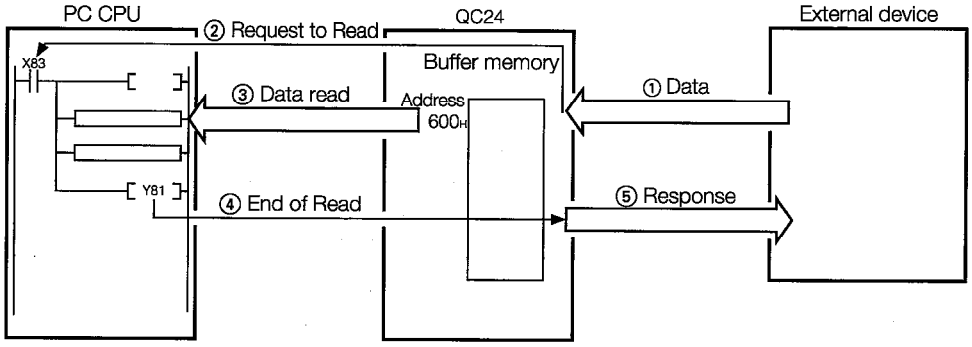
  Check the QC24 LEDs (Section 6.3), then check the settings.
- Is the number of characters input from the keyboard as send data an odd number ?

  (Sinve the send data count is in word units, key-in an even number of characters.)
- Refer to the Serial Communications Module Use's Manual for the contents of the displayed error code and take appropriate action.

After resetting the settings, reset the PC CPU and communicate again.

6.6.4 Summary of data reception from external device using bidirectional protocol

The following uses an image diagram to outline the data reception from the external device using the bidirectional protocol described in Section 6.6.3.



① When data is sent from the external device, the QC24 stores the receive data to its buffer memory.

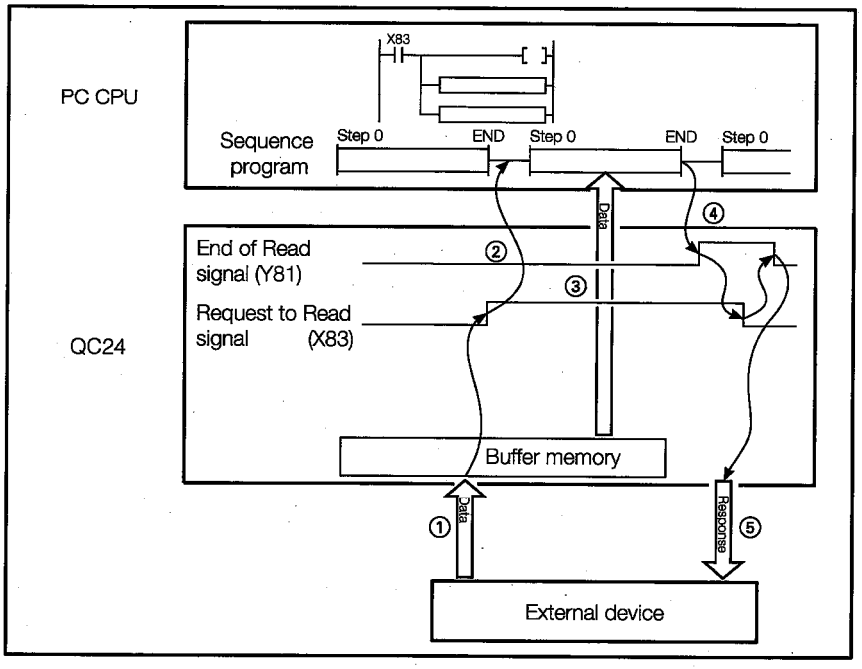
② The QC24 turns ON the Request to Read signal (X83) to the PC CPU.

③ The PC CPU uses the sequence program FROM instruction (*BIDIN instruction) to read the receive data from the QC24.

④ At the end of data read processing, the sequence program turns ON the End of Read signal (Y81).

⑤ When the End of Read signal (Y81) is turned ON, the QC24 sends a response to the external device and turns off the Request to Read signal (X83) and ends one data reception.

* : The BIDIN instructions automatically turns the QC24 Request to Read and End of Read signals ON and OFF by internal processing. The sequence program does not have to turn these signals ON and OFF.



