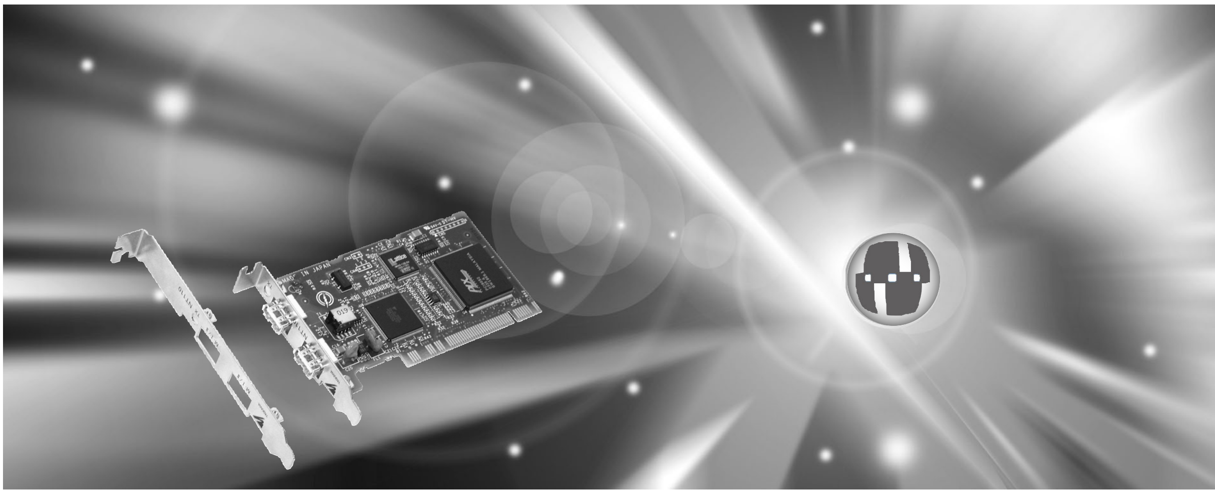
PCI-compliant MECHATROLINK-Ⅱ Interface Card
USER'S MANUAL

MODEL: JAPMC-NT110, NT111



YASKAWA

MANUAL NO. SIEP C880700 17B

Using this Manual

Please read this manual to ensure correct usage of the MECHATROLINK-II Interface Cards JAPMC-NT110 and JAPMC-NT111. Keep this manual in a safe place for future reference.

Note: In this manual, JAPMC-NT110 represents both the JAPMC-NT110 and the JAPMC-NT111 cards. For more information about the differences between the cards, refer to *1.1.2 External Appearance and Model Number Location*.

■ Basic Terms

Unless otherwise specified, the following definitions are used:

• NT110, NT111	: Generic terms for the PCI-compliant MECHATROLINK-II Interface Cards, the JAPMC-NT110 and the JAMPC-NT111
• MECHATROLINK	: Generic term for Motion Network MECHATROLINK-I and MECHATROLINK-II
• PC	: Generic term for personal computer or other general-purpose computers
• Master	: Primary station composed of MECHATROLINK-II Interface Card
• Slave	: Secondary station composed of MECHATROLINK-II compatible devices

■ Manual Configuration

Read the chapters of this manual as required by the purpose.

Chapter	Selecting Models and Peripheral Devices	Studying Specifications and Ratings	Designing the System	Installation and Wiring	Trial Operation	Maintenance and Inspection
Chapter 1 NT110 Overview	–	–	Applicable	–	–	–
Chapter 2 MECHATROLINK-II System	–	–	Applicable	–	Applicable	–
Chapter 3 NT110 Details	Applicable	Applicable	Applicable	Applicable	Applicable	Applicable
Chapter 4 Mounting and Connecting the NT110	Applicable	–	Applicable	Applicable	Applicable	Applicable
Chapter 5 Communication and Control Function	Applicable	–	Applicable	Applicable	Applicable	Applicable
Chapter 6 Communication Software	Applicable	–	Applicable	–	Applicable	Applicable

■ Visual Aids

The following aids are used to indicate certain types of information for easier reference.

IMPORTANT Indicates important information that should be memorized, including precautions such as alarm displays to avoid damaging the devices.

INFO Indicates supplemental information.

◀ EXAMPLE ▶ Indicates application examples.

TERMS Describes technical terms that are difficult to understand, or appear in the text without an explanation being given.

■ Indication of Reverse Signals

In this manual, the names of reverse signals (ones that are valid when low) are written with a forward slash (/) before the signal name, as shown in the following example:

- $\overline{\text{DATA}}$ = /DATA

■ Copyrights

- Windows and Windows2000/XP are registered trademarks of the Microsoft Corporation.
- Other product names and company names are the trademarks or registered trademarks of the respective company. "TM" and the ® mark do not appear with product or company names in this manual.

■ Related Manuals

Refer to the following related manuals as required.

Manual Name	Manual Number	Contents
Σ-III Series SGM□S/SGDS User's Manual For MECHATROLINK-II communications	SIEPS80000011	Describes the models, capacities, selection methods, ratings, characteristics, diagrams, cables, peripheral devices, wiring, panel installation, trial operation, adjustment, function application methods, maintenance, inspection, and MECHATROLINK communication of the Σ-III Series SERVOPACKs and servomotors.
Σ-II Series SGDH MECHATROLINK-II Application Module User's Manual	SIEPC71080001	Describes the MECHATROLINK communication of the JUSP-NS115 Application Module mounted in the Σ-II Series SERVOPACK.
Machine Controller MP900/MP2000 Series MECHATORLINK System User's Manual	SIE-C887-5.1	Describes how to use the MECHATROLINK-I and MECHATORLINK-II communications and MECHATROLINK-compatible devices.

Safety Information

The following conventions are used to indicate precautions in this manual. Failure to heed precautions provided in this manual can result in serious or possibly even fatal injury or damage to the products or to related equipment and systems.

⚠ WARNING

Indicates precautions that, if not heeded, could possibly result in loss of life or serious injury.

⚠ CAUTION

Indicates precautions that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

In some situations, the precautions indicated could have serious consequences if not heeded.

🛇 PROHIBITED

Indicates prohibited actions that must not be performed. For example, this symbol would be used as follows to indicate that fire is prohibited: 🛇 .

❗ MANDATORY

Indicates compulsory actions that must be performed. For example, this symbol would be used as follows to indicate that grounding is compulsory: ⏚ .

Safety Precautions

The following precautions are for checking products on delivery, storage, transportation, installation, wiring, operation, maintenance, inspection, and disposal. These precautions are important and must be observed.

⚠ WARNING

- Before starting operation in combination with the machine, ensure that an emergency stop procedure has been provided and is working correctly.

  There is a risk of injury.
- Do not touch anything inside the MECHATROLINK devices.

  There is a risk of electrical shock.
- Always keep the front cover attached when power is being supplied.

  There is a risk of electrical shock.
- Observe all procedures and precautions given in this manual for trial operation.

  Operating mistakes while the servomotor and machine are connected can cause damage to the machine or even accidents resulting in injury or death.
- Do not remove the front cover, cables, connector, or options while power is being supplied.

  There is a risk of electrical shock.
- Do not allow installation, disassembly, or repairs to be performed by anyone other than specified personnel.

  There is a risk of electrical shock or injury.
- Do not damage, pull on, apply excessive force to, place heavy objects on, or pinch cables.

  There is a risk of electrical shock, operational failure or burning of the Interface Card.
- Do not attempt to modify the Interface Card in any way.

  There is a risk of injury or device damage.
- Do not approach the machine when there is a momentary interruption to the power supply. When power is restored, the machine may start operation suddenly. Provide suitable safety measures to protect people when operation restarts.

  There is a risk of injury.

■ Storage and Transportation

⚠ CAUTION

- Do not store or install the Interface Card in the following locations.

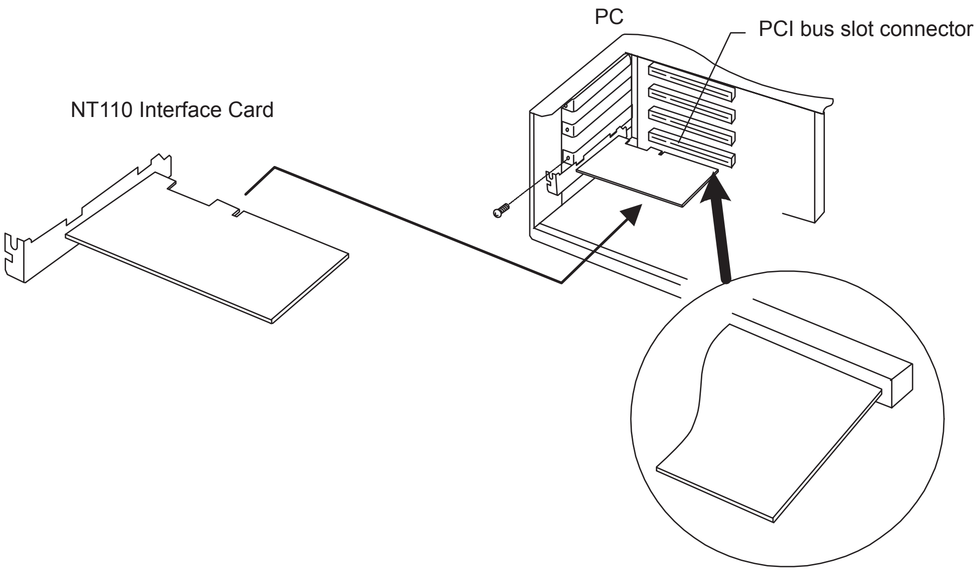
  There is a risk of fire, electrical shock, or device damage.
  - Direct sunlight
  - Ambient temperature exceeds the storage or operating conditions
  - Ambient humidity exceeds the storage or operating conditions
  - Rapid changes in temperature or locations subject to condensation
  - Corrosive or flammable gas
  - Excessive dust, dirt, salt, or metallic powder
  - Water, oil, or chemicals
  - Vibration or shock
- Do not overload the Interface Card during transportation.

  There is a risk of injury or an accident.

■ Installation

⚠ CAUTION

- Mount the NT110 Interface Card in a PCI bus slot in the PC.



- Before touching the NT110 Interface Card, discharge static electricity from your body to prevent the Card from being damaged by static electricity.
- If mounting or removing the Card, turn OFF the host PC and unplug the power-supply cable to ensure safety.
- Fully insert the card so that it touches the back of the PCI slot. The card and the host PC may malfunction or fail to operate if the card is not fully inserted. If the card does not slide smoothly into the slot, do not force it. Instead, remove the card from the slot and reinsert it.
- When handling the card, hold it by its edges. Do not touch the components or solder. If the card is handled incorrectly, injury may occur and the card or the host PC may fail to operate.
- Depending on the model of the host PC, the Card may need to be fixed with screws after it is mounted in the PCI slot. Refer to the operation manual of the host PC for the mounting method. If the Card is not mounted securely, the Card and the host PC may malfunction or fail to operate.
- Insert and fix the cable connectors for the Card, servodrive, and I/O Modules. The Card, servodrive, and I/O Modules may malfunction or fail to operate if the cables are not correctly connected.
- Never use the Interface Card in locations subject to water, corrosive atmospheres, or flammable gas, or near burnable objects.

  Failure to observe this caution may result in electric shock or fire.
- Do not subject the Interface Card to strong shock.

  Failure to observe this caution may result in malfunction of the Interface Card.

■ Wiring

⚠ CAUTION

- Check the wiring to be sure it has been performed correctly.

  There is a risk of motor run-away, injury, or an accident.
- Always use a power supply of the specified voltage.

  There is a risk of burning.
- In places with poor power supply conditions, take all steps necessary to ensure that the input power supply is within the specified voltage range.

  There is a risk of device damage.
- Install breakers and other safety measures to provide protection against shorts in external wiring.

  There is a risk of fire.
- Provide sufficient shielding when using the Interface Card in the following locations.

  There is a risk of device damage.
  - Noise, such as from static electricity
  - Strong electromagnetic or magnetic fields
  - Radiation
  - Near power lines

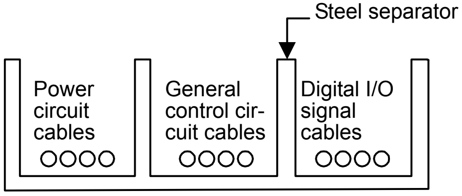
■ Selecting, Separating, and Laying External Cables

⚠ CAUTION

- Consider the following items when selecting the I/O signal lines (external cables) to connect the MECHATROLINK device to external devices.
  - Mechanical strength
  - Noise interference
  - Wiring distance
  - Signal voltage, etc.
- Separate the I/O signal lines from the power lines both inside and outside the control box to reduce the influence of noise from the power lines.

  If the I/O signal lines and power lines are not separated properly, malfunctioning may result.

Example of Separated External Cables



■ Maintenance and Inspection

⚠ CAUTION

- Do not attempt to disassemble the MECHATROLINK device.

  There is a risk of electrical shock or injury.
- Do not change wiring while power is being supplied.

  There is a risk of electrical shock or injury.

■ Disposal

⚠ CAUTION

- Dispose of the Interface Card as general industrial waste.

CONTENTS

5 Communication and Control Functions

6 Optional Communication Software

1

NT110 Overview

1.1 Checking the NT110

1.1.1 Upon Delivery

Check the following items as soon as the product is delivered.

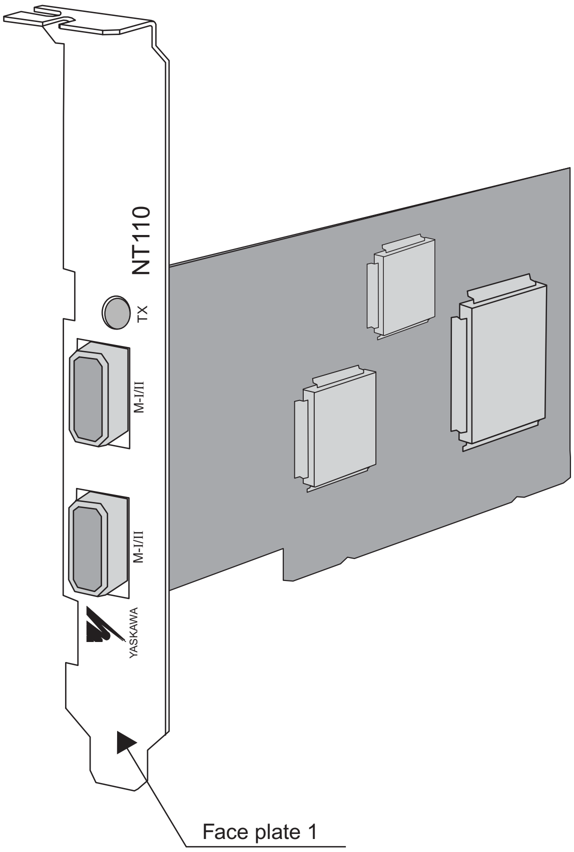
Check Item	Remarks
Is the delivered product the one that was ordered?	Check the model number given on the Interface Card to be sure the correct product was delivered. (Refer to the following pages.)
Is there any damage?	Check the overall appearance, and check for damage or scratches that may have occurred during shipping.

If any of the above items are faulty or incorrect, contact your Yaskawa representative or the dealer from whom you purchased the products.

1.1.2 External Appearance and Model Number Location

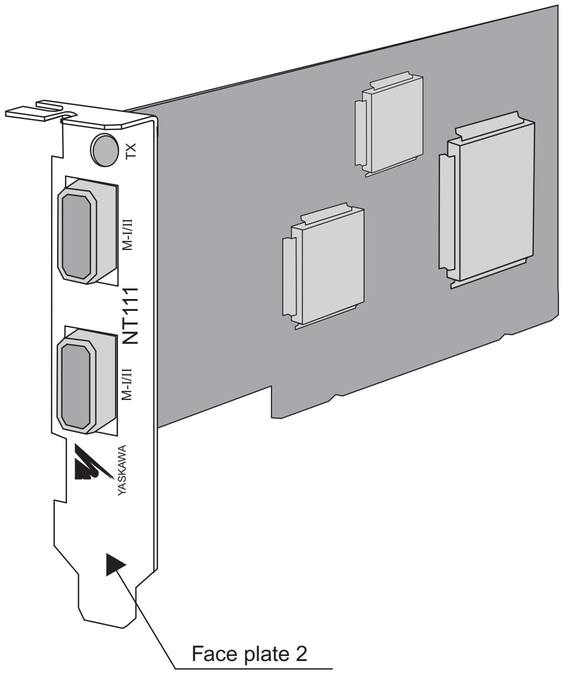
(1) JAPMC-NT110

The NT110 has installation face plate 1 mounted to it.



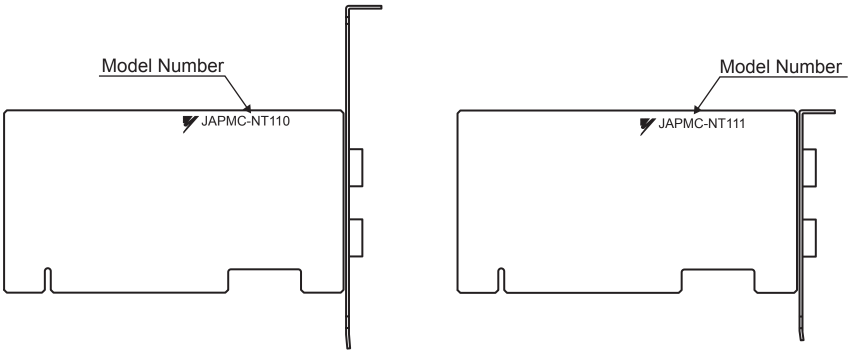
(2) JAPMC-NT111

The NT111 has Face plate 2 attached to it.



(3) Location of Model Number

The following illustrations show the location of the model number.



JAPMC-NT110　　　JAPMC-NT111

1.2 Model Numbers and Component Names

1.2.1 Reading Model Numbers

The model numbers used to select and confirm the NT110s are interpreted as shown here.

JAPMC - NT11 □

Face plate
0: Face plate 1 (with standard PCI)
1: Face plate 2 (with low-profile PCI)

1.2.2 Component Names

The names of the NT110 components are given in the following figure using the NT110 with a standard PCI as an example.

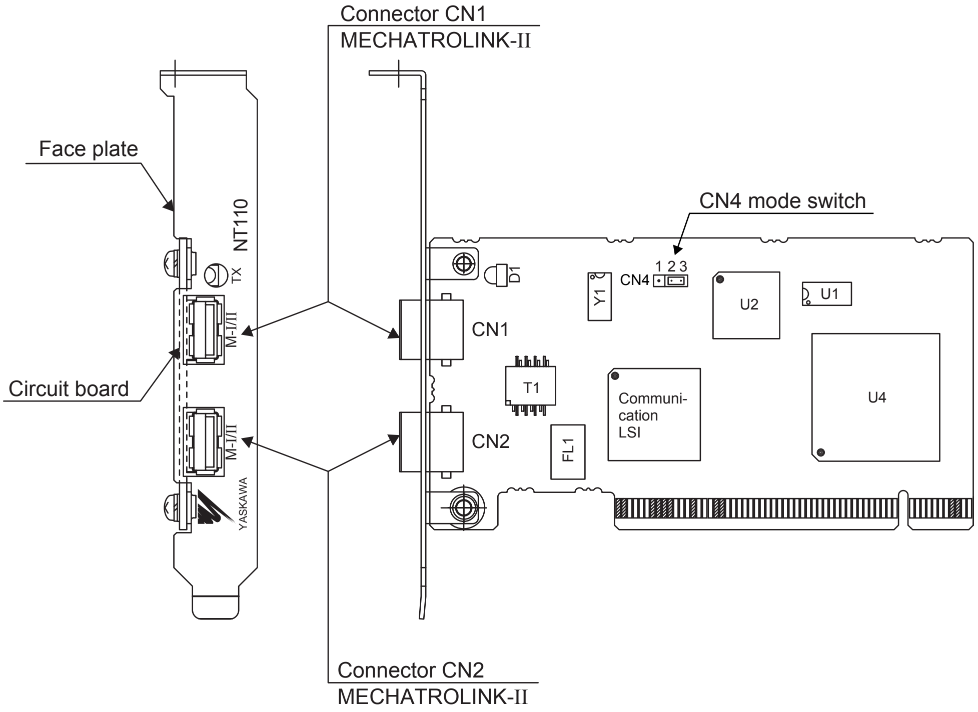


Fig. 1.1 JAPMC-NT110 Component Names

2

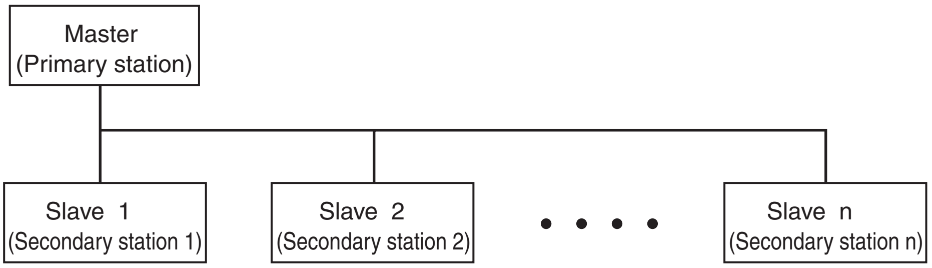
MECHATROLINK-II System

2.1 Overview of MECHATROLINK-II System

This section provides an overview of the MECHATROLINK-II System and describes its features.

2.1.1 MECHATROLINK-II System with the NT110

A MECHATROLINK-II System with the NT110 uses a master-slave system.



Note: The maximum number of stations, n, is 30. A Repeater is required for systems with 15 stations or more or 16 stations or more. (Refer to *2.2.3 Maximum Number of Slaves*.)

The MECHATROLINK-II System performs cyclic bidirectional communication for 17-byte/32-byte data between one master (the primary station) and up to 30 slaves (secondary stations) at a communication cycle of between 250 µs to 8 ms.

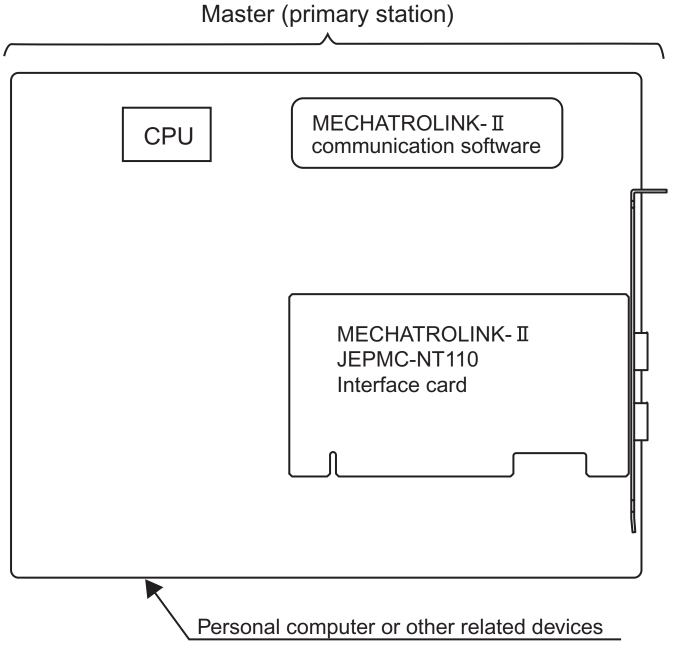
IMPORTANT

The above communication cycle may be restricted by the slaves that are connected.

MECHATROLINK-II communication is effective for locus control between multiple axes with complete synchronization for all stations. The MECHATROLINK-II System consists of the master and slaves described next.

(a) Master

The master is a PC equipped with software for executing MECHATROLINK-II communication in which a JAPMC-NT110 Interface Card is mounted to PC via a PCI-compliant bus.



The communication software must be created by the user.

However, Yaskawa can provide some communication software at an extra charge.

For details, refer to *Chapter 6 Optional Communication Software*.

(b) Slaves

Refer to *2.2.2 MECHATROLINK-II compatible devices* for details on applicable slaves.

2.1.2 Features of a MECHATROLINK-II System with the NT110

The NT110 has the following features as a MECHATROLINK-II Interface Card connected to a PCI-compliant bus.

- The NT110 contains a MECHATROLINK-II communication LSI, facilitating control of communication.
- The NT110 can be directly accessed to the communication LSI (JL-080) by using a PCI-compliant bus.
- The MECHATROLINK-II System performs bidirectional communication between one primary station (the master) and up to 30 secondary stations (slaves).
- If the NT110 is used, data can be exchanged between the primary station and secondary stations through the built-in 2-port RAM so that data communication are automatically checked for errors, reducing the load on the PC software.
- The communication line is connected using only one, 2-conductor twisted-pair cable. A pulse transformer is used for isolation. Therefore, high-speed, highly reliable communication can be achieved with simple connections.

2.2 MECHATROLINK-II System Configuration

2.2.1 MECHATROLINK-II System Configuration Example

A MECHATROLINK-II System configuration example using the NT110 is shown in *Fig. 2.1 MECHATROLINK-II System Configuration Example*. Up to 30 slaves can be connected to one master. If there are 15 stations or more, or 16 stations or more, a Repeater must be used.

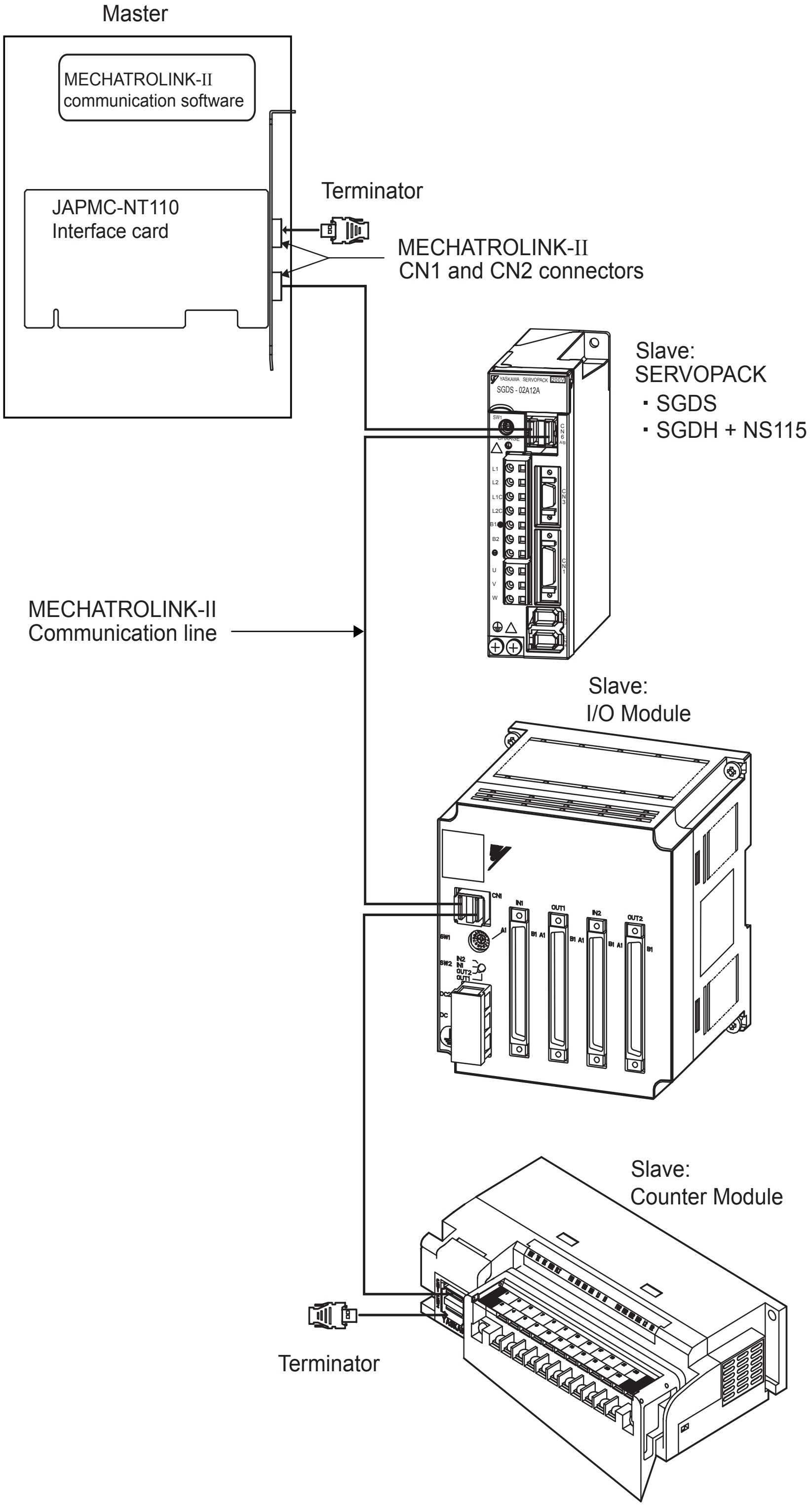

Master
MECHATROLINK-II communication software
JAPMC-NT110 Interface card
Terminator
MECHATROLINK-II CN1 and CN2 connectors
Slave: SERVOPACK
・SGDS
・SGDH + NS115
MECHATROLINK-II Communication line
Slave: I/O Module
Slave: Counter Module
Terminator

Fig. 2.1 MECHATROLINK-II System Configuration Example

2.2.2 MECHATROLINK-II compatible devices

Devices compatible with MECHATROLINK-II are listed in the following table.

Classification	Model Number	Contents
SERVOPACK	SGDH-□□□E + JUSP-NS115	SGDH SERVOPACK + NS115 MECHATROLINK-II Interface Module
	SGDS-□□□12□□ SGDS-□□□15□□	SGDS SERVOPACK
I/O Module	JEPMC-IO2310	64-point I/O Module 24 VDC, 64 inputs, 64 outputs
Counter Module	JEPMC-PL2900	Counter Module Reversible counters, 2 channels
Pulse Output Module	JEPMC-PL2910	Pulse Output Module Pulse outputs, 2 channels

2.2.3 Maximum Number of Slaves

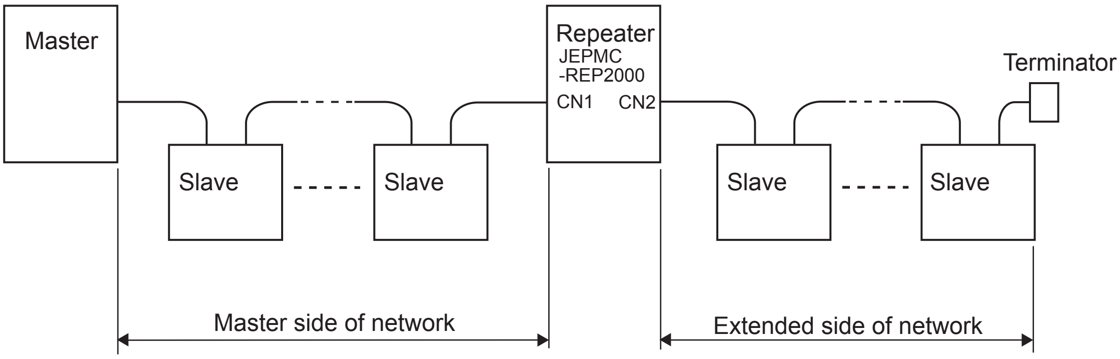
A maximum of 30 slaves can be connected to one master using the NT110.

If a certain number of slaves is exceeded, however, a Repeater is required.

The restrictions in the number of stations that can be connected on the master side of the network and the extended side of the network are listed in *Table 2.1 Restrictions in the Number of Slave Stations* based on the length of MECHATROLINK-II cable.

Table 2.1 Restrictions in the Number of Slave Stations

Type	Cable Length	Number of Slave Stations
Master side of network	30 m max.	15 stations max.
	50 m max.	14 stations max.
Extended side of network	30 m max.	16 stations max.
	50 m max.	15 stations max.



Note: 1. Regardless of the above conditions, no network can contain more than 30 slave stations. This restriction is based on the specifications of the master.

2. The Repeater is not counted as one of the stations.

3. The minimum distance between stations if 0.5 m regardless of whether a Repeater is used.

Use a Repeater with the following model number.

- JEPMC-REP2000

3

NT110 Details

3.1 Specifications and Dimension Diagrams

3.1.1 NT110 Specifications

The specifications of the NT110 are listed in the following table.

Item		Specifications
Board Size		PCI-compliant size (approx. 65 × 120 mm)
PCI Bus Connector		PCI-compliant bus
Bus Data Width		16 bits
Operation Timing		Conforms to PCI bus.
Input Power Supply		5 V (±5 %), 400 mA maximum
MECHATROLINK Port		1 port
MECHATROLINK Connector		USB type A
Communication Clock Frequency		Selected from 4 or 10 MHz.
Communication Cycle		Selected from 250 µs to 8 ms. (The connected devices, however, may limit the communication cycle.)
Number of Transmission Bytes		Selected from 17 or 32 bytes.
Environmental Conditions	Operating Temperature	0 to 45°C
	Storage Temperature	–25 to 85°C
	Humidity	30% to 95% (with no condensation)
	Vibration Resistance	15.7 m/s^2
	Shock Resistance	98 m/s^2

3.1.2 Mode Switch Specifications

The following table shows the specifications of the CN4 mode switch.

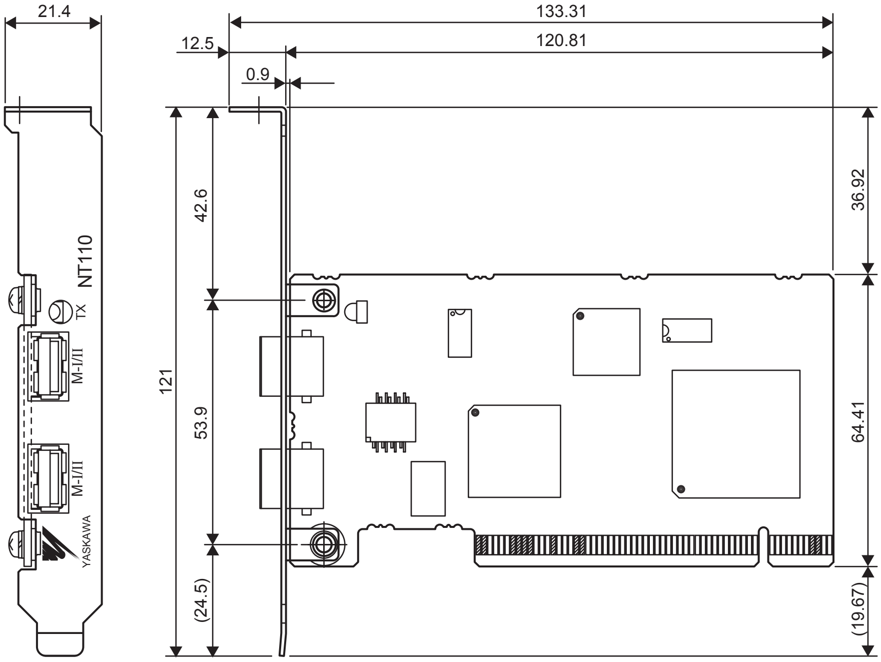
For using the mode switch, refer to *6.2.5 Using API*.

Short Pin	Name	Status	Mode
1 and 2	Slot	Open	Slot 1
		Short	Slot 2
2 and 3	Reserved	OFF	–
		ON	–

3.1.3 NT110 Dimension Diagrams

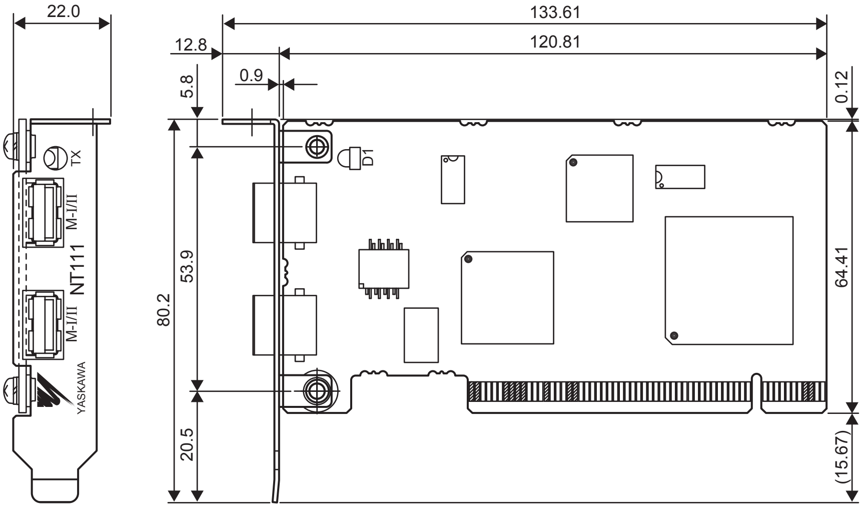
Dimension diagrams of the two models of NT110 are provided below.

