

PLC Serial Communication

This course is for participants who will use a MELSEC-Q series serial communication module for the first time.

Introduction Purpose of the Course

This course explains the basics of a serial communication module that is compatible with the MELSEC-Q series programmable controller, and is designed for those who will use the module for the first time.

By taking this course, a participant will understand the data communication mechanism, specifications, settings and the start-up method of the serial communication module.

This course requires the basic knowledge of MELSEC-Q series programmable controllers, sequence programs, and GX Works2.

Taking the following courses is recommended before starting this course.

1. MELSEC-Q Series Basics Course
2. GX Works2 Basics Course
3. Intelligent Function Module Course

Introduction Course Structure

The contents of this course are as follows.
We recommend that you start from Chapter 1.

Chapter 1 - Serial Communication Basics

Explains the serial communication basics.

Chapter 2 - Serial Communication Module Details

Explains the serial communication module types, component names and functions of a module, and connection methods.

Chapter 3 – Initial Configuration

Explains how to setup a serial communication module and how to program it using dedicated instructions.

Chapter 4 - Troubleshooting





Explains the network diagnostic procedure for troubleshooting.

Final Test

Passing grade: 60% and higher

Introduction How to Use This e-Learning Tool



Go to the next page		Go to the next page.
Back to the previous page		Back to the previous page.
Move to the desired page		"Table of Contents" will be displayed, enabling you to navigate to the desired page.
Exit the learning		Exit the learning. Window such as "Contents" screen and the learning will be closed.

Introduction Cautions for Use



Safety precautions

When you learn by using actual products, please carefully read the safety precautions in the corresponding manuals.

Precautions in this course

- The displayed screens of the software version that you use may differ from those in this course.

This course uses the following software version:

- GX Works2 Version 1.493P

Chapter 1 Serial Communication Basics

Chapter 1 describes the serial communication module basics.

In Chapter 1, you will understand how a serial communication module is used, its main functions, and its data communication method.

1.1 Communication Parameters

1.2 Communication Protocols

1.3 Flow Control

1.4 Interface Types

1.5 Data Division

1.6 Summary

Basic knowledge of serial communication

Serial communication is a mature technology that has been used for many years. It is still popular today as a data communication method for devices such as a measuring instrument and bar code reader. One reason of the popularity is their inexpensive parts.

This course features RS-232, a representative interface for serial communication.

In serial communication with a serial communication module, various device types can be connected comparatively freely. However, the communication specifications of the connected device (3rd party device) must be fully understood to establish normal communication.

Communication specifications are roughly classified into the following:

- **Communication parameters**
- **Communication protocol**
- **Flow control**

Both of the communicating devices must satisfy the communication specifications at the design stage.

Below are communication parameters that are important to serial communication:

Number of data bits

An alphanumeric character is expressed in 7 bits. Therefore, when sending only a numeric or alphabetical character, data size can be reduced by selecting 7 bits.

Parity bit

This needs to be set to detect data corruption caused by noise, etc.

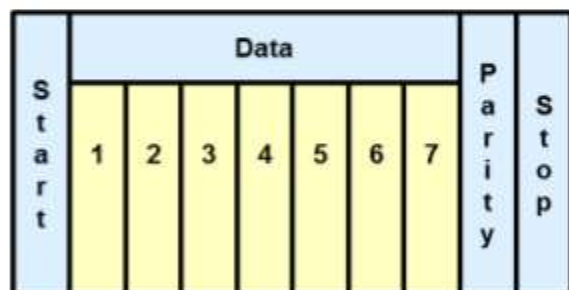
Stop bit

This bit indicates the end of data.

Bit rate

The bit rate is the number of bits sent per second. This is also called the transmission speed.

A higher bit rate means a shorter transmission time. Adjust the bit rate when the communication is affected by noise, etc.



All of the above parameters must be set the same at both of the communicating devices. The parameters of many devices are non-changeable. Therefore, check the specifications of the 3rd party device and adjust the serial communication modules communication parameters.

A communication protocol is a set of conventions adopted by the devices connected to a network.

Examples of communication protocols (rules) include:

- When data has been received normally, a specific code is returned to report a normal reception.
- When an error has occurred, an error code is sent to report the error occurrence.

Since these communication protocols are determined by the connected 3rd party device, the specifications of the device must be checked.

To set a communication protocol for a serial communication module, the user can use the "predefined protocol support function" of GX Works2 (details are given in later chapters), and simply select the communication protocol from the existing protocol options.

New protocols can also be added if the desired protocol is not found. Doing so allows data to be sent or received automatically via compatible 3rd party devices, without using sequence programs.

Flow control is a procedure that ensures the data receiving side receives all the transmitted data. Flow control is roughly classified into two types: hardware flow control and software flow control.

Hardware flow control

Adjusts data send timing by using a flow control line, which is installed separately from the signal line, in the same cable. Using the flow control line, data receive information is returned to the source.

The serial communication module uses DTR/DSR hardware flow control. Connection with an RTS/CTS control device is possible but such connections must be carefully designed.

Software flow control

Adjusts data send timing by using specific codes. When using this method, data receive information is returned to the source.

The Xon/Xoff control, which is a representative software flow control type, is the same as the DC1/DC3 control, which is an option selectable at GX Works2.

Some devices do not support flow control. In such cases, the serial communication module should perform operations such as:

- Adjust the send interval.
- Detect when the receiving side fails to receive data, and if that happens, discard the unreceived data.

1.4

Interface Types

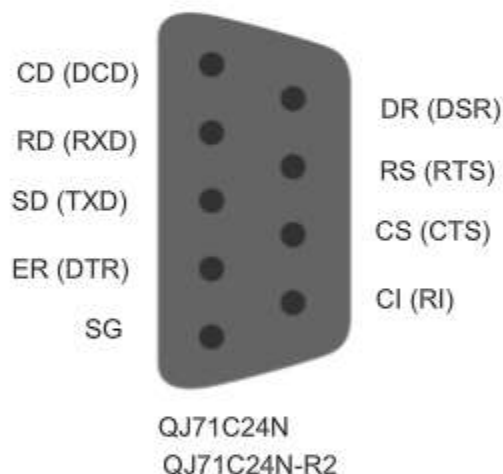
RS-232

The RS-232 interface is often connected via a D-sub connector. A function is assigned to each contact pin according to the RS-232 standard.

Note that the RS-232 compatible serial port of a personal computer, etc. is a male port with protruding pins, but the RS-232 port of a programmable controller is a female port.

A signal cable consists of a communication line and a control line. Which of the two lines is used depends on the communication specifications of the 3rd party device.

If the desired wiring is not commercially available, the connector must be configured to accept such wiring.

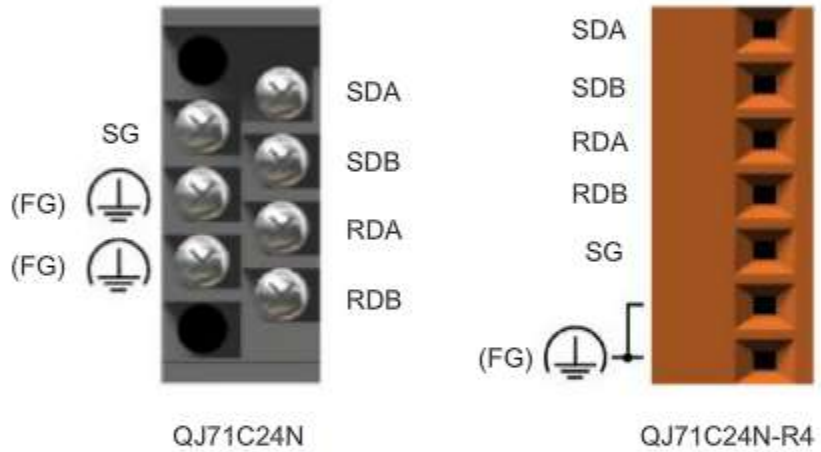


Pin number	Signal code	Signal function	Signal direction Module <=> 3rd party device
1	CD (DCD)	Detection of data channel-receiving carrier	←
2	RD (RXD)	Received data	←
3	SD (TXD)	Sent data	→
4	ER (DTR)	Data terminal ready	→
5	SG	Signal ground	↔
6	DR (DSR)	Data set ready	←
7	RS (RTS)	Request to send	→
8	CS (CTS)	Clear to send	←
9	CI (RI)	Ring indicator	←

1.4 Interface Types

RS-422 and RS-485

When these interfaces are used, devices communicate by differential signals. For differential signals, a pair of signal lines is used for one signal. Differential signals are comparatively resistant to noise and suitable for long-distance transmission. As no control line is used, software flow control is used when flow control is required. The RS-422 interface uses one signal line for sending data and another for receiving. The RS-485 interface uses one signal line for both transmission and reception.



Signal code	Signal name	Signal direction Module <=> 3rd party device
SDA	Sent data (+)	→
SDB	Sent data (-)	→
RDA	Received data (+)	←
RDB	Received data (-)	←
SG	Signal ground	↔
FG	Frame ground	↔
FG	Frame ground	↔

This course explains the highly versatile RS-232 interface.

1.5

Data Division

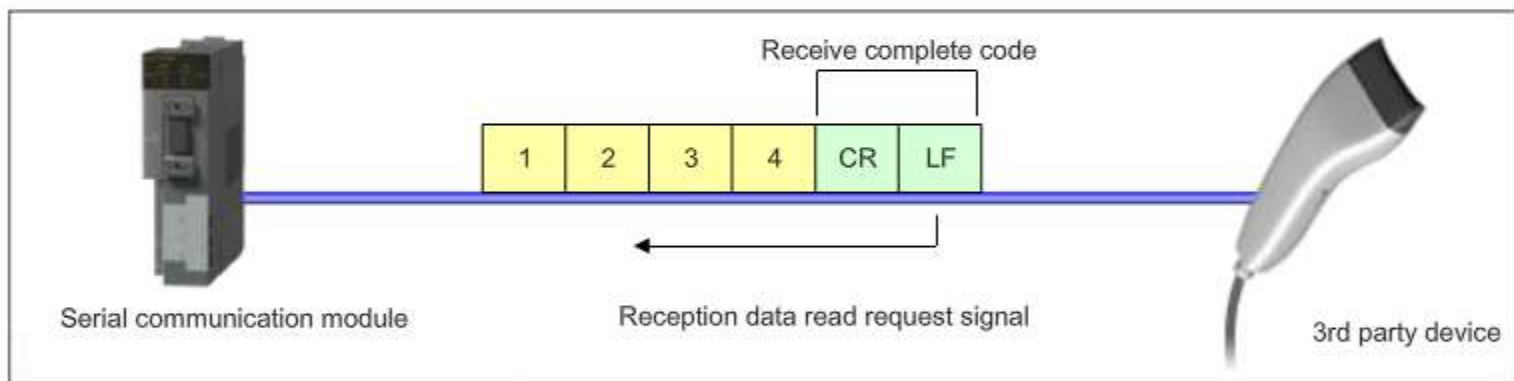
When data is received, it is usually divided into parts of a certain length.

There are two data division methods: division by the number of data and division by a receive complete code. Each method depends on the communication specifications of the 3rd party device, therefore make sure to confirm the specifications.

If necessary, a receive complete code and the received data count can be changed from their default settings.

Receiving variable-length data using a receive complete code

This method is used to receive data with varying lengths from a 3rd party device. Before data is sent from a 3rd party device, a receive complete code (CR+LF or one-byte data), which is specified by the serial communication module, is added to the end of the message.



The example system in this course receives data using a receive complete code.