MEMO

APPENDIXES

APPENDIX 1 BUFFER MEMORY ALLOCATION TABLE

Address and Objective I/F		Application	Name		QC24 Default Value	Objective Protocol		
CH1	CH2					Dedicated	Non Procedure	Bidirectional
0H (0)		System settings	LED OFF request (LED No.5~13)		0	○		
1H (1)			LED OFF request (LED No.16~29)					
2H (2)			EEPROM access	Write/read/delete directive		○		—
3H (3)				Frame No. directive				
4H (4)				Write/read/delete result storage				
5H (5)				Registered data byte count storage				
6H (6) to 2DH (45)				User frame *40 words				
2EH (46) to 8FH (143)			System area (unusable)		0	—		
90H (144)	130H (304)		Mode switching	Switching mode No. designation	0 (Switching not requested)	○		
91H (145)	131H (305)			Switching specifications designation	0			
92H (146)	132H (306)		System area (unusable)		0	—		
93H (147)	133H (307)		Transmission control	DTR/DSR, DC control designation	0 (DTR/DSR control)	○		
94H (148)	134H (308)			DC1/DC3 code designation	1311H (11H/13H)			
95H (149)	135H (309)			DC2/DC4 code designation	1412H (12H/14H)			
96H (150)	136H (310)		Word/byte designation (message data length units)		0 (Word units)	○ [On-demand]	○	
97H (151)	137H (311)		RS-232C CD pin check designation		1 (Do not check)	○		
98H (152)	138H (312)		RS-232C communications system designation		0 (full-duplex communications)	○		
99H (153)	139H (313)		Half-duplex communications	Simultaneously transmission priority/nonpriority designation	0 (Priority transmission)	○		
9AH (154)	13AH (314)			Retransmission method designation	0 (Do not retransmit)			
9BH (155)	13BH (315)		Simultaneous transmission data valid/invalid designation		0 (Send data, receive data valid)	—		○
9CH (156)	13CH (316)		No reception watchdog timer (timer 0) designation		FA0H (4000 bytes)	○		
9DH (157)	13DH (317)		Response watchdog timer (timer 1) designation		32H (5 seconds)	○	—	○ (Send only)
9EH (158)	13EH (318)		Send watchdog timer (timer 2) designation		708H (3 minutes)	○	—	○
9FH (159)	13FH (319)		System area (unusable)		0	—		

○: Read/write area, △: Read only area, —: System area/area unused with given protocol

Address and Objective I/F		Application	Name			QC24 Default Value	Objective Protocol		
							Dedicated	Non Procedure	Bidirectional
CH1	CH2								
A0H (160)	140H (320)	System settings	On-demand	Buffer memory head address designation		CH1: 400H CH2: 800H	○	—	
A1H (161)	141H (321)			Data length designation		0			
A2H (162)	142H (322)		Send/receive setting	Send buffer memory head address designation		CH1: 400H CH2: 800H	—	○	
A3H (163)	143H (323)			Send buffer memory length designation		200H			
A4H (164)	144H (324)			Receive end data count designation		1FFH	—	○	—
A5H (165)	145H (325)			Receive end code designation		0D0AH (CR, LF)			
A6H (166)	146H (326)			Receive buffer memory head address designation		CH1: 600H CH2: A00H	—	○	
A7H (167)	147H (327)			Receive buffer memory length designation		200H			
A8H (168)	148H (328)			Receive data clear request		0 (No clear request)	—	○	—
A9H (169)	149H (329)		On-demand user frame designation	Header frame No. designation	(1st)	0 (None)	○	—	
AAH (170)	14AH (330)				(2nd)				
ABH (171)	14BH (331)			Trailer frame No. designation	(1st)	0 (None)			
ACH (172)	14CH (332)				(2nd)				
ADH (173)	14DH (333)		Receive user frame designation	User frame use/do not use designation		0 (Do not use)	—	○	—
AEH (174)	14EH (334)			Header frame No. designation	(1st)	0 (None)			
AFH (175)	14FH (335)				(2nd)				
B0H (176)	150H (336)				(3st)				
B1H (177)	151H (337)				(4th)				
B2H (178)	152H (338)			Trailer frame No. designation	(1st)	0DH			
B3H (179)	153H (339)				(2nd)	0AH			
B4H (180)	154H (340)				(3rd)	0 (None)			
B5H (181)	155H (341)				(4th)				
B6H (182)	156H (342)		Send user frame designation	Sending user frame No.		0 (Do not send)	—	○	—
B7H (183)	157H (343)			Schedule designation	CR/LF output designation	0 (Do not output)			
B8H (184)	158H (344)				Output destination pointer designation	0 (Output not designated)			
B9H (185)	159H (345)				Number of outputs designation	0 (None)			
BAH (186)	15AH (346)			Output frame No. designation	(1st)	0 (None)			
BBH (187) to 11DH (285)	15BH (347) to 1BDH (445)				(2nd) to (100th)	0 (None)			
11EH (286)	1BEH (446)		Message wait time designation (Used in communications other than A compatible frame)			0 (No wait time)	○	—	
11FH (287)	1BFH (447)		Send transparent code designation			0 (None)	—	○	
120H (288)	1C0H (448)		Receive transparent code designation			0 (None)			
121H (289)	1C1H (449)		ASCII-BIN conversion designation			0 (Do not convert)			
122H (290) to 12FH (303)	1C2H (450) to 1FFH (511)		System area (unusable)			0	—		