(a) JAPMC-NT110 Dimension Diagram



Unit: mm

(b) JAPMC-NT111 Dimension Diagram

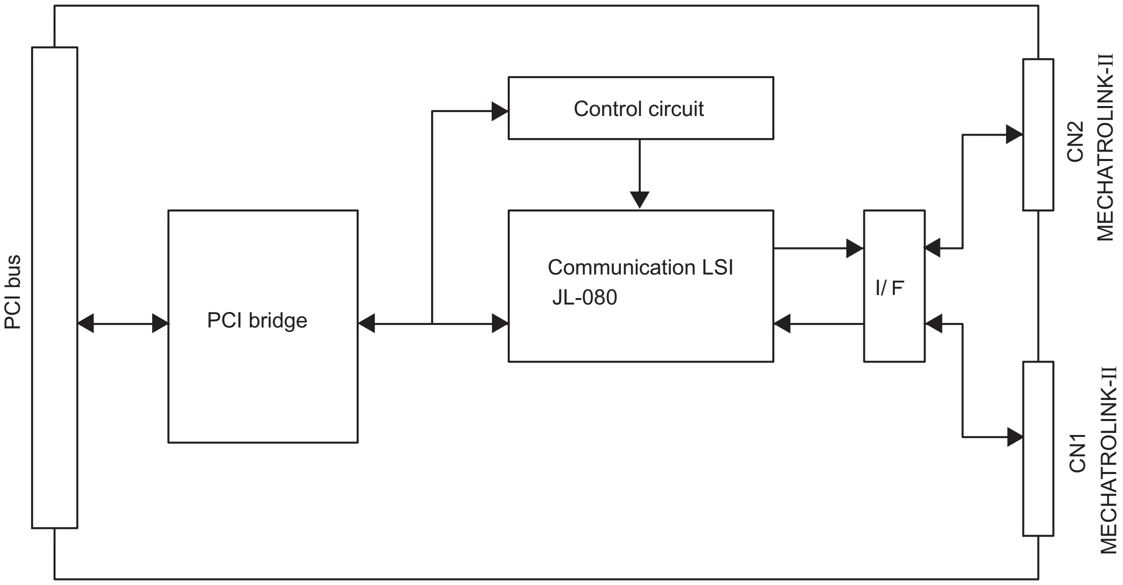


Unit: mm

3.2 Internal Block Diagrams and Connectors

3.2.1 Internal Block Diagrams

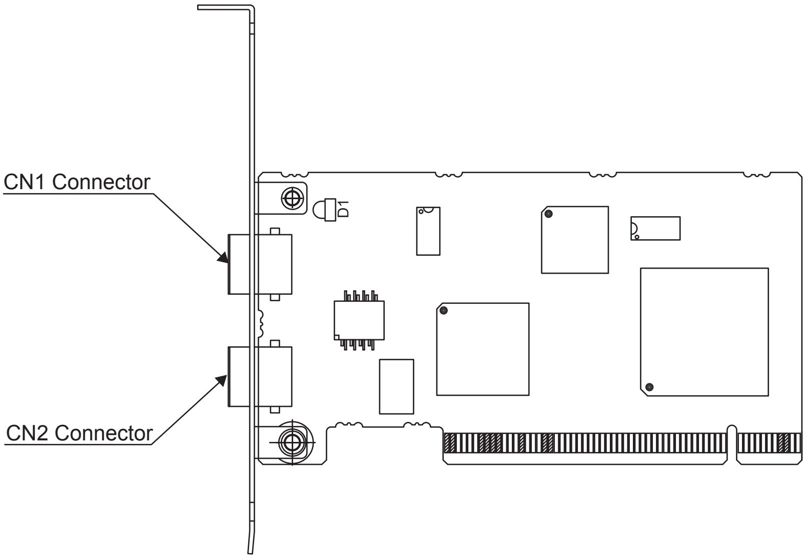
The following figure shows an internal block diagram of the NT110.



3.2.2 Applicable Connector

The following tables shows the connectors used with the NT110.

Connector Name	Applicable Connector	Application
CN1, CN2	USB type A	MECHATROLINK



- CN1 and CN2: MECHATROLINK

Pin No.	Signal Name	Remarks
1	NC	Not used.
2	/DATA	Signal, negative side.
3	DATA	Signal, positive side.
4	Shield	Connect the shield wire.

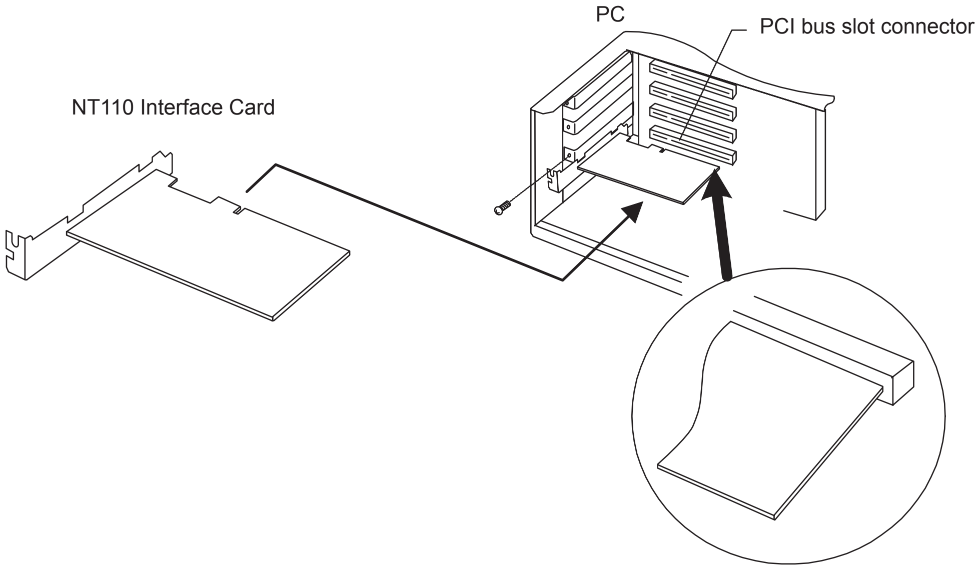
4

Mounting and Connecting the NT110

4.1 Mounting the NT110

4.1.1 Installing the NT110 Board in the PC

- Mount the NT110 Interface Card in a PCI bus slot in the PC.



- Before touching the NT110 Interface Card, discharge static electricity from your body to prevent the Card from being damaged by static electricity.
- If mounting or removing the Card, turn OFF the host PC and unplug the power-supply cable to ensure safety.
- Fully insert the card so that it touches the back of the PCI slot. The card and the host PC may malfunction or fail to operate if the card is not fully inserted. If the card does not slide smoothly into the slot, do not force it. Instead, remove the card from the slot and reinsert it.
- When handling the card, hold it by its edges. Do not touch the components or solder. If the card is handled incorrectly, injury may occur and the card or the host PC may fail to operate.
- Depending on the model of the host PC, the Card may need to be fixed with screws after it is mounted in the PCI slot. Refer to the operation manual of the host PC for the mounting method. If the Card is not mounted securely, the Card and the host PC may malfunction or fail to operate.
- Insert and fix the cable connectors for the Card, servodrive, and I/O Modules. The Card, servodrive, and I/O Modules may malfunction or fail to operate if the cables are not correctly connected.

4.2 Connecting MECHATROLINK-II Devices

4.2.1 MECHATROLINK-II Connectors

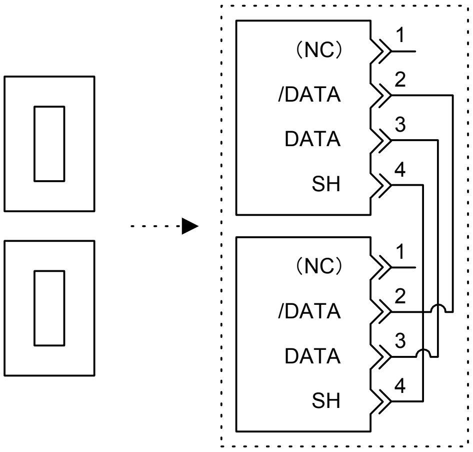
(1) Connector Shape

The shape of the MECHATROLINK-II connectors on the NT110 is shown in the following table.

Number of Connectors	Connector Shape	Connector Name
2		CN1
		CN2

(2) Internal Connections

The two MECHATROLINK-II connector ports are internally connected as shown in the following figure.



Connect a Terminator JEPMC-W6022 (for MECHATROLINK-II) to any connector that is not being used.

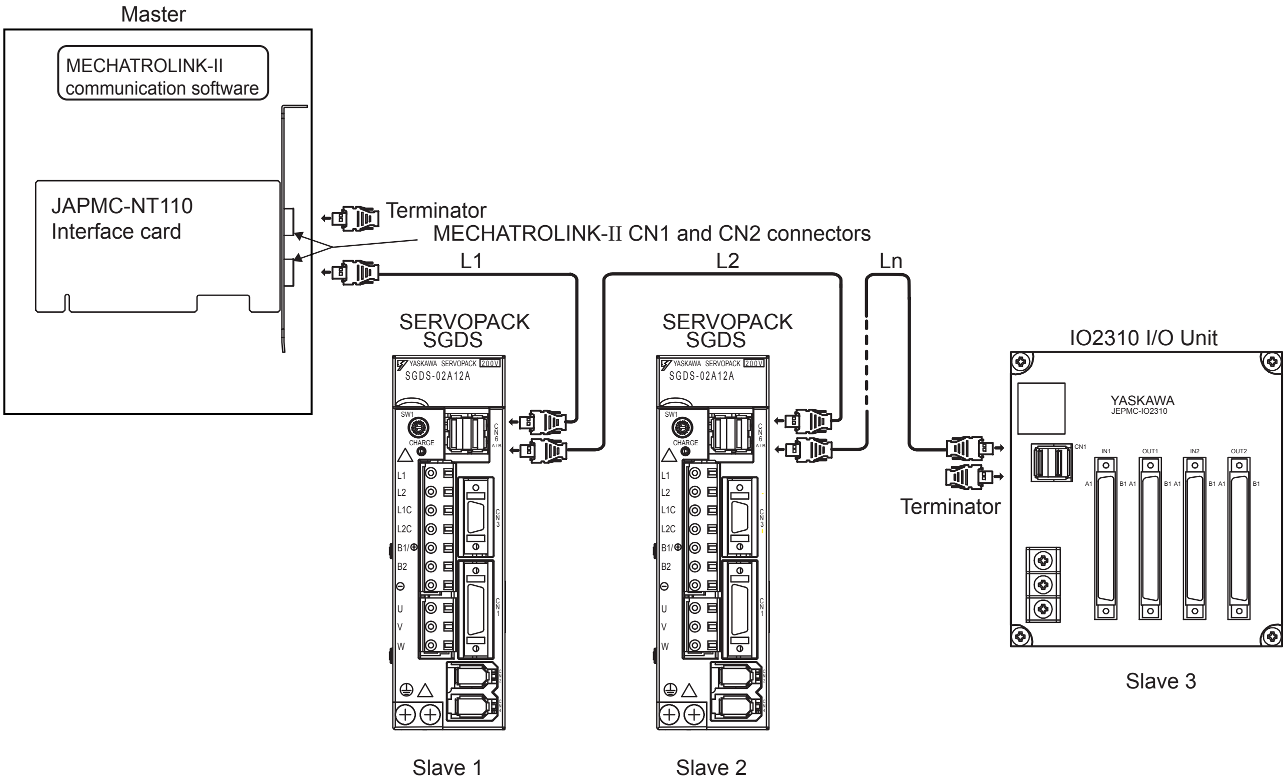
4.2.2 MECHATROLINK-II Cables

The following standard MECHATROLINK-II Cables are available from Yaskawa.

Name and Specifications	Model Number	Length (m)
MECHATROLINK-II Cable USB connector - USB connector	JEPMC-W6002-A5	0.5
	JEPMC-W6002-01	1
	JEPMC-W6002-03	3
	JEPMC-W6002-05	5
	JEPMC-W6002-10	10
	JEPMC-W6002-20	20
	JEPMC-W6002-30	30
MECHATROLINK-II Cable USB connector - USB connector (with core)	JEPMC-W6003-A5	0.5
	JEPMC-W6003-01	1
	JEPMC-W6003-03	3
	JEPMC-W6003-05	5
	JEPMC-W6003-10	10
	JEPMC-W6003-20	20
	JEPMC-W6003-30	30
Terminator (terminating resistance)	JEPMC-W6022	–

4.2.3 MECHATROLINK-II System Connection Example

An example of MECHATROLINK-II communications connections using a JAPMC-NT110 Interface Card is shown in the following figure.



Either connector CN1 or CN2 can be used.

Connect a Terminator to each end of the network.

5

Communication and Control Functions

5.1 Communication Functions

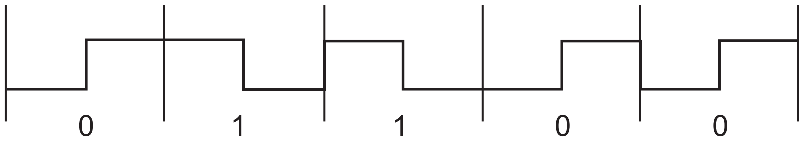
5.1.1 Frame Format

The MECHATROLINK-II frame format is shown in the following figure.

Preamble	Flags	Station address	Controls	Data	FCS	Flags
16 bits	8 bits	8 bits	8 bits	n bits（16 or 31 bytes)	16 bits	8 bits

5.1.2 Manchester Code

The MECHATROLINK-II communication LSI (JL-080) contains a converter circuit for Manchester code. Manchester code is a self-synchronizing code, which contains a clock signal in the data.



5.2 Control Functions

5.2.1 Communication LSI Interrupt Signal

(1) Interrupt Signal

An interrupt signal (/INT1) is output every 250 µs to 8 ms at the start of the communication cycle to synchronize communication.

The communication cycle, however, may be restricted by the performance of the slaves that are connected.

IMPORTANT

Set the communication cycle (250 µs to 8 ms) using the MECHATROLINK-II communication software in the master.

6

Optional Communication Software

6.1 NT110 Access API CD-ROM

Upon request, Yaskawa will provide an NT110 Access API CD-ROM for Windows RTX5.1.1, *NT110 Access API CPMC-NTA700* and an NT110 Access API CD-ROM for Windows2000/XP, *NT110 Access API CPMC-NTA701* at an extra charge. This CD-ROM includes information required to create software for MECHATROLINK-II communications if using the NT110 card.

6.1.1 Contents of NT110 Access API CD-ROM

The CPMC-NTA700 contains the files listed in the following table.

(1) For RTX

Type	Folder Name/File Name			Contents
Files (7 files)	NT110LicenseAgreement_Eng.pdf			NT110 Access API License Agreement (in English and Japanese)
	NT110LicenseAgreement_Jpn.pdf			
	NT110.rtdll			RTDLL driver for NT110
	NT110_rt.lib			API static link library
	jl080def.h			API definition header
	jl080gbl.h			API definition header for external reference
	readme_EN.txt, readme_JP.txt (in English and Japanese)			Manual for usage of the sample application
Folders (2 folders)	Sample_rtx _1thread	jl080def.h		Sample files for the CPMC-NTA700 (Parts of the files are the same as in the product software.)
		jl080gbl.h		
		NT110.rtdll		
		NT110_rt.lib		
		mcha_cmd.h		
		mst_smpl.c		
		NT110_SAMPLE_RTX.dsp		
		NT110_SAMPLE_RTX.dsw		
		Release	NT110_SAMPLE_RTX.rtss	Executable Test Program
	doc	NT110UsersMan_JP.pdf		PDF files of this manual (in English and Japanese)
		NT110UsersMan_Eng.pdf		

(2) For Windows2000/XP

Type	Folder Name/File Name			Contents
Files (6 files)	NT110LicenseAgreement_Eng.pdf			NT110 Access API License Agreement (in English and Japanese)
	NT110LicenseAgreement_Jpn.pdf			
	NT110_rt.lib			API static link library
	jl080def.h			API definition header
	jl080gbl.h			API definition header for external reference
	readme_EN.txt, readme_JP.txt (in English and Japanese)			Manual for usage of the sample application
Folders (3 folders)	Drv	NT110.inf		Windows API driver for NT110
		NT110.sys		
	NT110_ SAMPLE_ WIN	NT110.lib		Sample files for the CPMC-NTA700 (Parts of the files are the same as the product software.)
		jl080def.h		
		jl080gbl.h		
		mcha_cmd.h		
		mst_smpl.c		
		NT110_SAMPLE_WIN.dsp		
		NT110_SAMPLE_WIN.dsw		
		RtApiMain.h		
		SubFunc.h		
		Release	NT110_SAMPLE_WIN.exe	Executable Test Program
	doc	NT110UsersMan_JP.pdf		PDF files of this manual (in English and Japanese)
		NT110UsersMan_Eng.pdf		

6.1.2 Overview of Manuals

The contents and application of the files (manuals) in the *doc* folder are listed in the following table.

File Name	Document Name	Document No. (Number of Pages)	Application
NT110 UsersMan_Eng.pdf	PCI-compliant MECHATROLINK-II Interface Card JAPMC-NT110, NT111 User's Manual	SIEPC88070017	Describes details on application methods of the NT110 Interface Card.

6.2 NT110 Access API

This section outlines the API used with the NT110 card.

The API acts as a C1 Master (MECHATROLINK-II Master) and provides access to the NT110 interface board for MECHATROLINK-II communications.

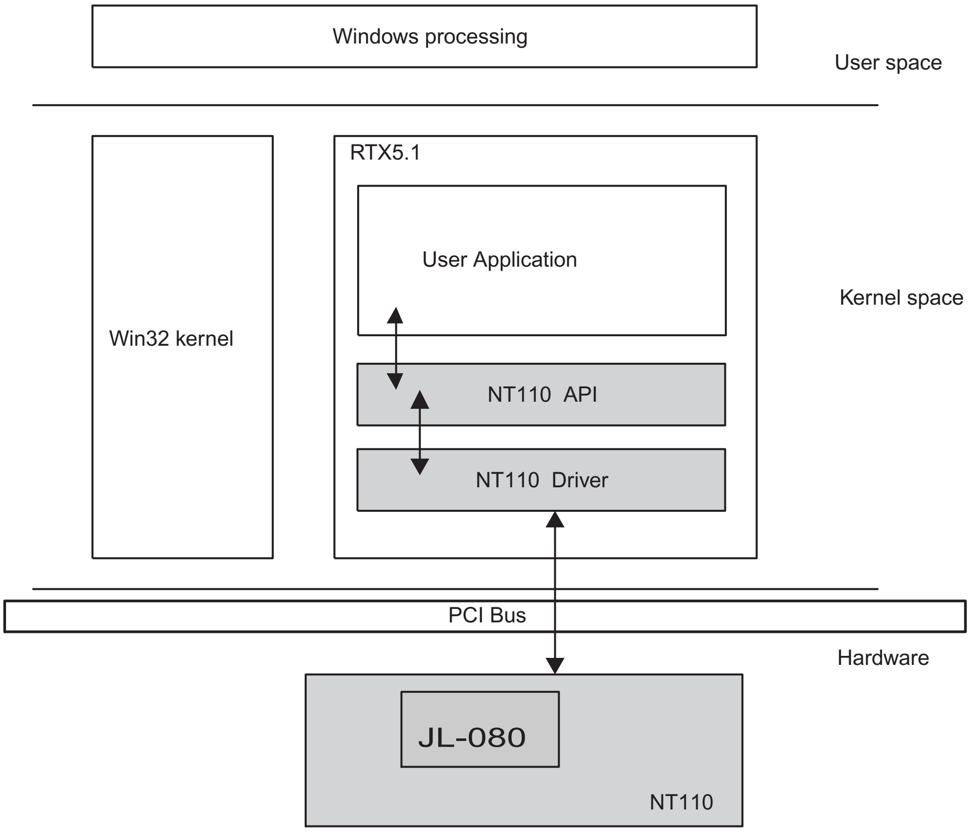
6.2.1 Applicable Range

The applicable range of the NT110 API is as follows:

- Host mode
- C1 master station
- MECHATROLINK-I/II
- OS: Windows 2000/XP + RTX5.1.1 or Windows 2000/XP

6.2.2 Software Configuration

The following illustration shows the software configuration of Master devices with the NT110.



Note: The sections shown in gray are provided by the Yaskawa Electric Corporation.

6.2.3 File Configuration

(1) API for RTX

The following file provides the NT110 API for the RTX. The configuration is as follows.

(a) File configuration

• RTDLL driver for NT110	NT110.rtdll
• API static link library	NT110_rt.lib
• API definition header	jl080def.h
• API definition header for external reference	jl080gbl.h
• Sample program source	mst_smpl.c
• Sample program header	mcha_cmd.h
• Sample program project file	NT110_SAMPLE_RTX.dsp
• Sample program project work space	NT110_SAMPLE_RTX.dsw

(b) Function configuration

- RXT Initial Setting API

yns_rtx_set_priority()	Sets the priority value of interrupt handler in RTX.
yns_rtx_set_interval()	Sets the polling cycle to detect interrupt signal from NT110.

- Communication Initialization Service API

yns_sequent_init()	Initializes JL-080 communication.
yns_check_ram()	Checks JL-080 RAM.
yns_reset_board()	Resets the software.

- Link Communication (Data Exchange) Service API

yns_read_ldata()	Reads channel data.
yns_write_ldata()	Writes channel data.
yns_read_ldata2()	Reads channel data including control field.
yns_write_ldata2()	Writes channel data including control field.
yns_begin_link_acc()	Starts link access. (Refreshes host-monitoring WDT (watchdog timer).)
yns_end_link_acc()	Ends link access.
yns_read_comstat()	Checks JL-080 status.
yns_link_exchange()	Exchanges all channel data.
yns_set_netclk()	Writes network time (required only when using a C2 master station).
yns_wait_interrupt()	Processes interrupt waiting.

(2) API for Windows2000/XP

The following file provides the NT110 API for the Windows2000 and XP. The configuration is as follows.

(a) File configuration

• Windows API for NT110	NT110.sys, NT110.inf
• API static link library	NT110.lib
• API definition header	jl080def.h
• API definition header for external reference	jl080gbl.h
• Sample program	mst_smpl.c
• Sample program header	mcha_cmd.h, Rt Api Main.h, Sub Func.h
• Sample program project file	NT110_SAMPLE_WIN.dsp
• Sample program project work space	NT110_SAMPLE_WIN.dsw

(b) Function configuration

- Driver loading/unloading API

nt110_drv_load()	Initializes the NT110 driver.
nt110_drv_unload()	Ends the NT110 driver.

- Communication Initialization Service API

yns_sequent_init()	Initializes JL-080 communication.
yns_check_ram()	Checks JL-080 RAM.
yns_reset_board()	Resets the software.

- Link Communication (Data Exchange) Service API

yns_read_ldata()	Reads channel data.
yns_write_ldata()	Writes channel data.
yns_read_ldata2()	Reads channel data including control field.
yns_write_ldata2()	Writes channel data including control field.
yns_begin_link_acc()	Starts link access. (Refreshes host-monitoring WDT (watchdog timer).)
yns_end_link_acc()	Ends link access.
yns_read_comstat()	Checks JL-080 status.
yns_link_exchange()	Exchanges all channel data.
yns_set_netclk()	Writes network time (required only when using a C2 master station.)
yns_wait_interrupt()	Processes interrupt waiting.

6.2.4 Using Drivers

(1) RTX Driver

To load the RTDLL driver NT110.rtdll for the NT110, execute "RTSSrun /d nt110.rtdll" from the Console. Or, use the RTSSrun Utility to select ***Register RTDLL***, ***nt110.rtdll***, and click the **OK Button**.

After loading the driver, use *yns_rtx_set_priority()* to set the thread priority of the driver interrupt handler.

(2) Windows2000/XP Driver

The following two files are required to install the NT110 Windows driver.

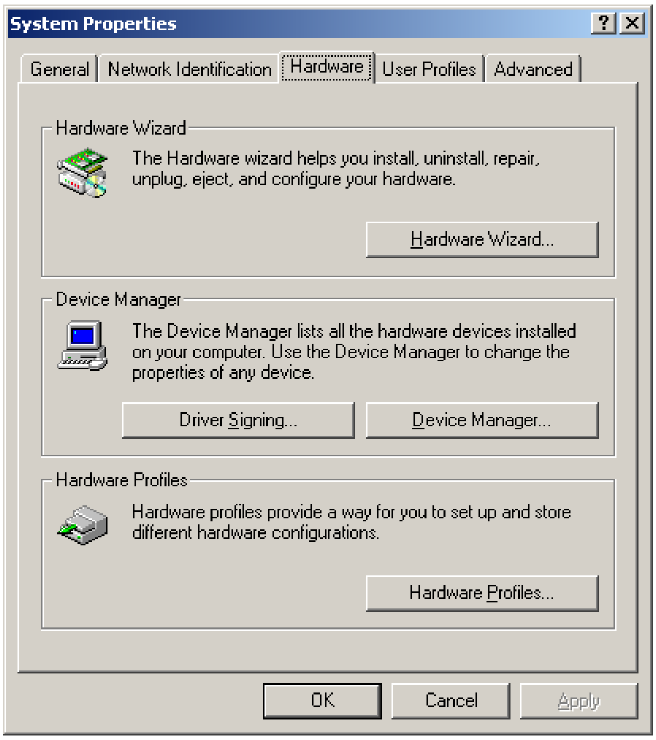
- NT110.sys
- NT110.inf

(a) Installing the NT110 Windows driver

For the personal computer in which no drivers for Yaskawa products such as MP2100 series have been installed, start the installation procedures from step 4.

On the personal computer in which a driver for Yaskawa products has been installed, the wizard to upgrade the device drivers may not start because the personal computer does not recognize the NT110 Windows driver as a new device. In such case, start the procedures from step 1.

1. Select ***System Properties*** from ***Control Panel***, and then select ***Device Manager.***

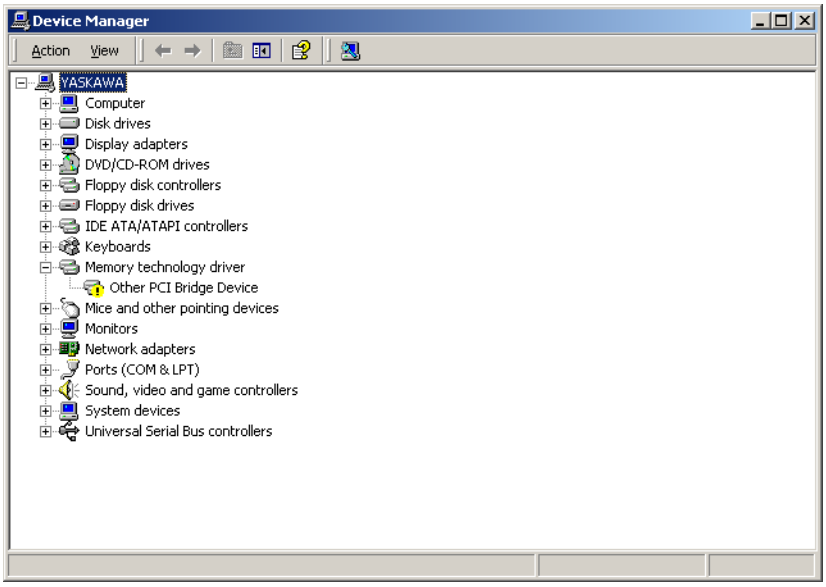


The Hardware wizard helps you install, uninstall, repair, unplug, eject, and configure your hardware.

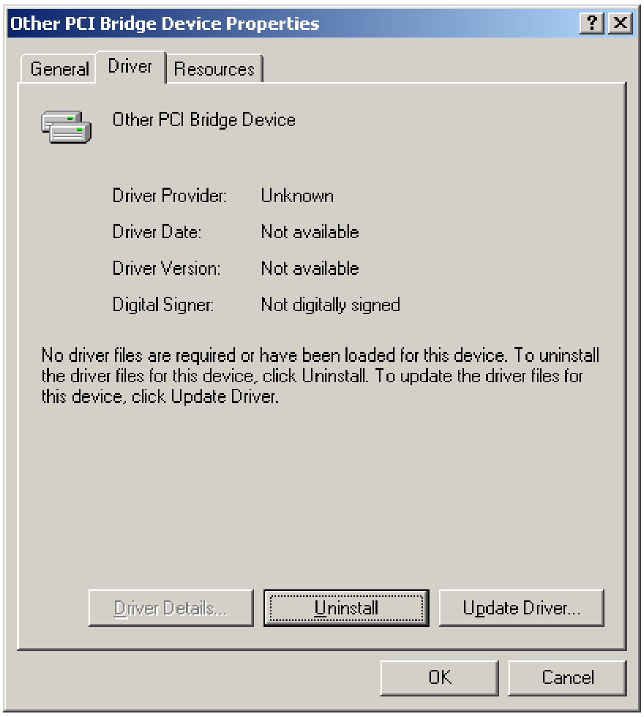
The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Hardware profiles provide a way for you to set up and store different hardware configurations.

2. Open the properties of the Other PCI Bridge Device in one of the following manners.
   - Double-click ***Other PCI Bridge Device***.
   - Right-click ***Other PCI Bridge Device***, and then select ***Properties***.



3. Click the **Update Driver Button**.



Other PCI Bridge Device

Driver Provider: Unknown

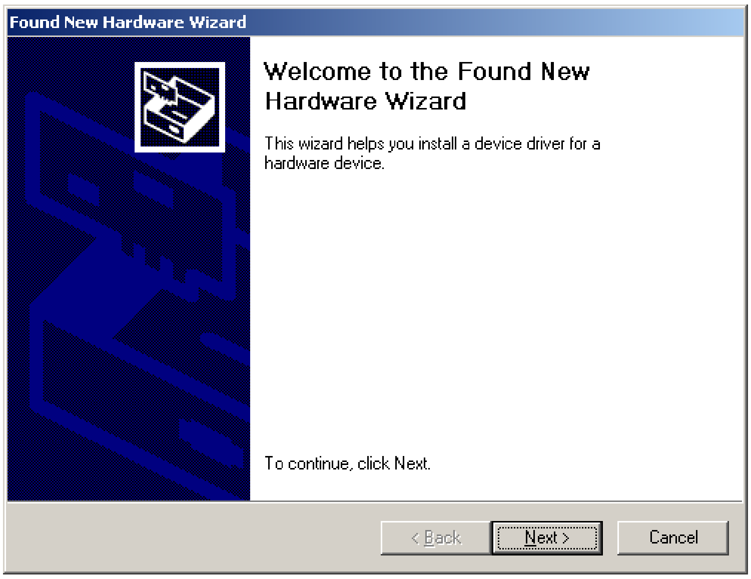
Driver Date: Not available

Driver Version: Not available

Digital Signer: Not digitally signed

No driver files are required or have been loaded for this device. To uninstall the driver files for this device, click Uninstall. To update the driver files for this device, click Update Driver.

4. Click the **Next Button**.



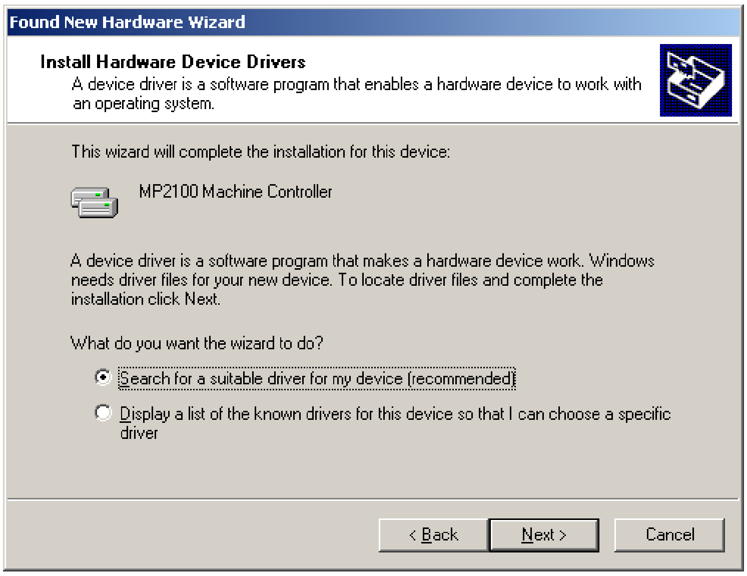
Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install a device driver for a hardware device.

To continue, click Next.

5. Select ***Search for a suitable driver for my device [recommended]***, and then click the ***Next Button***.



Found New Hardware Wizard

Install Hardware Device Drivers

A device driver is a software program that enables a hardware device to work with an operating system.

This wizard will complete the installation for this device:

MP2100 Machine Controller

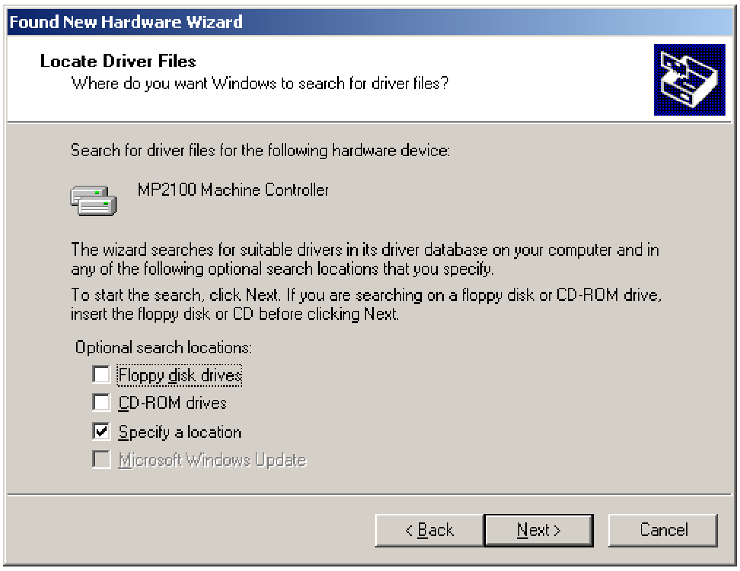
A device driver is a software program that makes a hardware device work. Windows needs driver files for your new device. To locate driver files and complete the installation click Next.

What do you want the wizard to do?

● Search for a suitable driver for my device [recommended]

○ Display a list of the known drivers for this device so that I can choose a specific driver

6. Select ***Specify a location*** check box, and then click the **Next Button**.



Found New Hardware Wizard

Locate Driver Files
Where do you want Windows to search for driver files?

Search for driver files for the following hardware device:

MP2100 Machine Controller

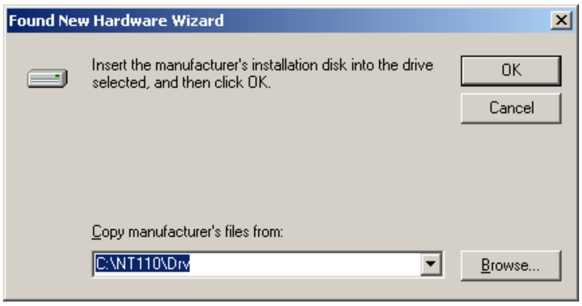
The wizard searches for suitable drivers in its driver database on your computer and in any of the following optional search locations that you specify.

To start the search, click Next. If you are searching on a floppy disk or CD-ROM drive, insert the floppy disk or CD before clicking Next.

Optional search locations:

- □ Floppy disk drives
- □ CD-ROM drives
- ☑ Specify a location
- □ Microsoft Windows Update

7. Click the **Browse Button** to select a directory of the driver, and then click the **OK Button**.



Found New Hardware Wizard

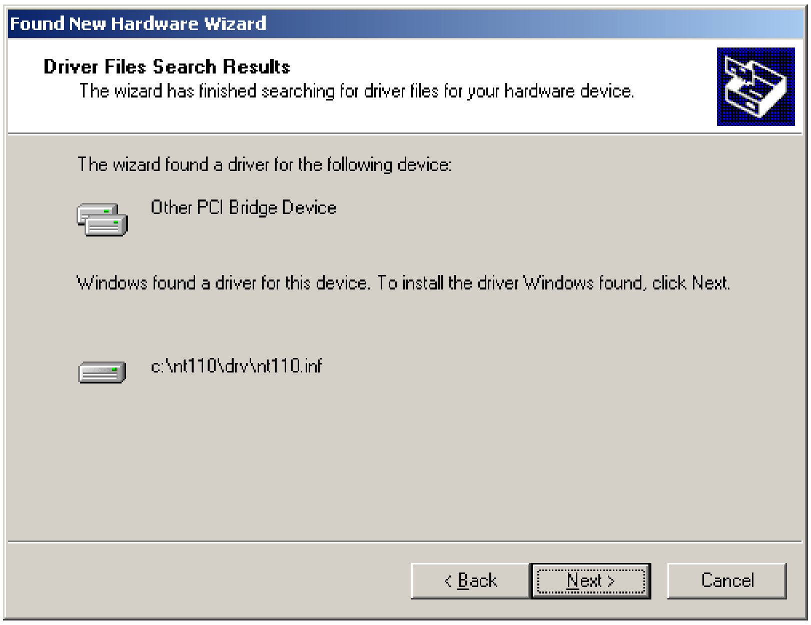
Insert the manufacturer's installation disk into the drive selected, and then click OK.

Copy manufacturer's files from:

C:\NT110\Drv

- When using the personal computer in which no drivers for Yaskawa products have been installed, proceed to Step 8.
- When using the personal computer in which a driver for Yaskawa product has been installed, proceed to Step 9.

8. Check if the driver that Windows found is the NT110 Windows driver. If yes, click the **Next Button**, and proceed to Step 11.



Found New Hardware Wizard

Driver Files Search Results
The wizard has finished searching for driver files for your hardware device.

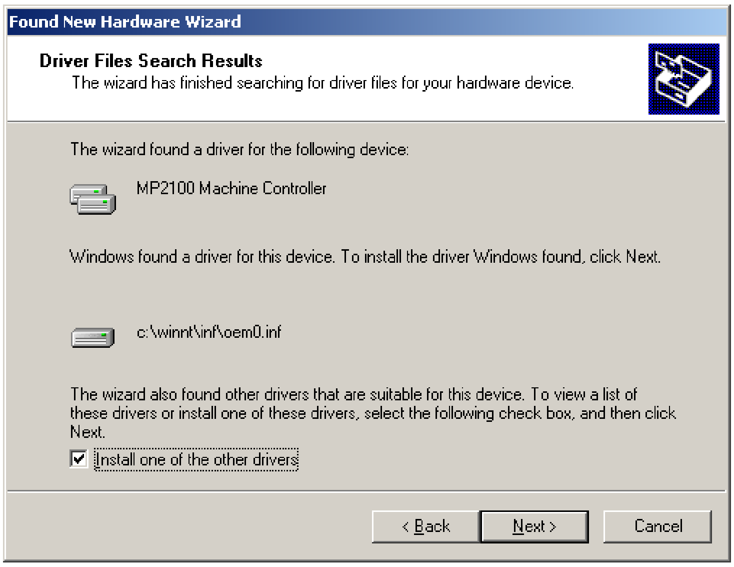
The wizard found a driver for the following device:

Other PCI Bridge Device

Windows found a driver for this device. To install the driver Windows found, click Next.

c:\nt110\drv\nt110.inf

9. Select ***Install one of the other drivers*** check box, and then click the **Next Button**.



Found New Hardware Wizard

Driver Files Search Results

The wizard has finished searching for driver files for your hardware device.

The wizard found a driver for the following device:

MP2100 Machine Controller

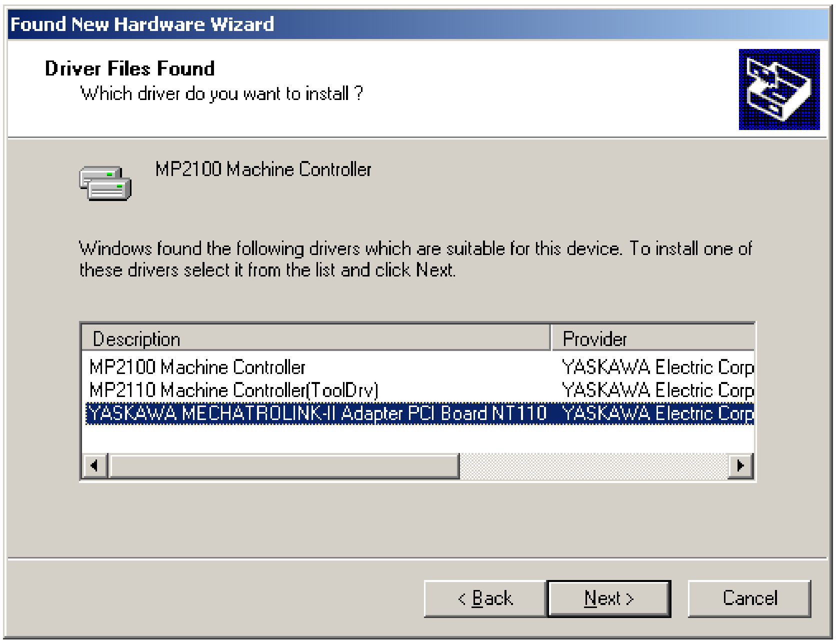
Windows found a driver for this device. To install the driver Windows found, click Next.

c:\winnt\inf\oem0.inf

The wizard also found other drivers that are suitable for this device. To view a list of these drivers or install one of these drivers, select the following check box, and then click Next.

☑ Install one of the other drivers

10.Select the NT110 driver, and then click the **Next Button**.



Found New Hardware Wizard

Driver Files Found

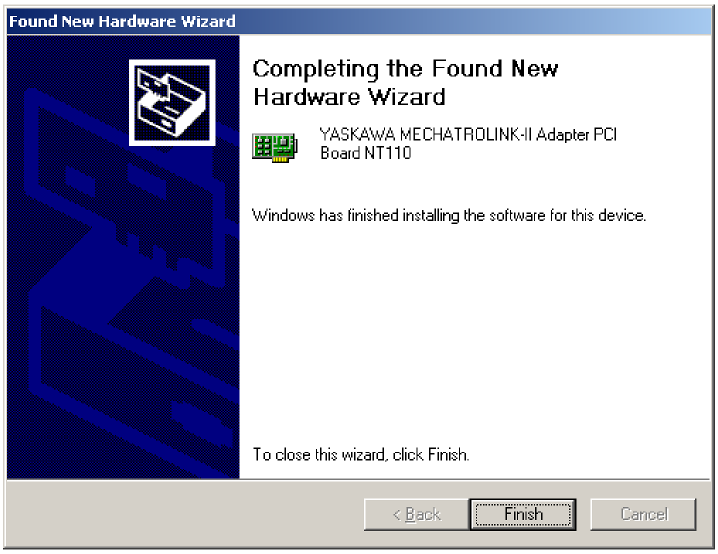
Which driver do you want to install ?

MP2100 Machine Controller

Windows found the following drivers which are suitable for this device. To install one of these drivers select it from the list and click Next.

Description	Provider
MP2100 Machine Controller	YASKAWA Electric Corp
MP2110 Machine Controller(ToolDrv)	YASKAWA Electric Corp
YASKAWA MECHATROLINK-II Adapter PCI Board NT110	YASKAWA Electric Corp

11. After confirming that the installation of the NT110 has completed, click the **Finish Button**.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

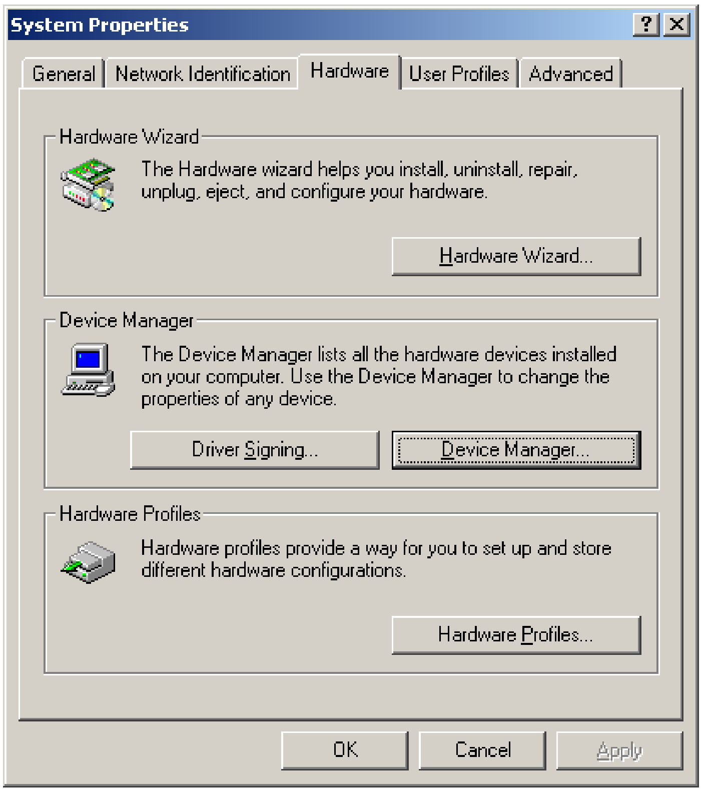
YASKAWA MECHATROLINK-II Adapter PCI Board NT110

Windows has finished installing the software for this device.

To close this wizard, click Finish.

(b) Confirming the Installed Driver

1. Select ***System Properties*** from ***Control Panel***, and then select ***Device Manager***.



System Properties

Hardware Wizard

The Hardware wizard helps you install, uninstall, repair, unplug, eject, and configure your hardware.

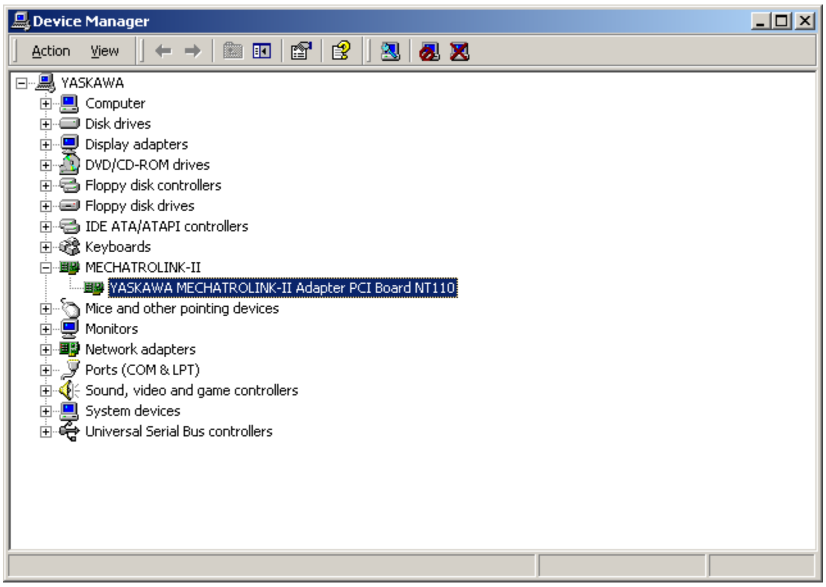
Device Manager

The Device Manager lists all the hardware devices installed on your computer. Use the Device Manager to change the properties of any device.

Hardware Profiles

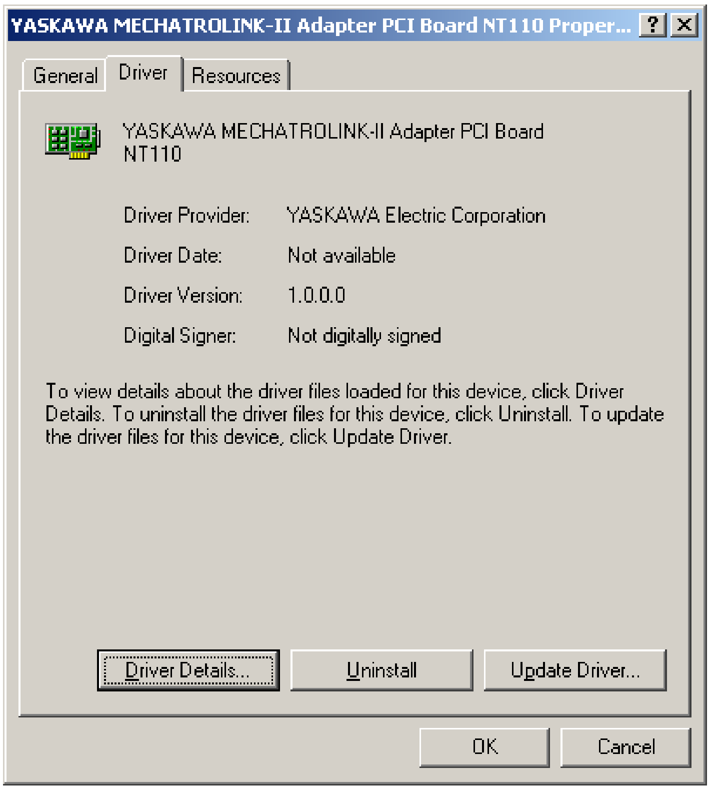
Hardware profiles provide a way for you to set up and store different hardware configurations.

2. Open the properties of the Other PCI Bridge Device in one of the following manners.
   - Double-click ***Other PCI Bridge Device***.
   - Right-click ***Other PCI Bridge Device***, and then select ***Properties***.



YASKAWA
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Floppy disk controllers
- Floppy disk drives
- IDE ATA/ATAPI controllers
- Keyboards
- MECHATROLINK-II
  - YASKAWA MECHATROLINK-II Adapter PCI Board NT110
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

3. Check driver name.



YASKAWA MECHATROLINK-II Adapter PCI Board NT110

Driver Provider: YASKAWA Electric Corporation

Driver Date: Not available

Driver Version: 1.0.0.0

Digital Signer: Not digitally signed

To view details about the driver files loaded for this device, click Driver Details. To uninstall the driver files for this device, click Uninstall. To update the driver files for this device, click Update Driver.

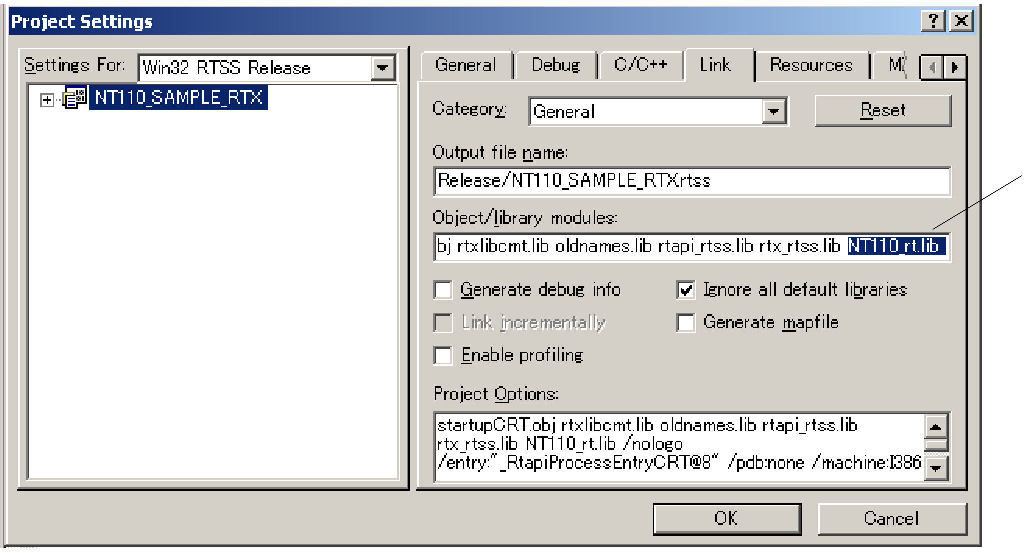
6.2.5 Using API

(1) Include Files

The program using API requires linking the NT110_rt.lib of the LIB file.

(a) For RTX

Sets the link of API for NT110 as follows.



Project Settings

Settings For: Win32 RTSS Release

NT110_SAMPLE_RTX

Category: General

Output file name: Release/NT110_SAMPLE_RTXrtss

Object/library modules: bj rtxlibcmt.lib oldnames.lib rtapi_rtss.lib rtx_rtss.lib NT110_rt.lib

□ Generate debug info ☑ Ignore all default libraries
□ Link incrementally □ Generate mapfile
□ Enable profiling

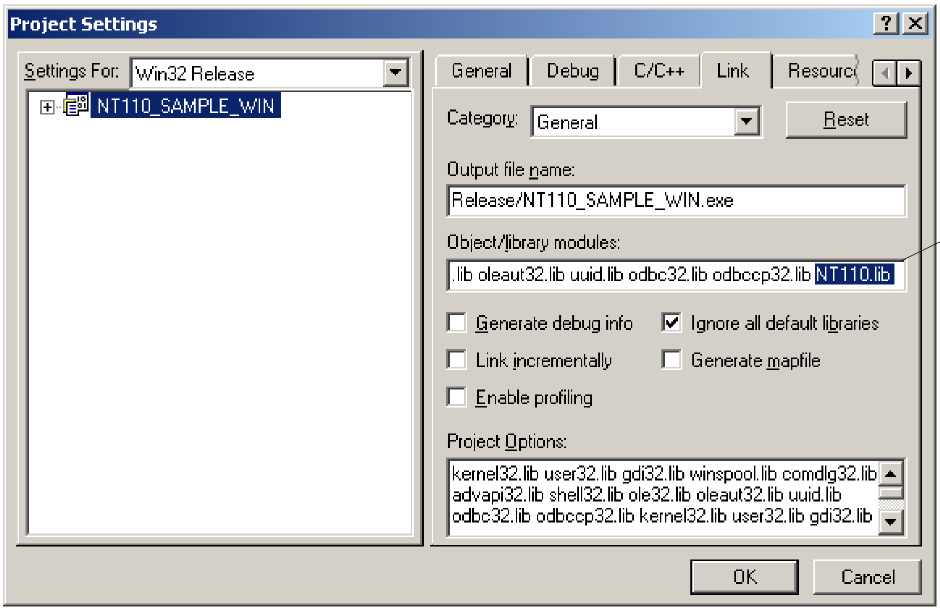
Project Options:

```
startupCRT.obj rtxlibcmt.lib oldnames.lib rtapi_rtss.lib
rtx_rtss.lib NT110_rt.lib /nologo
/entry:"_RtapiProcessEntryCRT@8" /pdb:none /machine:I386
```

Add NT110_rt.lib.

(b) For Windows2000/XP

Set the link of API for NT110 as follows.



Project Settings

Settings For: Win32 Release

NT110_SAMPLE_WIN

Category: General

Output file name: Release/NT110_SAMPLE_WIN.exe

Object/library modules: .lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib NT110.lib

□ Generate debug info ☑ Ignore all default libraries
□ Link incrementally □ Generate mapfile
□ Enable profiling

Project Options:

```
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib kernel32.lib user32.lib gdi32.lib
```

Add NT110.lib.

(2) Access Limit

When accessing one NT110, only one thread can call the API for link communication.

Several threads cannot access one NT110 simultaneously.

(3) Precautions on using RTX

When using the RTX, change the settings as follows.

The settings differ from the default setting.

- Run time library: multi-thread run time
- Structure alignment: 8 bytes
- Character string support: ASCII only

(4) NT110 Insertion Limitations

NT110 can select slot 1 or 2 by mode switch setting. For setting the mode switch, refer to *3.1.2 Mode Switch Specifications*.

When using API for RTX, do not insert several NT110s into one personal computer.

The reason is: The API for RTX is in static buffer mode. When processing the second NT110 after processing first NT110, the API cannot exchange data at the set timing. Refer to *6.3.5 Data Exchange Access Timing*.

When using the API for Windows, API is in the alternating buffer mode. So, two NT110s can be inserted into one personal computer.

6.2.6 Setting RTX

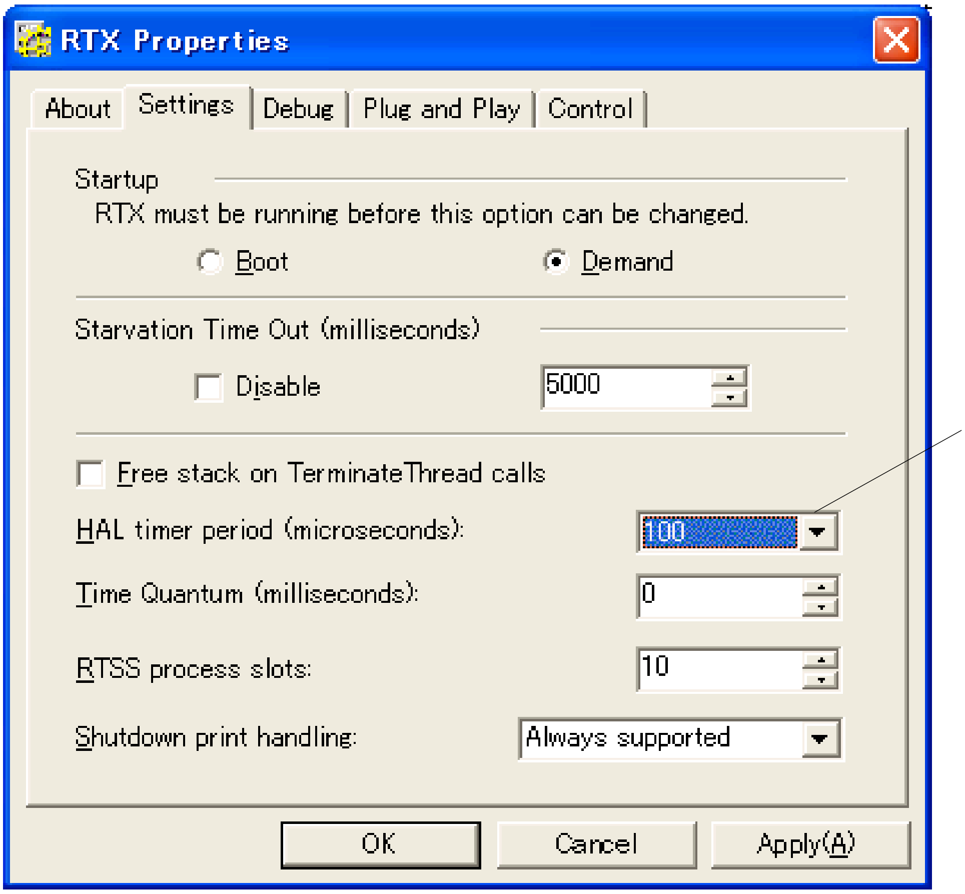
Set the processing time in the RXT properties to the minimum value of 100 µs.

Open the RTX properties as follows:

Start → Program → VenturCom RTX → RTX → RTX Properties.

IMPORTANT

If the RTX processing time exceeds the NT110 processing time, RXT cannot synchronize with NT110 and the communication error occurs.



Startup

RTX must be running before this option can be changed.

○ Boot ● Demand

Starvation Time Out (milliseconds)

□ Disable 5000

□ Free stack on TerminateThread calls

HAL timer period (microseconds): 100

Time Quantum (milliseconds): 0

RTSS process slots: 10

Shutdown print handling: Always supported

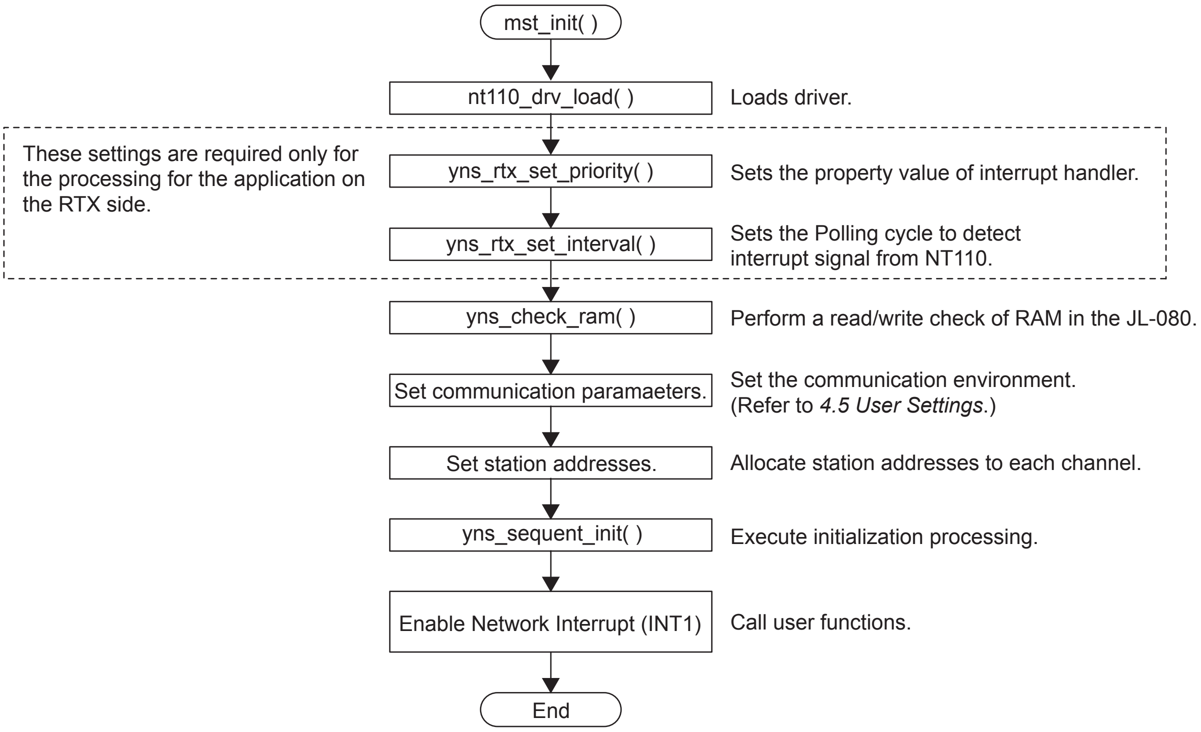
Set to 100.
(Default setting is 500.)

6.3 Processing Overview

This section outlines an overview of the processing of the NT110 API.

6.3.1 Communication Initialization Procedure

Initializes the JL-080 and makes network settings.


mst_init( )
nt110_drv_load( )
Loads driver.
These settings are required only for the processing for the application on the RTX side.
yns_rtx_set_priority( )
Sets the property value of interrupt handler.
yns_rtx_set_interval( )
Sets the Polling cycle to detect interrupt signal from NT110.
yns_check_ram( )
Perform a read/write check of RAM in the JL-080.
Set communication paramaeters.
Set the communication environment. (Refer to *4.5 User Settings*.)
Set station addresses.
Allocate station addresses to each channel.
yns_sequent_init( )
Execute initialization processing.
Enable Network Interrupt (INT1)
Call user functions.
End

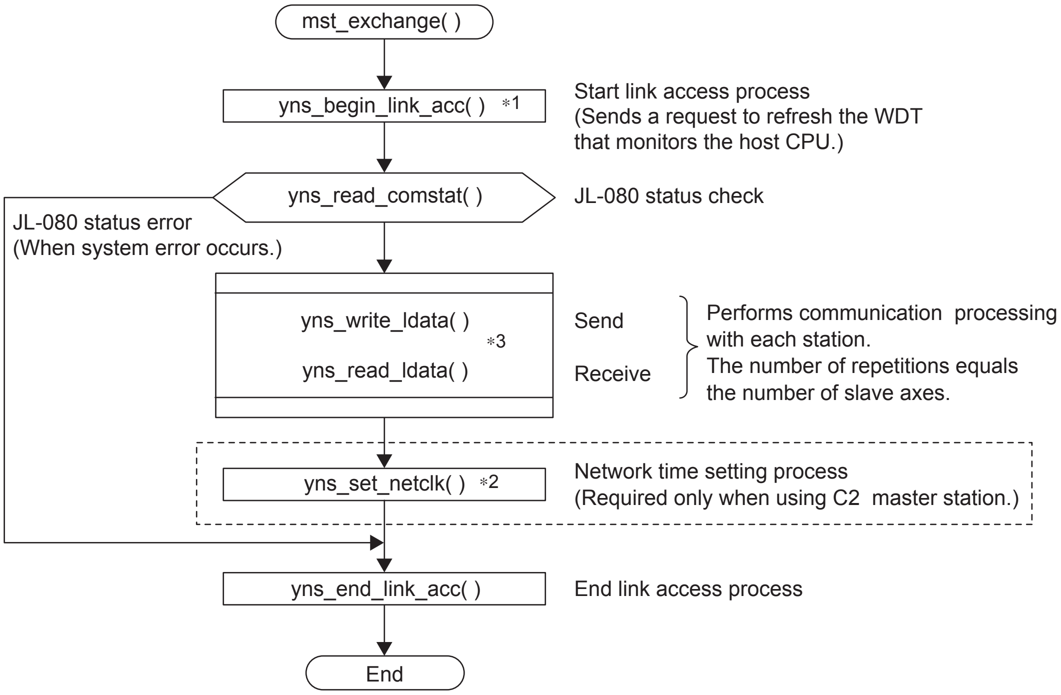
6.3.2 End Processing

When the application ends, unload the driver. Carry out *nt110_drv_unload()*.

6.3.3 Link Data Exchange Procedure

The following flowchart shows the flow of link data exchange processing. In each communication cycle, the send data is written to the JL-080 and the receive data is read. Also, the JL-080's status is checked.

Perform the processing at the beginning of a communication cycle interrupt (INT1).


mst_exchange( )
yns_begin_link_acc( ) *1
Start link access process
(Sends a request to refresh the WDT that monitors the host CPU.)
yns_read_comstat( )
JL-080 status check
JL-080 status error
(When system error occurs.)
yns_write_ldata( )
*3
yns_read_ldata( )
Send
Receive
Performs communication processing with each station.
The number of repetitions equals the number of slave axes.
yns_set_netclk( ) *2
Network time setting process
(Required only when using C2 master station.)
yns_end_link_acc( )
End link access process
End

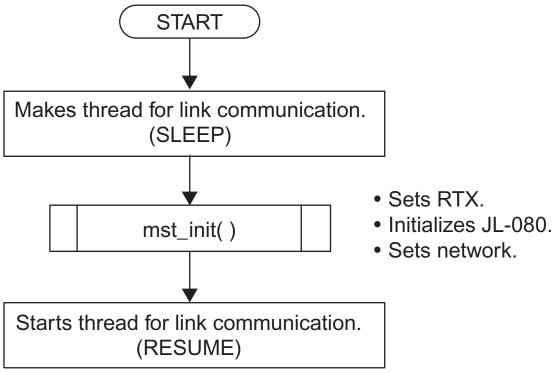
* 1. Execute this function at the beginning and end of the link data transfer processing. Refer to *6.4 User Settings* when using the WDT that monitors the host CPU.

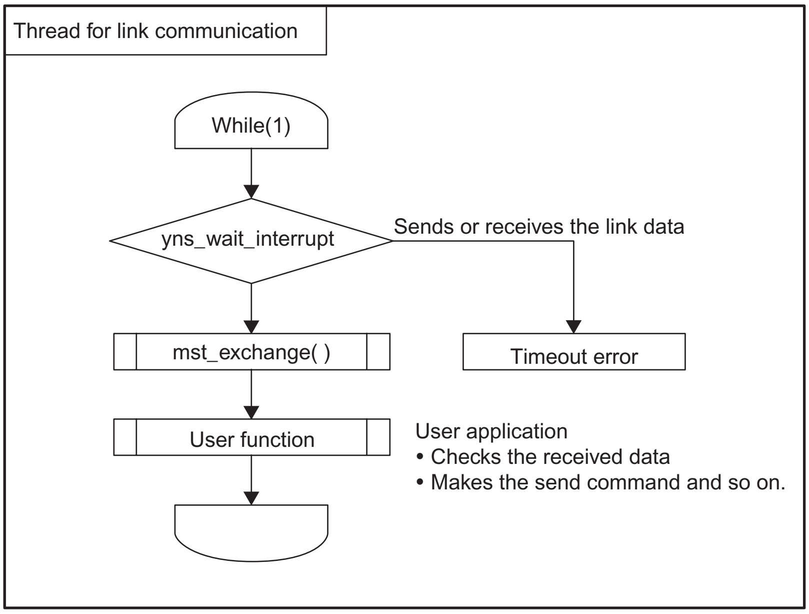
* 2. This process is required only when a C2 master station is connected.

* 3. When access in the control field of link data, use yns_write_ldata2() and yns_read_ldata2().

6.3.4 Processing Examples

The following examples show the sequence of process from NT110 initialization to link communication.




Thread for link communication
While(1)
yns_wait_interrupt
Sends or receives the link data
mst_exchange( )
Timeout error
User function
User application
• Checks the received data
• Makes the send command and so on.

INFO

If using the RTX API, intensive memory access will occur immediately after starting Windows. As a result, processing of communications for the NT110 Interface Card (i.e., the processing for the application on the RTX side) may not match with the timing of the communications cycle, and a reception error may result.

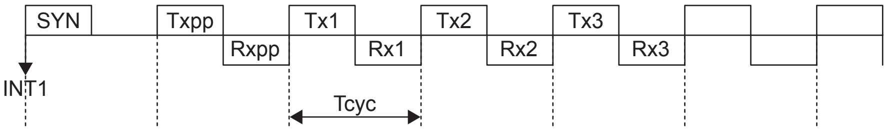
After starting Windows, first access the memory and then start the processing for NT110 communications.

However, using the API for Windows 2000 or XP that provides asynchronous processing will not cause the reception error.

6.3.5 Data Exchange Access Timing

In static buffer mode, data exchange between JL-080 and host CPU connected through bus must be synchronized with the transmission sequence of MECHATROLINK-II. The condition is explained as follows.

The following figure shows the transmission sequence and the interrupt signal for the host CPU.



SYN: Synchronous frame

Txpp/Rxpp: Transmission/reception channel for C2 master station

Txn: Transmission frame for the number n slave station

Rxn: Reception frame for the number n slave station

Tcyc: Time slot (17-byte mode: 60µs, 32-byte mode: 90µs)

INT1: Synchronized interrupt signal for host CPU

Data access to the JL-080 starts at INT1 as trigger signal. Data exchange for the number n slave station between JL-080 and host CPU must be completed before the transmission sequence starts for the number n slave station on the MECHATROLINK-II network.

For example, data exchange for the second slave station must be completed within 3 Tcycs form INT1.

6.4 User Settings

This section explains the communication parameters that must be changed by the user when initializing communication. Set the communication parameters in the members of the YENET_USER_PAR structure shown in the following table.

struct YENET_USER_PAR

Type	Member Name	Description
unsigned short	slot	Connection slot (1: fixed)
unsigned short	mod	Communication mode setting
unsigned short	ma	C1 master station address
unsigned short	ma_max	Max. number of connectable slave stations
unsigned short	t_mcyc	Transmission cycle
unsigned short	t_cyc	Time slot width
unsigned short	byte	Number of transmission bytes
unsigned short	dev	INT1 dividing ratio (for INT2 output)
unsigned short	max_rtry	Max. number of retries
unsigned short	c2m_ch	Time slot for C2 master station
unsigned short	wdt	WDT function setting for host monitoring

6.5 Communication Parameter Details

- mod

Set the logical OR of the following defined symbols. The sample program has the settings for the MECHATROLINK-II (MECHATROLINK-II protocol, 10 Mbps, and C1 master station).

Symbol	Description	
YN_MOD_PROCEL_1NS	Communication protocol setting [*1]	MECHATROLINK-I compatible
YN_MOD_PROCEL_2NS		MECHATROLINK-II
YN_MOD_SPEED_4M	Baud rate setting [*1]	4 Mbps
YN_MOD_SPEED_10M		10 Mbps
YN_MOD_BMOD_STATIC	Buffer mode setting [*1, *2]	Static buffer
YN_MOD_BMOD_ALT		Alternating buffer
YN_MOD_BMOD_RALT		Alternating buffer only for reception
YN_MOD_TYPE_MASTER	Operating mode setting [*1]	C1 master station
YN_MOD_TYPE_SLAVE		Slave station
YN_MOD_ESYNCS	Synchronized with the external synchronization input (RTCIL). If this setting is not specified, operation will be synchronized with the internal clock.	
YN_MOD_INT_FRS	Causes a synchronous interrupt to be generated by the internal timer. (Required when used as a C1 master station.)	

* 1. These settings are mandatory.

- Example 1: MECHATROLINK-I Settings (MECHATROLINK-I protocol, 4 Mbps, static buffer, and master station)
- mod = YN_MOD_PROCEL_1NS | YN_MOD_SPEED_4M | YN_MOD_BMOD_STATIC | YN_MOD_TYPE_MASTER | YN_MOD_INT_FRS;
- Example 2: MECHATROLINK-II Settings (MECHATROLINK-II protocol, 10 Mbps, static buffer, and master station)
- mod = YN_MOD_PROCEL_2NS | YN_MOD_SPEED_10M | YN_MOD_BMOD_STATIC | YN_MOD_TYPE_MASTER | YN_MOD_INT_FRS;

* 2. If using Windows XP or 2000, transmission data cannot be read or written in synchronization with the transmission cycle because Windows XP or 2000 does not have the capability for processing in real time. We recommend using an alternating buffer.

- ma (Local station address)

01 Hex (C1 master station) Fixed

- ma_max (Max. number of connectable slave stations)

Setting range: 1 to 30 Units: Stations
(with MECHATROLINK-I mode: 14 [stations])

- t_mcyc (Transmission cycle)

Setting range: 500 to 32,000 (125 µs to 8 ms) Units: 0.25 µs
(with MECHATROLINK-I mode: 2000 (2000 [µs])

- t_cyc (Time slot width)

With MECHATROLINK-II communication 17-byte mode: 240 (60 µs)
With MECHATROLINK-II communication 32-byte mode: 360 (90 µs)
Units: 0.25 µs
(with MECHATROLINK-I mode: 130 (130 [µs])

- byte (Number of MECHATROLINK field transmission bytes)

With MECHATROLINK-II communication 17-byte mode: 17
With MECHATROLINK-II communication 32-byte mode: 32
Units: Bytes
(with MECHATROLINK-I mode: 17 [bytes])

- dev (INT1 dividing ratio for INT2 output)

0: Dividing ratio = 2
1: Dividing ratio = 4
2: Dividing ratio = 8
3: Dividing ratio = 16

- max_rtry (Max. number of retries)

Sets the maximum number of channels for retries.
Setting range: 0 to 7 Units: Channels
(with MECHATROLINK-I mode: 0 [channels])

- c2m_ch (Time slot for C2 master station)

0: Do not reserve channel for C2 master station.
1: Reserve channel for C2 master station.
(with MECHATROLINK-I mode: 0)

Note: C2 master station: Network configurator, etc.

- wdt (WDT function setting of JL-080 for host monitoring)

Sets the time to wait before stopping communication after the last WDT refresh request is received from the host CPU.
Setting range: 0 (WDT function is canceled.)
1 to 15 (10 to 150 ms) Units: 10 ms
(with MECHATROLINK-I mode: 0 [ms])

6.6 Error Processing

This section describes the errors that may be returned by the various communication driver functions during initialization or link communication. The errors can be divided into three categories: Communication errors, system errors, and user errors. The master station must perform appropriate error processing for each error category to satisfy the system specifications.

- Communication Errors:
  These errors are caused by factors such as external noise.
  The frequency of occurrence depends upon the operating environment. The error processing is required in the master station software.
- System Errors:
  System errors can be caused by hardware failures or application problems.
- User Errors:
  User error means an incorrect setting in the master station.
  (The setting must be checked and corrected.)

6.6.1 Initialization Errors

Initialization Function Error Details

Function Name	Error	Value	Description	Error Category
yns_check_ram	ERROR_RAM_CHECK	-11	The data written to the RAM area used by the JL-080 does not match the data read from the same area.	System
yns_sequent_init	ERROR_INIT_CHANNEL	-1	Initialization of the CHANNEL_INFO structure's buffer was not completed correctly.	System
	ERROR_INVALID_PROTOCOL	-2	An invalid protocol was selected. Check the *mod* setting.	User
	ERROR_INVALID_BMODE	-3	An invalid buffer mode was selected. Check the *mod* setting.	User
	ERROR_INVALID_PARAMETER	-4	A user parameter setting is incorrect.	User
	ERROR_NOT_READY	-5	The JL-080 reset operation failed.	System
	ERROR_POLL_TIMEOUT_REG	-6	The communication mode setting process was not completed correctly.	System
	ERROR_POLL_TIMEOUT_PAR	-7	The communication parameter setting process was not completed correctly.	System
	ERROR_POLL_TIMEOUT_ENA	-8	The JL-080's initialization sequence end process was not completed correctly.	System
	ERROR_RAM_SIZE_OVER	-9	The internal RAM area used by the JL-080 is insufficient. (Default: 4 KB)	User
	ERROR_SLOTCH_OVER	-12	Station number exceeds the maximum station number. Change the setting to satisfy the following condition: t_mcyc > (ma_max+c2m_ch+max_rtry+1)*t_cyc	User
yns_reset_board	ERROR_BOARD_RESET_FAIL	-13	Failed the soft reset of the board.	System

6.6.2 Link Communication Errors

Link Communication Function Error Details

Function Name	Error	Bit Position	Value	Description	Error Category
yns_begin_link_acc	ERROR_ALT_BUFFER	–	-21	While operating in alternating buffer mode, it was not possible to switch the alternating buffer (toggling failed) at the beginning of a communication interrupt. (This error will not occur in static buffer mode.)	System
yns_read_ldata yns_read_ldata2	ERROR_INVALID_ST_NO	–	-20	An out-of-range station number was set.	User
	ERROR_RECV_DATA	–	-24	Either the receive data was not received or a data reception error occurred.	Communication
	ERROR_REQ_SIZE	–	-23	The data offset setting exceeds the length of the link data.	User
yns_write_ldata yns_write_ldata2	ERROR_INVALID_ST_NO	–	-20	An out-of-range station number was set.	User
	ERROR_REQ_SIZE	–	-23	The data offset setting exceeds the length of the link data.	User
yns_end_link_acc	ERROR_TOGGLE_BUFFER	–	-22	The alternating buffer switch (toggle) request could not be set. When there is an error, the JL-080 reuses the buffer used in the last cycle and performs cyclic transmission. (This error will not occur in static buffer mode.)	Communication
yns_wait_interrupt	ERROR_WAIT_TIMEOUT	–	-25	Interrupt waiting status time out.	Communication
yns_read_comstat	WDTOVR	Bit F	1	A timeout occurred in the JL-080's built-in WDT. This error can occur when the host WDT is enabled in the communication parameters.	System
	EWDTOVR	Bit E	1	A timeout occurred in the JL-080's external WDT. This error indicates a JL-080 malfunction.	System
	RFO_ERR	Bit C	1	An overrun error occurred in the JL-080's internal FIFO reception.	System
	TUR_ERR	Bit B	1	An underrun error occurred in the JL-080's internal FIFO transmission.	System
	TGL_ERR	Bit A	1	The alternating buffer switch (toggle) request could not be set. When there is an error, the JL-080 reuses the buffer used in the last cycle and performs cyclic transmission.	User
	SPT_ERR	Bit 5	1	A short-packet reception error occurred (received data of 2 bytes or less). The ABT_ERR error will be set at the same time.	Communication
	ALG_ERR	Bit 4	1	A reception alignment error occurred (received data that was not in an 8-bit format). The ABT_ERR error will be set at the same time.	Communication
	TMCYCOVR	Bit 3	1	This error occurs when data transmission and reception could not be performed for all of the stations within the T_MCYC transmission cycle time. Possible causes of this error include incorrect settings in the number of slave stations (MA_MAX), transmission cycle (T_MCYC), or time slot width (T_CYC) of communication parameters.	User
	ABT_ERR	Bit 1	1	Data reception was canceled by ABORT reception.	Communication
	CRC_ERR	Bit 0	1	A CRC error occurred during data reception.	Communication

6.6.3 Driver Setting Errors

Function Name	Error	Value	Description	Error Category
nt110_drv_load	ERROR_DRV_LOAD	-16	Failed to initialize the driver for NT110.	User
nt110_drv_unload	ERROR_DRV_UNLOAD	-17	Failed to end the driver for NT110.	User

6.6.4 RTX Initialization Errors

Function Name	Error	Value	Description	Error Category
yns_rtx_set_priority	ERROR_PRIORITY	-14	Property is set out of setting range.	User
yns_rtx_set_interval	ERROR_INTERVAL	-15	Interval is set out of setting range.	User

6.7 API References

List of API References

Function Name	Description
(1) yns_sequent_init	Initializes the JL-080 and initializes communication with user parameters.
(2) yns_begin_link_acc	Executes the link communication start processing.
(3) yns_link_exchange	Performs data transmission and reception with all slave stations that have been set.
(4) yns_read_ldata	Reads data from the specified slave station's link data area.
(5) yns_read_ldata2	Reads data including control field from the specified slave station's link data area.
(6) yns_write_ldata	Writes data to the specified slave station's link data area.
(7) yns_write_ldata2	Writes data including control field to the specified slave station's link data area.
(8) yns_read_comstat	Checks the JL-080's status.
(9) yns_end_link_acc	Executes the link communication end processing.
(10) yns_set_netclk	Refreshes the network time.
(11) yns_check_ram	Performs a read/write check on the RAM area used by the JL-080.
(12) yns_wait_interrupt	Waits the interrupt from NT110.
(13) yns_reset_board	Executes the soft reset of the NT110.
(14) yns_rtx_set_priority	Sets the priority value of interrupt handler in RTX.
(15) yns_rtx_set_interval	Sets polling cycle to detect interrupt signal from NT110.
(16) nt110_drv_load	Executes the initialization processing of the NT110 (LOAD).
(17) nt110_drv_unload	Executes the end processing of the NT110 (UNLOAD).

(1) yns_sequent_init

Initializes the JL-080 and internal data of the driver with user parameters.

```
int yns_sequent_init (CHANNEL_INFO *chbuffp,
                      YENET_USER_PAR *usr_par, UCHAR *adrp);
```

chbuffp Leading address of the JL-080 access definition structure

usr_par Communication parameters set by the user
(See *6.4 User Settings* for details.)

adrp Station address corresponding to the station number

- Explanation

1. Initializes the JL-080 access definition structure (*chbuffp*).
2. Initializes the JL-080 to be accessed and enables link communication. Initialization is performed with the communication parameters set in *usr_par* and station address set in *adrp*.

   Use the following method to set the addresses to avoid duplicating the addresses set in *adrp*.
   (ma_max: Max. number of connectable slave stations)

Parameter being Set	Channel Number	Address Setting
adrp[0]	0	FF Hex (synchronous frame)
adrp[1]	1	21H to 3FH (Inverters) 41H to 5FH (Servos) 61H to 7FH (I/Os)
adrp[2]	2	
:	:	
adrp[ma_max]	ma_max	

- Return Value

The value OK (0) will be returned when the JL-080 has correctly completed the communication start request.

If an error occurs, one of the following values will be returned.

ERROR_INIT_CHANNEL	: chbuffp initialization incomplete
ERROR_INVALID_PROTOCOL	: Protocol not supported
ERROR_INVALID_PARAMETER	: Parameter setting error
ERROR_INVALID_BMODE	: Buffer mode not supported
ERROR_NOT_READY JL-080	: Reset failed
ERROR_POLL_TIMEOUT_REG JL-080	: Register settings failed
ERROR_POLL_TIMEOUT_PAR JL-080	: Communication parameter settings failed
ERROR_POLL_TIMEOUT_ENA JL-080	: Communication startup failed

INFO

This function must be used to initialize the JL-080 and the internal data of the driver before establishing link communications. Also, the JL-080 access definition structure (chbuffp) is set by default to be used by other service drivers. If calling other service functions, use the JL-080 access driver set with this function.

(2) yns_begin_link_acc

Performs startup processing for the link communication.

```
int yns_begin_link_acc (CHANNEL_INFO *chbuffp);
```

chbuffp Leading address of the JL-080 access definition structure

- Explanation

Recognizes the JL-080's communication cycle start interrupt and sends a watchdog refresh request to the JL-080. When this function ends with an error, that cycle's write data and read data are not reliable.

- Return Value

The value OK (0) will be returned when the function ends correctly.
A non-zero value will be returned when the function ends with an error. (Refer to *6.6 Error Processing* for details.)

INFO

1. This function must be executed each time before link communication processing.
2. Initialization processing must be completed by the yns_sequent_init() function before starting this function.

(3) yns_link_exchange

Performs data transmission/reception with all slave stations that have been set.

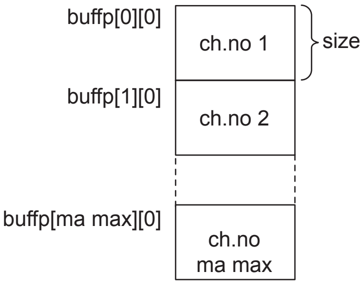
```
int yns_link_exchange (CHANNEL_INFO *chbuffp, USHORT *rbuffp,
                       USHORT *sbuffp, UINT size, USHORT *ests);
```

chbuffp	Leading address of the JL-080 access definition structure
rbuffp	Leading address of the reception buffer
sbuffp	Leading address of the send buffer
size	Number of send/receive words per station
ests	Destination for storage of the JL-080 error status

- Explanation

Sends and receives data for the maximum number of connectable slave stations (ma_max) specified in *yns_sequent_init*. For each station, the amount of receive data specified by the *size* parameter are copied to *rbuffp* and the contents of *sbuffp* are written to the JL-080.

The communication buffer's storage source is normally defined as buffp[ma_max][size]. When this function is called, pass along the buffer's leading address. The following diagram shows the data contents.



- Return Value

The value OK (0) will be returned when the function ends correctly.

A non-zero value will be returned when the function ends with an error. (Refer to *6.6 Error Processing* for details.)

INFO

1. Initialization processing must be completed by the yns_sequent_init() function before starting this function.
2. Do not call this function more than once in a communication cycle.

(4) yns_read_ldata

Reads the specified slave station's link receive data.

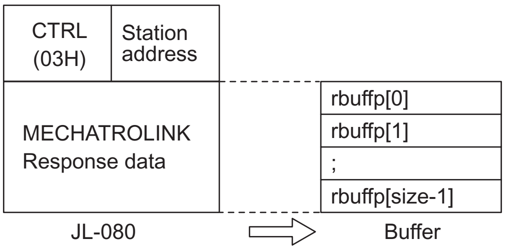
```
int yns_read_ldata (CHANNEL_INFO chbuffp,
                    UINT st_no, USHORT *rbuffp, UINT size);
```

chbuffp	Leading address of the JL-080 access definition structure
st_no	Channel number where receive data will be read
rbuffp	Location to store the receive data that is read
size	Read size (number of words)

• Explanation

Reads data from the station specified by *st_no* in the JL-080 specified by *chbuffp*. The amount of data to be read is specified by *size* and the copy destination is specified by *rbuffp*.

The following diagram shows how the receive data is input in *rbuffp*.



• Return Value

When the function ends correctly, the return value indicates the number of words of data that were received.

When an error occurs, the value will be negative. (Refer to *6.6 Error Processing* for details.)

INFO

1. Initialization processing must be completed by the yns_sequent_init() function before starting this function.
2. Do not call up this function more than once in a communication cycle.

(5) yns_read_ldata2

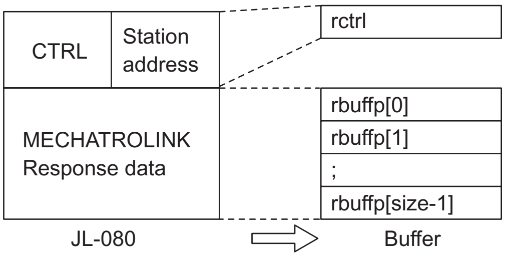
Reads the header data (control field and station address) and channel's link receive data in the specified slave station.

int yns_read_ldata2 (CHANNEL_INFO chbuffp, UINT st_no,
USHORT *rctrl, USHORT *rbuffp, UINT size);

chbuffp	Leading address of the JL-080 access definition structure
st_no	Channel number where receive data will be read
rctrl	Location to store the received header data
rbuffp	Location to store the receive data that is read
size	Read size (number of words)

- Explanation

Reads data from the station specified by *st_no* in the JL-080 specified by *chbuffp*. The amount of data to be read is specified by *size* and the copy destination is specified by *rbuffp*. The control field and station address are stored in rctrl. If the receive error occurs, 00FFH is stored in rctrl.



- Return Value

When the function ends correctly, the return value indicates the number of words of data that were received.

When an error occurs, the value will be negative. (Refer to *6.6 Error Processing* for details.)

INFO

1. Initialization processing must be completed by the yns_sequent_init() function before starting this function.
2. Do not call up this function more than once in a communication cycle.

(6) yns_write_ldata

Writes link send data to the specified slave station.

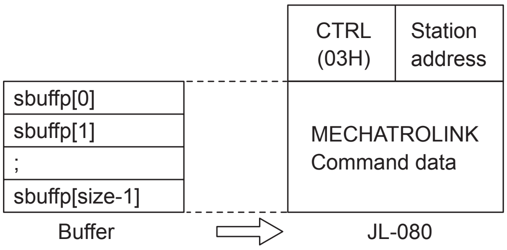
int yns_write_ldata (CHANNEL_INFO chbuffp,
UINT st_no, USHORT *sbuffp, UINT size);

chbuffp	Leading address of the JL-080 access definition structure
st_no	Channel number where send data will be written
sbuffp	Location where send data is stored
size	Write size (number of words)

- Explanation

Writes data to the station specified by *st_no* in the JL-080 specified by *chbuffp*. The amount of data to be written is specified by *size* and the source buffer is specified by *sbuffp*.

The following diagram shows how to set the send data in *sbuffp*.



- Return Value

When the function ends correctly, the return value indicates the number of words of data that were written.

When an error occurs, the value will be negative. (Refer to *6.6 Error Processing* for details.)

INFO Initialization processing must be completed by the yns_sequent_init() function before starting this function.

(7) yns_write_ldata2

Writes the header data (control field and station address) and link send data to the specified slave station.

int yns_write_ldata2 (CHANNEL_INFO chbuffp,
UINT st_no, USHORT sctrl, USHORT *sbuffp, UINT size);

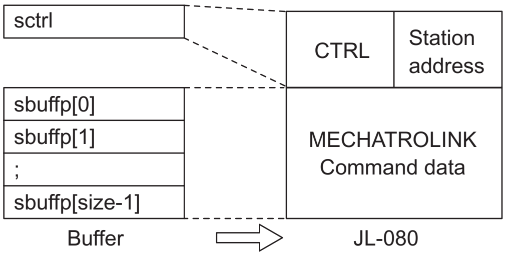
chbuffp	Leading address of the JL-080 access definition structure
st_no	Channel number where send data will be written
sctrl	Send header data
sbuffp	Location where send data is stored
size	Write size (number of words)

- Explanation

Writes data to the station specified by *st_no* in the JL-080 specified by *chbuffp*.

The amount of data to be written is specified by *size* and the source buffer is specified by *sbuffp*. Writes the control field (CTRL) and station address specified by sctrl in the JL-080.

The following diagram shows how to set the send data in *sbuffp*.



- Return Value

When the function ends correctly, the return value indicates the number of words of data that were written.

When an error occurs, the value will be negative. (Refer to *6.6 Error Processing* for details.)

INFO

1. Initialization processing must be completed by the yns_sequent_init() function before starting this function.
2. Before changing the station address, initialize the JL-080 again. If the station address is changed under communication, the change will be invalid.

(8) yns_read_comstat

Checks the status of the JL-080.

USHORT yns_read_comstat (CHANNEL_INFO *chbuffp);

chbuffp	Leading address of the JL-080 access definition structure
err_info	JL-080's error information

- Explanation

Checks the status of the JL-080.
When an error is detected, the cause of the error is indicated in the return value.

- Return Value

The value OK (0) will be returned when the function ends correctly.

A non-zero value will be returned when there is an error. (Refer to *6.6 Error Processing* for details.)

INFO

Initialization processing must be completed by the yns_sequent_init() function before starting this function.

(9) yns_end_link_acc

Performs end processing for link communication.

int yns_end_link_acc (CHANNEL_INFO *chbuffp);

chbuffp	Leading address of the JL-080 access definition structure

- Explanation

Executes a link data send request.

- Return Value

The value OK (0) will be returned when the function ends correctly.

A non-zero value will be returned when there is an error. (Refer to *6.6 Error Processing* for details.)

INFO

1. Always execute this function every cycle after executing link communication.
2. Initialization processing must be completed by the yns_sequent_init() function before starting this function.

(10) yns_set_netclk

Refreshes the network time.

USHORT yns_set_netclk (CHANNEL_INFO *chbuffp, UINT cycle);

chbuffp	Leading address of the JL-080 access definition structure
cycle	Communication cycle (multiple of transmission cycle) Set the master station's link communication cycle as a multiple of the transmission cycle.

- Explanation

Increases the network time in the JL-080 specified in *chbuffp* by the amount specified by *cycle*, and writes the network time in the global data.

- Return Value

Returns the network time that was written.

INFO

1. Always execute this function every cycle when using a C2 master station.
2. Initialization processing must be completed by the yns_sequent_init() function before starting this function.

(11) yns_check_ram

Performs a read/write check on the RAM area used by the JL-080.

int yns_check_ram (USHORT slot, UINT ofst_adr, UINT size, USHORT chk_data);

slot	Slot number to check RAM
ofst_adr	Starting address for RAM check (0 to 2,048 when using built-in RAM, in words)
size	Amount of RAM to check (words)
chk_data	Data to write to RAM

- Explanation

Writes the amount of *chk_data* specified by *size* to the RAM area starting at the address specified by *ofst_adr* in the NT110 specified in *slot*. The original *chk_data* is compared to the data read from RAM to check whether the data was written and read from the RAM correctly.

- Return Value

The value OK (0) will be returned when the function ends correctly.

An ERROR _RAM_CHECK (-11) will be returned when a verification error occurred.

(12) yns_wait_interrupt

Device waits for an interruption from the NT110 board.

int yns_wait_interrupt (USHORT slot, USHORT type, DWORD timeout);

slot	Select the slot number of the NT110 board which will wait for interruption.
type	Type of interruption (Set YN_INTS_TIM1.)
timeout	Timeout of the wait for interruption (0x00 to 0xFFFFFFFF)

- Explanation

Waits for an interruption from the NT110 board at specified slot.

(Global interruption for every communication cycle)

- Return Value

The value OK (0) will be returned when the function ends correctly.
An ERROR _WAIT_TIMEOUT(-23) will be returned when an interrupt timeout error occurred.

(13) yns_reset_board

Used to carry out a soft reset of the NT110 board.

int yns_reset_board (USHORT slot,);

slot	Select the slot number of the NT110 board which is to be reset.

- Explanation

Resets the NT110 board at specified slot.

Before resetting, disconnect the connected slave.

- Return Value

The value OK (0) will be returned when the function ends correctly.
An ERROR _BOARD_RESET_FAIL(-13) will be returned when a board reset error occurred.

(14) yns_rtx_set_priority

Used to set the priority value of the interruption handler in the RTX.

This API function cannot be used with Windows2000 or XP. This is only for the RTX API.

ins yns_rtx_set_priority (USHORT priority,);

priority	Select the priority value (0 to127). A large value has greater priority than a small value.

- Explanation

Used to set the priority value of interruption handler in the RTX.

- Return Value

The value OK (0) will be returned when the function ends correctly.

An ERROR _PRIORITY(-14) will be returned when a priority setting error occurred.

(15) yns_rtx_set_interval

Used to set the polling cycle to detect Interrupt signal from the NT110 board in the RTX.

This API function cannot be used with Windows2000 or XP. This is only for the RTX API.

int yns_rtx_set_interval (USHORT interval,);

interval	Select the interval value. (Units: 1= 0.1 ms)

- Explanation

Used to set the polling cycle to detect the Interrupt signal from the NT110 in the RTX.

Calling is not required when the Interrupt signal is used in the RTX.

- Return Value

The value OK (0) will be returned when the function ends correctly.
An ERROR _INTRVL(-15) will be returned when a interval setting error occurred.

(16) nt110_drv_load

(a) API for RTX

Used to initialize the NT110 driver.

ULONG nt110_drv_load();

- Explanation

Initializes the NT110 driver.

- Return Value

The value OK (0) will be returned when the function ends correctly.

An ERROR_DRV_LOAD (-16) will be returned when a driver initialization error occurred.

(b) API for Windows2000 and XP

Used to initialize the NT110 driver.

ULONG nt110_drv_load (USHORT slot,);

slot	Select the slot number, 1 or 2, of the NT110 driver to be initialized.

- Explanation

Initializes the NT110 driver of the selected slot number.

- Return Value

The value OK (0) will be returned when the function ends correctly.

An ERROR_DRV_LOAD(-16) will be returned when a driver initialization error occurred.

(17) nt110_drv_unload

(a) API for RTX

Used to end the NT110 driver.

ULONG nt110_drv_unload( );

- Explanation

Ends the NT110 driver.

- Return Value

The value OK (0) will be returned when the function ends correctly.

An ERROR_DRV_UNLOAD(-17) will be returned when a driver end error occurred.

(b) API for Windows2000 and XP

Used to end the NT110 driver.

ULONG nt110_drv_unload (USHORT slot,);

slot	Select the slot number, 1 or 2, of the NT110 driver to end.

- Explanation

Ends the NT110 driver of the selected slot number.

- Return Value

The value OK (0) will be returned when the function ends correctly.

An ERROR_DRV_UNLOAD(-17) will be returned when a driver end error occurred.

6.8 Data Definition

struct CHANNEL_INFO

Type	Member Name	Description
YENET_HOST_IF_REGS*	mod	Leading address of JL-080 interface register
unsigned short	slot	Connection slot (1: fixed)
unsigned short	ma	C1 master station address
unsigned short	ma_max	Max. number of connectable slave stations
unsigned short	link_sz	Number of JL-080 internal link words
unsigned short	buff_mod	Buffer mode
unsigned short	alt_buf	Currently-used buffer
unsigned short	ubp-data	UBP registered data
unsigned short	link_sadr[2]	link buffer offset address
unsigned short	link_radr[2]	link buffer offset address
unsigned short	pars_data	Leading address of communication parameter area
unsigned short	c1msg_sz	C1 message size
unsigned short	c2msg_sz	C2 message size
unsigned short	c1msg_adr	Leading address of C1 message buffer
unsigned short	c2msg_adr	Leading address of C2 message buffer
unsigned short	msgpkt_sz	Size of data packet in each message transmission operation
unsigned short	check_code	Check code for buffer checking
unsigned short	net_clk	Network time

The information required to access the JL-080 is set in the CHANNEL_INFO structure, so do not overwrite the variables directly. When two or more JL-080 LSIs are used with a single CPU, each JL-080 needs its own CHANNEL_INFO structure.

struct YENET_USER_PAR

Type	Member Name	Description
unsigned short	slot	Connection slot (1: fixed)
unsigned short	mod	Communication mode setting
unsigned short	ma	C1 master station address
unsigned short	ma_max	Max. number of connectable slave stations
unsigned short	t_mcyc	Transmission cycle (0.25 µs)
unsigned short	t_cyc	Time slot width (0.25 µs)
unsigned short	byte	Number of transmission bytes
unsigned short	dev	INT1 dividing ratio (for INT2 output)
unsigned short	max_rtry	Max. number of retries
unsigned short	c2m_ch	Time slot for C2 master station
unsigned short	wdt	WDT function setting for host monitoring

Use within the yns_sequent_init() function.

Refer to *6.4 User Settings* for more details.

6.9 Test Program

6.9.1 Test Program Files

(1) RTX Driver

The contents of the files in the folder named *Sample_rtx_1thread* are listed in the following table.

File Name	Contents
\Release\NT110_SAMPLE_RTX.rtss	Executable Test Program
readme_EN.txt	MECHATROLINK Test Program Application Procedures (Sample)

To start the Test Program, execute the NT110_SAMPLE_RTX.rtss.

(2) Windows2000/XP Driver

The contents of the files in the folder named *NT110_SAMPLE_WIN* are listed in the following table.

File Name	Contents
\Release\NT110_SAMPLE_WIN.exe	Executable Test Program
readme_EN.txt	MECHATROLINK Test Program Application Procedures (Sample)

To start the Test Program, execute the NT110_SAMPLE_WIN.exe.

6.9.2 Using the Test Program

The Test Program named *NT110_SAMPLE_RTX.rtss* operates in a RTX5.1.1 environment and the test program named *NT110_SAMPLE_WIN.exe* operates in a Windows2000/XP environment. They are the test software for sending MECHATROLINK-II commands. These executable Test Programs can be included in a PC in which the NT110 is mounted so that the PC can operate as a master.

The file called *readme_EN.txt* contains the following manual, which provides specific application methods for using these Test Programs. Always read this manual before attempting to use the Test Program.

- MECHATROLINK Test Program Application Procedures (Sample)

6.10 Software License Agreement

The contents of JAPMC-NT110 Access API Software License Agreement (NT110 LicenseAgreement_Eng.pdf) are as follows in the NT110 Access API.

Name of the software: JAPMC-NT110 Access API

Software License Agreement

Yaskawa Electric Corporation (hereinafter referred to as "YEC") shall license the non-transferable and non-exclusive right to use this software provided with this Agreement to the customers (hereinafter referred to as "Recipient") who bought this product, under the condition that Recipient consents the following terms and conditions.

Accordingly, it shall be deemed to conclude this Agreement when Recipient opens this DISK PACKAGE. YEC should like to request Recipient to keep this Agreement.

Article 1 DEFINITION

The following terms have the following meanings respectively in this Agreement.

1. SOFTWARE means the computer program in the DISK PACKAGE and the amended computer program YEC may provide in the future according to sub-article 3 of Article 4.

   REFERENCE INFORMATION means all of the printed documentation in the DISK PACKAGE or the printed documentation YEC may provide in the future.

   PRODUCT means the software, reference documentation and duplicates of software in all YEC provided.

Article 2 COPYRIGHT

The right of possession and the copyright shall be reserved to YEC and shall be protected by the law of copyright and international treaty. However, Recipient shall be able to perform each of the following actions.

The action to duplicate only one copy of the software as the back-up

The action to edit or modify the software for purpose of trying compile this software in order to make run time object under the Recipient development environment

The action to install the object which is compiled under the Recipient development environment, to the computer with JAPMC-NT110 (No limit of number)

Receiver shall not be able to use, manufacture, delivery or publication etc. using this software in case of no description in this Agreement.
All of the right except being licensed clearly shall be reserved to YEC.

Article 3 PROHIBITION

Recipient shall consent to prohibit the following each sub-article.

Duplicating the software except the above article

Reengineer this software

Assigning or lending or sub-licensing this software to the third party

Renting or leasing or sub-licensing or assigning this software to the third party

Transferring the status in this Agreement

Removing or unclearing the indications such as YEC’s copyright or trade mark

Article 4 SCOPE OF WARRANTY

YEC shall exchange the software free of charge in the limitation within 90 days (including 90) from the day Recipient bought it, if there are physical defects in the material or relevant documentation.

YEC shall not warrant the quality or the function of this software to fit the Recipient’s purpose.

Receiver shall have the responsibility for selecting, introducing, using and its result.

YEC shall provide such amended computer program or relevant information in the limitation within 1 year from the day Recipient bought it, if YEC amended the software.

YEC, however, shall decide the necessity or the time to provide amended computer program or relevant information.

YEC shall not warrant the quality of the compiled object under Recipient development environment. And YEC shall not warrant the behavior of the JAPMC-NT110 with this compiled object. This is because it has the possibility that a difference occurs in the object after it is compiled by the Recipient development environment (compiler, compilation option, optimization, etc.).

Article 5 DURATION

This Agreement shall be into force at the time Recipient opens the DISK PACKAGE.

In the event that Recipient breaches each of the provision in this Agreement or infringes YEC’s copy right, YEC shall have the right to cancel this Agreement and to terminate the use of the software by Recipient at the time when the notice arrives to Recipient from YEC.

In the event that Recipient decides to stop the use of this software, this Agreement shall automatically be terminated.

In the event that this Agreement is canceled or terminated, Recipient shall consent to destruct this software.

Article 6 NON-DISCLOSURE OF TRADE SECRET

Recipient and YEC each other shall keep to be confident of the trade secret known in the process of concluding or carrying this Agreement to the third party.

Article 7 GENERAL PROVISIONS

This Agreement constitutes the complete and exclusive agreement between YEC and Recipient with respect to the subject matter hereof, and supersedes all prior oral or written understandings, communications or agreements not specifically incorporated herein.

Recipient and YEC shall submit to settle the exclusive jurisdiction of the Tokyo.

District Court of Japan with respect to any dispute arising from this Agreement.

Revision History

The revision dates and numbers of the revised manuals are given on the bottom of the back cover.

MANUAL NO. SIEP C880700 17A
© Printed in Japan June 2004 04-6

Date of printing
Date of original publication

Date of Printing	Rev. No.	Section	Revised Contents
June 2004	−		First edition
July 2004	◇1	Chapter 6	Addition: Description of Windows2000/XP
		6.2.4	Revision: Description of using drivers
		6.2.5	Addition: (4) NT110 Insertion Limitations
		6.3.1	Revision: Communication initialization procedure
		6.6.3	Addition: Driver setting error
		6.7	Addition: List of API references, (16) nt110_drv_load, and (17) nt110_drv_unload
		Back cover	Revision: Address

PCI-compliant MECHATROLINK-Ⅱ Interface Card

USER'S MANUAL

IRUMA BUSINESS CENTER
480, Kamifujisawa, Iruma, Saitama 358-8555, Japan
Phone 81-4-2962-5696 Fax 81-4-2962-6138

YASKAWA ELECTRIC AMERICA, INC.
2121 Norman Drive South, Waukegan, IL 60085, U.S.A.
Phone 1-847-887-7000 Fax 1-847-887-7370

MOTOMAN INC. HEADQUARTERS
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-937-847-6200 Fax 1-937-847-6277

YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTD.A.
Avenida Fagundes Filho, 620 Bairro Saude-Sao Päulo-SP, Brazil CEP: 04304-000
Phone 55-11-5071-2552 Fax 55-11-5581-8795

YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-569-312

Motoman Robotics Europe AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-48800 Fax 46-486-41410

Motoman Robotec GmbH
Kammerfeldstraβe1, 85391 Allershausen, Germany
Phone 49-8166-90-100 Fax 49-8166-90-103

YASKAWA ELECTRIC UK LTD.
1 Hunt Hill Orchardton Woods Cumbernauld, G68 9LF, United Kingdom
Phone 44-1236-735000 Fax 44-1236-458182

YASKAWA ELECTRIC KOREA CORPORATION
Kfpa Bldg #1201, 35-4 Youido-dong, Yeongdungpo-Ku, Seoul 150-010, Korea
Phone 82-2-784-7844 Fax 82-2-784-8495

YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-6282-3003 Fax 65-6289-3003

YASKAWA ELECTRIC (SHANGHAI) CO., LTD.
No.18 Xizang Zhong Road. Room 1805, Harbour Ring Plaza Shanghai 20000, China
Phone 86-21-5385-2200 Fax 86-21-5385-3299

YATEC ENGINEERING CORPORATION
4F., No.49 Wu Kong 6 Rd, Wu-Ku Industrial Park, Taipei, Taiwan
Phone 886-2-2298-3676 Fax 886-2-2298-3677

YASKAWA ELECTRIC (HK) COMPANY LIMITED
Rm. 2909-10, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2803-2385 Fax 852-2547-5773

BEIJING OFFICE
Room No. 301 Office Building of Beijing International Club, 21
Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-6532-1850 Fax 86-10-6532-1851

TAIPEI OFFICE
9F, 16, Nanking E. Rd., Sec. 3, Taipei, Taiwan
Phone 886-2-2502-5003 Fax 886-2-2505-1280

SHANGHAI YASKAWA-TONGJI M & E CO., LTD.
27 Hui He Road Shanghai China 200437
Phone 86-21-6553-6060 Fax 86-21-5588-1190

BEIJING YASKAWA BEIKE AUTOMATION ENGINEERING CO., LTD.
30 Xue Yuan Road, Haidian, Beijing P.R. China Post Code: 100083
Phone 86-10-6233-2782 Fax 86-10-6232-1536

SHOUGANG MOTOMAN ROBOT CO., LTD.
7, Yongchang-North Street, Beijing Economic Technological Investment & Development Area,
Beijing 100076, P.R. China
Phone 86-10-6788-0551 Fax 86-10-6788-2878

YASKAWA

YASKAWA ELECTRIC CORPORATION

In the event that the end user of this product is to be the military and said product is to be employed in any weapons systems or the manufacture thereof, the export will fall under the relevant regulations as stipulated in the Foreign Exchange and Foreign Trade Regulations. Therefore, be sure to follow all procedures and submit all relevant documentation according to any and all rules, regulations and laws that may apply.

Specifications are subject to change without notice
for ongoing product modifications and improvements.

MANUAL NO. SIEP C880700 17B

Printed in Japan July 2004 04-6 ◇1
04-1⑤