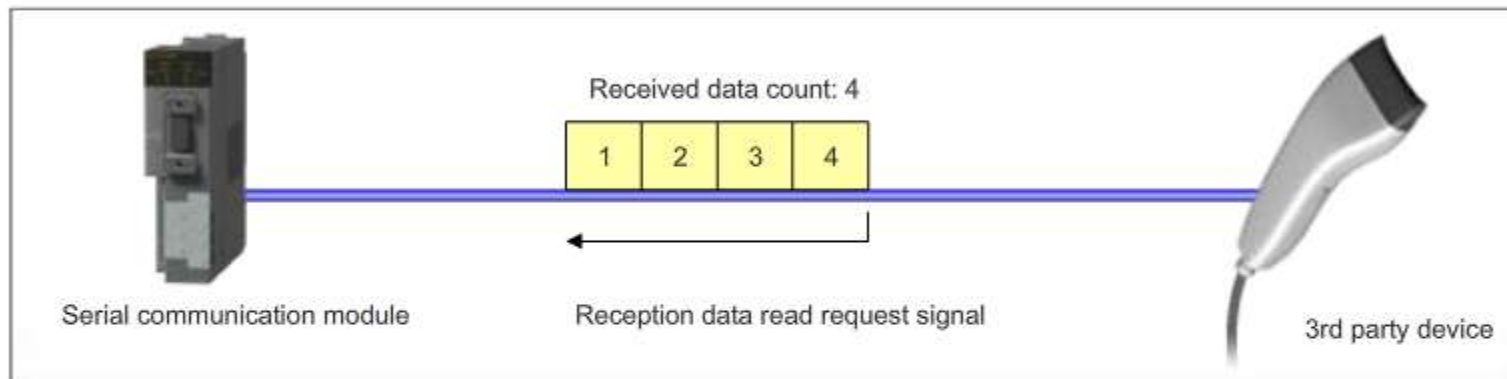
1.5

Data Division

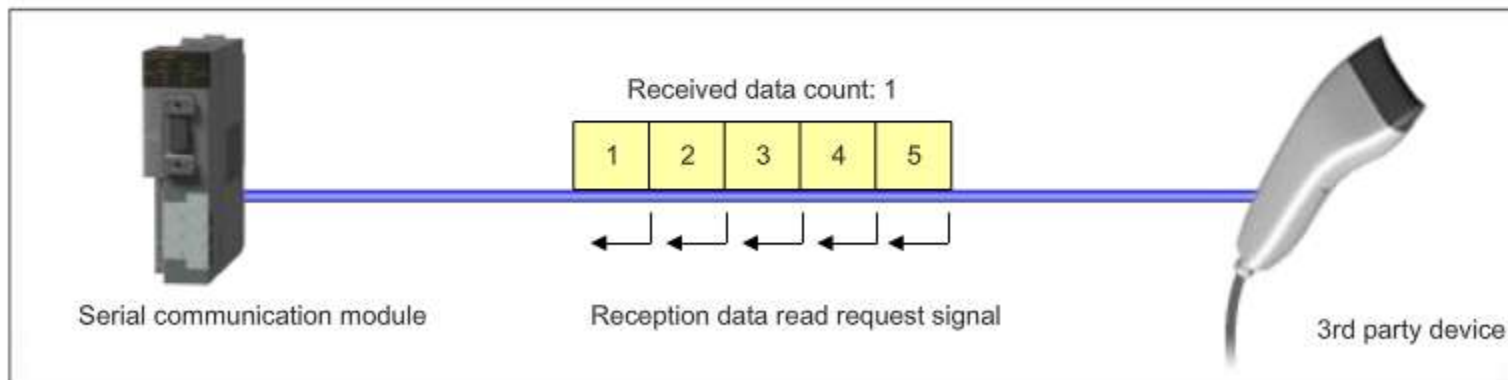
Receiving fixed-length data using the received data count

This method is used to receive data with a fixed length. Since the data length is fixed by the 3rd party device, a receive complete code is unnecessary.

The 3rd party device sends the data amount that is specified by the received data count setting of the serial communication module.

**Advanced technique: receiving variable-length data with no receive complete code**

If a receive complete code is not added to the data with varying lengths sent from the 3rd party device, the data is received and processed byte by byte.



In this chapter, you have learned:

- Communication parameters
- Communication protocols
- Flow control
- Interface types
- Data division

Important points

Communication parameters	Important parameters in serial communication are the number of data bits, parity bit, stop bit, and bit rate.
Fixed length and variable length	Communication protocols handle two types of data: fixed-length data and variable-length data.
Flow control	Flow control is roughly classified into two types: hardware flow control and software flow control.
Interface type	Interfaces of a serial communication module are RS-232, RS-422, and RS-485.
Data division	The received data is divided by the receive data count or a receive complete code.

Chapter 2 Serial Communication Module Details

Chapter 2 describes the serial communication module types, component names and functionality of a module, and the connection methods.

2.1 Serial Communication Module Types

2.2 Communication Cable Connection

2.3 Serial Communication Module Communication Protocols

2.4 Serial Communication Module Configuration

2.5 Summary

2.1

Serial Communication Module Types

This section describes the serial communication module types, the component names of a module, and its LED indicators.

Serial communication module

A serial communication module is an intelligent function module. A serial communication module connects an external device such as a measuring instrument and bar code reader, to a Q-series CPU module through its RS-232 or RS-422/485 interface, which are typical serial communication interfaces, to enable data communication between the connected devices.

Each module provides two communication channels that can be simultaneously used.

Three module types, with different combinations of interfaces, are available.

QJ71C24N



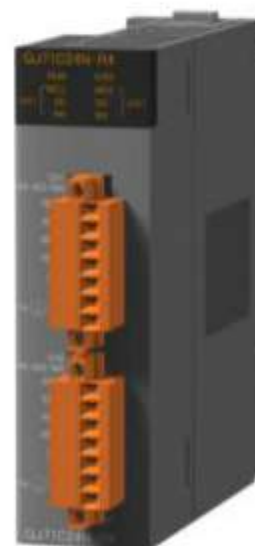
RS-232: 1 channel
RS-422/485: 1 channel

QJ71C24N-R2



RS-232: 2 channels

QJ71C24N-R4



RS-422/485: 2 channels

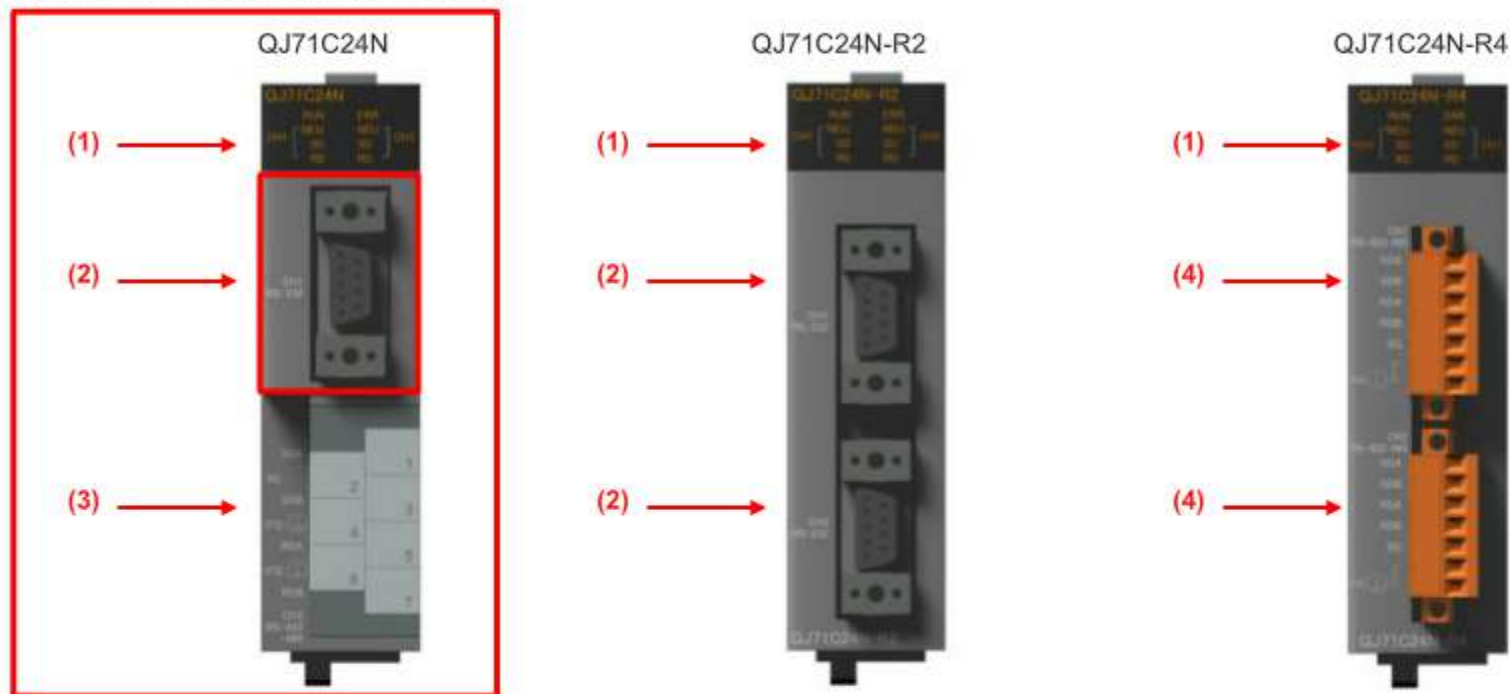
This course uses the QJ71C24N single channel RS-232 interface as an example.

2.1.1

Serial communication module components

This section describes the serial communication module components and their functionality.

Component names and functionality



No.	Name	Function
(1)	LED indicators	Please refer to the list of LED indicators on the next page.
(2)	RS-232 interface	For serial communication with a 3rd party device (D-sub, 9-pin, female connector)
(3)	RS-422/485 interface	For serial communication with a 3rd party device (2-piece terminal block*)
(4)	RS-422/485 interface	For serial communication with a 3rd party device (2-piece plug-in connector socket block*)

* The 2-piece terminal block and the 2-piece plug-in connector socket block can be removed by loosening their screws.
Each terminal block can be replaced on the module easily without removing the wires in the case of a module breakdown.

2.1.2

LED indicators and their functions

This section describes the functionality of the LED indicators that are on a serial communication module.

LED indicators



CH	LED indicator name	Function	On or flashing	Off	Corresponding protocol			
					MC	Non procedure	Bi-directional	Predefined
-	RUN	Indicates normal operation	Normal	Abnormal, reset	Valid	Valid	Valid	Valid
	ERR	Indicates an error *1	Error	Normal				
CH1/2	NEU	Indicates neutral status *2	Waiting to receive an MC command	Receiving an MC command	Valid	Invalid (off)	Invalid (off)	Invalid (off)
	SD	Indicates send status	Sending data	Not sending data	Valid	Valid	Valid	Valid
	RD	Indicates receive status	Receiving data	Not receiving data				

*1 This indicator turns on when an error occurs in the hardware or data communication of a serial communication module.

*2 This indicator indicates status of data communication by the MC protocol.

On: Waiting to receive a command from the 3rd party device

Off: A command from the 3rd party device is being received or processed.

2.2

Communication Cable Connection

This section shows connection examples with serial communication modules.

2.2.1

Connecting the RS-232 interface to a device

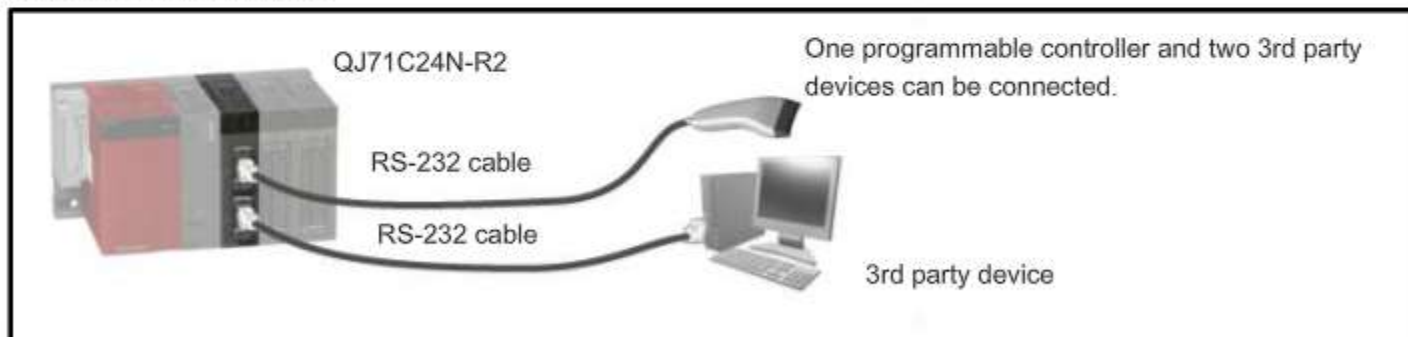
Below are the connection examples for the RS-232 interface, its 3rd party device and the QJ71C24N and QJ71C24N-R2.

Connection example

When QJ71C24N is used



When QJ71C24N-R2 is used



2.2.2

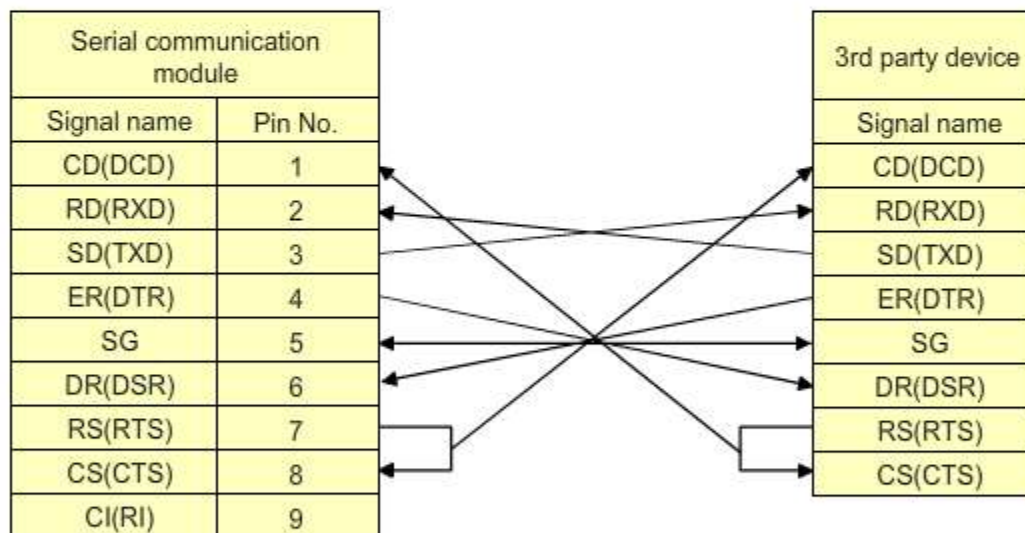
Wiring for the RS-232 control signals

Click the buttons below to visualize the corresponding wiring examples.

The 3rd party device turns on/off the CD signal.
DTR/DSR control and DC code control are supported.

The 3rd party device does not turn on/off the CD signal.
DTR/DSR control and DC code control are supported.

The 3rd party device does not turn on/off the CD signal.
DC code control is supported.



- The flow control method of the 3rd party device is employed by the both devices.
- If the 3rd party device has a wiring example for the Mitsubishi serial communication module, follow that example.

2.3

Serial Communication Module Communication Protocols

Below are the communication protocols available to a serial communication module.

Protocol	Details	Control direction
Non procedure protocol	Any data can be exchanged between a 3rd party device and a CPU module in any message format and by any transmission procedure. A message can also be created flexibly according to the specifications of the 3rd party device. Select this protocol when data communication needs to be established according to the protocol of the 3rd party device, such as a measuring instrument or a bar code reader.	From the programmable controller to the 3rd party device (Active)
Predefined protocol	Data communication based on the 3rd party device's protocol is established using the "predefined protocol function". To set a protocol, select a predefined protocol from the communication protocol library, or create a new one, or edit an existing protocol. The selected protocol is written on the flash ROM of the serial communication module and executed by "dedicated instruction (CPRTCL)". Details of the predefined protocol support function are given in Chapter 3.	
MC protocol	MC protocol is the communication method for programmable controllers. With this method, a 3rd party device reads or writes the device data and programs of a CPU module via a serial communication module. If a 3rd party device can send or receive data by the MC protocol, it can access a CPU module.	From the 3rd party device to the programmable controller (Passive)
Bi-directional protocol	This simple predefined protocol allows external devices such as personal computers, to send and receive data comparatively easily. A programmable controller uses dedicated instructions (BIDIN, BIDOUT) to respond to the external device.	

Active: A programmable controller gives instructions to its 3rd party device and receives a response.

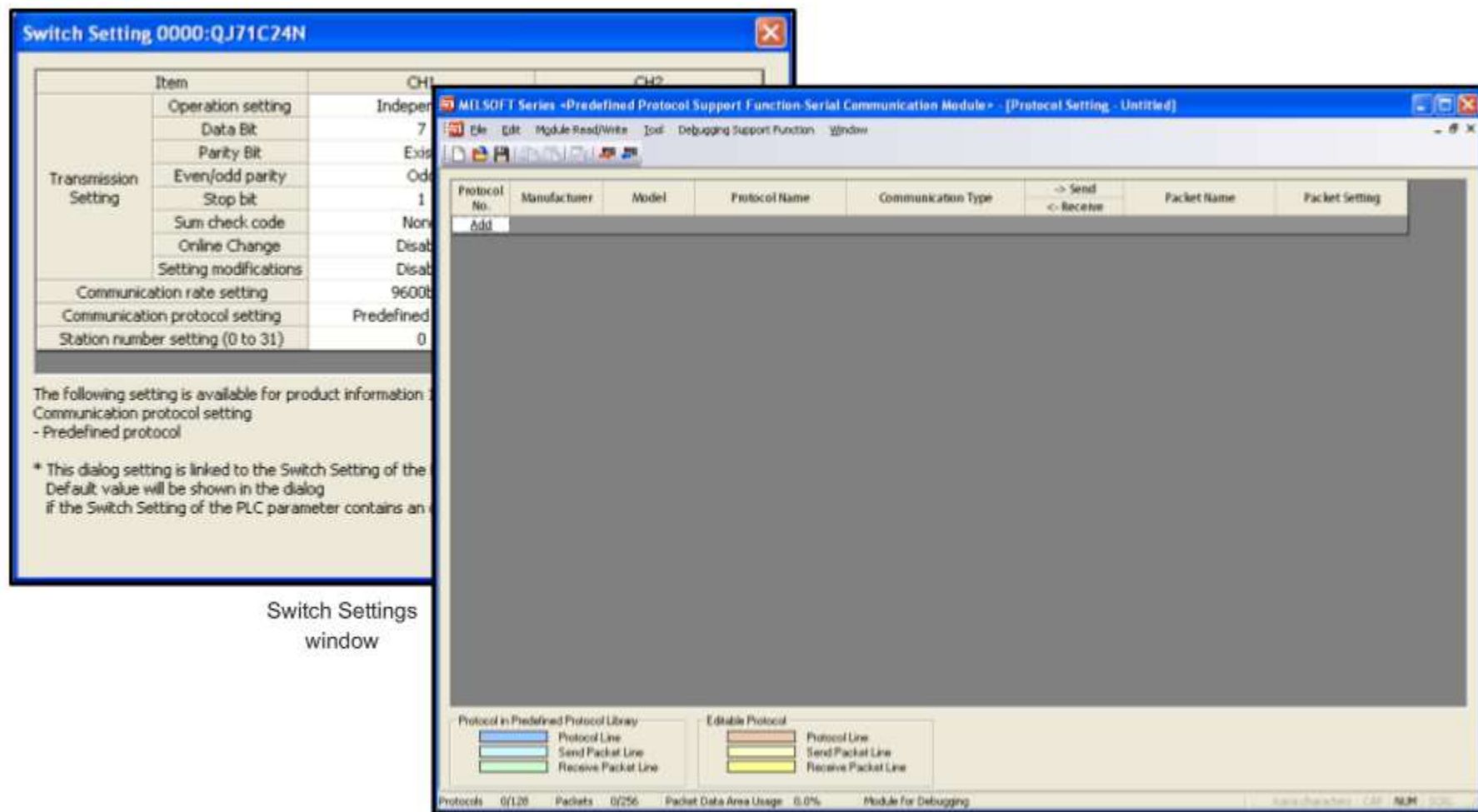
Passive: A programmable controller receives instructions from the 3rd party device and returns the value and status saved in its devices as responses.

The example system in this course uses "predefined protocol".

2.4

Serial Communication Module Configuration

GX Works2 is useful in configuring initial settings and registering predefined protocols (predefined protocol support function) to the serial communication modules. Please refer to Chapter 3 for details.



Switch Settings
window

Predefined Protocol Support
Function window

In this chapter, you have learned:

- Serial communication modules types
- Communication cable connection
- Serial communication module communication protocols
- Serial communication module configuration

Important points

Data communication protocols	The data communication protocols available to a serial communication module are: non procedure protocol, bi-directional protocol, MC protocol, and predefined protocol.
Predefined protocol	The "predefined protocol support function" creates a predefined protocol based on the 3rd party device's protocol
Connection method	<ul style="list-style-type: none">• QJ71C24N can be connected to a 3rd party device via an RS-232 or RS422/485 interface.• QJ71C24N-R2 can be connected to two 3rd party devices via an RS-232 interface.

Chapter 3 Initial Configuration

Chapter 3 describes how to setup a serial communication module for its initial operation. This chapter especially focuses on the programing method that uses dedicated instructions.

All the knowledge required to operate a serial communication module (system configuration, connection method, and various settings and operations of a serial communication module) are covered in this chapter.

3.1 Settings Before Operation and Setting Procedure

3.2 Parameter Settings

3.3 Parameter Writing

3.4 Predefined Protocol Support Function

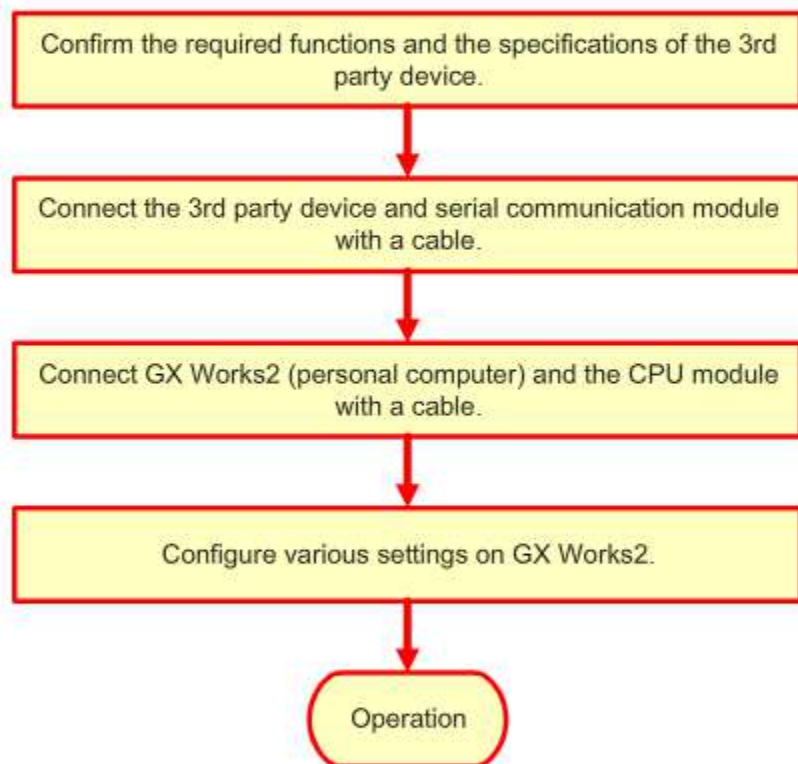
3.5 Dedicated Instructions

3.6 Summary

3.1 Settings Before Operation and Setting Procedure

This section describes the system structure containing a connected 3rd party device, as well as the serial communication module settings and cable connection methods.

The set-up procedure for a serial communication module is shown below.



...

Specifications of the bar code reader used in the example system	
Interface	RS-232
Baud rate	9600 bps
Data bit	7 bits
Parity bit	Present
Parity	Odd number
Stop bit	1 bit
Receive complete code	CR+LF

3.1.1

Example system structure

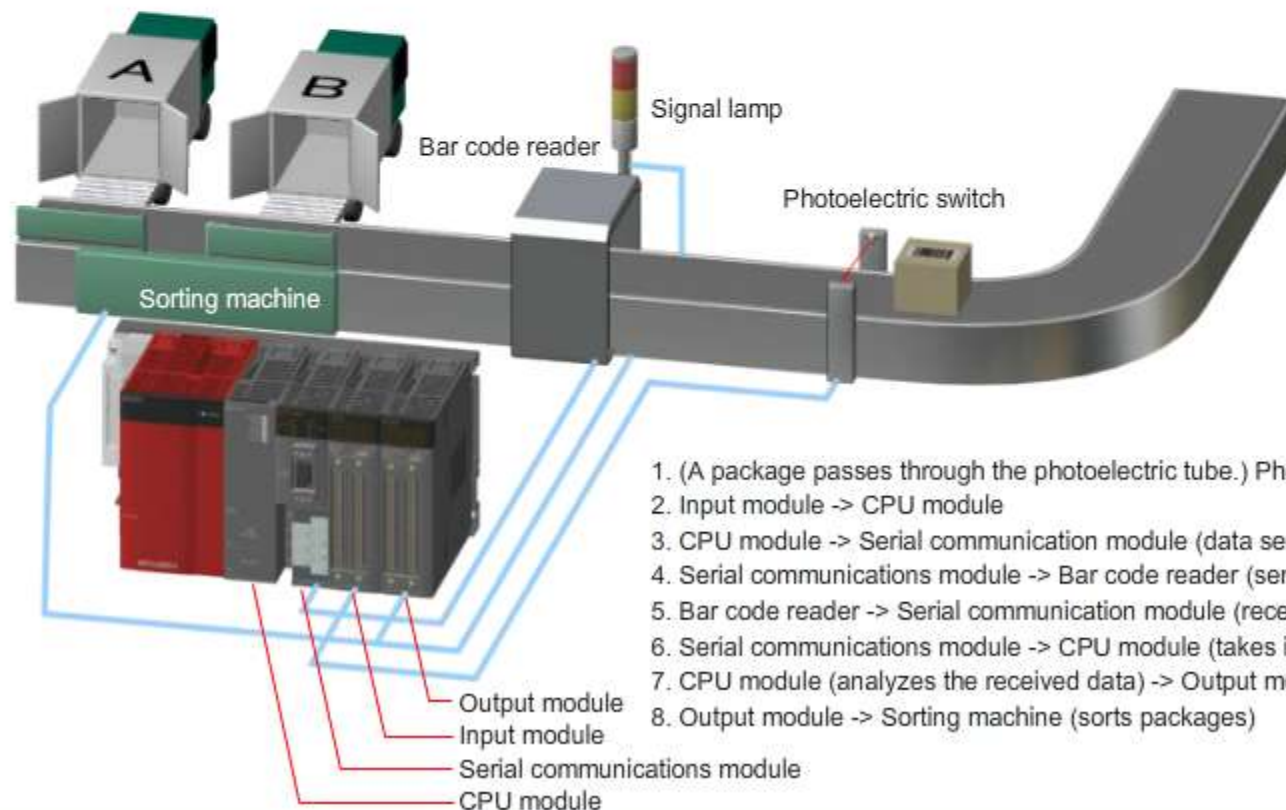
The example system shown below has the following structure and performs the following operations:

Structure

- A bar code reader and a signal lamp are installed close to each other.
- The bar code reader is connected with programmable controllers including a serial communication module via an RS-232 interface.

Operation

- A package moving on a conveyor is detected.
- After the detection, the bar code reader reads the bar code on the package.
- The read data is sent as a variable-length data with a receive complete code [CR+LF] appended, to the serial communication module.
- The read data is then saved in the CPU module devices.



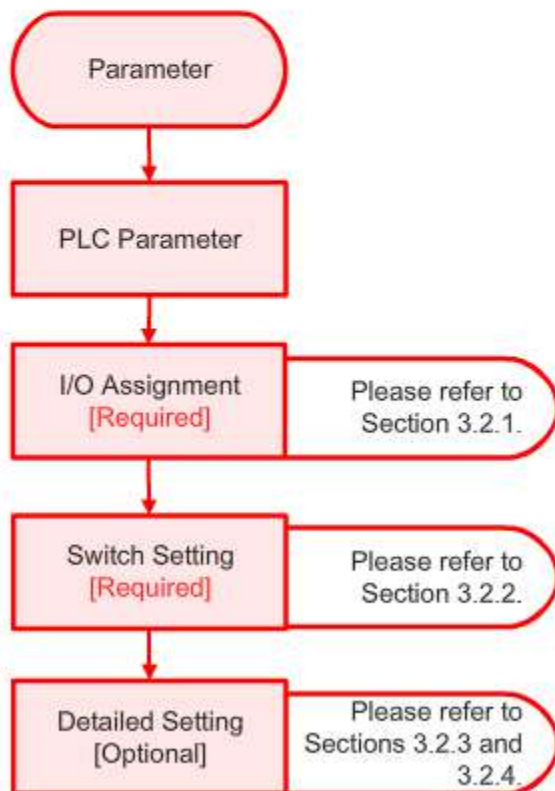
3.2

Parameter Settings

Various settings need to be configured through GX Works2 to establish data communication with a 3rd party device.

Parameter setting overview

- The model, installation slot No., start I/O number, etc. of the serial communication module are set in "I/O Assignment".
- The transmission speed, communication speed, etc. of the serial communication module are set for each channel in "Switch Setting".
- The control method is set in "Detailed Setting" in accordance with the control target of the serial communication module.



3.2.1

I/O assignment setting

The model, installation slot No., start I/O number, etc. of the serial communication module to be installed on the base unit are configured in "New Module" window.

To add a new module in GX Works2, select "PLC Parameter" - "I/O Assignment" - "New Module".

The screenshot shows the "New Module" dialog box with the following fields and annotations:

- Module Selection:**
 - Module Type:** A dropdown menu showing "Serial Communication/Modem Interface Module". A yellow callout bubble points to it with the text "Select 'Serial Communication/Modem Interface Module'".
 - Module Name:** A dropdown menu showing "QJ71C24N". A yellow callout bubble points to it with the text "Select 'QJ71C24N'".
- Mount Position:**
 - Base No.:** A dropdown menu showing "Main Base".
 - Mounted Slot No.:** A numeric input field showing "0". A yellow callout bubble points to it with the text "Set '0' for Mounted Slot No.". To the right of this field is a button labeled "Acknowledge I/O Assignment".
 - Specify start XY address:** A checked checkbox, followed by a text field containing "0000", a dropdown menu showing "(H)", and a text field containing "1 Slot Occur [32 points]".
- Title setting:**
 - Title:** An empty text field.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

New Module window

3.2.2

Switch settings

The transmission speed, communication speed, etc. of a serial communication module are set for each channel in "Switch Setting". In GX Works2, select "Intelligent Function Module" - "0000: QJ71C24N" - "Switch Setting".

Switch Setting 0000:QJ71C24N

Use only "CH1" in this example system.

Item	CH1	CH2
Operation setting	Independent	Independent
Data Bit	7	7
Parity Bit	Exist	Select "Exist"
Even/odd parity	Odd	Odd
Stop bit	1	1
Sum check code	None	None
Online Change	Disable	Disable
Setting modifications	Disable	Disable
Communication rate setting	9600bps	Select "9600bps".
Communication protocol setting	Predefined protocol	Select "Predefined protocol".
Station number setting (0 to 31)	0	

Switch Setting window

Item	Item setting details
Transmission Setting	Operation setting
	Set whether two channels are used separately or linked for data communication.
	Data bit
	Set the bit length of one character in the communication data.
	Parity bit
	Set whether to add a parity bit to the communication data.
	Even/odd parity
	Set whether to add an odd or even parity bit.
Transmission Setting	Stop bit
	Set the stop bit length of the data exchanged with the 3rd party device.
Transmission Setting	Sum check code
	Set whether to add a sum check code to sent and received messages.
Transmission Setting	Online change
	Set whether to write while the CPU module is in the "RUN" state.
Transmission Setting	Setting modifications
	Set whether to permit changes to the settings after the module has started up.
Communication rate setting	
Set the speed of communication with the 3rd party device.	
Communication protocol setting	
Set the details of communication with the 3rd party device.	
Station number setting (0 to 31)	
Set the station number set by the 3rd party device when MC protocol is used.	

3.2.3

Changing the word/byte unit

Set the unit of sent/received data to word or byte.

The default unit is word. This setting needs to be changed in order to handle data in byte units.

In GX Works2, select "Intelligent Function Module" - "Various Controls Specification".

For specification of communication control	The user can change the communications method to match the specifications of the external device.	
Word/byte units specification	1:Byte Unit	0:Word Unit
CD terminal check specification (for RS-232)	0:Word Unit	1:Not Check
	1:Byte Unit	

Various Control Specification window

3.2.4

Changing the receive data count and receive complete code

The received data (size) count and the data receive complete code can be configured.
In GX Works2, select "Intelligent Function Module" - "Various Controls Specification".

Reception method	Received data count Default value: 511 (1FFH) words	Receive complete code Default value: CR+LF
Variable length	<p>To receive data equal to or smaller than the default value, use this setting as it is.</p> <p>To receive data exceeding the default value, change this setting together with other settings.</p> <p>For details, please refer to the corresponding manual of the serial communication module.</p>	To use a receive complete code other than the default value, change this setting.
Fixed length	Change the setting according to the length of the received data.	Change to "Not specified (FFFFH)".

Setting example for a fixed-length (10 words) data

<input type="checkbox"/> For data reception		For data transmission using the non procedure protocol, register system setting values.	
Received data count specification	10 (Ah)	511 (1FFh)	
Receive complete code specification	65535 (FFFFh)	3338 (D0Ah)	

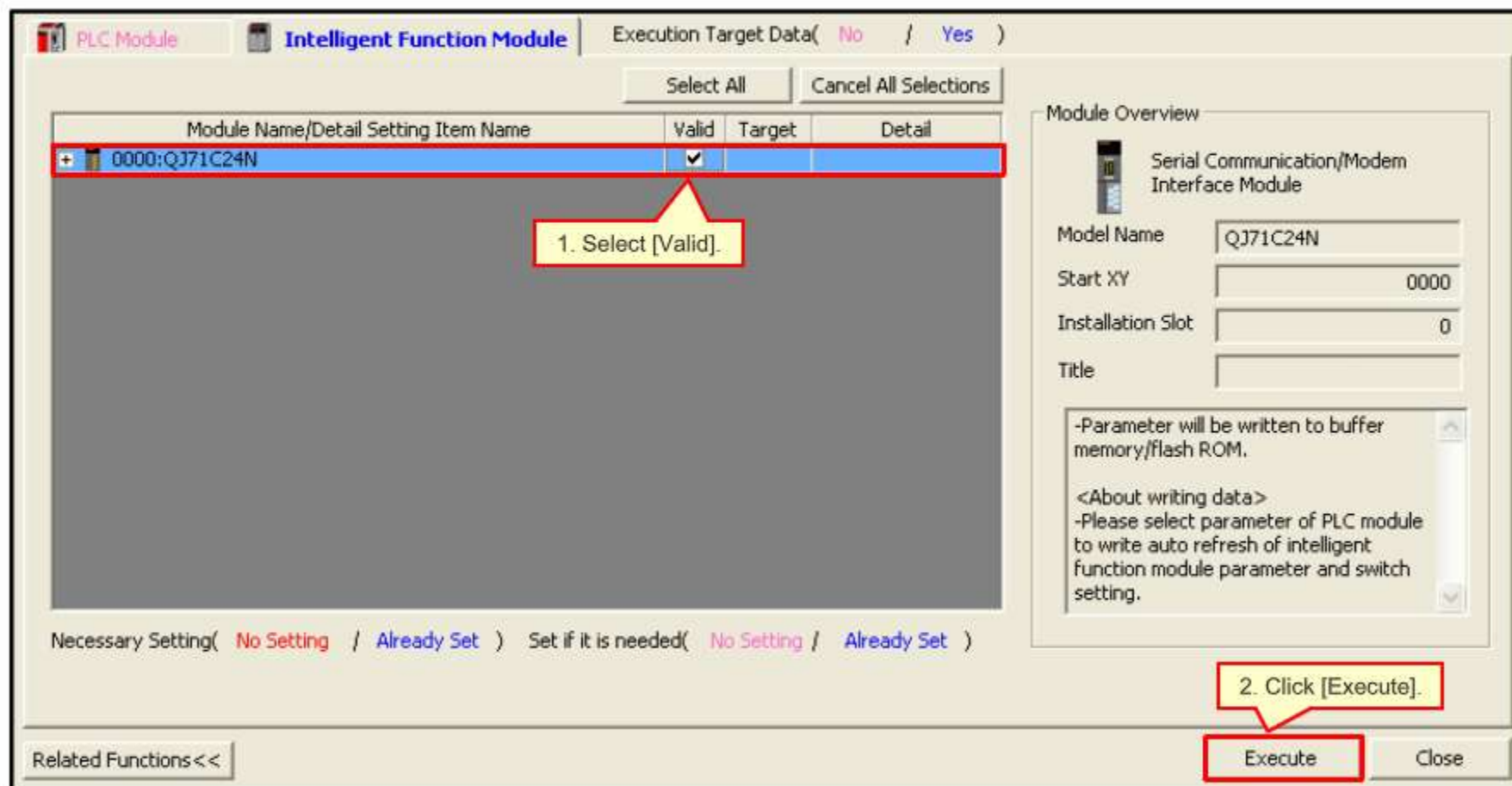
Various Control Specification window

3.3

Parameter Writing

The switch settings and various control specifications that have been configured in GX Works2 need to be written to the serial communication module.

In GX Works2, select "Online" – "Write to PLC" - "Intelligent Function Module" tab.



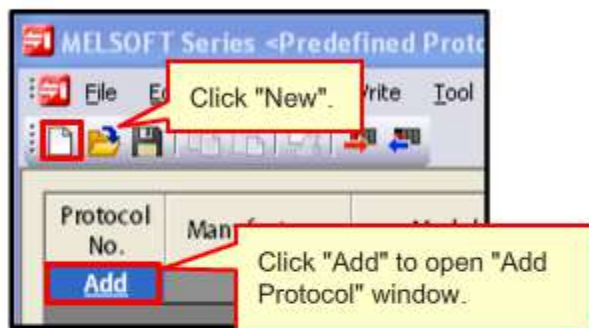
Write to PLC window

3.4

Predefined Protocol Support Function

The "predefined protocol support function" of GX Works2 enables protocol communication with a 3rd party device by using simple sequence programs containing dedicated instructions. The predefined protocol support function reduces the program size and program-creation time as compared to using individual sequence programs.

In GX Works2, select "Tool" – "Predefined Protocol Support Function" - "Serial Communication Module" to open the "Predefined Protocol Support Function" window.



Predefined Protocol Support
Function window

Some predefined protocols are already in GX Works2, but if the protocol of the 3rd party device is not found, new protocol can be created.

(1) When the predefined protocol is already in GX Works2

Select the manufacturer, model, and protocol name in the "Add Protocol" window.

(2) When the predefined protocol is not found in GX Works2

Create a new predefined protocol.

In the example system of this course, the predefined protocol will be newly created according to the 3rd party device.

3.4.1

Adding a protocol

(1) When the predefined protocol is already in GX Works2

When the desired predefined protocol already exists, select the manufacturer and model in the "Add Protocol" window to register it.

Add Protocol

Adds new protocol.

Selection of Protocol Type to Add

Type: Predefined Protocol Library Reference

* Select from Predefined Protocol Library.
Please select manufacturer, model and protocol name from Protocol to Add.

Protocol to Add

Protocol No.	Manufacturer	Model	Protocol Name
1	Cognex	DataMan100	GET:Common Prtcol

Set Protocol No., which will be specified in predefined protocol dedicated instructions.

The number can be selected from 1 to 128.

Select the manufacturer, model, and protocol name of the 3rd party device.

Add Protocol window

3.4.1

Adding a protocol

(2) When the predefined protocol is not found in GX Works2

On the "Add Protocol" window, select "Add New" at Type.

Adds new protocol.

Selection of Protocol Type to Add

Type : Add New

Reference

* Create new protocol.

Protocol to Add

Protocol No.	Manufacturer	Model	Protocol Name
1			

OK Cancel

Add Protocol window

Set Protocol No., which will be specified in predefined protocol dedicated instructions.

The number can be selected from 1 to 128.

3.4.2

Protocol setting

Set the information of the newly added predefined protocol and the details of the communication data.

Set the information about the 3rd party device and the newly added protocol. Double-click this area to open "Protocol Detailed Setting" window. Please refer to the next page for details.

Protocol No.	Manufacturer	Model	Protocol Name	Communication Type	Packet Name	Packet Setting
1				Send&Receive		
						Element Unset
						Element Unset

Protocol in Predefined Protocol Library

- Protocol Line
- Send Packet Line
- Receive Packet Line

Editable Protocol

- Protocol Line
- Send Packet Line
- Receive Packet Line

Protocols: 1/128 Packets: 2/256 Packet Data Area Usage: 0.0% Module for Debugging

Predefined Protocol Support Function window

3.4.2

Protocol setting



Forward

Detailed protocol settings

Set the information of the connected device, protocol, and data communication.

Protocol Detailed Setting

Connected Device Information

Manufacturer:

Type:

Model:

Version: (0000 to FFFF)

Explanation:

Protocol Setting Information

Protocol No.:

Protocol Name:

Communication Type:

Receive Setting

Clear OS area [receive data area] before protocol execution: ☐ Enable ☒ Disable

Receive Wait Time: x 100ms [Setting Range] 0 to 30000 (0: Infinite Wait)

Send Setting

Number of Retries: Times [Setting Range] 0 to 10

Retry Interval: x 10ms [Setting Range] 0 to 30000

Standby Time: x 10ms [Setting Range] 0 to 30000

Monitoring Time: x 100ms [Setting Range] 0 to 3000 (0: Infinite Wait)

Communication Parameter Batch Setting

OK Cancel

Set the information about the connected device.

Select whether to clear the module's OS area (received data area) before executing a program by the protocol.

Set the number of retries when the transmission from the module is not completed within the "monitoring time".

Set the time period for which the module waits before transmitting the data instructed by the predefined protocol.

Set the protocol information.

Set the data reception waiting time period of the serial communication module.

Set the time until the next retry.

Set the time period from the module goes into the "Sending" state until transmission is completed.

Protocol Detailed Setting window

3.4.3

Packet setting

The data that is exchanged in one communication link with the 3rd party device is called a "packet", and a packet consists of different elements. The packet configuration can be set in "Packet Setting".

Communication Type	-> Send	Packet Name	Packet Setting
	<- Receive		
Send&Receive			
	->		Element Unset
	<- (1)		Element Unset

Click "Element Unset" to display the "Packet Setting" window.
When the communication type is "->Send <- Receive", set the packet for sending and receiving.

Predefined protocol support function window

Packet Setting

Protocol No. Protocol Name

Packet Type Packet Name

Element List

Element No.	Element Type	Element Name	Element Setting
<div> <div>Set the packet name.</div> <div> <div>Click "Add New" to add a new packet element.</div> <div> <div>Select the packet elements to add. The elements are described in the subsequent pages.</div> <div> <div> <div>Header</div> <div>Terminator</div> <div>Length</div> <div>Static Data</div> <div>Non-conversion Variable</div> <div>Conversion Variable</div> <div>Check Code</div> </div> <div> <div>OK</div> <div>Cancel</div> </div> </div> </div> </div> </div>			

Packet Setting window

3.4.4

Packet element type



Forward

Header

A specific code or character string can be added to the head of a packet.

- When transmitted: The specified code or character string is sent.
- When received: The header is verified against the received data.

Terminator

A code or character string can be added to indicate the end of a packet.

Static data

A specific code or character string, such as a command, can be included in a packet.

- When transmitted: The specified code or character string is sent.
- When received: Received data is verified.

Set the element name.

Element Setting - Header(Send)

Element Name

Code Type: ASCII String

Setting Value (0 byte)

Select the data type of the setting value. (ASCII string / ASCII control code / HEX)

Set the data in 1 to 50 bytes.

Code type	Setting example
ASCII string	HEADER
ASCII control code	STX, ETX*
HEX (hexadecimal)	FFFF

[Setting Range] 1 to 50

OK Cancel

Element Setting window (header, terminator, static data)

* STX: Start of text, ETX: End of text

3.4.4

Packet element type

Length

An element indicating the data length can be included in a packet.

- When transmitted: Data length of the specified range is automatically calculated, added to the packet, and sent.
- When received: The received data is checked against the data length information (value) contained in the received data.

The screenshot shows a dialog box titled "Element Setting - Length(Send)". It contains several input fields and dropdown menus. Red callout boxes provide instructions for each field:

- Element Name:** Set the element name.
- Code Type:** Select the format of the data length. (ASCII hexadecimal / ASCII decimal / HEX). The current selection is "ASCII Hexadecimal".
- Data Length:** Select data length between 1 and 4. The current selection is "1".
- Data Flow:** Select the data flow order when the data length is not "1". The current selection is "-".
- Calculating Range (Start):** Select the start and end of the range where data length is calculated. Select by the packet element number. The current selection is "1".
- Calculating Range (End):** Select the start and end of the range where data length is calculated. Select by the packet element number. The current selection is "1".

At the bottom of the dialog box are "OK" and "Cancel" buttons.

Element Setting window (length)

3.4.4

Packet element type

Non-conversion variable

Use a non-conversion variable when:

- Data in a device or the buffer memory is sent as it is without data conversion.
- Part of a received packet is stored in a device or buffer memory without data conversion.

Element Setting - Non-conversion Variable(Send)

Set the name of an element that specifies the data storage area.

Set the data length. When the data length is varied, set the maximum data length.

Select whether to conduct the byte swap.

When the data length is fixed, set the start address of the device in which a variable is stored. The end address is set automatically.

When the data length is varied, this area is set automatically according to the setting at Send Data Storage Area.

Select "Fixed Length" or "Variable Length".

Select "Lower Byte + Upper Byte" or "Lower Byte Only".

Set here only when "Variable Length" is selected.

Set the start address of the devices in which the sent/received data length of the element is stored.

Element Setting window (non-conversion variable)

The dialog box contains the following fields and controls:

- Element Name:** A text input field.
- Fixed Length/Variable Length:** A dropdown menu currently set to "Fixed Length".
- Data Length/Maximum Data Length:** A text input field with the value "1". A note indicates the setting range is 1 to 2048.
- Unit of Stored Data:** A dropdown menu currently set to "Lower Byte + Upper Byte".
- Byte Swap:** A dropdown menu currently set to "Disable (Lower -> Upper)".
- Data Storage Area Specification:** A section containing:
 - Send Data Storage Area:** A text input field with "(1 Word)" next to it.
 - Device Symbol:** A text input field with a list of symbols: B, D, W, R, ZR, G (Buffer Memory).
- Buttons:** "OK" and "Cancel" at the bottom right.

3.4.4

Packet element type

Conversion variable

The data in the device or buffer memory is sent after being converted, and received data is converted and then stored in the device or buffer memory. This data conversion process does not require a sequence program and reduces the total program size and programming time.

Set the name of an element that specifies the data storage area.

Select "Fixed Number of Data" or "Variable Number of Data".

Select the number of digits "1 to 10" or "Variable Number of Digits".

Determine how many words of the data in the data storage area are handled as one set of data.
"Word"/"Double word"

- When data is sent

"HEX -> ASCII hexadecimal"
"HEX -> ASCII decimal"

- When data is received

"ASCII hexadecimal -> HEX"
"ASCII decimal -> HEX"

Set the data quantity
(1 to 256).

Select a digit character "-" or "0".
When the number of digits is "Variable Number of Digits", this item is disabled and "-" is displayed.

Element Setting window (conversion variable)

(Continued on the next page)

3.4.4

Packet element type

(Continued from
the previous page)

Element Setting - Conversion Variable(Send)

Element Name: _____

Conversion: HEX->ASCII Decimal

Fixed Number of Data/Variable Number of Data: Fixed Number of Data

Number of Send Data: 1 [Setting Range] 1 to 256

Number of Send Digits of Data: 5

Blank-padded Character at Send: 0

Conversion Unit: Word

Sign: Unsigned

Sign Character: -

Number of Decimals: No Decimal Point

Delimiter: No Delimiter

Data Storage Area Specification: _____

Send Data Storage Area: _____ (1 Word)

OK Cancel

Select "Unsigned" or "Signed".

Select "No Decimal Point", "1 to 9", or "Variable Point".

When "Signed" is selected at Sign, select "None", "+", "0", or "-".*

Select "No Delimiter", "One-byte Comma", or "Space".

Set here only when "Variable Number of Data" is selected.

Set the start address of the devices in which the quantity of sent/received data of the element is stored.

- When the data length is fixed, set the start address of the device in which a variable is stored. The end address is set automatically.
- When the data length is varied, this area is set automatically according to the setting at Send Data Storage Area.

Element Setting window (conversion variable)

* Select "+". Negative values always need the "-" symbol.

3.4.4

Packet element type

Check code

An element that checks for incorrect data can be included in a packet.

The check code can be added to a transmitting packet or used against a reception packet. The check code calculation is automatically performed at data reception/transmission.

The dialog box is titled "Element Setting - Check Code(Send)". It contains the following fields and options:

- Element Name:** A text input field for naming the element.
- Processing Method:** A dropdown menu currently set to "Horizontal Parity".
- Code Type:** A dropdown menu currently set to "ASCII Hexadecimal".
- Data Length:** A dropdown menu currently set to "1".
- Data Flow:** A text input field currently containing a hyphen "-".
- Complement Calculation:** A dropdown menu currently set to "No Complement Calculation".
- Calculating Range (Start):** A dropdown menu currently set to "1".
- Calculating Range (End):** A dropdown menu currently set to "1".

At the bottom of the dialog are "OK" and "Cancel" buttons.

Set the element name.

Select the send/receive format.

ASCII Hexadecimal /
ASCII Decimal / HEX

If Data Length is set to
other than "1", set here.

Select the start and end
of the calculation range.
Set by the packet
element number.

Select the calculation method.

Horizontal Parity / Sum Check
/ 16-bit CRC (for MODBUS)

Set the data length
between 1 and 4.

Select "No Complement
Calculation", "One's
Complement", or "Two's
Complement".

Element Setting window (check code)

3.4.5

Setting of the example system

This section explains the packets sent/received by the predefined protocol in the example system.

(1) Send packet

The send packet contains the command character string for instructing a bar code read.

It is composed of the header character string "M", command character string "TR" (static data, ASCII character) and packet end code "CR+LF" (terminator, ASCII character).

Protocol No.	1	Protocol Name	Bar code reader
Packet Type	Send Packet	Packet Name	BR read trigger
Element List			
Element No.	Element Type	Element Name	Element Setting
1	Header	Header	"M"(2Byte)
2	Static Data	Trigger	"TR"(2Byte)
3	Terminator	Footer	"CRLF"(4Byte)

Packet Setting window
(send packet)

(2) Receive packet

The receive packet contains the country ID code (JPN/USA) that has been read by the bar code reader.

The receive packet is composed of the number of country ID code characters "3" (static data, ASCII character), the country ID code (non-conversion variable, ASCII character), and a packet end code "CR+LF" (terminator, ASCII character). After the packet is received, the country ID code is stored in the devices "D600" and "D601".

Protocol No.	1	Protocol Name	Bar code reader
Packet Type	Receive Packet	Packet Name	BR read data output
Packet No.	1		
Element List			
Element No.	Element Type	Element Name	Element Setting
1	Header	Header	"M"(2Byte)
2	Static Data	# of char.	"3"(1 Byte)
3	Non-conversion Variable	Read data	[D600-D601](Fixed Length/3Byte/Lower/Upper Byte/No Swap)
4	Terminator	Footer	"CRLF"(4Byte)

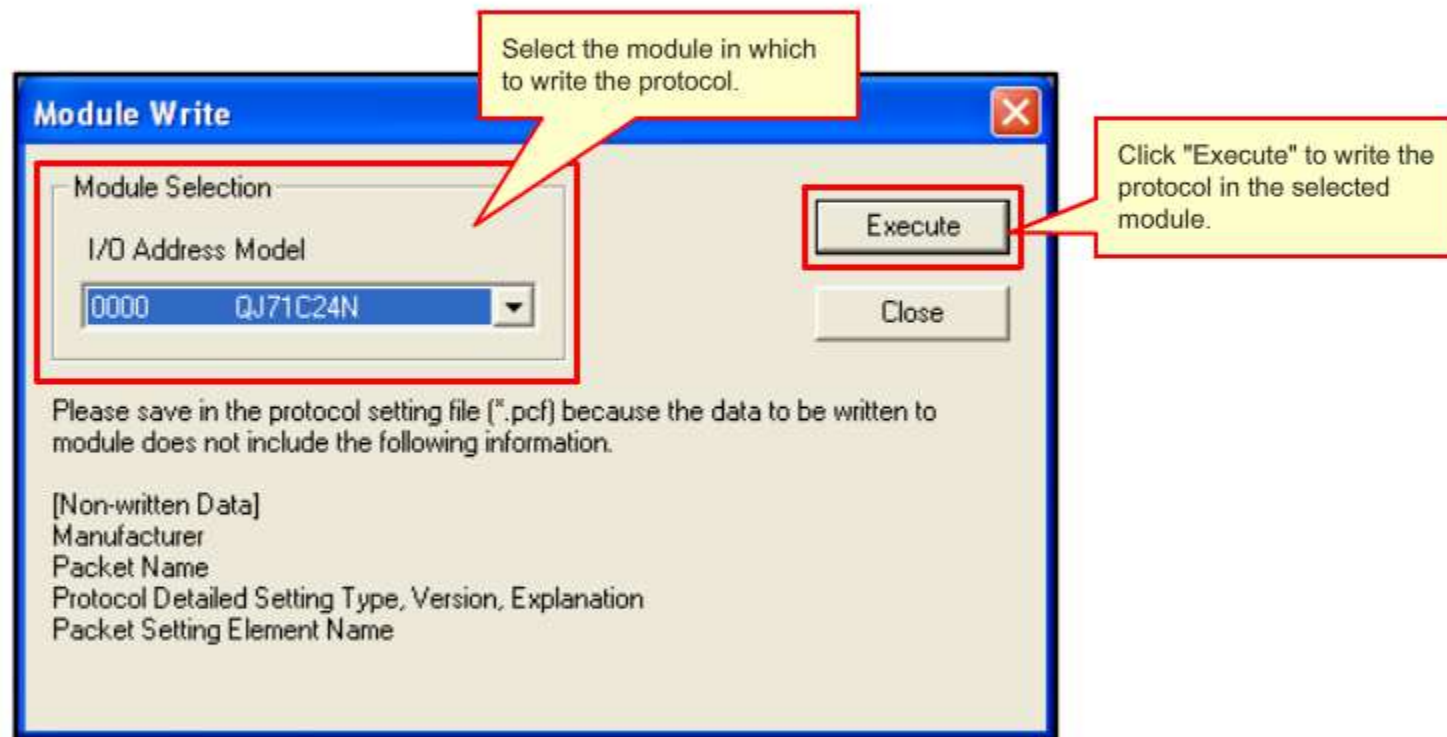
Packet Setting window
(receive packet)

3.4.6

Saving and writing created protocols

To save the created protocol in a protocol setting file, select "File" - "Save as" in the Predefined Protocol Support Function window.

The created protocol needs to be written in the serial communication module.
Select "Online" - "Module Write" in the Predefined Protocol Support Function window.



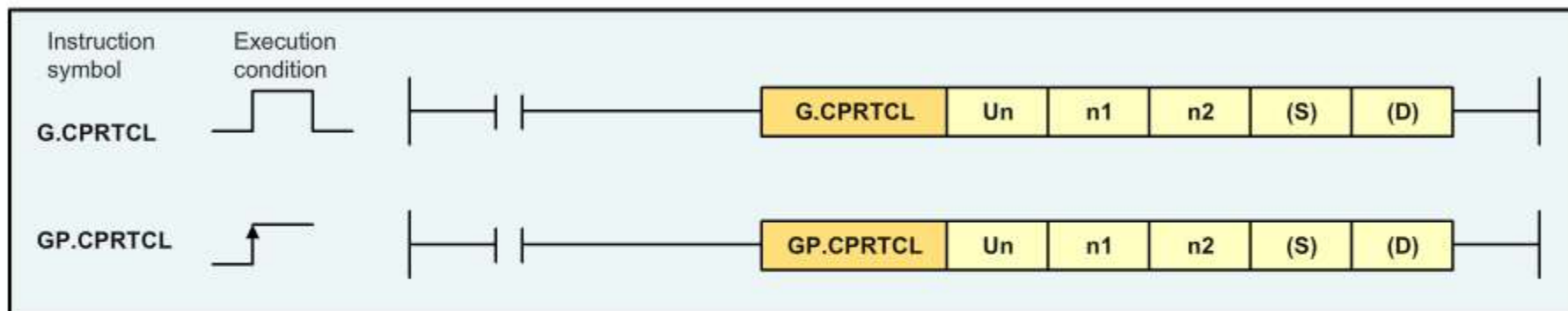
Module Write window

3.5

Dedicated Instructions

Dedicated instructions of sequence programs can be used to execute the predefined protocol, which has been written in the module.

Dedicated instruction



Setting data

Setting data	Details	Setting by	Data type	Value for the example system
Un	Start I/O signal of the module (00 to FE: First two digits of the three-digit I/O signal)	User	BIN 16 bits	Set the module installation slot 0.
n1	Channel for communicating with a 3rd party device 1: Channel 1 (CH1 side) 2: Channel 2 (CH2 side)	User	BIN 16 bits device name	Set "1" to use Channel 1
n2	Continuous protocol execution count (1 to 8)	User	BIN 16 bits device name	Number of protocols processed at a time. Set "1".
(S)	Start number of the device in which control data is stored.	User, system	Device name	Set "D500".
(D)	Device number of the bit device to be turned on when execution is completed.	System	Bit	"M1000"

3.5

Dedicated Instructions

Control data

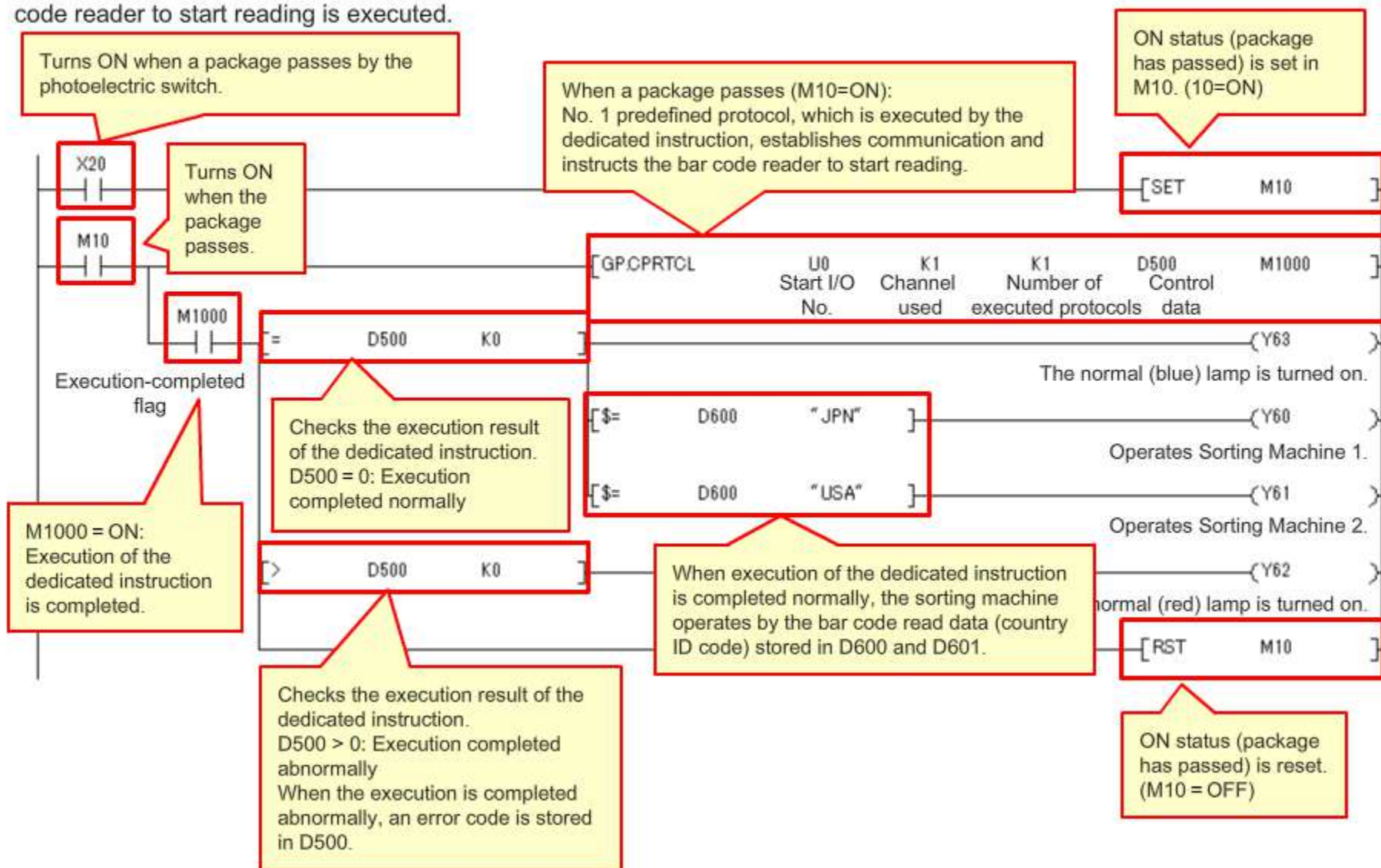
Control data is the data area storing the parameters to be executed by the GP.CPRTCL instruction.
The execution results are also saved here.

Setting data	Item	Set data	Setting range	Set by	Value for the example system
(S) + 0 = D500	Execution result	Execution result of the G (P).CPRTCL instruction. When multiple predefined protocols are executed, the execution result of the last executed predefined protocol is stored. 0: Normal Value other than 0: Error code	—	System	"0" denotes normal response. When error, an error code is written automatically by the system.
(S) + 1 = D501	Receive result	Number of executed predefined protocols. A protocol that has caused an error is also included in the number of executed protocols. "0" is stored when there is an error in the setting data or control data settings.	1 to 8	System	A normal response, "1", is written automatically by the system.
(S) + 2 = D502	Protocol No. to be executed	The protocol No. to be executed first, or the protocol No. of a functional protocol.	1 to 128 201 to 207	User	Write "1" in D503 because only the protocol No.1 is used.
-		-			
(S) + 9 = D509		The protocol number to be executed at the 8th order, or the protocol No. of a functional protocol.			

3.5.1

Sequence program example

An example of a sequence program using dedicated instructions is shown below.
When a package passes by the photoelectric switch, the predefined protocol setting that instructs the bar code reader to start reading is executed.



In this chapter, you have learned:

- Settings before operation and setting procedure
- Setting parameters using GX Works2
- Predefined protocol support function
- Dedicated instructions
- Sequence program example

Important points

Setting parameters using GX Works2	The switch settings and various control settings are configured using GX Works2. GX Works2 also configures the necessary settings to a serial communication module that will be installed on a programmable controller.
Parameter writing	The switch setting and various control settings that have been configured by GX Works2 need to be written in a serial communication module.
Predefined protocol support function	The "predefined protocol support function" of GX Works2 enables data communication with a 3rd party device in accordance with the 3rd party device's protocol. The function uses simple sequence programs containing dedicated instructions.
Dedicated instructions	The predefined protocol written in the flash ROM can be executed using the dedicated instructions (CPRTCL).

Chapter 4 Troubleshooting

[Forward](#)

Chapter 4 describes network diagnostics for problems.

4.1 Troubleshooting

4.2 Summary

Below are the details of the errors that can occur in data communication between a serial communication module and a 3rd party device, and corrective actions for the errors.

Problem	Possible cause	Corrective action	Reference
ERR LED turns on.	<ul style="list-style-type: none"> A communication error has occurred. 	<ul style="list-style-type: none"> Check the error code on the system monitor and remove the cause of the error. 	Section 4.1.1
"RD" does not flash when the 3rd party device sends a message.	<ul style="list-style-type: none"> The send control signal of the 3rd party device is off. 	<ul style="list-style-type: none"> Adjust the wiring so that the CTS signal on the 3rd party device is ready. 	--
"SD" does not flash when a send request is transmitted from the serial communication module.	<ul style="list-style-type: none"> The RS-232 control signals, "DSR" or "CTS", are off. 	<ul style="list-style-type: none"> Check the status of each RS-232 control signal. 	Section 4.1.2
Although "RD" flash after the 3rd party device sends a message, the receive and read request signal (X3/XA) of the serial communication module does not turn on.	<ul style="list-style-type: none"> The predefined protocol setting is incorrect. The 3rd party device did not add the receive complete code. 	<ul style="list-style-type: none"> Check the predefined protocol setting. 	Section 3.2.2
		<ul style="list-style-type: none"> Check the sent/received data using the circuit trace function. 	Section 4.1.3

4.1.1

Checking the error codes on the system monitor

Error codes can be confirmed on the system monitor.

In GX Works2, select "Diagnostics" - "System Monitor".

System Monitor window

The screenshot shows the 'System Monitor' window in GX Works2. The 'Main Base' is Q65B. The 'Operation to Selected Module' section shows the selected module is QJ71C24N. The 'Base Information List' table shows the main base and its extensions. The 'Module Information List (Main Base: Q65B)' table shows the status of various modules, including the selected QJ71C24N module. The 'Error Information' section shows the latest error code as 7FEF. The 'Error and Solution' section provides details about the error and a solution.

Base Information List

Base	Module	Base Model Name	Power Supply	Base Type	Slots	Installed Modules
	⚠	Q65B	Exist	Q	5	3
		Extension Base1				
		Extension Base2				
		Extension Base3				
		Extension Base4				
		Extension Base5				
		Extension Base6				
		Extension Base7				
Overall		1Base				

Module Information List (Main Base: Q65B)

Status	Base-Slot	Series	Model Name	Point	Parameter Type	Point	I/O Address	Network No. Station No.	Master PLC
	-	-	Power	-	Power	-	-	-	-
		CPU	Q06UDHCPU	-	CPU	-	-	-	-
⚠	0-0	Q	QJ71C24N	32Point	Intelli.	32Point	0000	-	-
	0-1	Q	QX40(-TS)	16Point	Input	16Point	0020	-	-
	0-2	Q	QY41P	32Point	Output	32Point	0030	-	-

Error Information

Latest Error Code: **7FEF**

Update Error History
Clear Error History

Error Clear

Display Format:
☒ HEX
☐ DEC

The error history is sequentially displayed from an old error. The latest error is displayed at the bottom line.

Error and Solution

Contents: Switch setting error
 * There is an error in the switch setting by the GX Works2.

Solution: * Write CPU to the parameter and reboot after correcting the setting value for the switch.

System Monitor window (module details)

4.1.2 Checking the signals on the state monitor

On the State Monitor window, a user can check the RS-232 control signal statuses. The status of each signal to/from the serial communication module can also be checked.

In GX Works2, select "Predefined Protocol Support Function" - "Debugging Support Function" - "State Monitor".

Object Module: I/O Address[00] Type[QJ71C24N] Channel[CH1] Monitor Stop Close

Signal | Error Information | Operation Setting Switch | Predefined Protocol Function

No.	Signal Description	Value
X00	CH1 Transmission normal completion	OFF
X01	CH1 Transmission abnormal completion	OFF
X02	CH1 Transmission processing	OFF
X03	CH1 Reception data read request	OFF
X04	CH1 Reception abnormal detection	OFF
X05	CH1 Protocol Execution Completion	OFF
X06	CH1 Mode switching	OFF
X0E	CH1 ERR. Occurrence	OFF
X10	Modem initialization completion	OFF
X11	Dialing	OFF
X12	Connection	OFF
X13	Initialization/connection abnormal completion	OFF
X14	Modem disconnection complete	OFF
X17	Flash ROM read completion	OFF
X18	Flash ROM write completion	OFF

No.	Signal Description	Value
Y00	CH1 Transmission request	OFF
Y01	CH1 Reception data read completion	OFF
Y02	CH1 Mode switching request	OFF
Y03	CH1 Protocol Execution Request	OFF
Y0E	CH1 ERR.clear request	OFF
Y10	Modem initialization request (standby request)	OFF
Y11	Connection request	OFF
Y12	Modem disconnection request	OFF

RS-232 Signal

RTS	●	CD	○
DSF	○	CS	○
DTF	●	RI	○

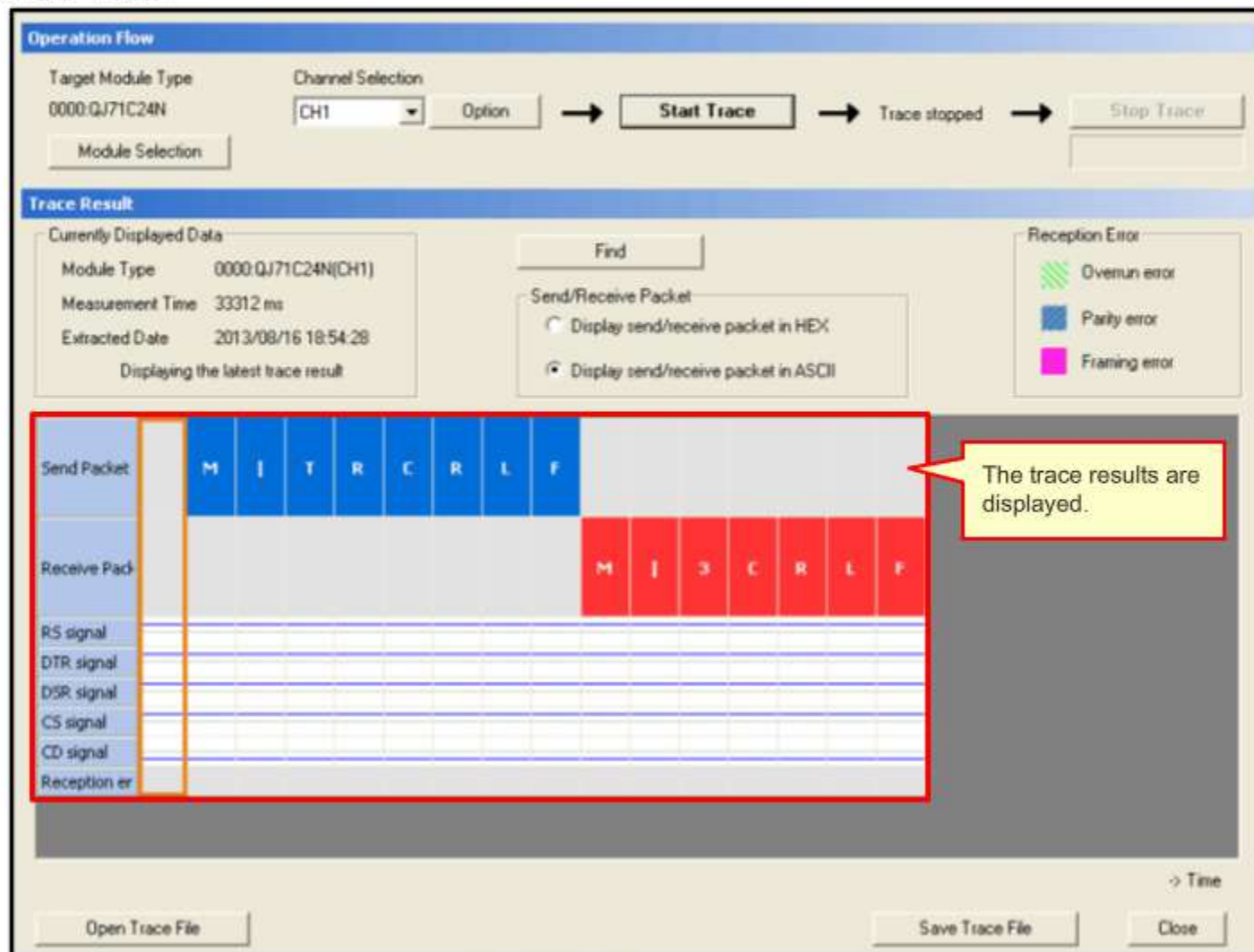
The ON/OFF status of each signal is shown by ● / ○.

4.1.3

Checking the sent/received data using the circuit trace

Check the sent/received data using the circuit trace function.

In GX Works2, select "Tool" - "Intelligent Function Module Tool" - "Serial Communication Module" - "Circuit Trace".



Circuit Trace window

4.2

Summary

In this chapter, you have learned:

- Troubleshooting

Important points

Checking the errors when ERR. LED is on	An error is indicated by the ERR. LED indicator on the serial communication module.
Checking the RS-232 control signal errors	The status of each signal can be checked on the state monitor.
Checking the errors using the circuit trace function	Using the circuit trace function, errors in sent/received data can be checked.

Test**Final Test**

Now that you have completed all of the lessons of the **PLC Serial Communication** Course, you are ready to take the final test. If you are unclear on any of the topics covered, please take this opportunity to review those topics.

There are a total of 11 questions (30 items) in this Final Test.

You can take the final test as many times as you like.

How to score the test

After selecting the answer, make sure to click the **Answer** button. Your answer will be lost if you proceed without clicking the Answer button. (Regarded as unanswered question.)

Score results

The number of correct answers, the number of questions, the percentage of correct answers, and the pass/fail result will appear on the score page.

Correct Answers : **2**

Total Questions : **9**

Percentage : **22%**

To pass the test, you have to answer **60%** of the questions correct.

Proceed

Review

Retry

- Click the **Proceed** button to exit the test.
- Click the **Review** button to review the test. (Correct answer check)
- Click the **Retry** button to retake the test again.

Test**Final Test 1****Network parameters**

Please select the correct term for each description.

(1) A bit that indicates the end of data. :

(2) A value that indicates the transmission speed, followed by the unit "bps". :

(3) A bit that indicates the head of data. :

Answer

Back

Test**Final Test 2****Flow control**

Please select the correct term for each description.

(1) A control method that adjusts data send timing using a flow control line installed separately from the signal line, in the same cable. :

--Select--



(2) A control method that adjusts data send timing using specific codes. :

--Select--



Answer

Back

Test**Final Test 3****RS-232 cable**

Please select the correct description about the RS-232 cable used for a serial communication module.

- ☐ Any RS-232 cross cable available on the market can be used.
- ☐ A cable must be carefully selected in accordance with the protocol of the 3rd party device.

[Answer](#)[Back](#)

Test**Final Test 4****Data reception procedure**

The table below lists data reception methods available to a serial communication module. Please select the correct data reception procedure for each description.

Characteristics of the data received from the 3rd party device	Data reception procedure
The data length is varied. The data has CR+LF added at the end.	<input type="text" value="--Select--"/>
The data length is fixed to 4 bytes.	<input type="text" value="--Select--"/>
The data length is varied. The data has no receive complete code.	<input type="text" value="--Select--"/>

[Answer](#)[Back](#)

Test**Final Test 5****Data exchange procedure**

The table below lists protocols available to a serial communication module.
Please select the correct protocol for each description.

Protocol	Description
--Select-- ▼	Data can be exchanged between the 3rd party device and a CPU module in any message format and by any communication protocol.
--Select-- ▼	Communication protocol for the Q series programmable controllers. By this protocol, the 3rd party device reads or writes the device data and programs of a CPU module via a serial communication module.
--Select-- ▼	This protocol is used when data communication needs to be established according to the protocol of the 3rd party device, such as a measuring instrument or a bar code reader.
--Select-- ▼	If the 3rd party device can send or receive data by MC protocol, it can access a CPU module.
--Select-- ▼	By using an existing simple protocol, data can be exchanged relatively easily with an external device such as a personal computer.
--Select-- ▼	Data communication by the 3rd party device's protocol is realized using the "predefined protocol function".

Answer

Back

Test

Final Test 6

Non procedure protocol

The following descriptions are about data communication by non procedure protocol.
Please select the correct terms to complete the sentences.

Description

To receive data in a by non procedure protocol, a receive complete code is used. To receive data, a received data count is used.

The receive complete code and received data count can be set to to receive data.

[Answer](#)[Back](#)

Test

Final Test 7

GX Works2

The table below explains the received data count and receive complete code settings in GX Works2.
Please select the correct values and terms to complete the table.

Data reception procedure	Received data count Default value: (--Select--) words	Receive complete code Default value: (--Select--)
Fixed length	<p>If the received data count is lower than the default value, the setting change is --Select-- .</p> <p>If the received data count is higher than the default value, the setting change is --Select-- .</p>	<p>If the receive complete code is different from the default value, the setting change is --Select-- .</p>
Variable length	<p>The setting change is required in accordance with the received data length.</p>	<p>The setting must be changed to "Not specified (FFFFH)".</p>

Answer

Back

Test**Final Test 8****Operation check 1**

Please select the sentence that correctly describes the RS-232 control signals, which are used between a serial communication module and its 3rd party device.

- ☐ The signal status can be checked from the "System Monitor" of GX Works2.
- ☐ The signal status can be checked from the "State Monitor" of GX Works2.
- ☐ The signal status can be checked from the "Circuit Trace" of GX Works2.

[Answer](#)[Back](#)

Test

Final Test 9



Operation check 2

The table below shows troubleshooting for data communication failure between a serial communication module and its 3rd party device.

Please select the correct item for each of the possible cause and corrective action.

Symptom	An external device transmitted a message and "RD" blinked, but the read request signal (X3/XA) from the serial communication module was not turned on.
Possible cause	Q1 (A) A communication error is occurring. (B) The transmission control signal is off at the 3rd party device. (C) Communication protocol is set incorrectly. The receive complete code was not added by the 3rd party device.
Corrective action	Q2 (D) Check the error code in the system monitor, and remove the error cause. (E) Check if the CS signal is on using the state monitor. (F) Check the communication protocol setting. Check the send/receive data with the circuit trace function.

Q1 Q2

Answer

Back

Predefined protocol support function 1

Please select the sentence that correctly describes the predefined protocol support function.

- ☐ This function enables protocol communication with a 3rd party device by using simple sequence programs containing dedicated instructions.
- ☐ This function enables automatic analysis of communication parameters transmitted from the 3rd party device so that a protocol suitable for the 3rd party device can be created.

[Answer](#)[Back](#)

Test**Final Test 11****Predefined protocol support function 2**

The sentences below respectively describe "non-conversion variable" and "conversion variable". Please select the correct term for each description.

(1) Data are sent and received without being converted. :

--Select-- ▼

(2) Data are sent and received after being converted.

This data conversion process does not require a sequence program and reduces the total program size and programming time. :

--Select-- ▼

Answer

Back

Test**Test Score**

You have completed the Final Test. Your results are as follows.
To end the Final Test, proceed to the next page.

Correct answers : **0**

Total questions : **11**

Percentage : **0%**

[Proceed](#)[Review](#)[Retry](#)

You failed the test.

You have completed the **PLC Serial Communication** Course.

Thank you for taking this course.

We hope you enjoyed the lessons and the information you acquired in this course will be useful in the future.

You can review the course as many times as you want.

Review

Close