○: Read/write area, △: Read only area, —: System area/area not used with given protocol

Address and Objective I/F CH1	Address and Objective I/F CH2	Application	Name		QC24 Default Value	Objective Protocol: Dedicated	Objective Protocol: Non Procedure	Objective Protocol: Bidirectional
200H (512)		System information	Station No. switch setting status storage		(Switch set No.)	△		
201H (513)			LED ON status storage (LED Nos. 5~13)		[Depends on the status of the module]			
202H (514)			LED ON status storage (LED Nos. 16~29)					
203H (515)			Switch setting error, mode switching error storage		0 (No errors)			
204H (516)			EEPROM access	Number of registered user frame storage	0 [Not registered. However, depending on the registering status after registering.]	△		—
205H (517) to 21DH (541)				User frame registered status storage (for registered No. check)				
21EH (542)			Number of registered default frames storage (OS ROM)					
21FH (543)			System area (unusable)		0	—		
220H (544)			EEPROM system settings write result storage		0	△		
221H (545) to 24FH (591)			System area (unusable)		0	—		
250H (592)	260H (608)		Mode switch setting status storage		[Depends on the switch setting]	△		
251H (593)	261H (609)		Transmission specifications switches setting status storage					
252H (594)	262H (610)		Mode setting status storage	Current set contents				
253H (595)	263H (611)		Transmission specifications setting status storage					
254H (596)	264H (612)		RS-232C control signal status storage		(Depends on the signal status)			
255H (597)	265H (613)		Transmission sequence status storage		0 (Wait to send and receive)			
256H (598)	266H (614)		On-demand executed result status		0 (Normal end)	○	—	
257H (599)	267H (615)		Data transmission result storage		0 (No errors)	○		
258H (600)	268H (616)		Data receive result storage					
259H (601)	269H (617)		System area (unusable)		0	—		
25AH (602)	26AH (618)		Dedicated protocol transmission error code storage		0 (No errors)	○	—	
25BH (603)	26BH (619)		Receive user frame storage (□th)		0 (Not received)	—	△	—
25CH (604) to 25FH (607)	26CH (620) to 3FFH (1023)		System area (unusable)		0	—		

○: Read/write area, △: Read only area, —: System area/area not used with given protocol

Address and Objective I/F CH1	Address and Objective I/F CH2	Application	Name		QC24 Default Value	Objective Protocol: Dedicated	Objective Protocol: Non Procedure	Objective Protocol: Bidirectional
400H (1024)		Default CH1 send/receive buffer memory	Send area	Send data count storage	0	○ [On-demand]	○	
401H (1025) to 5FFH (1535)				Send data storage * 511 words	0			
600H (1536)			Receive area	Receive data storage (number of data read)	0			
601H (1537) to 7FFH (2047)				Receive data storage * 511 words	0			
800H (2048)		Default CH2 send/receive buffer memory	Send area	Send data count storage	0	○ [On-demand]	○	
801H (2049) to 9FFH (2559)				Send data storage * 511 words	0			
A00H (2560)			Receive area	Receive data count storage	0			
A01H (2561) to BFFH (3071)				Receive data storage * 511 words	0			
C00H (3072) to 1AFFH (6911)		User	User area	* 3840 words	0	○		
1B00H (6912)		User frame registering [number registered to buffer memory]	Registered No. 8001H	Registered data byte count storage	0	○		—
1B01H (6913) to 1B28H (6952)				User frame storage * 40 words	0			
1B29H (6953)			Registered No. 8002H	Registered data byte count storage	0			
1B2AH (6954) to 1B51H (6933)				User frame storage * 40 words	0			
1B52H (6934) to 1FCDH (8141)			⋮		0			
1FCEH (8142)			Registered No. 801FH	Registered data byte count storage	0			
1FCFH (8143) to 1FF6H (8182)				User frame storage * 40 words	0			
1FF7H (8183) to 1FFFH (8191)		—	System area (unusable)		0	—		

○: Read/write area, △: Read only area, —: System area/area not used with given protocol

APPENDIX 2 PC CPU INPUT/OUTPUT SIGNALS TABLE

The following shows the QC24 input/output signals for the PC CPU.

The No. of the input/output signal of the slot into which the QC24 is inserted determines (n) added to the X and Y numbers.

(Example: When the QC24 was inserted into slot 0 of the basic base unit Xn0→X0)

(1) Input/output signals table

Signal Direction: QC24→PC CPU			Signal Direction: PC CPU→QC24		
Device No.	Signal Name		Device No.	Signal Name	
Xn0	CH1	Send normal end	Yn0	CH1	Request to send
Xn1		Send abnormal end	Yn1		End of receive data read
Xn2		Sending	Yn2		Mode switching request (processing stop request)
Xn3		Receive data read request	Yn3 to Yn6	(Unusable)	
Xn4		Receive error detected			
Xn5	———				
Xn6	CH1	Mode switching (initializing) *1			
Xn7	CH2	Send normal end	Yn7	CH2	Request to send
Xn8		Send abnormal end	Yn8		End of receive data read
Xn9		Sending	Yn9		Mode switching request (processing stop request)
XnA		Request to read receive data	YnA to YnD	(Unusable)	
XnB		Receive error detected			
XnC	———				
XnD	CH2	Mode switching (initializing) *1			
XnE	CH1.ERR LED ON		YnE	CH1.ERR LED OFF request	
XnF	CH2.ERR LED ON		YnF	CH2.ERR LED OFF request	
X (n+1) 0 to X (n+1) 6	———		Y (n+1) 0 to Y (n+1) 6	(Unusable)	
X (n+1) 7	End of EEPROM read		X (n+1) 7	Request to read EEPROM	
X (n+1) 8	End of EEPROM write		Y (n+1) 8	Request to write EEPROM	
X (n+1) 9	End of EEPROM system settings write		Y (n+1) 9	Request to write system setting to EEPROM	
X (n+1) A	CH1	Global signal	Y (n+1) A	(Unusable)	
X (n+1) B	CH2	Global signal	Y (n+1) B		
X (n+1) C	End of system setting default		Y (n+1) C	System setting default request	
X (n+1) D	———		Y (n+1) D to Y (n+1) F	(Unusable)	
X (n+1) E	QC24 Ready (accessible) *2				
X (n+1) F	Watchdog timer error *3				

*1 During mode switching (Xn6, XnD: ON), do not send a Request to Send signal to the objective interface. (The QC24 will stop send/receive processing.)

*2 The QC24 Ready signal (X(n+1)E) is turned ON when the PC CPU can access the QC24. (The QC24 Ready signal is turned ON about 1 second after the power is turned ON or the PC CPU is reset.)
Use this signal as a FROM/TO instruction or other interlock signal.

*3 The watchdog timer error signal (X(n+1)F) is turned ON when the QC24 cannot operate normally. When it is turned ON, the PC CPU must be reset.

IMPORTANT

The output signals designatged "(unusable)" in the table above are used by the system and cannot be used by the user.
If the user should use these signals, the functions of the QC24 cannot be guaranteed.

IMPORTANT

(1) System settings should be set up so that protective devices for the sequencer and safety circuits are installed externally.

(2) Printed circuit boards contain components that are susceptible to static electricity. If handling printed circuit boards directly, the following precautions should be taken:

① Make sure people and work benches or tables are grounded.

② Never directly touch conductive components or electrical parts of the product.

AJ71QC24 Serial Communications Module GUIDEBOOK

MODEL	A-QC24(R2/R4)-G-E
MODEL CODE	13JF11
IB(NA)66622-A(9603)MEE	

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-8310 TELEX : J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN