KEYENCE

D42GB

Chapter 1 OVERVIEW

Chapter 2 SPECIFICATIONS AND WIRING

Chapter 3 UNIT INSTALLATION AND MAINTENANCE

Chapter 4 LADDER PROGRAMMING

Chapter 5 CPU BUILT-IN FUNCTIONS

Chapter 6 EXPANSION I/O UNIT FUNCTIONS

- List of PLC Unit Errors
- Lists of CR and CM Devices
- Index

Programmable Controller

KV-8000 Series

User's Manual

Read this manual before using the product.
Keep this manual in a safe place for later reference.

- CPU units
- Power supply units
- Expansion I/O units
- Bus connection units

Frequently Asked Questions

FAQ

www.keyence.com/controlfaq

Supported CPU Units

- KV-8000
- KV-8000A

Preface

This manual describes how to connect to and maintain the KV-8000 Series and also describes the CPU built-in functions and how to create ladder programs. It also includes the specifications and usage methods of CPU units and expansion I/O units. For expansion special units other than those listed above, see the user's manual of the corresponding unit.
When you are writing programs, read the "Instruction Reference Manual," “ST Language Programming Manual,” "Script Programming Manual," and "KV STUDIO User's Manual" together with this manual.
Store this manual in a safe place so that you can retrieve it whenever necessary.
Provide this manual to the end-users of the device.

■ KV-8000 Series Manuals

All the following PDF manuals can be found and opened in the help file of KV STUDIO. In addition, the latest versions of the PDF manuals can be downloaded from the KEYENCE website.

Name	Description
KV-8000 Series CPU Unit User's Manual	This manual. This manual, which describes the system configuration and specifications of the KV-8000 Series. It also describes how to create ladder programs.
KV-8000/7000/5000/3000/1000 Series and KV Nano Series Instruction Reference Manual	This manual describes the instructions that are available in ladder programming.
KV-8000 Series ST Language Programming Manual	This document describes how to create an ST language program and the Operators, Control Statements, and Functions that can be used.
KV-8000/7000/5000/3000/1000 Series and KV Nano Series Script Programming Manual	This manual describes how to create script programs and the available operators, control sentences, and functions.
KV STUDIO User's Manual	This manual describes how to operate KV STUDIO.

KV-8000(A) Series CPU Function Version

The KV-8000(A) CPU function version differs according to the period in which it was shipped. You can check the CPU function version from KV STUDIO, the access window, and the control memory CM.
CPU function version can be upgraded by updating the system program from KV STUDIO or a memory card.

"System Program Update" (page 3-21)
"CPU Monitor" (page 5-134)
"Control Memory Entries CM" (page A-14)

■ Software Licenses and Copyright

This product contains open source software components.
For information about the open source software components, see "license_KV-8000.txt" in the "license" folder found in the KV STUDIO installation folder.

About KV-8000(A) CPU Function Version

The functions of the KV-8000 Series differ depending on the CPU function version.
The CPU function version can be checked from "Tools" → "Check CPU Function Version" → "PLC" in the KV STUDIO menu (* KV STUDIO Ver.11.1 or later is required).
In addition, the CPU function version can be updated from "Tools" → "Check CPU Function Version" → "PLC" → "System Program Update" in the KV STUDIO menu. (* KV STUDIO Ver.11.1 or later is required.)

■ Differences in CPU function by CPU function version

Release		2022/09	2022/06	2021/11
CPU function version		2.6	2.5	2.4
KV STUDIO compatible version		Ver.11.6 or later	Ver.11.5 or later	Ver.11.4 or later
CPU function	OPC UA server function	○	○	○
Supported expansion unit	KV-XH64EC/XH32EC	○	—	—

Release		2021/09	2021/07	2021/04	2020/04	2019/04
CPU function version		2.3	2.2	2.1	2.0	1.0
KV STUDIO compatible version		Ver.11.3 or later	Ver.11.2 or later	Ver.11.1 or later	Ver.11.0 or later	Ver.10.0 or later
Expansion unit	KV-XD02	○	○	—	—	—
CPU function	Device comment	Up to 128 characters	Up to 128 characters	Up to 128 characters	Up to 32 characters	Up to 32 characters
	Variable	○	○	○	○	—
	Array	Compatible with function block type	STRING type supported	—	—	—
	Structure	○	○	○	○	—
	Function block	Compatible with automatic data type conversion	○	○	○(Compatible with EN/ENO, instances)	○
	ST Language	○	○	○	—	—
Instruction / Script	Function block instructions	○	○	○	2 instructions added (FBCALL, FBSTRT)	Not added
	Data Move Instructions	○	○	○	1 instructions added (BBMOV)	—
	Data Processing Instructions	○	○	○	1 instructions added (BBCMP)	—
	Flow Instructions	○	○	1 instruction added (RETURN)	—	—
	Floating Point Instructions	○	○	2 instructions added (ROUND, DROUND)	—	—
	Size Acquisition Instructions	○	○	○	3 instructions added (ACNT,WSIZE, BSIZE)	—
	Suffixes	Compatible with .A	—	—	—	—

Safety Precautions

■ Symbols

This document contains notices that you should observe to ensure your own personal safety, as well as to protect the device and connected equipment. These notices are marked according to the level of danger as shown below.

DANGER	**Indicates a hazardous situation which, if not avoided, will result in death or serious injury.**
WARNING	**Indicates a hazardous situation which, if not avoided, could result in death or serious injury.**
CAUTION	**Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.**
NOTICE	**Indicates a situation which, if not avoided, could result in product damage as well as property damage.**

Important **Indicates cautions and limitations that must be followed during operation.**

Point **Indicates additional information on proper operation.**

Reference Indicates tips for better understanding or useful information.

Indicates a page to be referred to in this manual or other manuals.

■ General Precautions

DANGER	• **Do not use the device with the purpose of protecting human beings.** • **This device is not intended for use as an explosion-proof product. Do not use this device in a hazardous location or in a location that has a potentially explosive atmosphere.** • **Do not use this product in an application that may cause death, serious injury or serious property damage due to a failure with this product should occur, such as nuclear power plants, on aircraft, trains, ships, or vehicles, used within medical equipment, playground equipment, roller coasters and other rides, etc.**
WARNING	• **Output circuit and internal circuit malfunctions sometimes prevent control from being performed normally. Be sure to provide a safety circuit in control systems where circuit malfunctions may lead to fire or other serious accidents.** • **Provide a safety circuit that bypasses the PLC to enable failsafe operation of the entire system in the event that the PLC fails.** • **You must perform a sufficient risk assessment for the machine where this product is to be installed prior to installing this product. Provide appropriate protective fail-safe measures on the machine independent from this product in case a failure with this product should occur.**
CAUTION	• **Before you use this device, verify its functionality and performance at startup and during operation.** • **If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.**
NOTICE	• **Proceed with care when modifying the device, or when using it in a manner that falls outside of the ranges indicated in its specifications, since KEYENCE is unable to guarantee device functionality or performance in such situations.** • **Use this device in combination with other devices only after careful consideration, since the device may fail to satisfy its functionality and performance capabilities as a result of factors such as its usage conditions and the environment in which it is used.**

Precautions Related to Regulations and Standards

■ CE and UKCA Marking

Keyence Corporation has confirmed that this product complies with the essential requirements of the applicable EU Directive(s) and UK regulations, based on the following specifications. Be sure to consider the following specifications when using this product in the Member States of European Union and United Kingdom.

● EMC Directive (CE) and Electromagnetic Compatibility Regulations (UKCA)

- Applicable standards (BS) EN61131-2, Class A
- Be sure to install the KV-8000 Series in an IP54 or better conductive enclosure (such as an industrial control panel).
- Ground the enclosure (such as an industrial control panel) to FG (the enclosure cannot be positively grounded).
- Be sure to use shield cable for signal wire that is outside the enclosure (control panel) and earth shield cable to the enclosure (control panel).
  However, do not ground the serial communication cable connected to the CPU unit or KV-XL202 / XL402 and the shielded cable of USB cable.
- Please be sure to earth the FG terminal for unit that has FG terminal.
- For KV-XLE02, KV-XCM02, use STP cable with double shield of category 5e or more.
- Please wrap 2 turns of ferrite core*1 to USB cable connected to the CPU unit.
- For 24/0 VDC input of the pulse train positioning unit (KV-SH04PL), wrap the ferrite core*2 one turn around each cable connected to the pulse output terminal.
- Wrap a ferrite core*2 one turn around the cable connected to the serial encoder communication terminal of the high-speed counter unit (KV-SSC02).
  *1 Ferrite core used during our assessment : TDK ZCAT-3035-1330
  *2 Ferrite core used for evaluation of our company: NEC / TOKIN ESD-SR-150

However, these specifications do not guarantee that the end product in which this product is incorporated complies with the essential requirements of the EMC Directive and Electromagnetic Compatibility Regulations. The manufacturer of the end product is solely responsible for ensuring that the end product complies with the EMC directive.

Remarks regarding compliance

<Industrial control panel>

- If the industrial control panel has joints on its top, bottom, or similar locations, remove the coating present or carry out any other procedures that are necessary to maintain the conductivity.
- Use a thick grounding cable, so that the impedance will be low even at high frequencies, to ground the industrial control panel.
- Before touching the devices that are inside the industrial control panel, such as when performing maintenance, be sure to eliminate static electricity so that it does not affect the internal devices.

<Grounding the shielded cables>

- Ground the shielded cables close to the expansion units, and ensure that grounded cables do not receive electromagnetic induction from ungrounded cables.
- Remove a portion of the outer covering of the shielded cables, and then ground the stripped part of the shielded cables against a wide surface of the industrial control panel. (If you are using clamps or similar metal fixtures, remove the coating from the parts of the cables that will be in contact with the metal fixtures.)
- Do not solder the wires to the shielded cables, as grounding the wires will increase the high-frequency impedance, which will negate the shielding effect.

● Low-voltage directive

The low-voltage directive applies to the KV-PU1 power supply unit.

- Applicable standard EN60950-1
- Overvoltage category II
- Use the product in an environment that corresponds to pollution degree 2.
- Use this product at the altitude of 2000 m or less.
- Indoor use only.
- When installing the KV-PU1, be sure to install it in a conductive enclosure (e.g. an industrial control panel) with IP54 or higher.

■ UL Certificate

This product complies with the following UL and CSA standards and has been certified by UL.

UL file number: E207185
Category: NRAQ, NRAQ7

Be sure to consider the following specifications when using this product as a product specified by UL.

- Be sure to install it in a conductive enclosure (an industrial control panel) with IP54 or higher.
- Use stranded copper wire (twisted wire) with a wire gauge of AWG #12 to #22 and temperature rating of 105ºC or higher when wiring to the terminal block on the KV-PU1. The tightening torque is 1.4 N･m. (12.4 lb.-In)
- Use stranded copper wire (twisted wire) with a wire gauge of AWG #16 to #22 and temperature rating of 105ºC or higher when wiring to the DC power input terminal block on the CPU unit. The tightening torque is 0.5 N･m.
- When using KV-PU1, please do not supply 24V of electricity from the DC power terminal platform of the CPU unit.
- Use this product under pollution degree 2.
- Overvoltage category I (II when using KV-PU1)
- Use this product at the altitude of 2000 m or less.
- Indoor use only.
- Please confirm that circuit connected to input output terminal (connector, etc.) of this product corresponds to SELV.
- Use this product with one of the following power supplies.
  - UL/CSA certified power supply that provides Class 2 output as defined in the NFPA70（NEC: National Electrical Code) and CEC (Canadian Electrical Code).
  - UL/CSA certified power supply that has been evaluated as a Limited Power Source as defined in UL60950-1 and CAN/CSA-C22.2 No.60950-1.

■ KC Certificate (South Korea)

Class A equipment
This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

A급 기기 (업무용 방송통신기자재)

이 기기는 업무용 (A 급) 전자파적합기기로서 판매자 또는 사용자는 이 점을
주의하시기 바라며 , 가정외의 지역에서 사용하는 것을 목적으로 합니다 .

Precautions on Regulations and Standards (KV-5500/5000/3000 series)

■ CE Marking

Keyence Corporation has confirmed that this product complies with the essential requirements of the applicable EC Directives, based on the following specifications. Be sure to consider the following specifications when using this product in the Member States of European Union.

● EMC Directive

- Applicable standard EN61131-2, Class A
- When installing the KV-5500/5000/3000 Series, it must be installed in an conductive enclosure (industrial control panel) with IP54 or higher.
- Use a shielded cable for the signal lines located outside of the conductive enclosure (industrial control panel). The shielded cable must be grounded.
- Be sure to use the functional earthing terminal on the unit for grounding, if applicable.

These specifications do not give any guarantee that the end-product with this product incorporated complies with the essential requirements of EMC Directive. The manufacturer of the end-product is solely responsible for the compliance on the end-product itself according to EMC Directive.

Notes on compatibility

<The control panel>

- If seams exist within the top or bottom plate of the control panel, strip the coating to maintain electric potential.
- When grounding the control panel, use a thick grounding wire to create low impedance even at high frequencies.

<Grounding shielded cables>

- Do not ground shielded cables near the expansion unit, otherwise, the grounded cables will receive electromagnetic induction from cables that were present before grounding was implemented.
- After removing and exposing a section of the outer covering for the shielded cable, connect this part to the control panel in a wide area. (When using clamp brackets or other such tools, be sure to strip the paint from the contact)
- Do not ground electrical wires by soldering them to shielded wires, as this method will increase high-frequency impedance and nullify the shield effect.

● Low-voltage Directive

The Low-voltage Directive is applied to KV-B8RC and KV-B16RC. It is not applicable to any other models.

- Applicable standard EN61131-2 (KV-B8RC/KV-B16RC)
- Overvoltage Category II
- Pollution degree 2
- When installing the KV-5500/5000/3000 Series, it must be installed in an enclosure (industrial control panel) with IP54 or higher.
- The rated relay load of the KV-B16RC is 8A per common. In other words, when all eight channels are used, the maximum load per channel is 1A, and when four channels are used, the maximum load per channel is 2A.
- The KV-B8RC is an independent contact and the rated relay load is 2A per channel.

■ UL Certificate

The KV-5500/5000/3000 Series is an UL/C-UL Listed Product.
UL-certified products display the UL mark on the label found on the side of each product.

- UL File No. E207185
- Category NRAQ, NRAQ7

Be sure to consider the following specifications when using this product as an UL/C-UL Listed product.

- When installing the KV-5500/5000/3000 Series, it must be installed in an enclosure (Industrial control panel) with IP 54 or higher.
- For wiring to the terminal block on the special units except for the KV-DN20 or I/O units, use stranded copper wire having a gage of AWG#16 to #22 and temperature rating of 60°C or higher (105°C or higher for the KV-B16RC). The tightening torque is 0.56 N•m (4.94 lb-in).
- Use this product with one of the following power supplies.
  - UL/CSA certified power supply that provides Class 2 output as defined in the NFPA70（NEC: National Electrical Code) and CEC (Canadian Electrical Code).
  - UL/CSA certified power supply that has been evaluated as a Limited Power Source as defined in UL60950-1 and CAN/CSA-C22.2 No.60950-1.

■ KC Certificate (South Korea)

Class A equipment

This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

A급 기기 (업무용 방송통신기자재)

이 기기는 업무용 (A 급) 전자파적합기기로서 판매자 또는 사용자는 이 점을
주의하시기 바라며 , 가정외의 지역에서 사용하는 것을 목적으로 합니다 .

Manual Organization

Chapter 1	OVERVIEW	This chapter describes information such as the system configuration of the KV-8000 Series, communication channels with PCs, and the occupied areas of the units.
Chapter 2	SPECIFICATIONS AND WIRING	It describes specification and wiring of CPU unit, AC power unit, expansion I/O unit, bus connection unit of the KV-8000 series.
Chapter 3	UNIT INSTALLATION AND MAINTENANCE	This chapter describes how to install the KV-8000 Series as well as how to attach units, wiring precautions, and maintenance procedures.
Chapter 4	LADDER PROGRAMMING	This chapter describes the specifications and setup methods of the functions that you have to use when creating ladder programs. You can improve programming efficiency by using modules, function blocks, macros and local devices.
Chapter 5	CPU BUILT-IN FUNCTIONS	This chapter describes various functions available using the access window, memory card slot and each communication port of the KV-8000 Series CPU unit.
Chapter 6	EXPANSION I/O UNIT FUNCTIONS	This section describes the functions provided by the KV-SIR32XT expansion I/O
	APPENDICES	The appendices include lists of the KV-8000 Series control relays (CR) and control memory entries (CM), an error list, and a list of the instructions used in this manual.

Contents

Safety Precautions **2**
Manual Organization **7**
Contents **8**
Conventions Used In This Manual **16**
How to Use Mnemonic Lists 16
Terminology 17

Chapter 1 OVERVIEW

1-1 System Configuration **1-2**
System Configuration Overview 1-2
(1) Power Units 1-3
(2) CPU Units 1-3
(3) KV-8000/7000 Series Expansion Units 1-3
(4) Bus Connection Units 1-4
(5) KV-5000/3000 Series Expansion Units 1-4
(6) End Units 1-4
Connecting to Communication Devices 1-5
1-2 Unit Reservation **1-6**

Chapter 2 SPECIFICATIONS

2-1 Series Common Specifications **2-2**
General Specifications 2-2
Performance Specifications 2-3
Operation during Power Outages 2-4
2-2 CPU Unit **2-5**
Package Contents 2-5
Names and Functions of Parts 2-6
RUN-PRG Selector Switch 2-8
Wiring and Internal Circuit Diagram 2-8
Dimensions 2-10
2-3 AC Power Unit **2-11**
Names and Functions of Parts 2-11
Specification 2-12
How to Wire 2-12
Dimensions 2-13
2-4 Expansion I/O Unit of KV-8000/7000 series (KV-SIR32XT) **2-14**
Names and Functions of Parts 2-14
Specifications 2-15
Wiring and Internal Circuit Diagrams 2-16
Dimensions 2-17
2-5 Expansion Input Unit **2-18**
Name and Function of Each Part 2-18
Specification 2-19
Wiring and Internal Circuit Diagram 2-20

Dimensions ····· 2-24
2-6 Expansion Output Unit ····· 2-25
Names and Functions of Each Part ····· 2-25
Specification ····· 2-26
Wiring and Internal Circuit Diagram ····· 2-28
Dimensions ····· 2-33
2-7 Extension I/O Unit ····· 2-34
Name and function of the parts ····· 2-34
Technical specifications ····· 2-35
Wiring/internal circuit diagram ····· 2-36
Dimensions ····· 2-39
2-8 Bus Connection Unit ····· 2-40
Names and Functions of Parts ····· 2-40
Dimensions ····· 2-41
2-9 Expansion Unit ····· 2-42
Name and Function of Each Part ····· 2-42
Specification ····· 2-43
About Connection / Setting ····· 2-43
About Power Supply Switching ····· 2-44
Dimensions ····· 2-45

Chapter 3 UNIT INSTALLATION AND MAINTENANCE

3-1 Connecting Units ····· 3-2
Precautions When Connecting Units ····· 3-2
Constructing Units ····· 3-3
Installing Units on DIN Rails ····· 3-5
3-2 Unit Installation ····· 3-6
Installation Environment ····· 3-6
Installation Position ····· 3-6
3-3 Wiring Precautions ····· 3-8
Unit Wiring Precautions ····· 3-8
Connector Assembly and Wiring ····· 3-11
Precautions When Using the USB Port to Communicate with a PC or Other Peripheral ····· 3-15
3-4 Maintenance ····· 3-16
Inspection and Maintenance ····· 3-16
About Maintenance ····· 3-16
About the Restoration Method ····· 3-18
Restoration Method ····· 3-19
System Program Update ····· 3-21

Chapter 4 LADDER PROGRAMMING

4-1 What Is Sequence Control? 4-2
Types of Sequence Control 4-4
4-2 Programming Language 4-5
Ladder Program 4-5
ST Language 4-6
KV Script 4-7
Mnemonics 4-8
Suffixes 4-9
4-3 Projects 4-11
Project Configuration 4-11
Program Capacity 4-14
Project Protection Settings 4-16
4-4 Devices 4-20
Lists of Devices 4-20
Device notation method 4-24
Configuring Power Off Holding Settings (Latching) 4-26
Bit Devices 4-27
Word Devices 4-37
Index Modification 4-44
Indirect Specification 4-46
Processing Bit Devices in Units of Words 4-50
Bit device processing of word devices 4-52
Internal Registers 4-53
4-5 Variable 4-55
Variable Types 4-55
List of Variables and Data Types 4-56
Array 4-57
Structure 4-59
Variable Setting 4-63
Structure Definition 4-65
ACNT Get number of array elements 4-66
WSIZE Get word size 4-68
BSIZE Gets bit size 4-70
4-6 Constants 4-73
4-7 How Data Is Processed 4-75
Binary Data 4-75
Value Notation 4-76
Correlations between Value Notations 4-78
Character Codes 4-81
4-8 Program Structure and Operation 4-82
CPU Unit Operation 4-82
Scan Time 4-90
Program Configuration 4-93
4-9 Modules 4-98

Overview 4-98
Module Type 4-100
Module Execution Sequence 4-109
Module Configuration Examples 4-110
Module System Devices 4-113
Module Passwords 4-114
4-10 Function Blocks 4-117
Overview 4-117
Usage Procedure 4-119
Function Block Type 4-120
Setting Arguments 4-131
Placing Function Blocks 4-135
Function block usage example 4-137
Argument Abbreviation Settings 4-138
Argument Privacy Settings 4-139
Function Block Nests 4-140
Setting a Function Block Password 4-141
Setting Function Blocks to Read Only 4-142
Function Block System Devices 4-143
Function Block Program Properties 4-144
System Function Block 4-146
Register a function block library 4-147
FBCALL Instance type function block start 4-151
FBSTRT Instance type function block call 4-151
System function block for "KV-8000" 4-154
SR Bistable function block (Set priority) 4-155
RS Bistable function block (Reset priority) 4-157
R_TRIG Rising edge detector 4-159
F_TRIG Falling edge detector 4-160
CTU Up counter 4-161
CTD Down counter 4-163
CTUD Up/Down counter 4-165
TON_100MS 100ms On-delay 4-167
TON_10MS 10ms On-delay 4-169
TON_1MS 1ms On-delay 4-171
TON_10US 10μs On-delay 4-173
4-11 Macros 4-178
Overview 4-178
Usage Procedure 4-180
Macro Type 4-181
Setting Arguments 4-186
Macro System Devices 4-188
Bit That Is Turned ON during Macro Execution 4-189
Macro Passwords 4-191
Differences between Macros and Subroutines 4-195
4-12 Local Devices 4-200
How to Write Local Devices 4-200
List of Local Devices 4-201

Local Device Assignment 4-202
Local Device Input and Errors during Conversion 4-204
4-13 Interrupts 4-206
Overview 4-206
How to Execute Interrupt Programs 4-208
Interrupt Priority 4-209
Instructions That Cannot Be Used in Interrupt Programs 4-210
I/O Processing during Interrupts 4-212
Sample Program 4-214
Interrupt Processing Instructions 4-217
Sample Program 4-219
4-14 File Register 4-226
Overview 4-226
Bank Switchover 4-226
Defaults 4-227
File Register Setting 4-227
File Register Instructions 4-229
4-15 Programming Skills 4-236
From Installation to Operation 4-236
Program Creation Support 4-238
Creating Clear, Legible Programs 4-252
Key Points to Program Creation 4-258
How to Perform Debugging 4-265

Chapter 5 CPU BUILT-IN FUNCTIONS

5-1 Overview of Built-In Functions 5-2
What You Can Do with the Built-In Functions 5-2
5-2 Memory Card 5-3
What You Can Do with the Memory Card 5-3
About the Memory Card 5-6
Devices using memory cards 5-10
Writing/Reading Project Data 5-13
Auto Load Function 5-24
Storage Instructions 5-26
5-3 CPU Memory (User Memory) 5-27
About CPU Memory (user memory) 5-27
CPU memory 5-29
Devices used by the CPU memory 5-32
Writing/Reading Project Data 5-35
Auto Load Function 5-45
RAM mode 5-47
Storage Instructions 5-49
5-4 Recipe Load/Save Function 5-50
Overview of Recipe Load Function/Recipe Save Function 5-50
Procedure for generating / reading recipe data 5-52

Recipe load / save procedure 5-54
Recipe load completion code / recipe save completion code 5-56
5-5 Logging/Tracing 5-57
Overview 5-57
How to Use 5-59
Setup 5-60
CPU unit buffer capacity setting 5-70
About CSV File 5-71
Logging Instructions 5-73
LOGE Enable logging/tracing 5-74
LOGD Inhibited logging/tracing 5-74
TRGD Capture log data 5-77
Device List for Logging/Tracing 5-79
Logging/Tracing Function Precautions 5-83
5-6 Operation Recorder Function 5-86
Overview of operation recorder 5-86
Operation steps 5-87
Settings 5-88
Device recorder buffer capacity setting 5-97
Device list for operation recorder function 5-98
TRGR Device record data acquisition 5-105
Precautions for operation recorder function 5-107
5-7 USB Communication 5-109
How to Connect to a PC 5-109
Installing the USB Driver 5-109
5-8 Ethernet/EtherNet/IP Communication 5-110
The Role of CPU Built-In EtherNet/IP Communication Function 5-110
The Role of the OPC UA Server Function 5-110
5-9 Access Window 5-111
Operations that Can be Implemented Through Access Window 5-111
Unit Monitor 5-115
Device Mode 5-118
Confirm error details 5-126
Error Clear 5-127
Digital Trimmer 5-128
Unit Test 5-130
Date/Time 5-132
CPU Monitor 5-134
Alarm 5-136
Storage 5-139
Access Window Setup 5-151
CLEAR 5-153
Access Window Settings Used by KV STUDIO 5-156
Access Window Initial Menu Display Function 5-157
User Message 5-161
Operation Inhibited Setting 5-162
Key Lock Function 5-163

5-10 Inter-unit Synchronization Function 5-165
What is the Inter-unit Synchronization Function 5-165
Inter-unit Synchronous Module 5-166
Operation When the Inter-Unit Synchronization Module is Not Used 5-167
Refresh Timing Settings (For Expansion unit relays and DM only) 5-168
Inter-Unit Synchronization Refresh Devices 5-169
Setting the Inter-unit Sync Refresh Devices 5-169
CPU Device Write/Read 5-170
Setting CPU Device Write/Read 5-170
System Configuration of Inter-unit Synchronization 5-171
Cycle of Inter-unit Synchronization 5-171
Performance Specifications of Inter-unit Synchronization 5-172
Setting Method for Inter-unit Synchronization 5-173
Setting Steps for Inter-unit Synchronization 5-174
Inter-Unit Synchronization Tracing 5-176
5-11 Cam Switch 5-178
Cam Switch Function Overview 5-178
Cam Switch Instructions 5-178
MCMP Multi-stage comparator 5-179
ABSENC Absolute encoder 5-183
5-12 Clock Function 5-187

Chapter 6 EXPANSION I/O UNIT FUNCTIONS

6-1 I/O Functions 6-2
General I/O 6-2
6-2 Unit Interrupt Function 6-4
Setting Procedures 6-4
6-3 Inter-unit Synchronization Function 6-6
What is the Inter-unit Synchronization Function 6-6
Performance Overview when Running Inter-unit Synchronization Function 6-6
Unit Specification when Running Inter-unit Synchronization Function 6-8
6-4 Input Capture Function 6-9
6-5 Buffering Input Function 6-10
6-6 PWM Output Function 6-19
6-7 Exclusive Instruction for the Unit for KV-SIR32XT 6-23
List of Exclusive Instructions for the Unit for KV-SIR32XT 6-23
6-8 Exclusive Function for the Unit for KV-SIR32XT 6-30
List of Exclusive Function for the Unit for KV-SIR32XT 6-30
U_RDBF Reading out the buffering data 6-31
U_DISOVP Input overvoltage protection function configuration 6-32
U_DISOCP Output overcurrent protection function configuration 6-33
6-9 Access Window 6-37
Access Window 6-37
Configuration Screen 6-39
Monitor Screen 6-40

Unit Test ····· 6-41
6-10 Unit Monitor ····· 6-42
Summary of Unit Monitor ····· 6-42
Displaying Unit Monitor ····· 6-42
6-11 I/O Unit Reference Program ····· 6-43
6-12 I/O Unit Error Mode ····· 6-47

APPENDICES

1 List of PLC Unit Errors ····· A-2
2 Lists of CR and CM Devices ····· A-10
Control Relays (CR) ····· A-10
Control Memory Entries CM ····· A-14
3 ASCII Code Table ····· A-18
ASCII Code Table ····· A-18
4 List of Instructions ····· A-19
5 List of Unusable Instructions ····· A-21
6 List of Abbreviated Mnemonics ····· A-23
7 List of Shortcut Keys ····· A-25
Shortcut Keys That Are Valid in the Unit Editor ····· A-25
Shortcut Keys That Are Valid in KV STUDIO ····· A-27
List of GPPQ and GPPA Format Shortcut Keys ····· A-33
List of CX Format Shortcut Keys ····· A-34
List of Logix Format Shortcut Keys ····· A-35
How to Use the Key ····· A-37
8 List of Characters That Cannot Be Used ····· A-39
9 Precautions for Using the Existing KV Series Extension Unit ····· A-42
List of Compatible Equipment ····· A-42
Usage Restrictions ····· A-43
10 Precautions When Changing From One KV-8000 Series Model to Another ····· A-45
Model Replacement Procedure ····· A-45
Differences Related to Functions Such as CPU Special Hardware Functions ····· A-46
Device Range Comparison Table ····· A-48
Instructions That Have Differences ····· A-49
Differences between Control Relays and Control Memory Entries ····· A-49
Differences between variable and label ····· A-50
11 Index ····· A-52

Conventions Used In This Manual

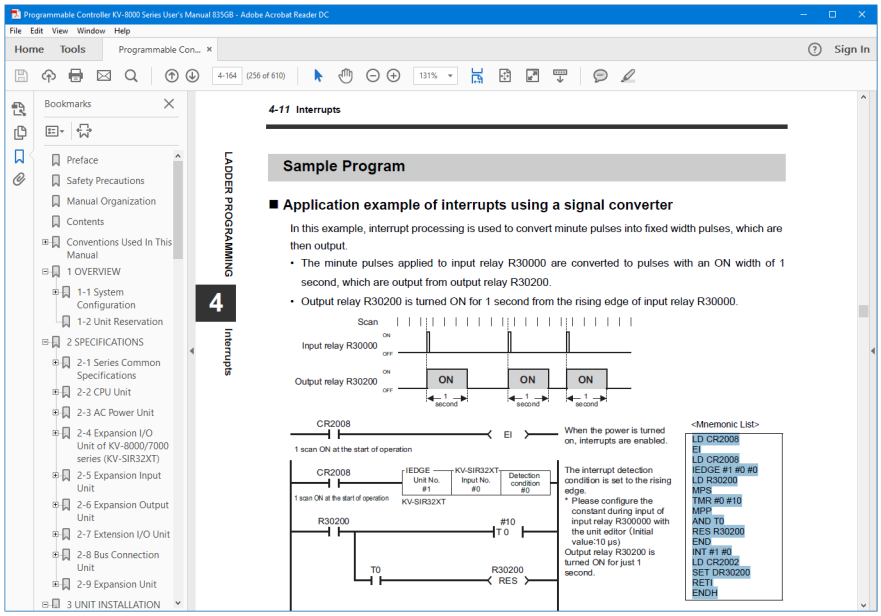
This section describes terminology used in this manual and how to use mnemonic lists.

How to Use Mnemonic Lists

Mnemonic lists are included on the pages in which sample ladder programs are introduced.
You can use mnemonic lists to easily enter ladder programs.

1 **Use Adobe Reader to display the manual.**

2 **Use the "Select" tool to copy a mnemonic list.**

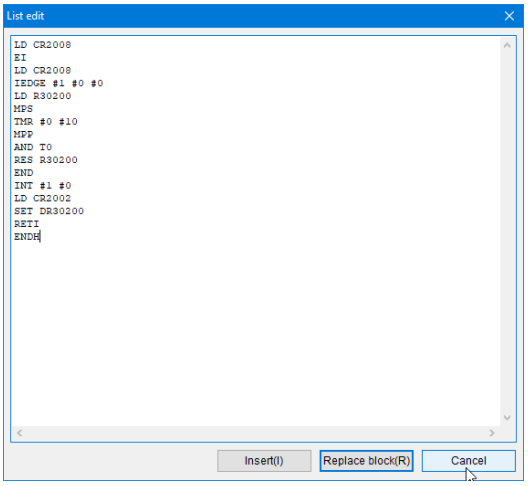


3 **In the ladder edit area of KV STUDIO, click the cell in which you want to insert a sample ladder program, then, on the "Edit" menu, click "Edit list" to display the "List edit" dialog box.**

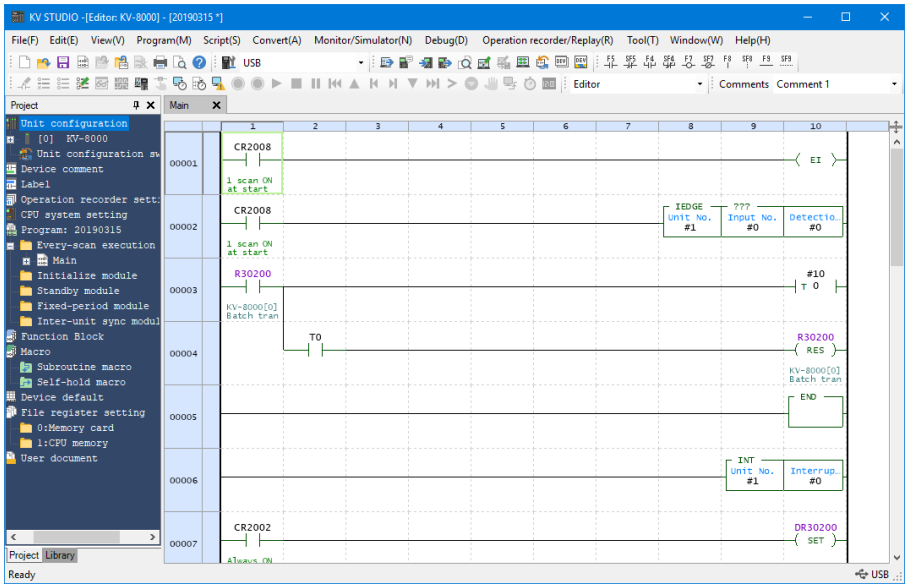
Alternative procedure

- Ctrl + D

Paste the copied mnemonic list into the "List edit" dialog box.
To paste the list, right-click the "List edit" dialog box, and then click "Paste."



4 **Click "Insert" to display the sample ladder program.**



Terminology

This manual uses the following terminology excluding some instances.

Terminology	Description
Programmable controller	This refers to electronic equipment that you can use to freely control equipment by changing programs. They are also called PLC (Programmable Logic Controller).
KV-8000 Series KV-8000	PLC KV-8000/8000A made by Keyence.
KV-7000 Series	PLC KV-7500/7300 made by Keyence.
KV-5000/3000 Series	PLC KV-5500/5000/3000 made by Keyence.
CPU unit	This refers to the KV-8000/7500/7300 and KV-5500/5000/3000.
Expansion unit	This refers to expansion I/O units and expansion special units, other than the CPU unit, that can be used with the KV-8000 Series.
Bus connection unit	This is a connection unit that is necessary when using expansion unit for 《KV-5000/3000 series》 with 《KV-8000》 series. The end unit is necessary at the right end when using the bus connection unit.
S-Unit	It is an expansion unit which corresponds to CPU inner bus of the KV-8000 series.
X-Unit	It is a high-performance expansion unit compatible with the KV-8000 series CPU inner bus.
CPU inner bus	It indicates high speed inner bus that corresponds to unit interruption from KV-8000 series CPU to the bus connection unit.
End unit	This refers to the unit that must be connected on the right of the other units when using KV-5000/3000 Series expansion units.
KV STUDIO	This refers to the software that supports the creation of KV-8000 Series programs.
Ladder program	This refers to programs that are created with KV STUDIO.

MEMO

1

OVERVIEW

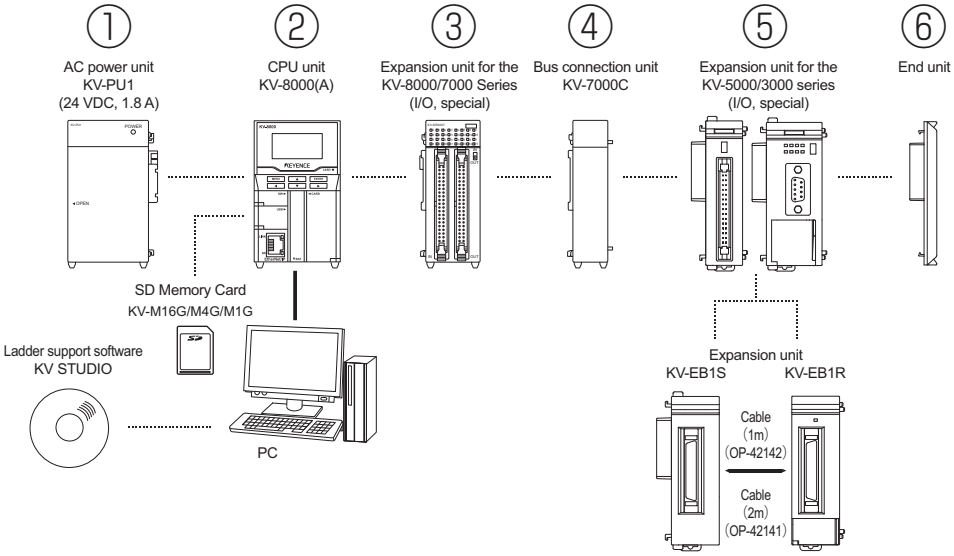
This chapter describes information such as the system configuration of the KV-8000 Series, communication channels with PCs, and the occupied areas of the units.

1-1 System Configuration..... 1-2
1-2 Unit Reservation..... 1-6

1-1 System Configuration

System Configuration Overview

Below shows the system configuration the KV-8000 Series is used in.
Be sure to turn each unit off before you connect them.



Point **The end unit is necessary at the right end only when using the bus connection unit, expansion unit of KV-5000/3000 series.**

(1) Power Units

■ Using 100 - 240 V AC

When the KV-8000 series runs at 100 - 240 V AC, the KV-PU1 AC power unit should be connected on the leftmost side. Each CPU unit should only be connected with one KV-PU1.

"2-3 AC Power Unit"

■ Using 24 V DC

When the KV-8000 series runs at 24 V DC, the KV-PU1 power unit is not used. Power (24 V DC) is supplied via the KV-8000 DC power input terminal block.

"2-2 CPU Unit"

(2) CPU Units

The AC power unit can be connected to the left side of CPU unit and expansion input/output unit and expansion special unit, KV-7000C can be connected to the right side. The end unit must be connected when expansion unit for KV-5000/3000 series is connected using the bus connection unit to the right side.

Reference

The end unit is necessary even when the expansion unit for KV-5000/3000 series is not connected when the bus connection is connected.

"2-2 CPU Unit"

The access window allows you to monitor device values, and read and write ladder programs and device data from/to SD memory cards or user memory.

"Chapter 5 CPU BUILT-IN FUNCTIONS"

With the KV-8000 Series, you can improve programming efficiency by using functions such as modules, function blocks, macros, local devices and KV scripts.

"Chapter 4 LADDER PROGRAMMING"
"Script Programming Manual"
"KV STUDIO User's Manual"

(3) KV-8000/7000 Series Expansion Units

The expansion special unit for KV-8000/7000 series will always be connected to the right side of CPU unit and left side of KV-7000C. (When the expansion unit of KV-5000/3000 series is not used, it is not necessary to connect KV-7000C.)
You can connect up to 16 KV-8000/7000 Series expansion units per CPU unit.

"2-4 Expansion I/O Unit of KV-8000/7000 series (KV-SIR32XT)"
"Expansion Special Unit User's Manuals"

(4) Bus Connection Units

Install a bus connection unit when using KV-5000/3000 Series expansion I/O units and expansion special units with the KV-8000 Series. Always attach the end unit to right side when using KV-7000C.

"2-8 Bus Connection Unit"

(5) KV-5000/3000 Series Expansion Units

When using expansion input/output unit and expansion special unit for KV-5000/3000 series by connecting to KV-8000 series, be sure to always use it by attaching to right side of bus connection unit. Be sure to connect an end unit on the right side of the right most expansion unit.
When two sets of expansion units are used, a total of 48 KV-8000/7000 Series expansion units can be connected. (excluding some expansion units)

"Expansion Special Unit User's Manuals for the KV-5000/3000 Series"

(6) End Units

Please connect to bus connection unit, expansion input output unit for KV-5000/3000 series and to the right side of expansion special unit. An error will occur when it is not connected and will not operate.

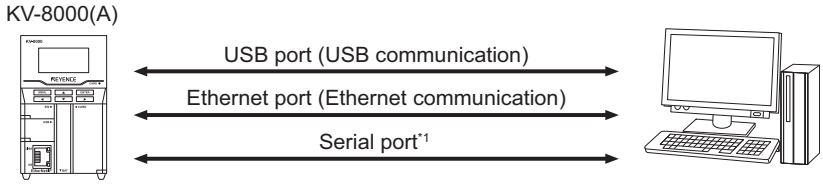
Point

The end unit is not necessary when CPU unit and expansion unit for KV-8000/7000 series is connected.

Connecting to Communication Devices

This section describes the communication channels that are used to perform communications between the CPU unit and a PC or touch panel.

■ When communicating with a PC

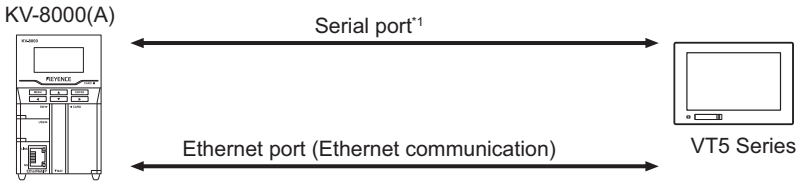


*1 A converter should be equipped separately on the serial communication unit and the PC side.

Point

Communications via VT, EtherNet/IP™ or other networks is also possible.
For details, see the "KV STUDIO User's Manual."

■ When communicating with a touch panel



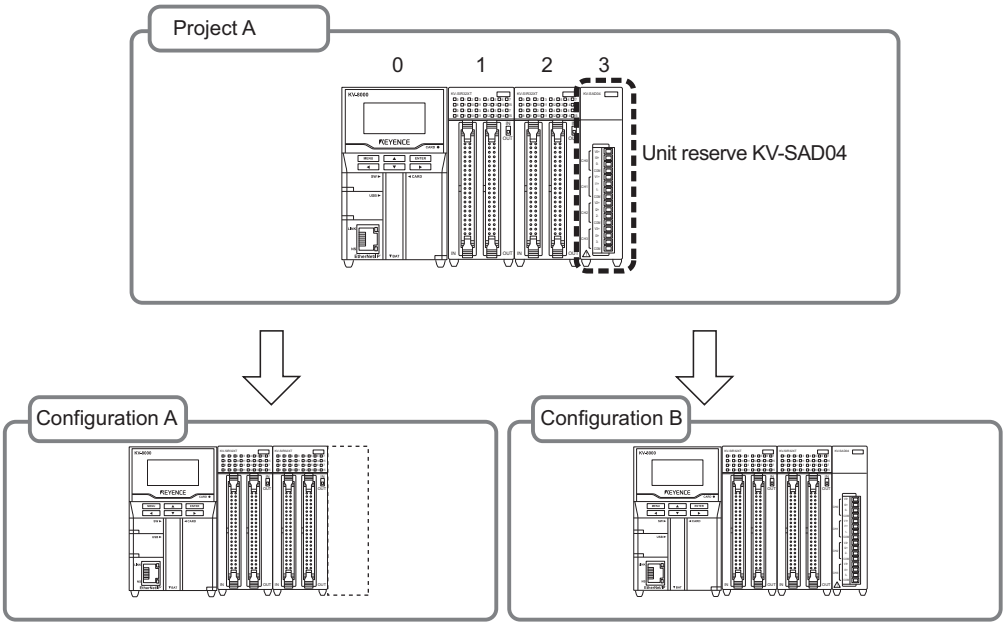
*1 A converter should be equipped separately on the serial communication unit and the PC side.

Point

- **A USB driver must be installed to use the USB interface.**
- **If your operating system is Windows 11/10/8/7, the driver will be automatically installed when you install KV STUDIO.**

1-2 Unit Reservation

You can use the unit reservation function to use the same project to manage units even if the unit configuration is different. For example, even if you have an installation in which an expansion analog input unit is attached as an option, you can use the same project to operate the system. This makes project management easy.



Operation is possible

Operation is possible

Point

- **When you make a reservation for a unit, if the unit is present in the actual configuration, a unit configuration error will not occur, and the system will operate normally.**
- **Regarding units other than the ones that you are making reservations for, if the information that you have used the Unit Editor to configure differs from the information of the devices that are actually connected, a number of units error or a unit classification error will occur, and you will not be able to operate the system.**
- **If you use the number of a unit that is not present in the actual configuration (a reserved unit) as an operand in an instruction (Function block instruction, etc.), an error will not occur during conversion, but an operation error (a minor error) will occur during execution. You can use the following methods to avoid calculation errors.**
  - **Include the expansion unit presence relay in a logical AND with the execution conditions.**
  - **Configure the CPU system settings (the operation when an error occurs) so that the CPU operation is set to continuation (to ignore errors).**
- **When expansion unit for KV-5000/3000 series is being used, there is an expansion unit which requires power switch on /off in order to reflect the modification of the unit number.**

Reference

- You can use the expansion unit n presence relays (CR4001 onwards) to check whether the expansion unit for each unit number is actually connected.
- You can add expansion units to the right of reserved units.
- If you make consecutive reservations for multiple instances of the same unit to connect less units in the actual configuration than the total number of units that you have specified, the connected units operate with unit numbers assigned from the left. For example, if you make reservations for unit numbers 1, 2, and 3 as KV-SIR32XT, but only use two KV-SIR32XT units, the connected units will be recognized as unit numbers 1 and 2 when they operate (you cannot have a gap between unit numbers such as assigning unit numbers 1 and 3).

2

SPECIFICATIONS

It describes specification and wiring of the CPU unit, AC power unit, expansion I/O unit, bus connection unit of KV-8000 series.

2-1 Series Common Specifications 2-2
2-2 CPU Unit 2-5
2-3 AC Power Unit.....2-11
2-4 Expansion I/O Unit of KV-8000/7000 series (KV-SIR32XT) 2-14
2-5 Expansion Input Unit 2-18
2-6 Expansion Output Unit 2-25
2-7 Extension I/O Unit 2-34
2-8 Bus Connection Unit 2-40
2-9 Expansion Unit 2-42

2-1 Series Common Specifications

This section provides the specifications of the KV-8000 Series and describes the operation of the KV-8000 Series when the power is turned off.

General Specifications

Item	Specification	
System configuration	When using a system configuration that includes KV-5000/3000 Series Expansion Units	When using a system configuration that uses only KV-8000/7000 Series Expansion Units
Power supply voltage	24 V DC (±10%)	24 V DC (-15% +20%)
Surrounding air temperature*1	0 to +50°C (no freezing)	0 to +55°C (no freezing)
Relative humidity	10 to 95% (no condensation)	5 to 95% (no condensation)
Storage temperature	-20 to +70°C	-25 to +75°C
Storage humidity	10 to 95% (no condensation)	5 to 95% (no condensation)
Operating atmosphere	As little dust and corrosive gases as possible	
Operating altitude	2000 m or below	
Pollution degree	2	
Overvoltage category	Ⅰ（Ⅱ when using KV-PU1）	
Noise immunity	1500Vp-p or above, pulse width 1μs, 50 ns (by noise simulator) Conforms to IEC standards (IEC61000-4-2/3/4/6)	
Withstand voltage	1500 V AC for 1 minute, between power supply terminal and I/O terminals and between all external terminals and case	
Insulation resistance	50 MΩ or more (500 V DC megger used to perform measurements between power terminal and input terminals, and between all external terminals and case)	

Vibration resistance*2	JIS B 3502 Conforms to IEC61131-2	Intermittent vibration			Number of scans
		Frequency	Acceleration	Amplitude	10 times (100 minutes) in each of the X, Y, and Z directions
		5 to 9 Hz	-	3.5 mm	
		9 to 150 Hz	9.8 m/s^2	-	
		Continuous vibration			
		Frequency	Acceleration	Amplitude	
		5 to 9 Hz	-	1.75 mm	
		9 to 150 Hz	4.9 m/s^2	-	

Shock resistance	Acceleration: 150 m/s^2. Application time: 11 ms. Triple in each of the X, Y, and Z directions.
Consumption current	CPU unit : 400 mA or below Bus connection unit : 30mA or below
Weight	CPU unit : Approx. 340 g Bus connection unit: Approx. 130 g KV-B1 (Battery): Approx. 10 g

*1 Indicated on the lower, central part of the unit in the industrial control panel.
*2 When installing DIN rail.

Performance Specifications

Item			Specification	
Calculation control method			Program storage method	
I/O control method			Refresh method	
Program language			Expanded ladder, KV script, mnemonic, ST Language	
Number of instructions			Base instructions: 80 kinds and 181 instructions Application instructions: 55 kinds and 73 instructions Calculation instructions: 128 kinds and 327 instructions Expansion instructions: 78 kinds and 133 instructions Total: 341 kinds and 714 instructions	
Instruction execution speed	Base instructions		0.96 ns minimum	
	Application instructions		5.75 ns minimum	
CPU memory capacity (project capacity + user memory capacity)			64MB	
Program capacity			Approx. 1500 k steps	
Maximum number of attachable I/O units*2			16 (KV-8000/7000 Series expansion units only) 48 (KV-8000/7000 Series expansion units, KV-5000/3000 Series expansion units (when using an expansion unit (KV-EB1))	
Maximum number of I/O points			When epxanded: 3072 points max. (When using two KV-EB1S/KV-EB1R units and 64-point units)	
Input relay		R	Total: 32,000 points 1 bit（R00000 to R199915）	
Output relay		R		
Internal auxiliary relay		R		
Link relay		B	32,768 points	1 bit (B0000 to B7FFF)
Internal auxiliary relay		MR	64000 points	1-bit（MR000000 to MR399915）
Latch relay		LR	16000 points	1-bit（LR00000 to LR99915）
Control relay		CR	1280 points	1-bit（CR0000 to CR7915）
Timer		T	4,000 points	32-bit (T0000 to T3999)
Counter		C	4,000 points	32-bit (C0000 to C3999)
Data memory		DM	65,535 points	16-bit (DM00000 to DM65534)
Expansion data memory		EM	65,535 points	16-bit (EM00000 to EM65534)
File register	Current bank	FM	524,288 points 16-bit	(FM00000 to FM32767 x 16 banks)
	Consecutive number mode	ZF		(ZF000000 to ZF524287)
Link register		W	32,768 points	16-bit (W0000 to W7FFF)
Temporary memory		TM	512 points	16-bit (TM000 to TM511)*3
Index register		Z	23 points	32-bit (Z1 to Z12)
Control memory		CM	7,600 points	16-bit (CM0000 to CM7599)
Number of comments / labels / variables stored	Device comments		Approx. 224,000	
	Labels*4		Approx. 285,000	
	Variable*4		Approx. 100,000*6	
Power off hold function	Program memory		Flash ROM can be rewritten 10,000 times	
	Devices*1		Nonvolatile RAM	
	Calender timer		Backup capacitor about 15 days (@25 degrees) (When using KB-B1 (battery) about 5years (@25 degrees))	
Self-diagnosis function			CPU error, RAM error, and other problems	

*1 You can set the target device by clicking "CPU system setting" and then "Power off holding" in KV STUDIO.
*2 This may be limited by the maximum number of installable I/O units.
*3 TM000 to TM003 are the areas to store the result of calculation instruction.
*4 Labels for CPU function version 1.0 and variables for CPU function version 2.0 or later.
*5 Z11～Z23 cannot be used since they are reserved for the system.
*6 Depending on the number of characters in the variable name, 100000 may not be defined.

Operation during Power Outages

Low supply voltage

- If the supply voltage decreases, the CPU unit stops, and the output is turned off.

Automatic recovery

- When the supply voltage recovers, operation is automatically restarted.

Temporary power outages

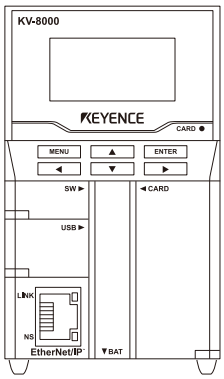
- The unit will stay on during momentary power outages that are shorter than the following. AC power: 20 ms, 24 V DC: 10 ms, 20.4 V DC: 5 ms.

Package Contents

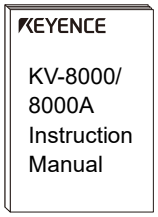
The package contains the following items. Before you use the product, make sure that the package contains everything it is supposed to contain.

■ Products packaged with KV-8000(A)

CPU unit
KV-8000(A)



Instruction
Manual



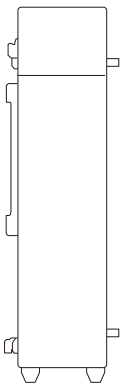
- Expansion unit connector cover
- Power unit connector cover
- Battery cover

* 3 types of covers are included in the package by installing on the CPU unit.

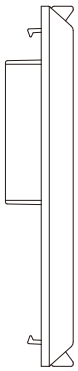
When using the extension unit for the KV-5000/3000 series, the bus connection unit KV-7000C should be equipped separately.
The items packaged with KV-7000C are as follows.

■ Products packaged with KV-7000C

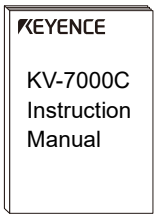
Bus connection unit



End unit



Instruction
Manual

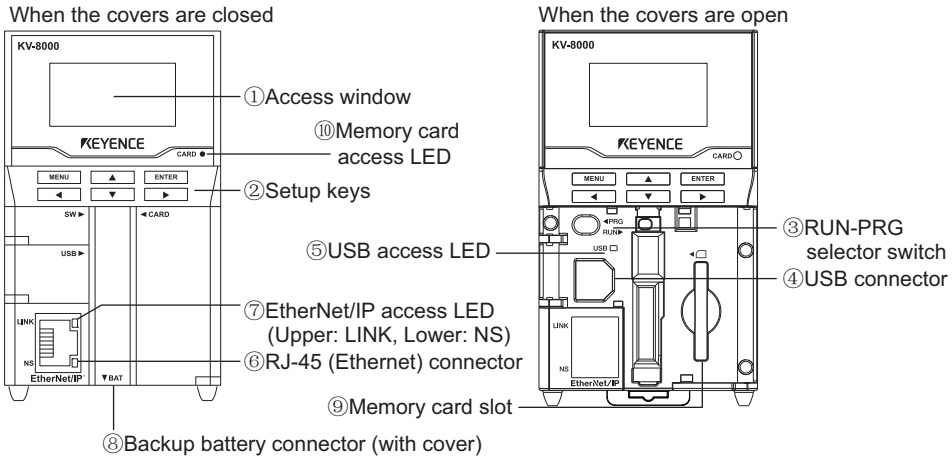


The utmost care has been exercised in packing the items, however, in the event that items malfunction or are damaged, contact your KEYENCE office.

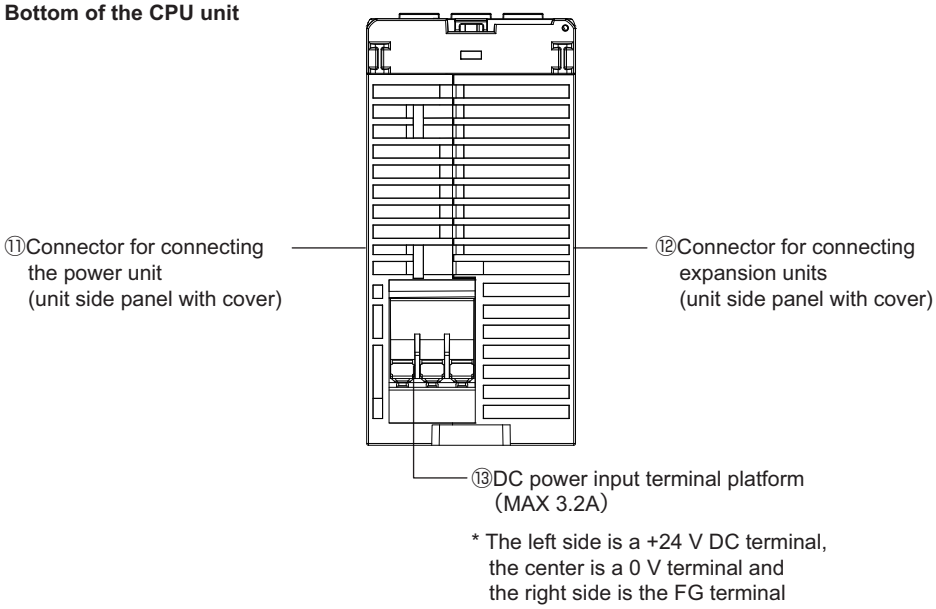
Names and Functions of Parts

■ KV-8000(A)

CPU unit front panel



Bottom of the CPU unit

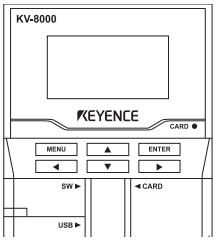


No.	Name	Function
①	Access window	Configure unit settings and monitor the unit. The backlight is used to indicate operation status. RUN: White PRG: Red
②	Setup keys	Used for access window operations.
③	RUN-PRG selector switch	Switch between RUN mode and PRG mode.
④	USB connector	For connecting with USB.
⑤	USB access LED	The lamp lights when data is received from/sent to a USB.
⑥	RJ-45 (Ethernet) connector	Connector used to connect the network cable (RJ-45 modular connector) (100BASE-TX).
⑦	EtherNet/IP access LED (Upper: LINK, Lower: NS)	LINK…Indicates the link status of the other party's machine. Green lamp on: link established Green lamp flashes: sending/receiving data Lamp off: link not established NS…Indicates data sending/receiving status Green lamp on: connection established Green lamp blinking: connection not established. Red lamp on: IP address overlapping Red lamp blinking: timeout has occurred on one or more connection Lamp off: IP address not set
⑧	Backup battery connector (with cover)	Used for installing KV-B1.
⑨	Memory card slot	Slot for inserting memory card.
⑩	LED indicating memory card access	Lit while the memory card is being accessed.
⑪	Connector for connecting the power unit (unit side panel with cover)	Connector for connecting the KV-PU1 (power unit).
⑫	Connector for connecting expansion units (unit side panel with cover)	Connector for connecting expansion units.
⑬	DC power input terminal block	It is a terminal block (M3 screw) for 24 V DC power input. (MAX3.2A)

NOTICE	**Do not connect USB cables to RJ-45 (Ethernet) connectors. (Failure to heed this caution may lead to damage.)**

RUN-PRG Selector Switch

There are two modes for CPU unit: RUN mode and PRG mode.
Mode can be selected through the RUN-PRG Selector Switch shown on the right.



RUN mode : run program.
PRG mode : stop program, all outputs OFF.

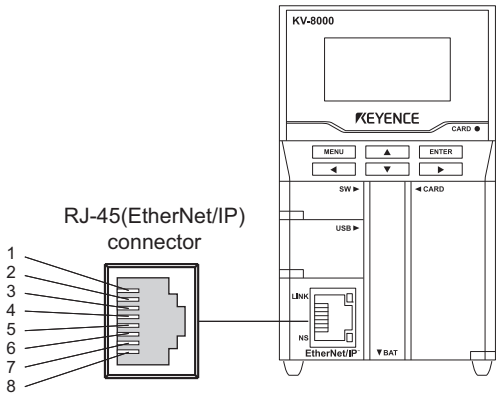
Point **In case of serious error, even if the switch is in RUN position, it will be switched to PRG mode.**

Reference When the switch is in RUN position, you can switch between PRG mode and RUN mode via KV STUDIO.

Wiring and Internal Circuit Diagram

■ KV-8000(A)

Ethernet communication can be realized via external equipment such as RJ-45 connector and PC or touch panel, etc. Besides, EtherNet/IP communication can be also realized.
"5-8 Ethernet/EtherNet/IP Communication", Page 5-110



Pin No.	MDI signal	Signal function
1	TD+	Send data(+)
2	TD-	Send data(-)
3	RD+	Receive data(+)
4	-	-
5	-	-
6	RD-	Receive data(-)
7	-	-
8	-	-

■ Ethernet communication specifications

Item	Specification
Baud rate	100 Mbps (100 BASE-TX)
Transmission medium	STP or category 5 or better UTP *1
Maximum cable length	100 m
Maximum number of connected hub stages	2
MDI/MDI-X auto switching function	available
Display	LINK（green） It displays link status with the other device. Light on：Link established.*4 Light off：Link unsuccessful. NS（green, red）*5 It displays the communication status of network. Light off：IP address is not configured. Green light on：More than 1 connection established. Green light blinking：Not a single connection is established. Red light on：IP address is overlapping. Red light blinking：1 or more connection (tag configuration) has timed out, of which KV-8000 is going to be the target *6.

KV-8000(A) conforms to the IEEE802.3 standard.

*1 STP refers to shielded twisted pair cable
UTP refers to unshielded twisted pair cable

*2 The maximum cable length is considered to be the distance between CPU unit and hub.

*3 When an Ethernet switch is used, there is no restriction on the number of cascading hubs.

*4 The established link is considered as the status in which power of communicating device is on, and is also properly connected to the Ethernet with cable.

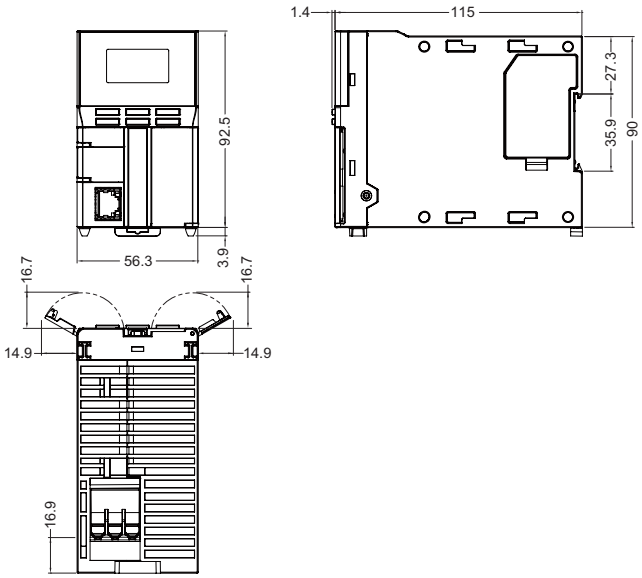
*5 The NS(green, red) lamp will turn on (during self test) in the order of NS green, NS red when power is inserted.

*6 The side in which the connection will open is the originator, and the side in which it will be opened is the target.

Dimensions

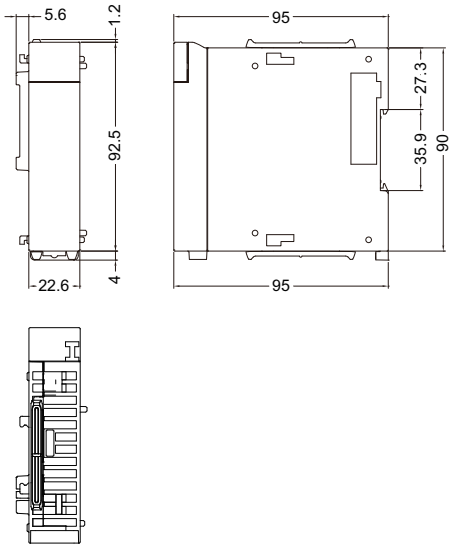
■ CPU Unit

Unit: mm



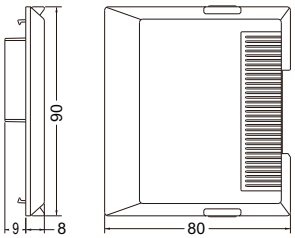
■ Bus connection unit

Unit: mm



■ End unit

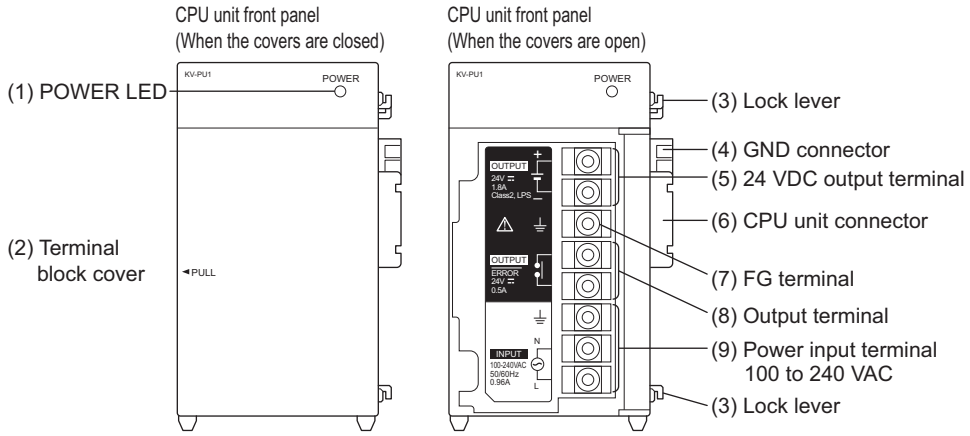
Unit: mm



2-3 AC Power Unit

This sections describes name, function and technical specification of each part of KV-PU1 AC power unit.

Names and Functions of Parts



No.	Name	Function
(1)	POWER LED	It will turn ON (green light) when 24VDC is being provided.
(2)	Terminal block cover	The protective cover of the terminal block.
(3)	Lock lever	This lever locks the connected units together.
(4)	GND connector	This connects connected units to GND.
(5)	24 VDC output terminal	24 V DC output terminal for extra power supply. A total of 1.8 A including (M3.5 screw) main power.
(6)	CPU unit connector	Connector for connecting CPU unit
(7)	FG terminal	FG terminal (M3.5 screws)
(8)	Output terminal	Terminal block used for error output. (M3.5 screws) It turns ON (includes during PRG) during normal operation, and turns OFF when serious error occurs.
(9)	Power input terminal 100 to 240 VAC	Connect this connector to a 100 -240 V AC (50/60Hz) power outlet. (M3.5 screws)

NOTICE	• **When over current (the POWER LED is OFF) occurs, please temporarily turn the power off and eliminate the reason for the fault.** • **When overvoltage occurs, output stops (the POWER LED turns off).** **Turn the power OFF for more than 3 minutes and then turn it back on again.** • **When using KV-PU1, do not supply power from the power terminal of CPU unit.**

Point

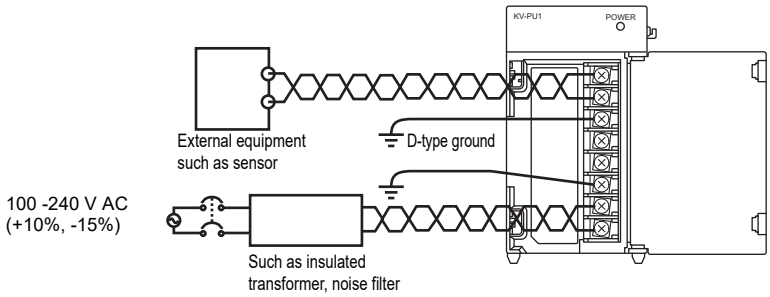
KV-PU1 must be connected to the left side of CPU unit.
In addition, when using KV-PU1, the power supply terminal of the CPU unit cannot be used as the supply source of 24V power.

Specification

Item		Specification
Input power voltage		100 -240 V AC(+10%,-15%)(50/60Hz)
Output voltage		24 V DC±10%
Output capacity		1.8 A (total for power of all units and operation power)
Internal consumption current		0.96A or below
Dwell time		20ms or below (with rated input and output condition)
Start-up time		Max. 3 s
Output mode		Relay (B contact)
Rated load		24 V DC, 0.5 A
ON resistance		50 mΩ or less
Response time	OFF → ON	10 ms or less
	ON → OFF	5 ms or less
Relay life		Electrical: 100,000 cycles or more (20 cycles/min) Mechanical: 20 million cycles or more
Relay replacement		Not possible
Weight		Approx. 300 g

How to Wire

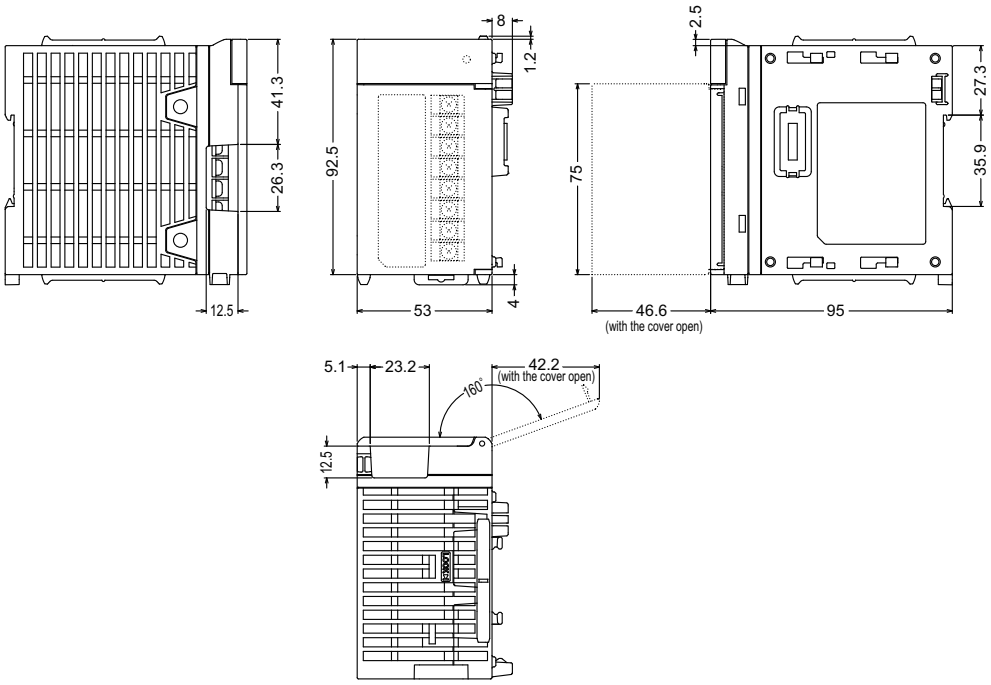
KV-PU1 AC power unit shall be wired as follows.



Point

- **To reduce line noise, please connect 1:1 insulated transformer and noise filter.**
- **To reduce induction influence, please twist the cable.**
- **The KV-PU1 has a built-in fuse as a power cutoff device.**
- **The primary and secondary circuits are reinforced and insulated in the equipment.**

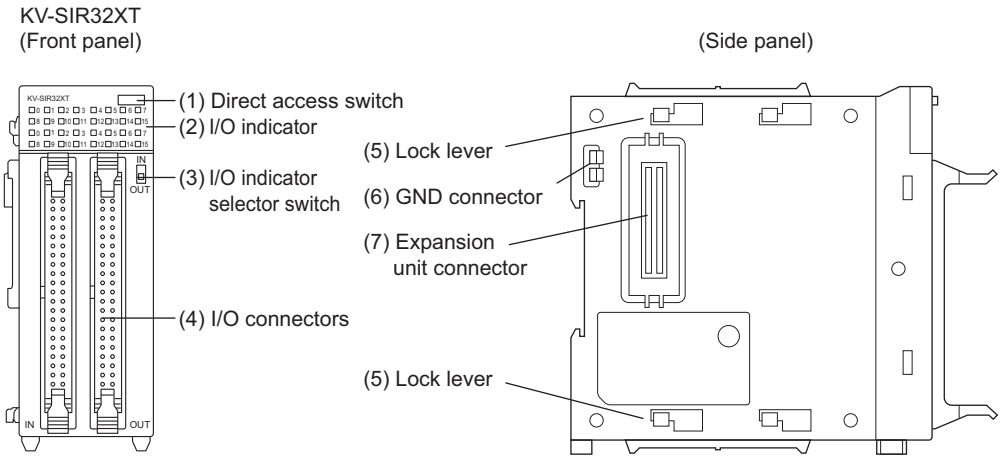
Dimensions


41.3
26.3
12.5
8
1.2
92.5
53
4
2.5
75
27.3
35.9
46.6
(with the cover open)
95
5.1
23.2
42.2
(with the cover open)
160°
12.5

2-4 Expansion I/O Unit of KV-8000/7000 series (KV-SIR32XT)

This section describes the names and functions of the parts of the KV-SIR32XT Expansion I/O Units. It also provides the specifications of the units.

Names and Functions of Parts



Number	Name	Function
(1)	Direct access switch	After setting the switch to ON, you can monitor expansion unit status from the CPU access window. In addition, you can also see the status of the expansion unit through the color of the switch. Lit in green: Normal operation Blinking in green: The switch is turned ON while the unit is selected and the LED is lit in green. Lit in red: A unit error has occurred (with some exceptions*1). Blinking in red: The switch is turned ON while the unit is selected and the LED is lit in red.
(2)	I/O indicators	These indicate the input/output status of the unit. Lit in red: Conducting Unlit: Not conducting
(3)	I/O indicator selector switch	This is a switch for switching between input and output indication. IN: Indicates unit input status. OUT: Indicates unit output status.
(4)	I/O connectors	These connectors are used to connect I/O units.
(5)	Lock lever	This lever locks the connected units together.
(6)	GND connector	This connects connected units to GND.
(7)	Expansion unit connector	This is used to connect expansion units for the CPU unit or the KV-8000/7000 Series.

*1 In the following instances, the LED will light in red even if a unit error has not occurred.
When a unit number error, unit type error or other error occurs at power on and CPU unit information is configured for one CPU, the CPU unit may have caused a system error.
In this case, an error will be displayed in the access window.

* An optional 40-pin connector for connecting the KV-SIR32XT is available.

Specifications

Model			KV-SIR32XT (32 + 32 points)	
External connection method			Connector (MIL standard)	
Input	Number of inputs		32	
	Input mode		24 V DC mode	5 V DC mode*2 (with overvoltage protection function*3)
	Maximum input voltage		28.8 V DC	6.0 V DC
	Rated input voltage		24 V DC/5.1 mA	5 V DC/8.8 mA
	Minimum ON voltage		19 V	3.5 V
	Maximum OFF current		1.5 mA	-
	Maximum OFF voltage		-	1.5 V
	Common method		16 points/1 common (2 terminals)*1	
	Input time constant			
	Input impedance		4.4 kΩ	350 Ω
Output	Number of outputs		32	
	Output mode		MOSFET(N-ch) (with overcurrent protection function)*4	
	Rated load		30 V DC, 0.2 A (1.6 A/common)	
	Leakage OFF current		100 μA or less	
	Residual ON voltage		0.5 V DC or less	
	Common method		16 points/1 common (2 terminals)*1	
	Runtime	OFF → ON	1 μs or less (with a load between 5 mA and 200 mA)	
		ON → OFF	5 μs or less (with a load between 5 mA and 200 mA)	
Internal current consumption			130 mA or less	
Weight			Approx. 190 g	

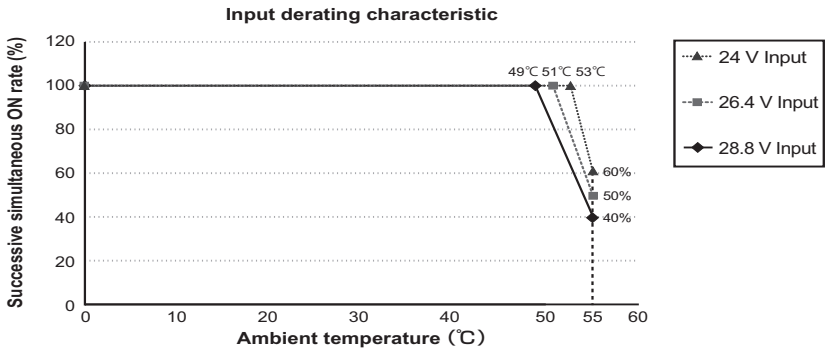
Constant configuration during input	TYP	MAX
1 μs	1.2 μs (0.7 μs)	2.6 μs
10 μs	2.4 μs (1.9 μs)	3.9 μs
20 μs	10 μs	12 μs
100 μs	73 μs	82 μs
500 μs	420 μs	450 μs
1 ms	840 μs	900 μs
5 ms	4.5 ms	4.8 ms
10 ms	9.0 ms	9.6 ms
50 ms	45 ms	48 ms

* during ONgOFF inside the parentheses

*1 The 4 COM consisting of input COM (COM0, COM1), output COM (COM2, COM3) are all independent. The 2 terminals in each COM is short-circuited inside.
*2 5VDC mode (configured with unit editor) can only be used from IN100 to IN115.
*3 When it detects overvoltage, it turns OFF all the input (IN100 to IN115) inside the common until the power is turned ON,OFF, or until PRG → RUN is executed.
*4 If it detects even a single point of overvoltage, all of the output within common (COM2：OUT200 to 215, COM3：OUT300 to 315) will repeat protection operation (output OFF) and automatic recovery until the cause is abolished.

NOTICE

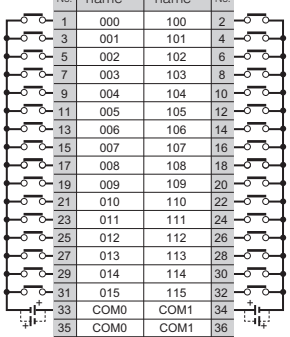
- **The unit may break down when the input point which was simultaneously turned ON has gone past the specification range.**
- **The relation between ambient temperature of KV-SIR32XT and the successive simultaneous ON rate will be displayed in the graph below.**



Wiring and Internal Circuit Diagrams

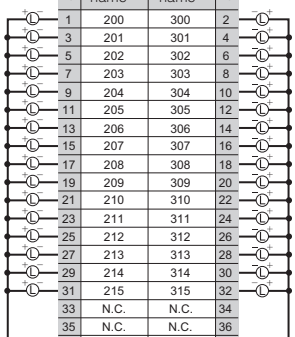
■ KV-SIR32XT I/O wiring diagram

Left side connectors



Pin No.	Signal name	Signal name	Pin No.
1	000	100	2
3	001	101	4
5	002	102	6
7	003	103	8
9	004	104	10
11	005	105	12
13	006	106	14
15	007	107	16
17	008	108	18
19	009	109	20
21	010	110	22
23	011	111	24
25	012	112	26
27	013	113	28
29	014	114	30
31	015	115	32
33	COM0	COM1	34
35	COM0	COM1	36
37	N.C.	N.C.	38
39	N.C.	N.C.	40

Right side connectors

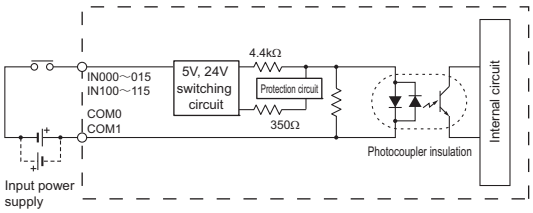


Pin No.	Signal name	Signal name	Pin No.
1	200	300	2
3	201	301	4
5	202	302	6
7	203	303	8
9	204	304	10
11	205	305	12
13	206	306	14
15	207	307	16
17	208	308	18
19	209	309	20
21	210	310	22
23	211	311	24
25	212	312	26
27	213	313	28
29	214	314	30
31	215	315	32
33	N.C.	N.C.	34
35	N.C.	N.C.	36
37	COM2	COM3	38
39	COM2	COM3	40

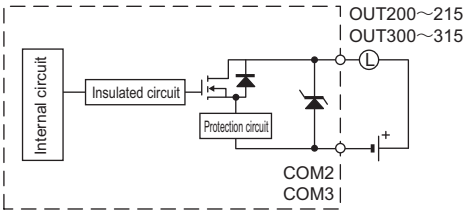
* The COM0 to COM3 are all independent.

* The 2 COM terminal are short-circuited inside.

■ KV-SIR32XT internal circuit diagram (input circuit)

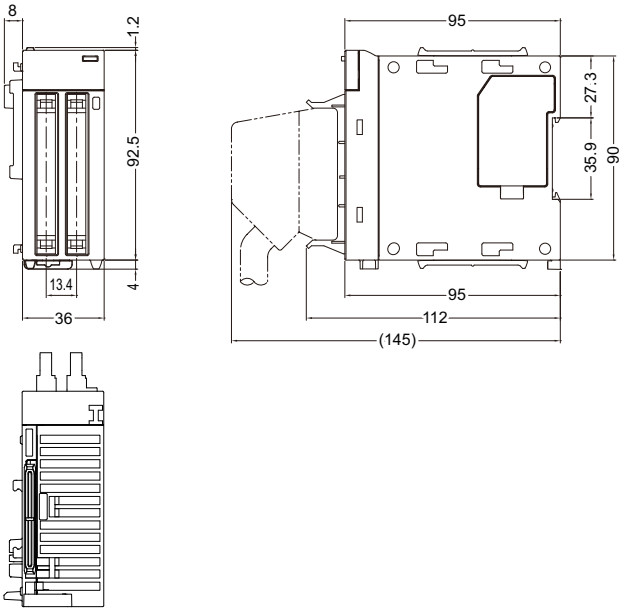


■ KV-SIR32XT internal circuit diagram (output circuit)



Dimensions

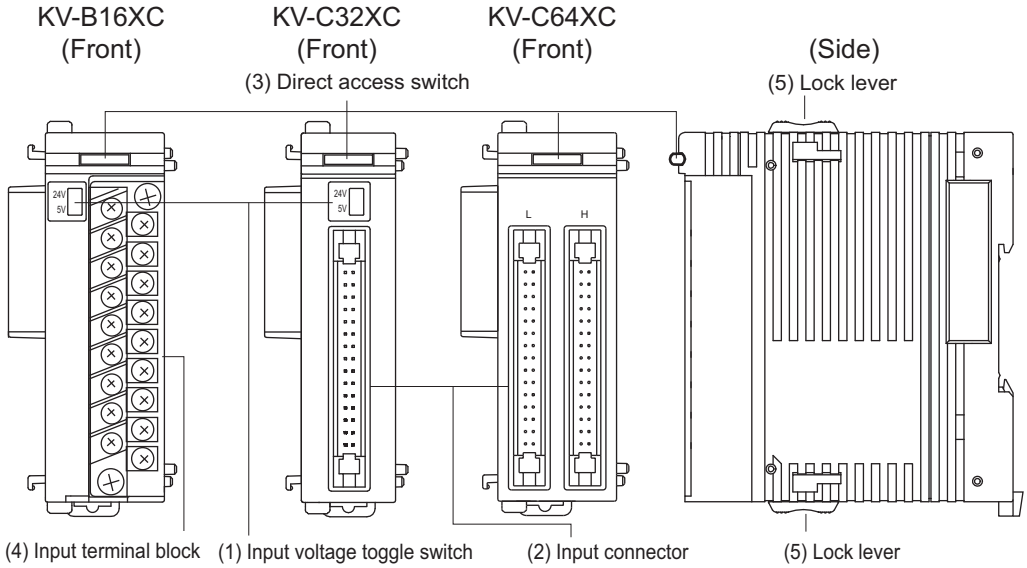
KV-SIR32XT



2-5 Expansion Input Unit

This chapter describes name, function and technical specification of every part of KV-5000/3000 series expansion input unit (KV-B16XC/C32XC/C64XC).

Name and Function of Each Part



No.	Name	Function
(1)	Input voltage selector switch	Toggle input voltage. 5V <=> 24V
(2)	Input connector	Input connector.
(3)	Direct access switch	After setting it to ON, you can monitor status of expansion unit via CPU access window. In addition, you can also know the status of the expansion unit through the color of the switch. Lit (green) : Normal operation (with some exceptions*1) Green LED blinking: when a unit is selected while the green LED is lit and the switch is turned on. Red LED lit: When a unit error has occurred (with some exceptions*2) Red lamp off: when turning ON switch and selecting unit under red lamp on status
(4)	Input terminal block	Input terminal block. Can be removed. (M3 screws)
(5)	Lock lever	Locks the connection of the unit.

*1 This LED lights (green) in the following instances even if operation is not normal.
(1) When the number of connected units differs from the unit information on the CPU Unit when the power is turned ON.
(2) When the model type connected differs from the unit information on the CPU Unit when the power is turned ON.
In this case, an error will be displayed in the access window.

*2 Red light will be ON in the following instance even if the unit is not in error state:
(1) When the power is turned on, and the CPU unit information becomes the configuration with one CPU.
In this case, an error will be displayed in the access window.

* KV-C32XC/C64XC 34-pin connector is supplied as an option.

Reference

Direct processing can be realized. Although KV-B16XA/C32XA/C64XA/C64XB can be also used, direct processing can not be realized.

"Direct processing", Page 4-83

Specification

Model	KV-B16XC (16 point)		KV-C32XC (32 point)		KV-C64XC (64 point)
External connection mode	Removable terminal block		Connector (MIL standard)*1		Connector (MIL standard)*1
Input terminal	24 VDC mode	5 VDC mode	24 VDC mode	5 VDC mode	24 VDC mode*4
Max. input voltage	26.4 VDC				
Rated input voltage	24 VDC, 5.3mA	5 VDC, 1mA	24 VDC, 5.3mA	5 VDC, 1mA	24 VDC , 4.1mA
Min. ON voltage	19V	3.5V	19V	3.5V	19V
Max. OFF current	1.5mA	—	1.5mA	—	1.5mA
Max. OFF voltage	—	1.5V	—	1.5V	—
Common point mode	16 point/1 common (2 terminals)*2		32 point/1 common (2 terminals)*2		32point/1 common (2 terminals)×2*3
Input time constant (4 level conversion)					
Input impedance	4.3kΩ				5.6kΩ
Internal current consumption	15mA or less				25mA or less
Weight	approx. 120g		approx. 110g		approx. 140g

Constant setting during input	OFF->ON		ON->OFF	
	TYP	MAX	TYP	MAX
25μs	25μs	65μs	75μs	120μs
300μs	275μs	365μs	275μs	420μs
1ms	1ms	1.2ms	1ms	1.2ms
10ms	10ms	12ms	10ms	12ms

* 1 Conform to MIL-C-83503 standard

* 2 2 COMs for KV-C32XC and KV-B16XC, connected internally.

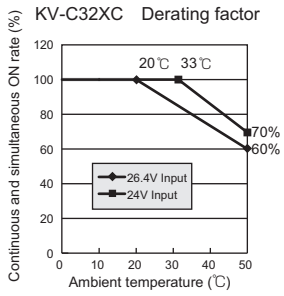
* 3 4 COMs for KV-C64XC, internal H side(2 points), L side(2 points), connected internally.

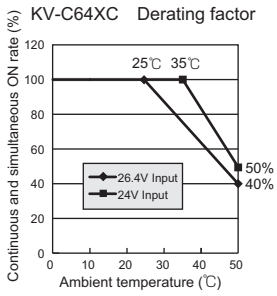
"KV-B16XC internal circuit diagram", Page 2-20 "KV-C32XC internal circuit diagram", Page 2-21
"KV-C64XC internal circuit diagram", Page 2-22

*4 All terminals support 2-wire connection (but 2-wire non-contact sensor EV Series not supported)

NOTICE

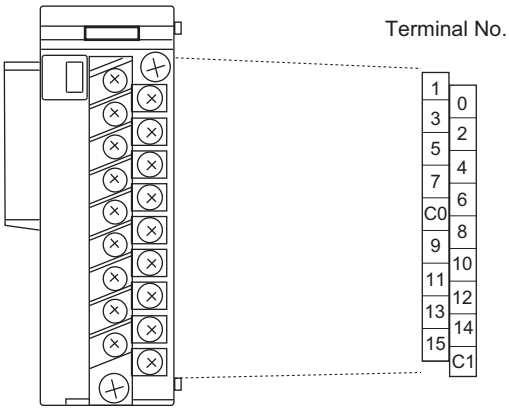
If the number of inputs in ON status exceeds the specified range, the unit may fail. Relationship between continuous simultaneous ON rate and ambient temperature of the KV-C32XC 2-wire terminal and KV-C64XC 2-wire terminal is shown as follows.



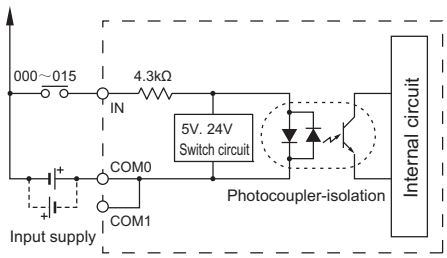


Wiring and Internal Circuit Diagram

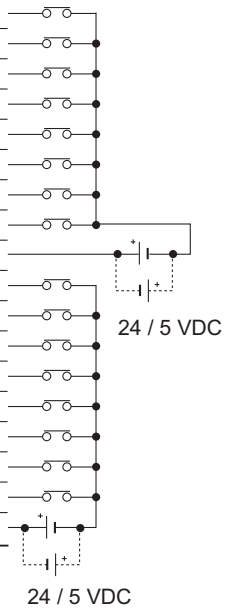
KV-B16XC Wiring diagram



KV-B16XC internal circuit diagram

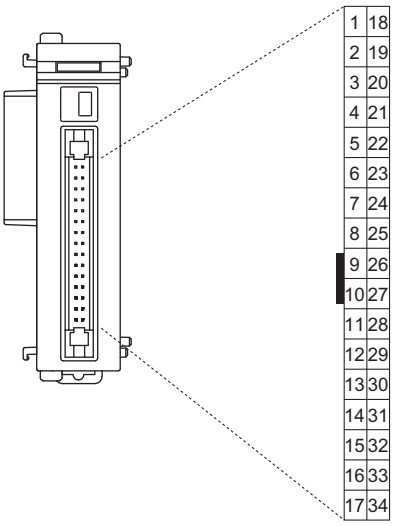


Terminal No.	Direction	Signal name
0	Input	000
1	Input	001
2	Input	002
3	Input	003
4	Input	004
5	Input	005
6	Input	006
7	Input	007
C0	Input	COM0
8	Input	008
9	Input	009
10	Input	010
11	Input	011
12	Input	012
13	Input	013
14	Input	014
15	Input	015
C1	Input	COM1

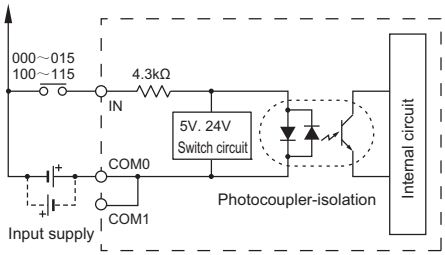


* COM circuit-shorted internally

KV-C32X Wiring diagram

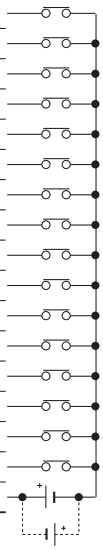


KV-C32XC internal circuit diagram



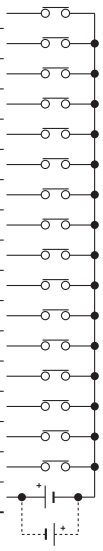
* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.

Pin No.	Direction	Signal name
1	Input	000
2	Input	001
3	Input	002
4	Input	003
5	Input	004
6	Input	005
7	Input	006
8	Input	007
9	Input	008
10	Input	009
11	Input	010
12	Input	011
13	Input	012
14	Input	013
15	Input	014
16	Input	015
17	Input	COM0



24 / 5 VDC

Pin No.	Direction	Signal name
18	Input	100
19	Input	101
20	Input	102
21	Input	103
22	Input	104
23	Input	105
24	Input	106
25	Input	107
26	Input	108
27	Input	109
28	Input	110
29	Input	111
30	Input	112
31	Input	113
32	Input	114
33	Input	115
34	Input	COM1

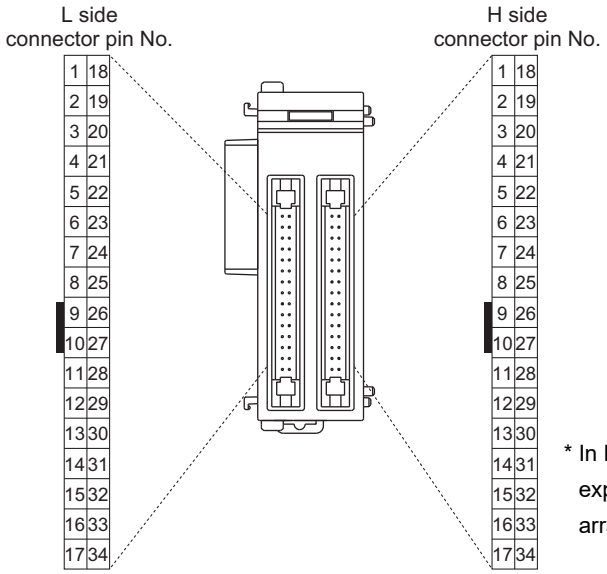


24 / 5 VDC

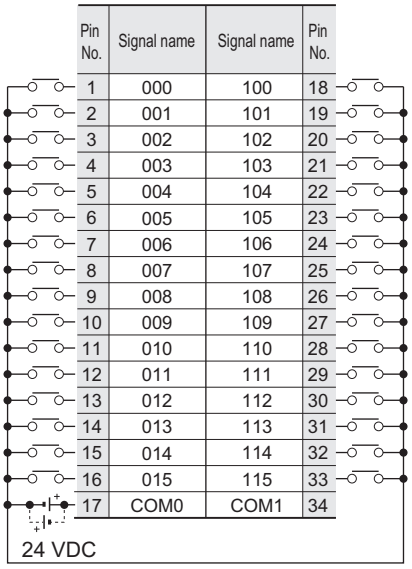
* COM circuit-shorted internally

NOTICE	**Input continuous and simultaneous ON rate should not exceed the range of technical specification. Otherwise, equipment damage may result.**

KV-C64XC Wiring diagram

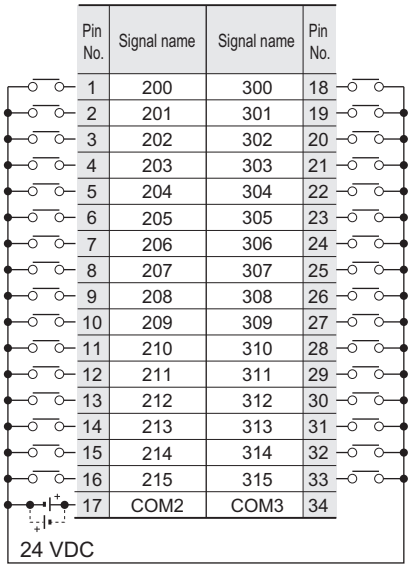


* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.



Pin No.	Signal name	Signal name	Pin No.
1	000	100	18
2	001	101	19
3	002	102	20
4	003	103	21
5	004	104	22
6	005	105	23
7	006	106	24
8	007	107	25
9	008	108	26
10	009	109	27
11	010	110	28
12	011	111	29
13	012	112	30
14	013	113	31
15	014	114	32
16	015	115	33
17	COM0	COM1	34

24 VDC

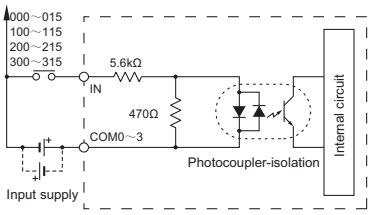


Pin No.	Signal name	Signal name	Pin No.
1	200	300	18
2	201	301	19
3	202	302	20
4	203	303	21
5	204	304	22
6	205	305	23
7	206	306	24
8	207	307	25
9	208	308	26
10	209	309	27
11	210	310	28
12	211	311	29
13	212	312	30
14	213	313	31
15	214	314	32
16	215	315	33
17	COM2	COM3	34

24 VDC

* COM0, COM1 circuit-shorted internally.
  COM2, COM3 circuit-shorted internally.
  (COM0, 1)-(COM2, 3) not circuit-shorted internally.

KV-C64XC internal circuit diagram

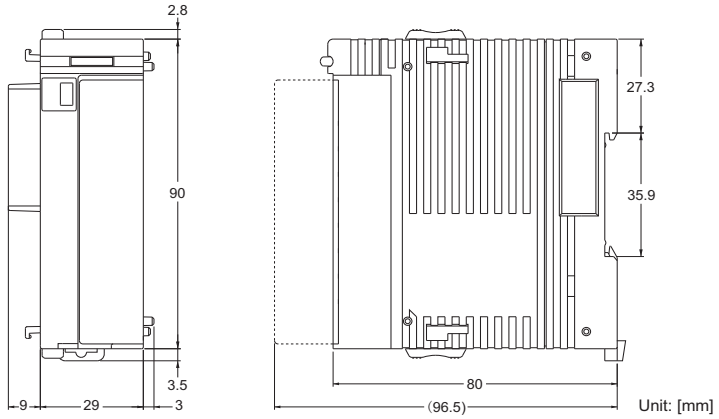


NOTICE	**Input continuous and simultaneous ON rate should not exceed the range of technical specification. Otherwise, equipment damage may result.**

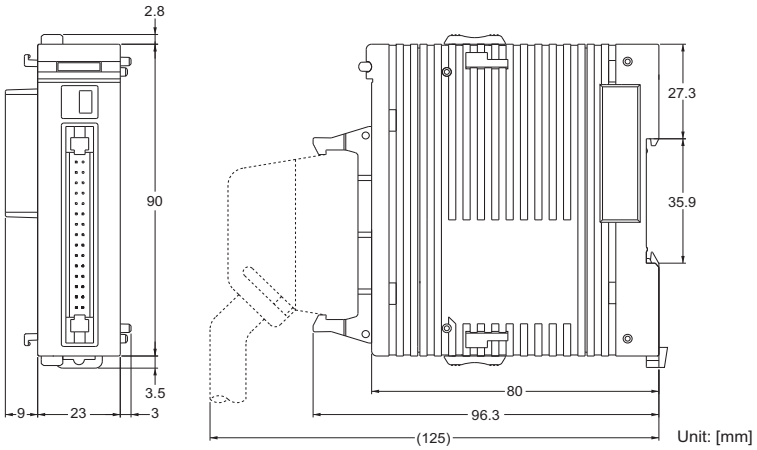
Point	**Please use OP- 42224 connector (34-pin) for 64-point unit.** **Do not use OP-23139.**

Dimensions

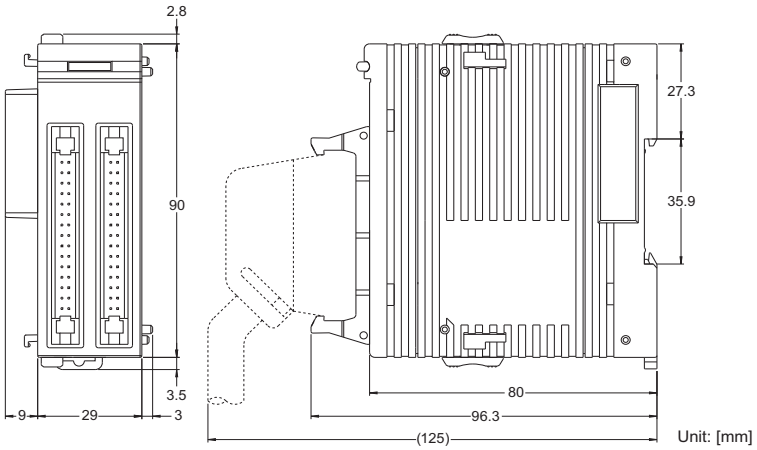
KV-B16XC



KV-C32XC



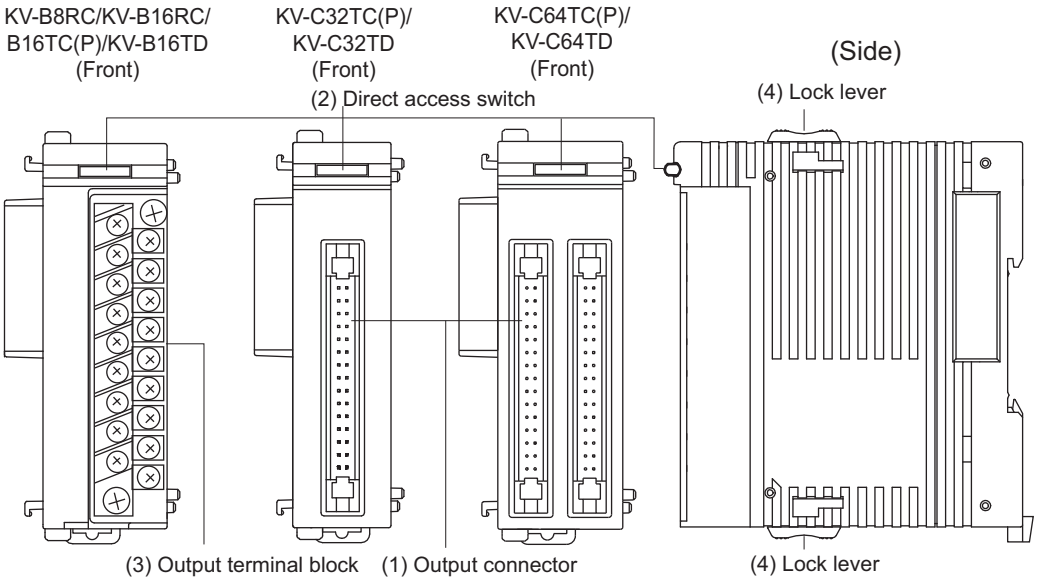
KV-C64XC



2-6 Expansion Output Unit

This section describes name, function and specification of each part of KV-5000/3000 series expansion input unit (KV-B8RC/KV-B16RC/KV-B16TC(P)/KV-C32TC(P)/KV-C64TC(P)/KV-B16TD/KV-C32TD/KV-C64TD).

Names and Functions of Each Part



No.	Name	Function
(1)	Output connector	Output connector.
(2)	Direct access switch	After setting the switch to ON, you can monitor the status of expansion unit through the CPU Access Window. In addition, you can also know the status of the expansion unit through the color of the switch. Lit (green) : Normal operation Green LED blinking: when a unit is selected while the green LED is lit and the switch is turned ON. Red LED lit: When a unit error has occurred (with some exceptions*1) Red light OFF: when a unit is selected while the red light is ON and the switch is turned ON.
(3)	Output terminal block	Output terminal block. Can be removed (M3 screws)
(4)	Lock lever	Locks the connection of the unit.

*1 Red light will be ON in the following instance even if the unit is not in error state:
(1) When the power is turned on, and the CPU unit information becomes the configuration with one CPU
In this case, an error will be displayed in the access window.

* KV-C32TC/C64TC/C32TCP/C64TCP/KV-C32TD/KV-C64TD 34-pin connector will be supplied as an option.

Reference

Direct processing can be realized. Although KV-B16RA/B16TA/C32TA(P)/C64TA(P) can also be used, direct processing can not be realized.

"Direct processing", Page 4-83

Specification

Type		KV-B8RC (8 point)	KV-B16RC (16 point)	KV-B16TCP (16 point)	KV-C32TCP (32 point)	KV-C64TCP (64 point)
Output mode		Relay		Transistor (PNP)		
External connection mode		Removable terminal block		Removable terminal block	Connector *1	
Rated load		250 VAC/30 VDC 2A (KV-B16RC is 8A/common point)		30 VDC 0.2A		
Leakage current in case of OFF		-		Below 100μA		
Remaining voltage in case of ON		-		Below 0.5V		
ON resistance		Below 50mΩ		-		
Common point mode		Independent	8 point/common point	16 point/ common point (2 terminals)*2	32 point/ common point (2 terminals)*2	64 point/ common point (4 terminals)*3
Response time	OFF→ON	Below 10ms		Below 10μs		Below 50μs
	ON→OFF			Below 200μs		
Internal consumption current		Below 65mA	Below 120mA	Below 30mA	Below 55mA	Below 100mA
Weight		About 160g	About 190g	About 130g	About 100g	About 140g
Relay life		Electrical above 100,000 times (20 times/min) Mechanical above 20 million times		-		
Relay replacement		Unallowed		-		

*1 According to MIL-C-83503

*2 Two common points are shorted in KV-B16TCP, C32TCP.

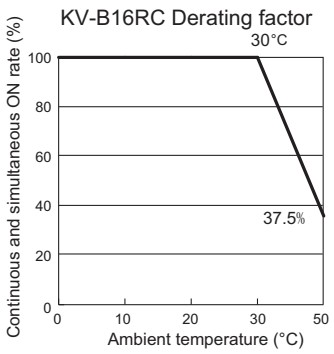
*3 4 common points are shorted in KV-C64TCP.

"KV-B8RC", "KV-B16RC internal circuit diagram", "KV-B16TCP internal circuit diagram", Page 2-29
"KV-C32TCP internal circuit diagram", Page 2-30
"KV-C64TCP internal circuit diagram", Page 2-32

Point **KV-C32TCP/KV-C64TCP is 1.6A/ common point.**

NOTICE

When the number of ON output points exceeds specified range, unit error may occur. Ambient temperature and continuous and simultaneous ON rate are shown as follows.



Type		KV-B16TC KV-B16TD (16 point)	KV-C32TC KV-C32TD (32 point)	KV-C64TC KV-C64TD (64 point)
Output mode		Transistor (NPN) (C type) MOSFET (N-ch) (with overcurrent protection function) (D type)		
External connection mode		Removable terminal block	Connector *1	
Rated load		30 VDC 0.2A(C type) 30 VDC 0.3A(D type)	30 VDC 0.2A*4	
Leakage current in case of OFF		Below 100μA		
Remaining voltage in case of ON		Below 0.5V		
Common point mode		16 point/common point (2 terminals)*2	32 point/common point (2 terminals)*2	64 point/common point (4 terminals)*3
Response time	OFF→ON	Below 10μs(C type) Below 100μs(D type)		Below 50μs(C type) Below 150μs(D type)
	ON→OFF	Below 200μs(C type) Below 300μs(D type)		
Internal consumption current		Below 35mA(C type) Below 45mA(D type)	Below 55mA(C type) Below 65mA(D type)	Below 100mA(C type) Below 120mA(D type)
Weight		About 130g	About 100g	About 140g

*1 According to MIL-C-83503

*2 Two common points of KV-B16TC, B16TD, C32TC, C32TD are shorted inside.

*3 4 common points are shorted in KV-C64TC, C64TD.

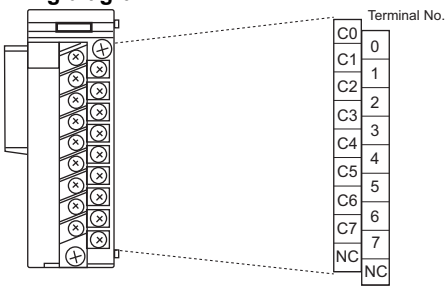
*4 The rated load for each terminal in the common is 1.6 A.

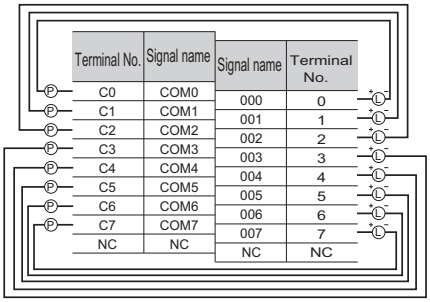
"KV-B16TC/B16TD internal circuit diagram", Page 2-29
"KV-C32TC/C32TD internal circuit diagram", Page 2-30
"KV-C64TC/C64TD internal circuit diagram", Page 2-31

Point **KV-C32T*/C64T* is 1.6A/common point.**

Wiring and Internal Circuit Diagram

KV-B8RC Wiring diagram

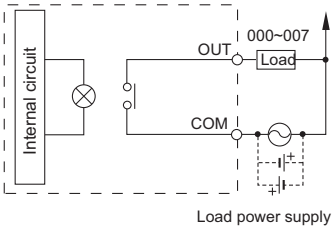




Terminal No.	Signal name	Signal name	Terminal No.
C0	COM0	000	0
C1	COM1	001	1
C2	COM2	002	2
C3	COM3	003	3
C4	COM4	004	4
C5	COM5	005	5
C6	COM6	006	6
C7	COM7	007	7
NC	NC	NC	NC

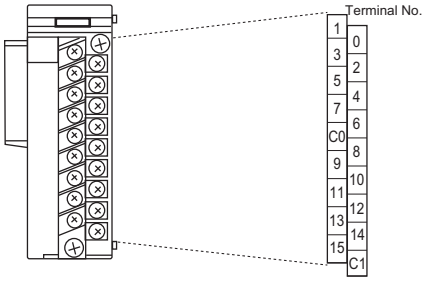
Ⓟ :

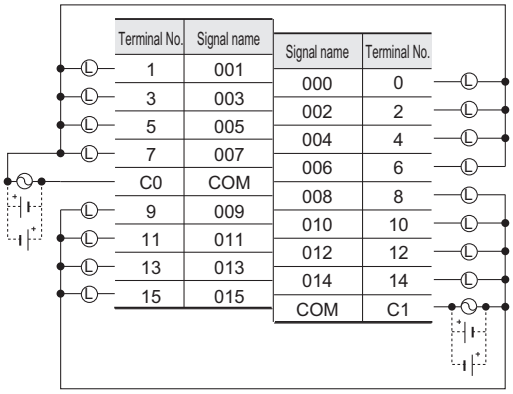
KV-B8RC
internal circuit diagram



* COM not circuit-shorted internally.

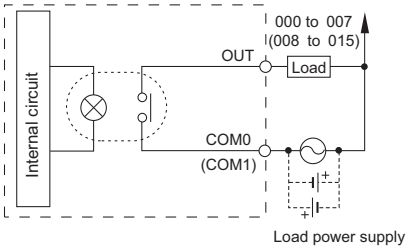
KV-B16RC/B16TC/B16TD/B16TCP Wiring diagram



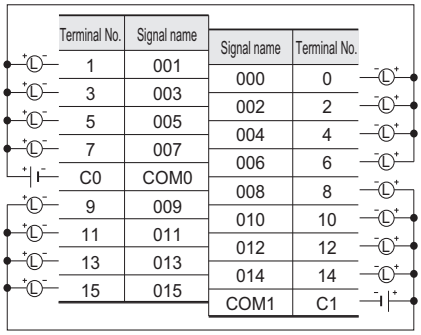


Terminal No.	Signal name	Signal name	Terminal No.
1	001	000	0
3	003	002	2
5	005	004	4
7	007	006	6
C0	COM	008	8
9	009	010	10
11	011	012	12
13	013	014	14
15	015	COM	C1

KV-B16RC
internal circuit diagram



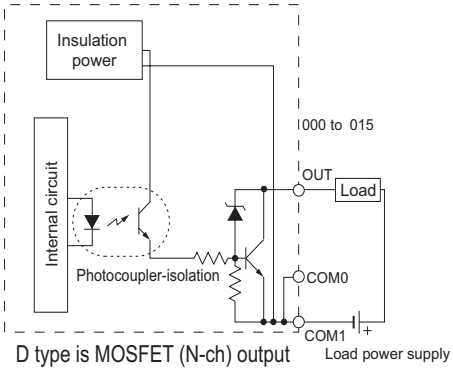
* COM not circuit-shorted internally.



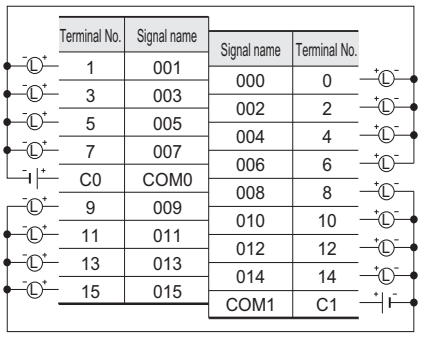
Terminal No.	Signal name	Signal name	Terminal No.
1	001	000	0
3	003	002	2
5	005	004	4
7	007	006	6
C0	COM0	008	8
9	009	010	10
11	011	012	12
13	013	014	14
15	015	COM1	C1

* COM circuit-shorted internally.

KV-B16TC/B16TD internal circuit diagram



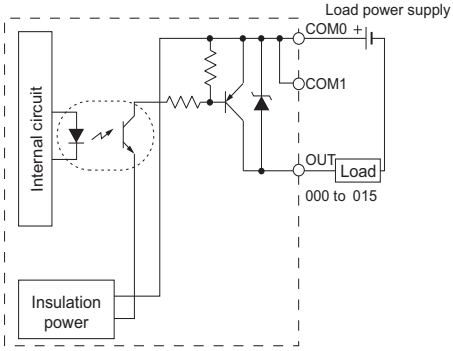
D type is MOSFET (N-ch) output



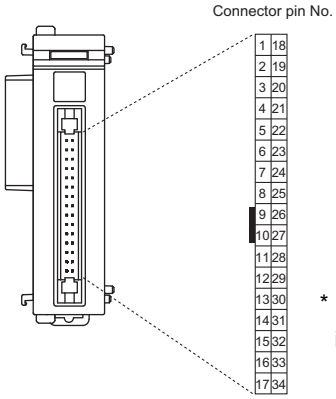
Terminal No.	Signal name	Signal name	Terminal No.
1	001	000	0
3	003	002	2
5	005	004	4
7	007	006	6
C0	COM0	008	8
9	009	010	10
11	011	012	12
13	013	014	14
15	015	COM1	C1

* COM circuit-shorted internally.

KV-B16TCP internal circuit diagram



KV-C32TC/C32TD/C32TCP Wiring diagram



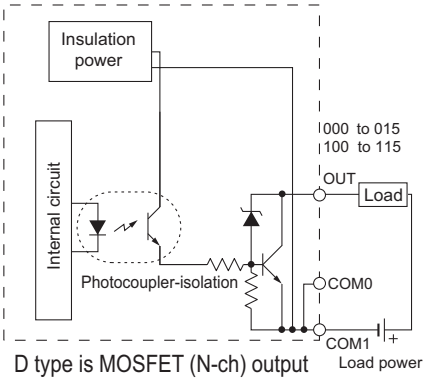
* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.

Pin No.	Signal name	Signal name	Pin No.
1	000	100	18
2	001	101	19
3	002	102	20
4	003	103	21
5	004	104	22
6	005	105	23
7	006	106	24
8	007	107	25
9	008	108	26
10	009	109	27
11	010	110	28
12	011	111	29
13	012	112	30
14	013	113	31
15	014	114	32
16	015	115	33
17	COM0	COM1	34

DC DC

* COM circuit-shorted internally.

KV-C32TC/C32TD internal circuit diagram



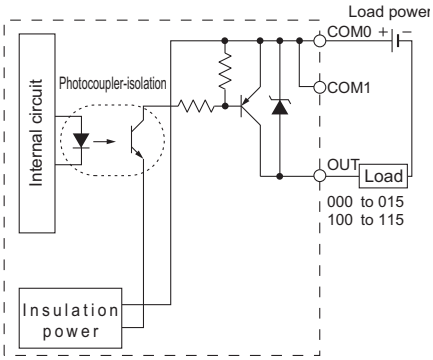
D type is MOSFET (N-ch) output

Pin No.	Signal name	Signal name	Pin No.
1	000	100	18
2	001	101	19
3	002	102	20
4	003	103	21
5	004	104	22
6	005	105	23
7	006	106	24
8	007	107	25
9	008	108	26
10	009	109	27
11	010	110	28
12	011	111	29
13	012	112	30
14	013	113	31
15	014	114	32
16	015	115	33
17	COM0	COM1	34

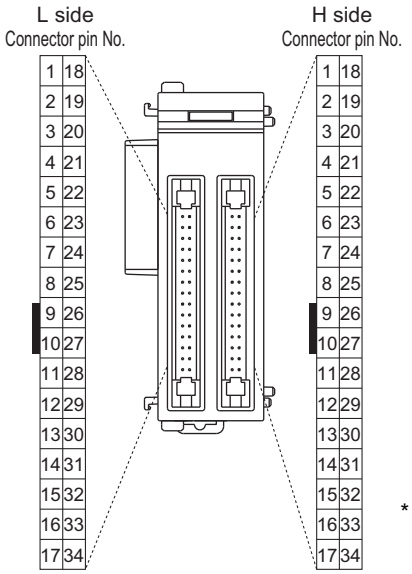
DC DC

* COM circuit-shorted internally.

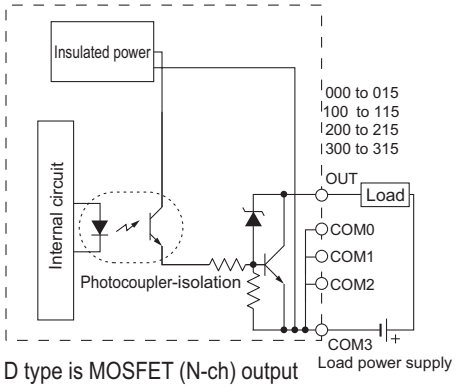
KV-C32TCP internal circuit diagram



KV-C64TC/C64TD Wiring diagram



KV-C64TC/C64TD internal circuit diagram



D type is MOSFET (N-ch) output

* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.

Pin No.	Signal name	Signal name	Pin No.
1	000	100	18
2	001	101	19
3	002	102	20
4	003	103	21
5	004	104	22
6	005	105	23
7	006	106	24
8	007	107	25
9	008	108	26
10	009	109	27
11	010	110	28
12	011	111	29
13	012	112	30
14	013	113	31
15	014	114	32
16	015	115	33
17	COM0	COM1	34

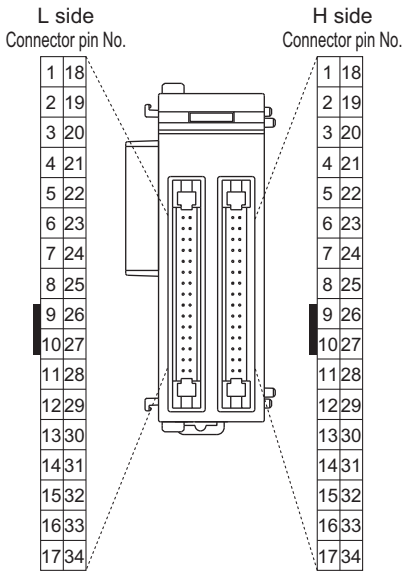
Pin No.	Signal name	Signal name	Pin No.
1	200	300	18
2	201	301	19
3	202	302	20
4	203	303	21
5	204	304	22
6	205	305	23
7	206	306	24
8	207	307	25
9	208	308	26
10	209	309	27
11	210	310	28
12	211	311	29
13	212	312	30
14	213	313	31
15	214	314	32
16	215	315	33
17	COM2	COM3	34

* COM circuit-shorted internally.

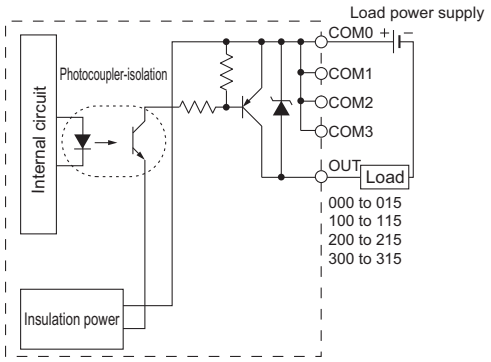
Point

Please use OP-42224 connector (34-pins) for 64-point unit.
Do not use OP-23139.

KV-C64TCP Wiring diagram



KV-C64TCP internal circuit diagram



* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.

Pin No.	Signal name	Signal name	Pin No.
1	000	100	18
2	001	101	19
3	002	102	20
4	003	103	21
5	004	104	22
6	005	105	23
7	006	106	24
8	007	107	25
9	008	108	26
10	009	109	27
11	010	110	28
12	011	111	29
13	012	112	30
14	013	113	31
15	014	114	32
16	015	115	33
17	COM0	COM1	34

DC DC

* COM circuit-shorted internally.

Pin No.	Signal name	Signal name	Pin No.
1	200	300	18
2	201	301	19
3	202	302	20
4	203	303	21
5	204	304	22
6	205	305	23
7	206	306	24
8	207	307	25
9	208	308	26
10	209	309	27
11	210	310	28
12	211	311	29
13	212	312	30
14	213	313	31
15	214	314	32
16	215	315	33
17	COM2	COM3	34

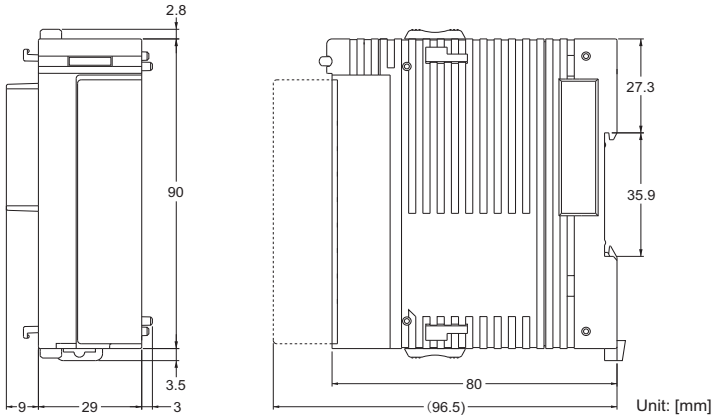
DC DC

Point

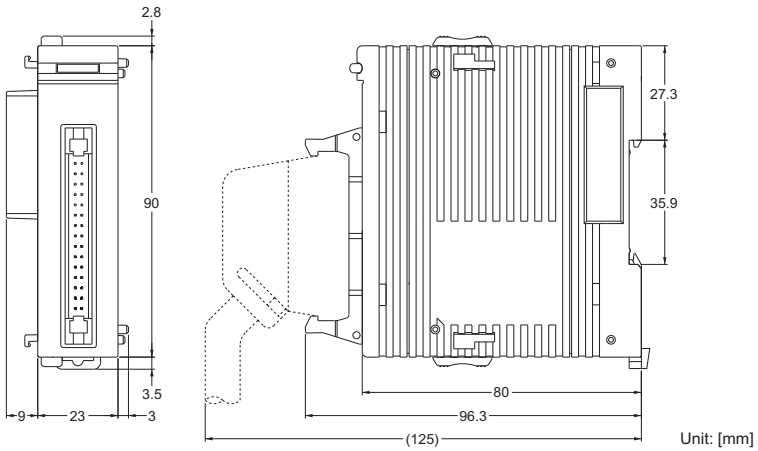
Please use OP-42224 connector (34-pin) for 64-point unit.
Do not use OP-23139.

Dimensions

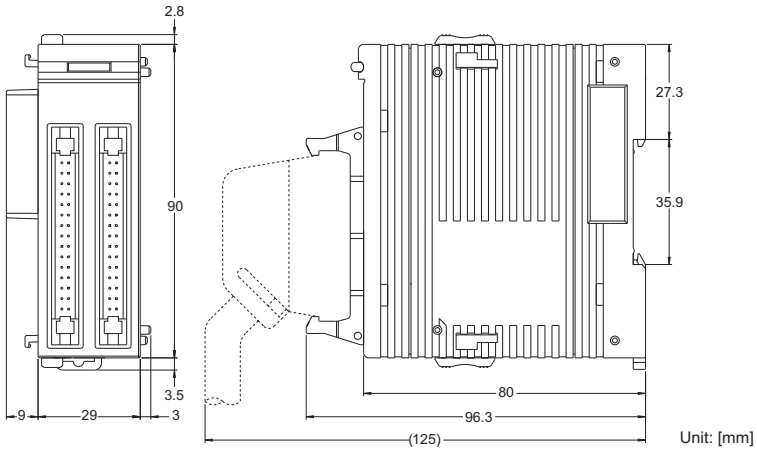
KV-B8RC/B16RC/B16TC/B16TD/B16TCP



KV-C32TC/C32TD/C32TCP



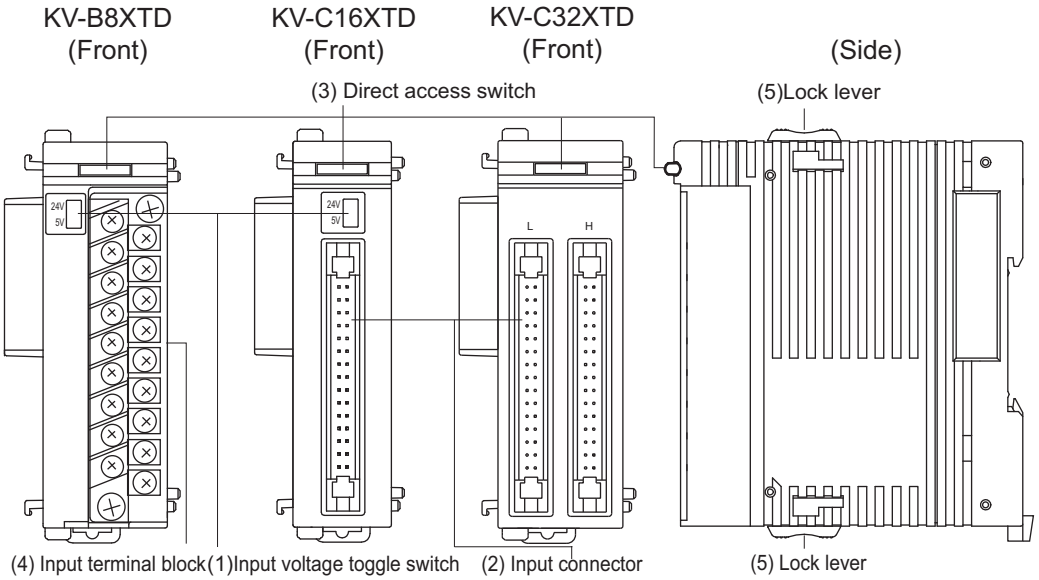
KV-C64TC/C64TD/C64TCP



2-7 Extension I/O Unit

Introduces name, function as well as technical specifications of KV-5000/3000 series expansion input unit (KV-B8XTD/C16XTD/C32XTD).

Name and function of the parts



No.	Name	Function
(1)	input voltage change-over switch	Switch input voltage. 5V <=> 24V
(2)	I/O connector	Connector for input/output.
(3)	direct access switch	After the switch is ON, state of the expansion unit may be monitored from CPU access window. Judge state of the expansion unit according to the switch color. Green lamp illuminates: normal operation Green lamp flashes: when green lamp illuminates, set the switch to ON, and select the unit Red lamp illuminates: when unit error occurs (excluding partial exception *1) Red lamp flashes: when red lamp illuminates, set the switch to ON, and select unit
(4)	I/O terminal block	Terminal block for input/output. Removable. (M3 screw)
(5)	Locking lever	Lock the unit connection.

*1 In the following cases, even if unit error does not occur, red lamp will also illuminates.
(1) when powered on, unit configuration of CPU unit only includes one CPU
Here, display error in the access window.

* 34-pin connector option is provided for KV-C16XTD/C32XTD connection.

Technical specifications

Type		KV-B8XTD (8 + 8 point)		KV-C16XTD(16 + 16 point)		KV-C32XTD(32 + 32 point)
External connection mode		Removable terminal block		Connector (MIL standard)*1		
Input	Number of input points	8 point		16 point		32 point
	Input mode	24 VDC mode	5 VDC mode	24 VDC mode	5 VDC mode	24 VDC mode*3
	Maximum input voltage	26.4 VDC				
	Rated input voltage	24 VDC/ 5.3mA	5 VDC/1mA	24 VDC/ 5.3mA	5 VDC/1mA	24 VDC/4.1mA
	Minimum ON voltage	19V	3.5V	19V	3.5V	19V
	Maximum OFF current	1.5mA	-	1.5mA	-	1.5mA
	Maximum OFF voltage	-	1.5V	-	1.5V	-
	Common point mode	8 point/common point (1 terminal)*4		16 point/common point (1 terminals) *4		32 point/common point (2 terminals)*2
	Input time constant					

Input time constant setting	OFF→ON		ON→OFF	
	TYP	MAX	TYP	MAX
25μs	25μs	65μs	75μs	120μs
300μs	275μs	365μs	275μs	420μs
1ms	1ms	1.2ms	1ms	1.2ms
10ms	10ms	12ms	10ms	12ms

Input	Input impedance	4.3kΩ				5.6kΩ
Output	Number of output point	8 point		16 point		32 point
	Output mode	MOSFET (N-ch) (with overcurrent protection function)				
	Rated load	DC30V 0.3A		DC30V 0.2A		
	Leakage current in case of OFF	Below 100μA				
	Remaining voltage in case of ON	Below 0.5V				
	Common point mode	8 point/common point (1 terminal)*4		16 point/common point (1 terminals) *4		32 point/common point (2 terminals) *2
	Run-time OFF→ON	Below 100μs				Below 150μs
	Run-time ON→OFF	Below 300μs				
Internal consumption current		Below 30mA		Below 40mA		Below 65mA
Weight		About 130g		About 110g		About 130g

*1 According to MIL-C-83503

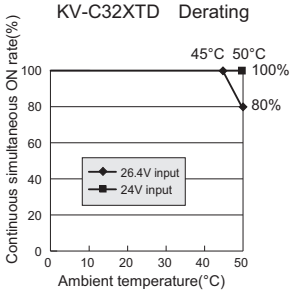
*2 KV-C32XTD has 2 point input COM and output COM, which are universal inside. Input COM and output COM are independent.

*3 All terminals support two-wire system (however, do not support two-wire system proximity sensor EV Series)

*4 Input COM and output COM are independent.

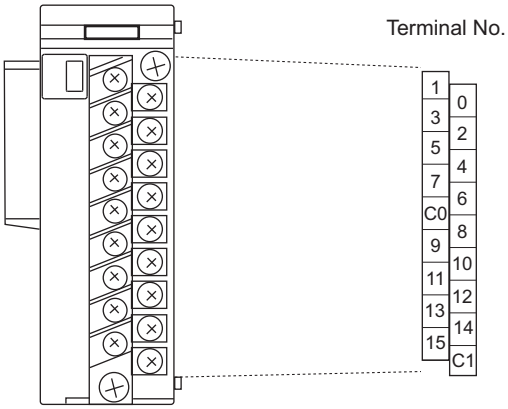
"KV-B8XTD internal circuit diagram", page 2-36 "KV-C16XTD internal circuit diagram", page 2-37
"KV-C32XTD internal circuit diagram", page 2-38

NOTICE	• **When number of simultaneous ON input points exceeds the specification scope, unit fault might occur.** • **The relationship of KV-C32XTD ambient temperature and continuous simultaneous ON rate is shown in the figure below.**



Wiring/internal circuit diagram

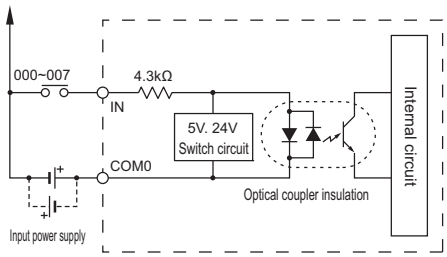
KV-B8XTD wiring diagram



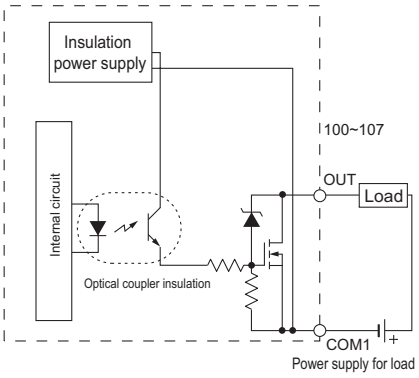
Terminal No.	Direction	Signal name	Signal name	Direction	Terminal No.
1	input	001	000	input	0
3	input	003	002	input	2
5	input	005	004	input	4
7	input	007	006	input	6
C0	input	COM0	100	output	8
9	output	101	102	output	10
11	output	103	104	output	12
13	output	105	106	output	14
15	output	107	COM1	output	C1

*COM is independent.

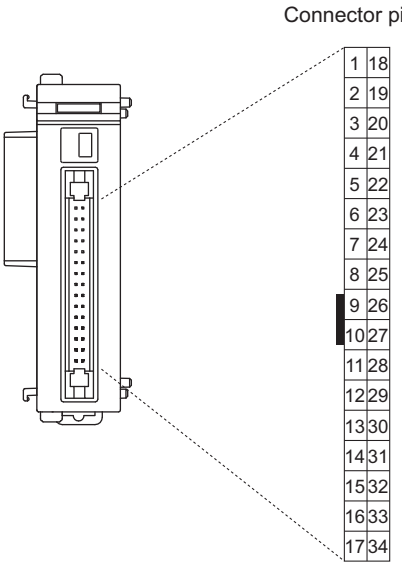
KV-B8XTD internal circuit diagram (input)



KV-B8XTD internal circuit diagram (output)

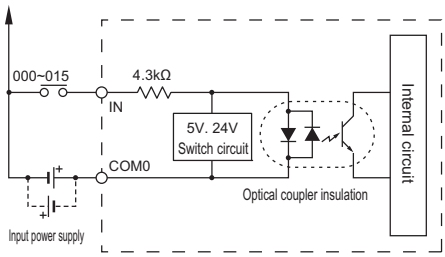


KV-C16XTD wiring diagram

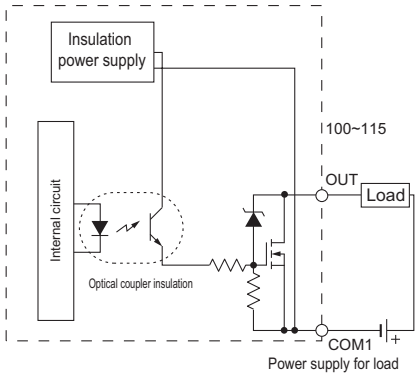


* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.

KV-C16XTD internal circuit diagram (Input)



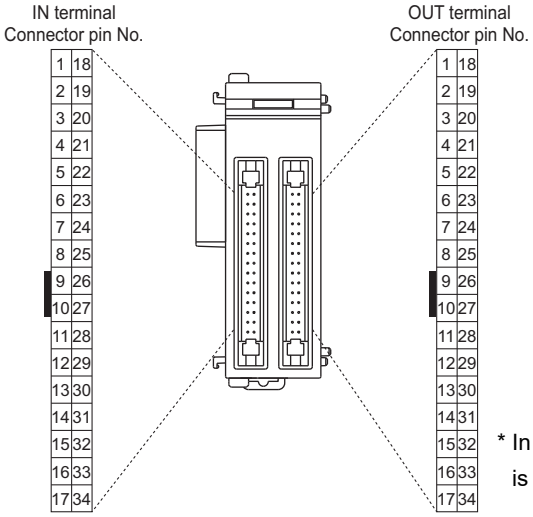
KV-C16XTD internal circuit diagram (output)



Pin No.	Direction	Signal name	Signal name	Direction	Pin No.
1	input	000	100	output	18
2	input	001	101	output	19
3	input	002	102	output	20
4	input	003	103	output	21
5	input	004	104	output	22
6	input	005	105	output	23
7	input	006	106	output	24
8	input	007	107	output	25
9	input	008	108	output	26
10	input	009	109	output	27
11	input	010	110	output	28
12	input	011	111	output	29
13	input	012	112	output	30
14	input	013	113	output	31
15	input	014	114	output	32
16	input	015	115	output	33
17	input	COM0	COM1	output	34

*COM is independent.

KV-C32XTD wiring diagram



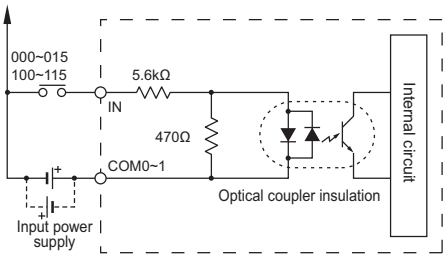
* In KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 vertically.

Pin No.	Direction	Signal name	Signal name	Direction	Pin No.
1	input	000	100	input	18
2	input	001	101	input	19
3	input	002	102	input	20
4	input	003	103	input	21
5	input	004	104	input	22
6	input	005	105	input	23
7	input	006	106	input	24
8	input	007	107	input	25
9	input	008	108	input	26
10	input	009	109	input	27
11	input	010	110	input	28
12	input	011	111	input	29
13	input	012	112	input	30
14	input	013	113	input	31
15	input	014	114	input	32
16	input	015	115	input	33
17	input	COM0	COM1	input	34

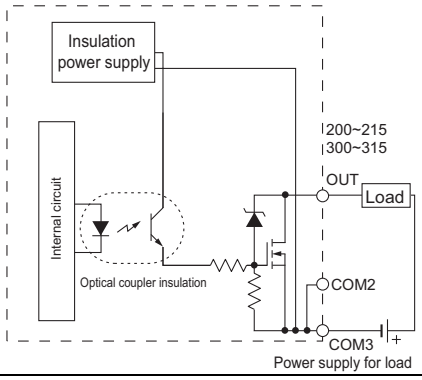
Pin No.	Direction	Signal name	Signal name	Direction	Pin No.
1	output	200	300	output	18
2	output	201	301	output	19
3	output	202	302	output	20
4	output	203	303	output	21
5	output	204	304	output	22
6	output	205	305	output	23
7	output	206	306	output	24
8	output	207	307	output	25
9	output	208	308	output	26
10	output	209	309	output	27
11	output	210	310	output	28
12	output	211	311	output	29
13	output	212	312	output	30
14	output	213	313	output	31
15	output	214	314	output	32
16	output	215	315	output	33
17	output	COM2	COM3	output	34

* Internal short circuit occurs on COM0, COM1.
Internal short circuit occurs on COM2, COM3.
Internal short circuit does not occur on (COM0,1) – (COM2,3).

KV-C32XTD internal circuit diagram (Input)



KV-C32XTD internal circuit diagram (output)

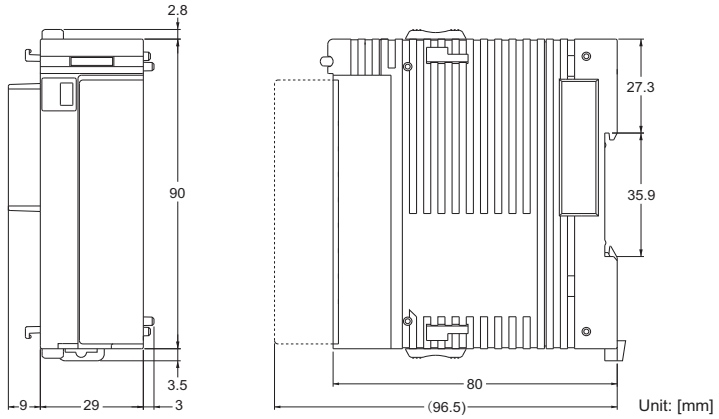


Point

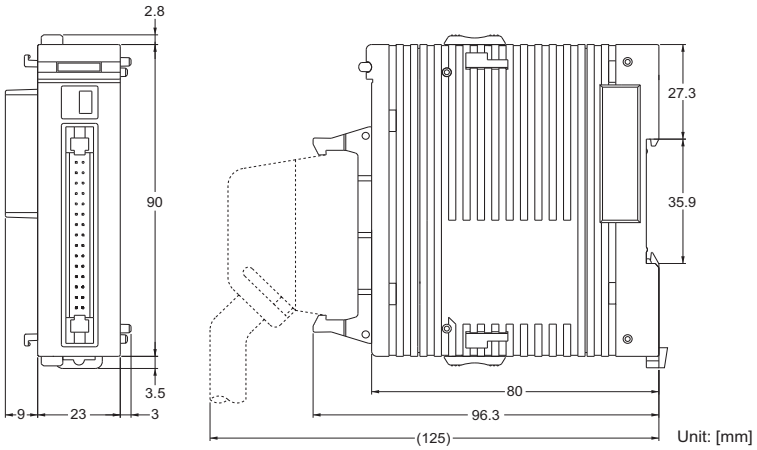
For 32 point I/O unit, please use OP-42224 connector (34 pin).
OP-23139 cannot be used.

Dimensions

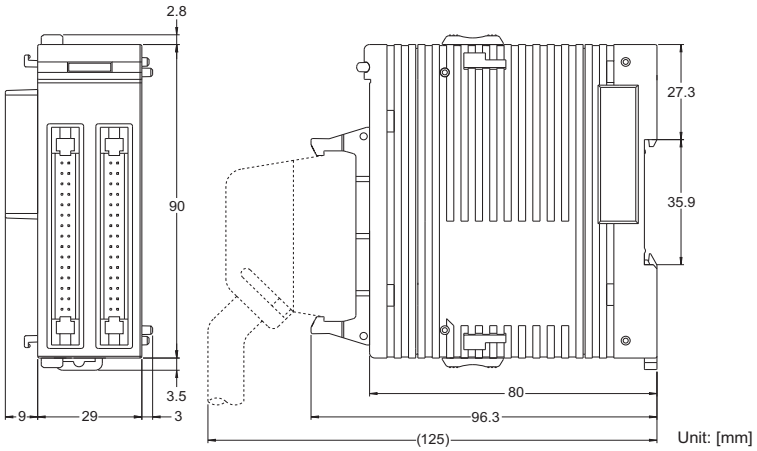
KV-B8XTD



KV-C16XTD



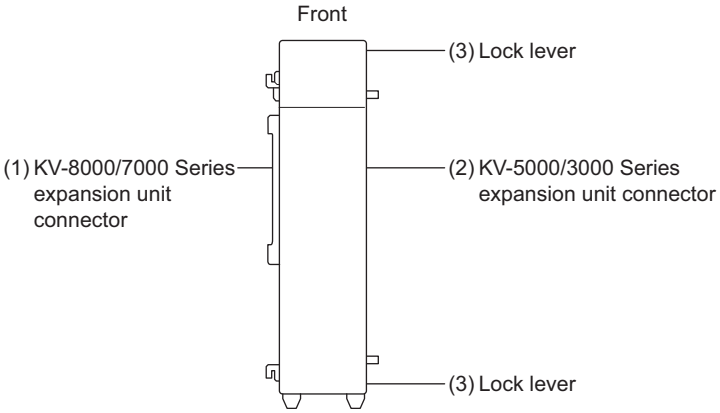
KV-C32XTD



2-8 Bus Connection Unit

It describes name and function of each portion of the bus connection unit (KV-7000C).

Names and Functions of Parts

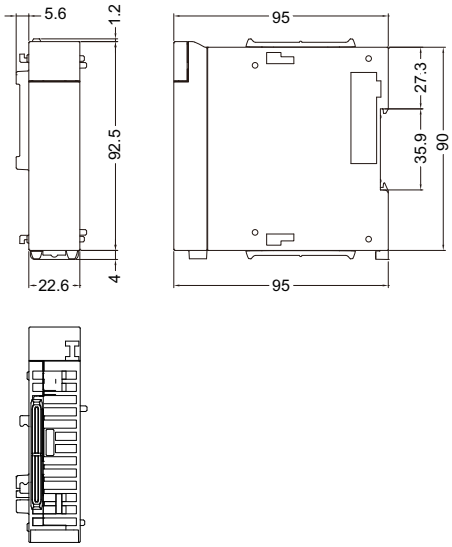


Number	Name	Function
(1)	KV-8000/7000 Series expansion unit connector	This connector is used to connect CPU units or expansion units.
(2)	KV-5000/3000 Series expansion unit connector	This connector is used to connect expansion units for the KV-5000/3000 Series.
(3)	Lock lever	This lever locks the connected units together.

Point

When bus connection unit is used, the end unit must be attached to the right end of system.

Dimensions

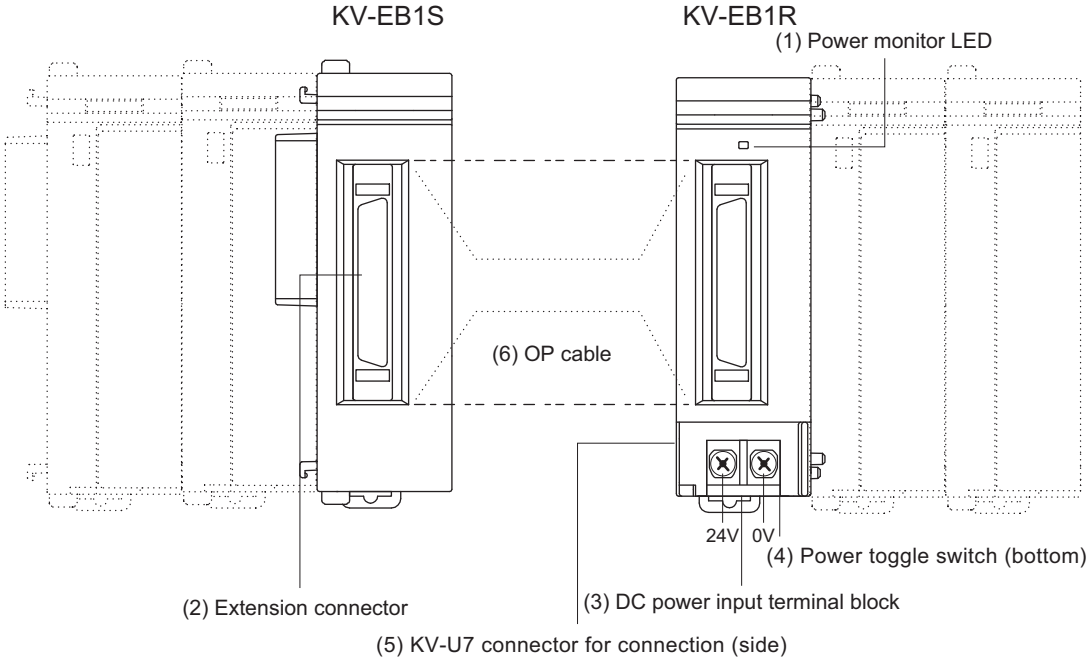


2-9 Expansion Unit

KV-EB1S/ KV-EB1R

This section describes name, function and technical specification of each part of KV-EB1S/KV-EB1R expansion unit.

Name and Function of Each Part



No.	Name	Function
(1)	Power monitor LED	Power monitor LED. *1
(2)	Extension connector	Extension cable connector.
(3)	DC power input terminal block	Terminal block for 24 VDC mains input. (Max. 3.5A)*2, *3
(4)	Power selector switch	Switch for toggle power to expansion unit. INT : When use the power on previous section. EXT : when using KV-EB1R or KV-U7 DC power.
(5)	KV-U7 connector	Connector for KV-U7 power unit *3, *4
(6)	Extension cable	Connect KV-EB1S and KV-EB1R. (OP-42142 1m), (OP-42141 2m)

*1 The orange or red LED may light up, due to mains voltage drop. Please check power supply.

*2 Ensure power connection not reversed. Although reverse connection protective circuit is designed, protection function is not started under power connection in some cases.
Please note the current consumption of expansion unit, guarantee current supplied from terminal block below 3.5A.

*3 When power capacity on KV-8000 side is under certain limit, please supply power from this point to expansion unit side.

*4 When connecting KV-U7, please note current consumption of expansion unit.

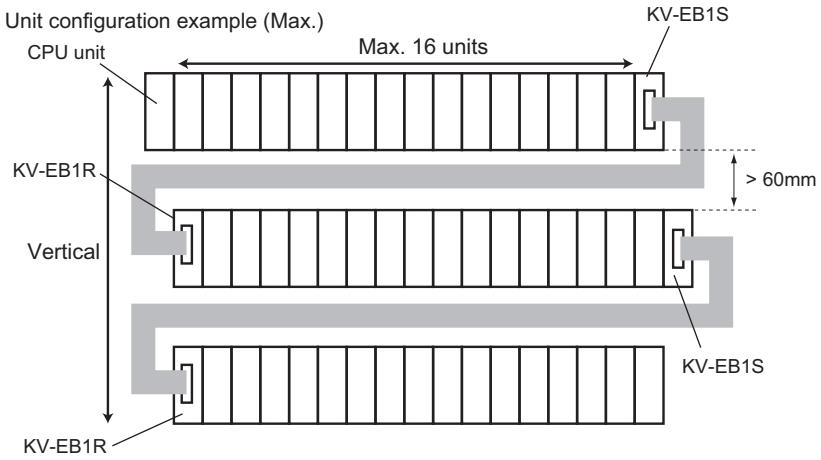
Point **When toggling power or switch, be sure to disconnect power.**

Specification

Model	KV-EB1S	KV-EB1R
Supply voltage	24 VDC±10%	
Internal current consumption	< 15mA	< 25mA
Weight	approx. 90g	approx. 115g

About Connection / Setting

KV-EB1S is connected to the rightmost end of the section before extension.
KV-EB1S is connected to the leftmost end of the section before extension.



■ About connection cable and the number of sections

- When using 1m extension cable (OP-42142)
  The maximal number of vertical sections is 3s (connect maximal 48 units)
- When using 2m extension cable (OP-42141)
  The maximal number of vertical sections is 2s (connect maximal 32 units)

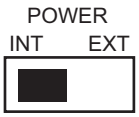
■ About setting

When using expansion unit, please consider of heat influence, the gap between the upper unit and the lower unit shall be 60mm above.
When temperature inside panel (temperature below the unit) exceeds environmental temperature used by the unit, please adopt the methods such as forced air cooling, enlarging surrounding gap or improving ventilation, etc. to reduce temperature.

About Power Supply Switching

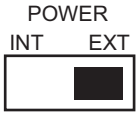
Power supply to expansion unit can be switched.

■ Use the power on previous section



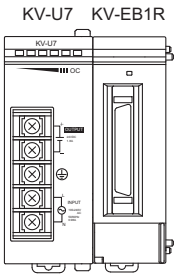
When using the power from the previous section supplied via OP cable, set power switch to "INT".
KV-U7 or other power need not to be used during operation. However, total current consumption should not exceed rating value of power.

■ Using external power



When using KV-U7 or other power, set supply power switch to "EXT".

● Using KV-U7



Connect KV-U7 to left side of KV-EB1R.

● Use separate power

Connect other power to DC power terminal block.

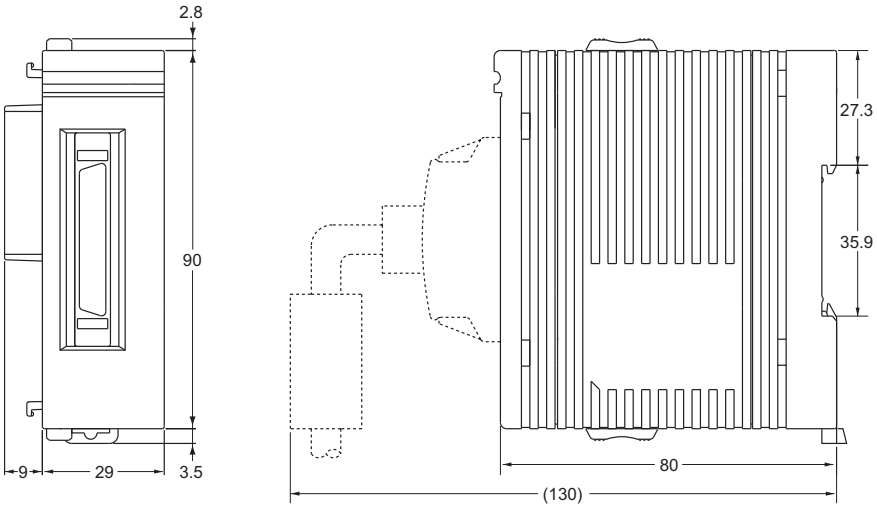
NOTICE	**When supplying power separately from KV-EB1R, please note the following points.** **• Turn ON power: please supply power from KV-EB1R side first.** **• Turn OFF power: Please disconnect power from CPU unit side first.** **(when an error occurs and after the error is cleared in access window or KV STUDIO , it will operate when switching from PROG- >RUN.)**

Point

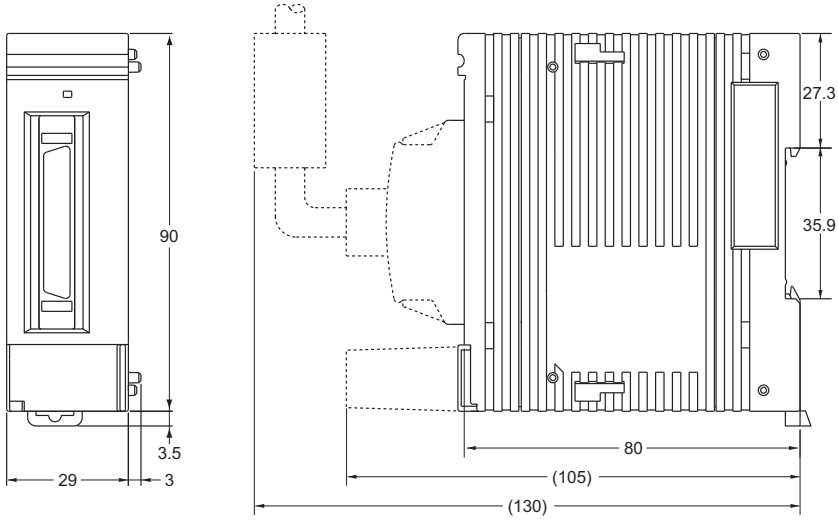
- **When using KV-EB1 for 3 section addition, and using other powers for each equipment, please wrap 2 turns of ferrite core on power cable (Kitagawa Industries: RFC-10). When using KV-U7, please adopt anti-noise measures such as FG grounding, etc.**

Dimensions

KV-EB1S



KV-EB1R



Unit: [mm]

3

UNIT INSTALLATION AND MAINTENANCE

This chapter describes how to install the KV-8000 Series as well as how to attach units, wiring precautions, and maintenance procedures.

3-1 Connecting Units 3-2
3-2 Unit Installation 3-6
3-3 Wiring Precautions 3-8
3-4 Maintenance 3-16

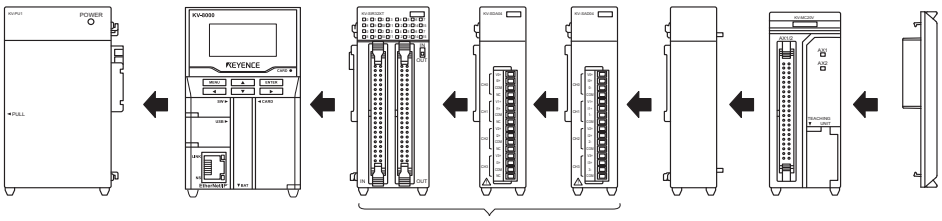
3-1 Connecting Units

This section describes how to construct units and how to install units on DIN rails and similar objects.

Precautions When Connecting Units

Be sure to observe the following point when connecting units.

- Be sure to turn the units off before connecting them.



Point

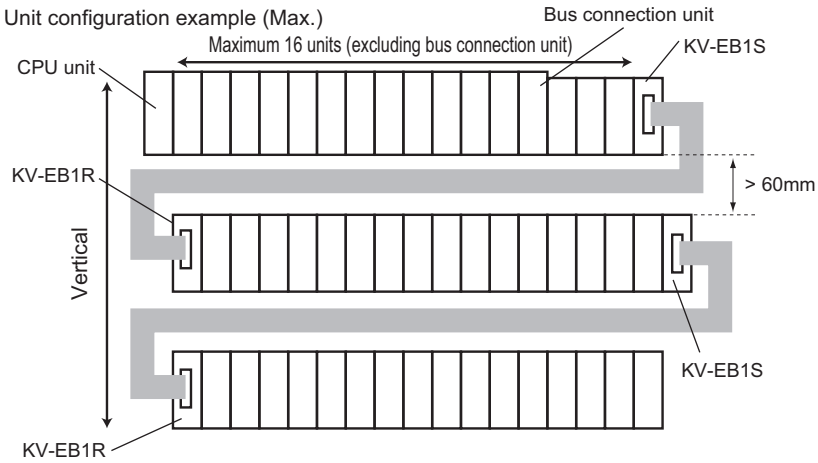
- **The CPU unit uses backup capacitors to store clock data. If the unit is not powered for an extended time period, clock data may be lost. (If lost, CPU error 85 "Clock data lost" occurs.) You may install an optional battery (KV-B1) to prevent the clock data from being lost.**
- **When the bus connection unit is used, the end unit is required at the right end. The no end unit error occurs when there is no end unit.**
- **When the bus connection unit is not used, please attach expansion unit joining connect cover to the unit at the right end during use.**

■ Maximum number of connected units

In 1 CPU unit, maximum of 48 units can be connected with a mix of expansion unit for KV-8000/7000 series, expansion unit for KV-5000/3000 series (omitting power unit, bus connection unit).

- Maximum of 16 expansion unit for KV-8000/7000 series can be connected to the CPU unit.
- The expansion unit for KV-5000/3000 series can be connected by using bus connection unit.
- When number of units connected adjacent to the CPU unit exceeds 16 units, 16 units to each step and a total of 48 units can be connected by using the expansion unit (expansion unit for KV-8000/7000 series is only connected to the first step).

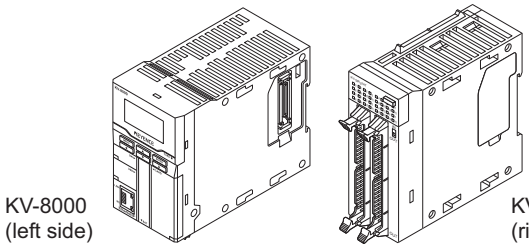
It may not be possible to connect 48 units, depending on the types of expansion unit used.
For details, refer to the manual of each unit.



Constructing Units

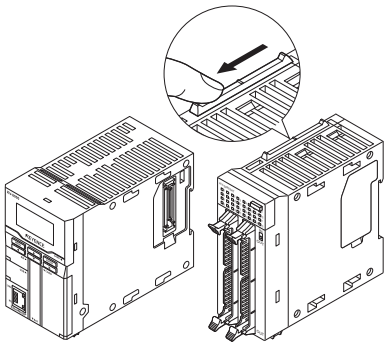
■ Connecting KV-8000/7000 Series expansion units

The connection of the CPU unit and KV-SIR32XT is describes here as an example.



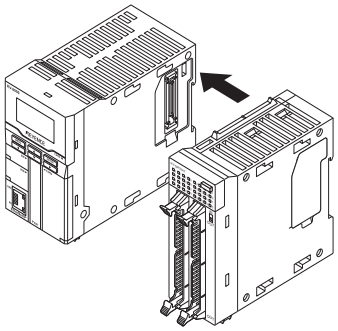
1 Release the lock levers at the upper and lower side of the right unit.

- Place your finger on the depression, and slide the lever toward the front of the unit to release.



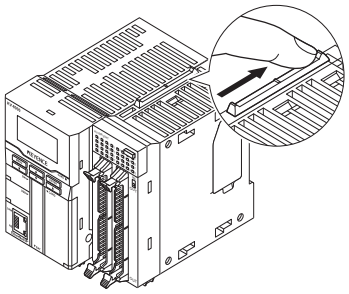
2 Insert the lock levers of the right unit into the lock lever slots on the left unit, and connect.

- The units can be connected easily by holding them parallel and lining up the connectors.



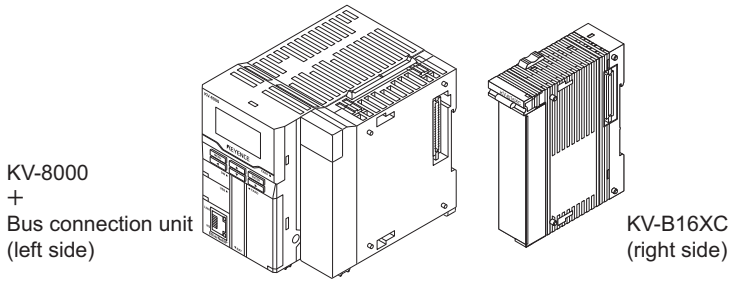
3 Lock the lock levers at the upper and lower side of the right unit.

- As in step 1, place your finger on the depression, and slide the lever toward the back of the unit to lock it.
  Then check that there is no gap between the units. The units have not been connected correctly if there is any gap between them.



■ Connecting KV-5000/3000 Series expansion units

The connection of the bus connection unit and KV-B16XC to the CPU unit is described here as an example.



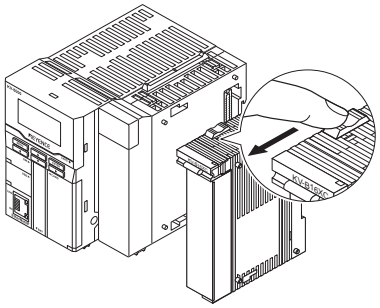
1 Remove the expansion unit connector cover attached to the right side of the KV-8000/ 7000 Series expansion unit.

2 Attach the bus connection unit.

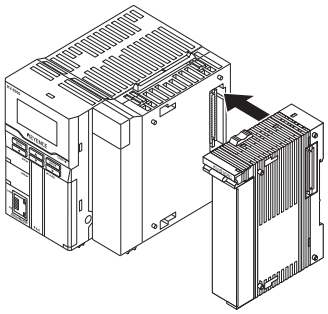
- Refer to the instructions for connecting a KV-8000/7000 Series expansion unit.

3 Release the lock lever on the KV-5000/3000 Series expansion unit.

- Place your finger on the lock lever, and slide the lever toward the front of the unit to release.

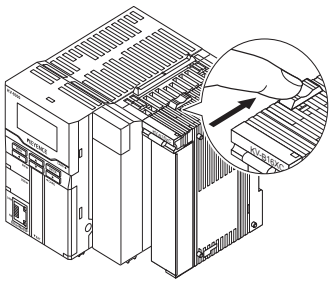


4 Insert the lock levers on the KV-5000/3000 Series Expansion Unit into the lock lever slots on the Bus Connection Unit to connect the units.



5 Lock the lock levers at the upper and lower side of the right unit.

- As in step 3, place your finger on the lock lever, and slide the lever toward the back of the unit to lock it. Then check that there is no gap between the units. The units have not been connected correctly if there is any gap between them.



6 Attach the end unit on the right side.

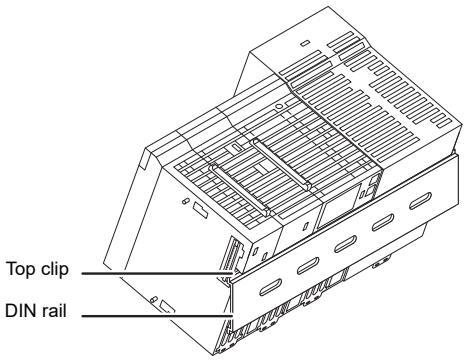
Point **When bus connection unit is used, an end unit unconnected error will occur if the end unit is not attached to the right end.**

Installing Units on DIN Rails

This section describes how to install connected units on and remove connected units from DIN rails.

■ Installing units

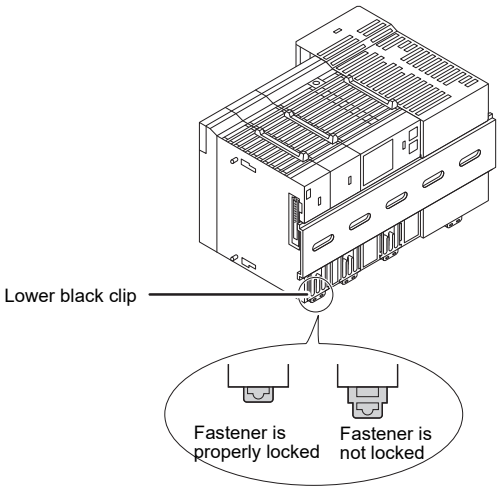
1 Hang the slot on the top of the DIN rail onto the DIN-rail fasteners on the top of each unit.



2 Insert the bottom part of the DIN rail into the DIN-rail fasteners on the bottom of each unit.

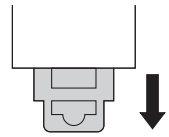
- The lower black fasteners are attached to a spring. Press down on these fasteners until you hear a click.
- In locations where it is difficult to install the units, first release all locks, install the units on the DIN rail, and then lock all the locks.

3 Check that the bottom DIN-rail fasteners are all inserted (locked).



■ Removing units

1 Pull the bottom DIN-rail fasteners down.



When rail is not attached

2 Follow the reverse of the installation procedure to remove the DIN rail from the units.

3-2 Unit Installation

This section describes precautions regarding the installation environment of the units and describes the panel mounting position.

Installation Environment

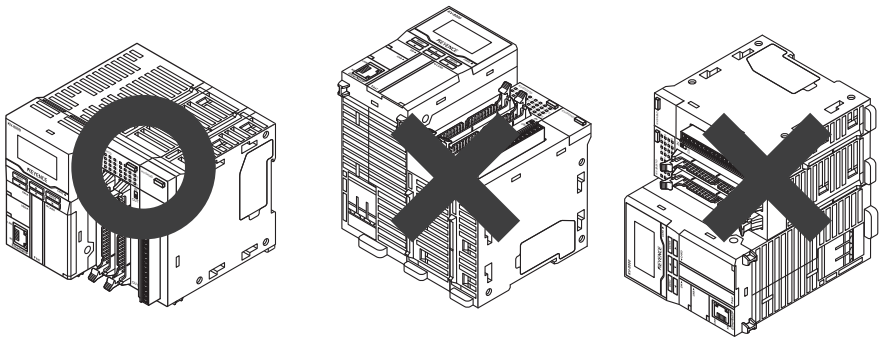
Don't install the device in the locations listed below.

- A location that is subject to direct sunshine
- A location with an ambient temperature that is not within the range of 0 to +55°C
- A location with an ambient humidity that is not within the range of 5 to 95%RH
- A location in which condensation forms because of sudden temperature changes
- A location with corrosive or combustible gas
- A location with heavy dust, salt, iron powder, or soot
- A location that is subject to direct vibration or shock
- A location that is subject to sprayed liquids such as water, oil, and drugs
- A location in which strong magnetic and electric fields occur
- A location at an altitude exceeding 2000 m

Installation Position

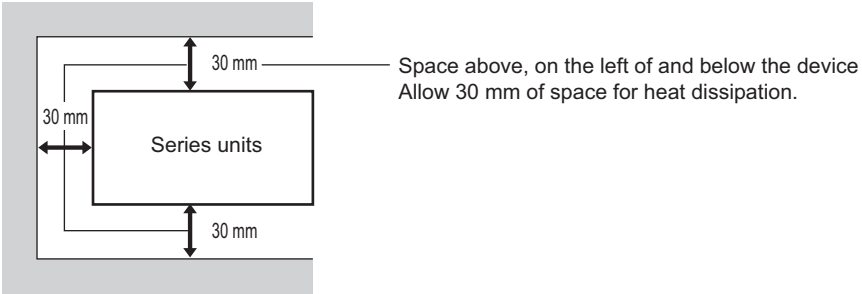
■ Device orientation

When you install the CPU unit in a panel, ensure that the front surface faces out.



■ Space surrounding the device

Install the device such that there is at least 30 mm of space between the device and the wall behind it and between the device and other equipment.



Point

If the panel temperature (the temperature indicated on the lower, central part of the unit) exceeds the ambient usage temperature, reduce the temperature by using methods such as forced air cooling and improving ventilation by increasing the space around the device.
When using an expansion unit, make sure that the upper and lower units are separated by 60 mm or more to minimize the effect of heat.

3-3 Wiring Precautions

This section describes how to assemble connectors, the wiring and internal circuits of each unit, and how to wire each unit.

Unit Wiring Precautions

Wiring precautions for I/O units and expansion I/O units are given below. Be sure to read these precautions before you wire units.

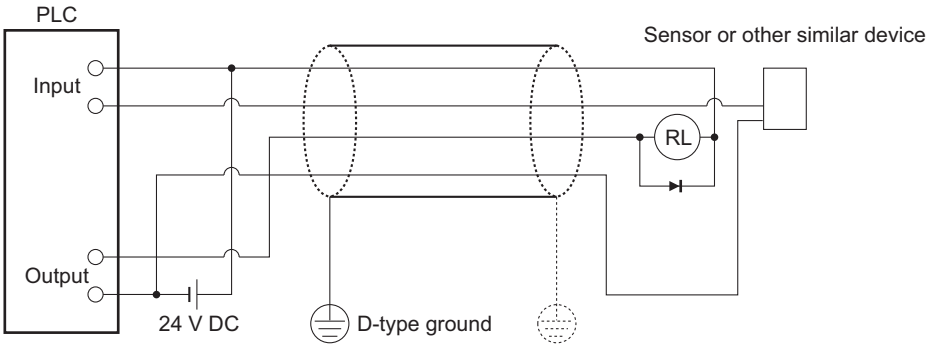
■ I/O unit wiring precautions

Note the following precautions when wiring I/O units.

- Wire the input and output cables separately.
- If I/O signal cables are close to power cables, the I/O signal cables may be influenced by high voltage and current, which may lead to operation errors. I/O signal cables should be laid at least 100 mm away from power cables.
- Lay I/O signal cables separately from 100 V AC and 200 V AC power cables.
- When using cable conduits during wiring, ensure that the cable conduits are grounded securely.
- Be sure to turn all devices off before performing wiring. Failure to do so may lead to electric shock and product damage.
- After performing wiring or similar tasks, be sure to attach the terminal cover that is included with the product before turning the power on and performing operations. Failure to do so may lead to electric shock.

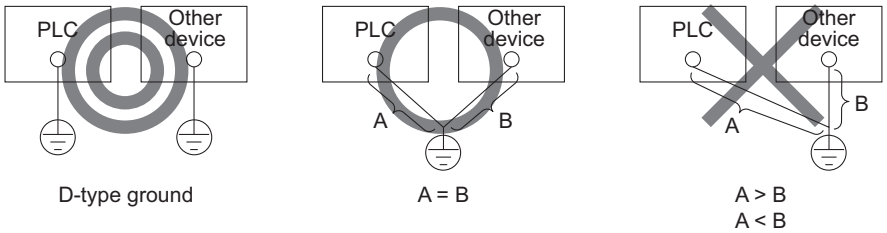
When you cannot lay cables separately from power cables

Use a shielded cable that houses all the wires, and ground the cable on the PLC side.
(Depending on the environment, you may have to ground the cable on the side opposite the PLC or on both sides.)



■ Grounding precautions

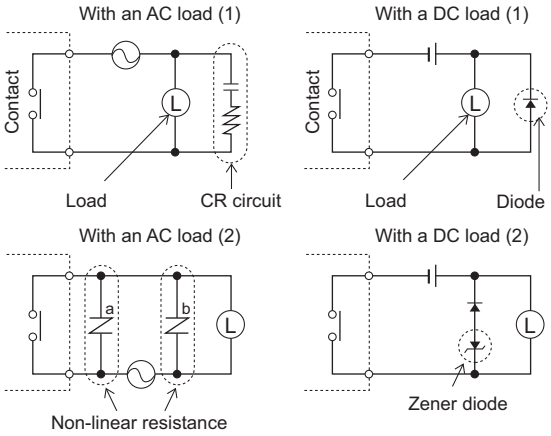
- Ground each device separately using D-type grounding facilities. In addition, ensure that the grounding resistance in this situation is 100Ω or less.
- If separate grounding is not possible, ground all the devices from a common point. However, in this situation, all cables must be the same length.
- Use the FG terminals to ground the devices (do not ground the power supply terminals).



■ Contact protection

When you are using clutches, motors, solenoids, or other inductive loads, an inrush current will flow when power is applied to the load, and inverting voltage will be generated when the load is turned off. Inrush current and inverting voltage are major factors in reducing the life of the contact. To reduce the occurrences of these phenomena, provide a contact protection circuit.

Contact protection circuit example



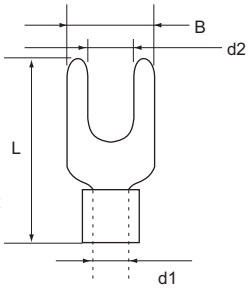
• Supply voltage 24 to 48 V To position b
100 to 200 V To position a
Attach the power supply so that it behaves in this manner.

- Use a load coil that has a rating that is less than or equal to the capacity of the contact.
- Use a diode that has a withstand voltage at least 10 times the circuit voltage and that has a forward current that is greater than or equal to the load current.
- Attach the diode, non-linear resistance, and CR circuit directly to the relay coil terminal.

■ AC power supply unit/power supply terminal block wiring

Use stranded copper wire (twisted wire) with a wire gauge of AWG #12 to #22 and a temperature rating of 105°C or higher when wiring to the terminal block on the KV-PU1. The tightening torque is 1.4 N·m. (12.4 lb.-In)
Use stranded copper wire (twisted wire) with a wire gauge of AWG #16 to #22 and temperature rating of 105°C or higher when wiring to the DC power input of the terminal block. The tightening torque is 0.5 N·m.
Use Y terminals that have the following dimensions.

Y terminal part dimensions
B : Y-shaped part outer size
d1 : Diameter of power line outlet
d2 : Y-shaped part inner width (part that the screw grips)
L : Total length



Compatible dimensions
B : 6 mm or less
L : 13 mm or less
d2 : 3.2 mm or more

Recommended products

Manufacturer	Model	B	d1	d2	L	Compatible Wire (Stranded)
Nippon Tanshi Co., Ltd.	VD1.25－3	5.5	1.7	3.3	11.5	0.25 to 1.65 mm^2 (AWG 22 to 16)
DAIDO SOLDERLESS TERMINAL MFG. CO., LTD.	F1.25－C3.5	6.0	1.7	3.7	12.8	0.25 to 1.65 mm^2 (AWG 22 to 16)
NICHIFU Co., Ltd.	0.3Y－3	5.2	1.0	3.2	12.0	0.3 to 0.5 mm^2 (AWG 22 to 20)
JST Mfg. Co., Ltd.	0.5－3A	5.0	1.0	3.2	12.5	0.2 to 0.5 mm^2 (AWG 26 to 22)
FUJI TERMINAL INDUSTRY CO., LTD.	V1.25－YASS3.5	5.7	3.3	3.8	20	0.25 to 1.65 mm^2 (AWG 22 to 16)

Reference

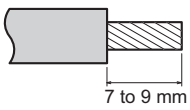
Hold the screwdriver so that it is perpendicular with the terminal block screws when wiring cables to the terminal block. Tightening the screws with the screwdriver held in any orientation other than perpendicular with the terminal block screws may damage the terminal block.

● Wiring for European terminals

The wiring of the European terminal block should use the AWG#12~#24 copper wire (stranded wire). The tightening torque is 0.5N·m.
For the cable termination, refer to the following.

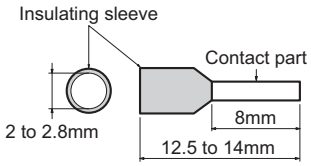
(1) When using stranded wires or single wires as is

(a) Twist the tip of stranded wires so that there are no loose wires.
(b) Do not solder plate the tips of single wires.



(2) When using rod terminals with insulating sleeves

Depending on the thickness of the wire's sheath, it may be difficult to insert the wire into the insulating sleeve. Therefore, use the outline diagram as a reference when you select wires.



Recommended manufacturer	Shape
Phoenix Contact	AI0.25-8BU(AWG24)
	AI0.34-8TQ(AWG22)
	AI0.5-8WH(AWG20)
	AI0.75-8GY(AWG18)

Connector Assembly and Wiring

When connecting equipment on each unit, please connect the wire from each equipment on the connector (female) on the unit side.

■ Precautions on wiring (OP-22184/51404/23139/42224)

The following points must be noted before wiring:

- The following matching cable must be used:

  Wire gauge : AWG24 to 22 (twisted wire)
  Cross-section area: 0.2 to 0.3mm^2
  Sheath O.D.: Ø1.5 to Ø1.1
- Top of wire shall be cut off straight to make it flat.
- After wiring, make sure that wire has been clamped completely by the fixed part, and inserted inside.
- No burr shall be left at cutting on the cutting part of wire (it's not necessary to strip the sheath).
- AWG28 to 26 fine line contact OP-30594 is supplied.

  (please use with OP-22184 and OP-51404, etc.)

  Wire gauge : AWG28 to 26
  Cross-section area: 0.08 to 0.14mm^2
  Sheath O.D.: Ø1.3 to Ø1.1

NOTICE	**The unit is made of synthetic resin. It may be dissolved after touching with strong solvents such as acetone, trichloroethylene and gasoline, etc. Besides, it may be damaged if too large force is applied in crimp connection. Be sure to take care.**

Point

Please use OP- 42224 connector (34-pin) for 64-point unit.
Do not use OP-23139.

■ Special-purpose crimping tool (OP-21734)

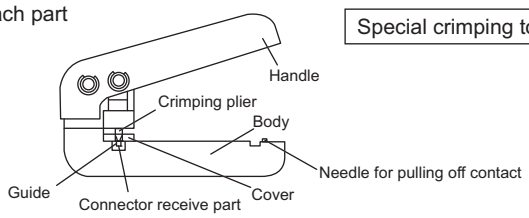
When crimping wire, using special crimping tool can simplify the operation.
Please prepare the following special crimping tool in advance.

Point

If no special crimping tool is available, it is necessary to solder.

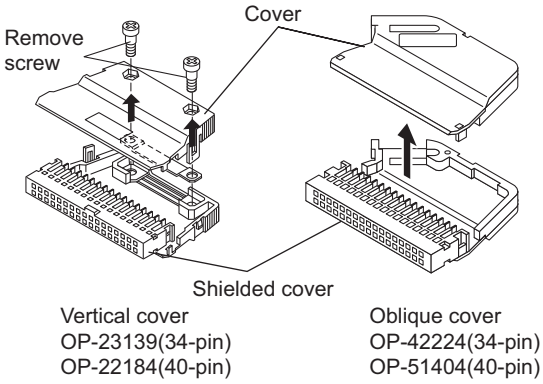
Name of each part

Special crimping tool: type OP-21734



■ About the Connector

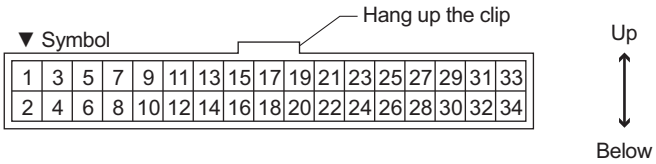
Upper and lower covers are installed on the connector. When wiring, firstly remove the upper and lower cover.



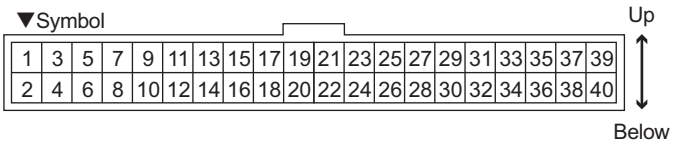
Connector terminal No.

Depending on the type of units, there are two kinds of connectors, 34-pin and 40-pin. The Terminal No. of connectors are as follows. During wiring operation, please check against "Wiring and Internal Circuit Diagrams of Each unit".

34-pin connector: type: OP-23139, OP-42224



40-pin connector: type:OP-22184, OP-51404

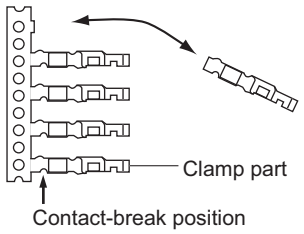


* When using KV-5000/3000 Series expansion units, Pin No. is arranged as 1, 2, 3 horizontally.

■ Connector wiring

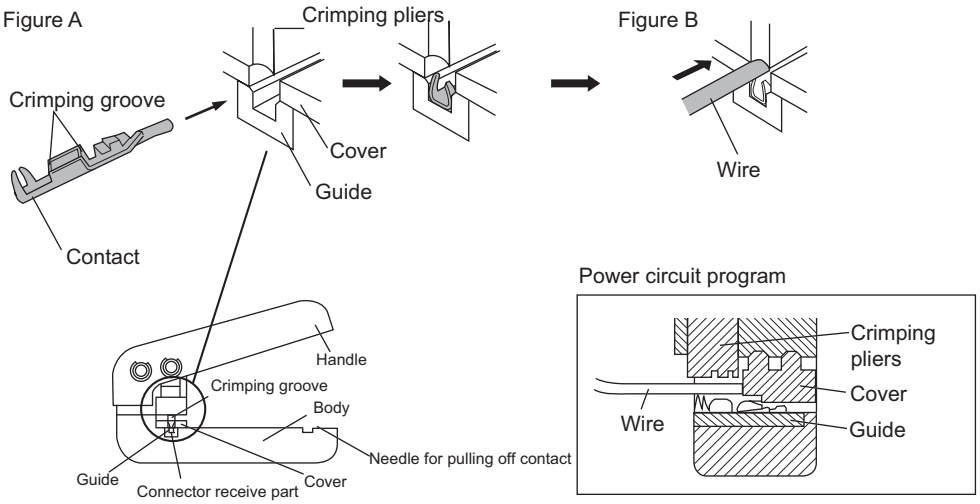
The wiring method of using special crimping tool is introduced below. Contact, shield sleeve and wire shall be prepared.

1 Break off it starting from the breakdown position at clamp part of contact toward the breaking-off direction of arrow.

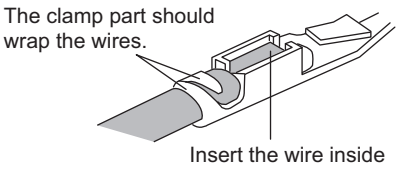


2 Insert contact into the contact stand of special crimping tool and contact directly (Fig. A).

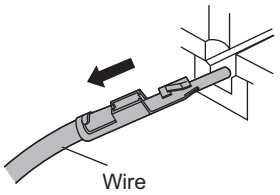
3 Insert the wire straight to contact crimping slot (Fig. B).



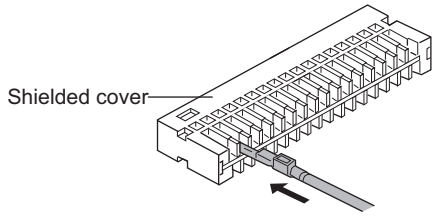
4 Hold the handle of special crimping tool, until the wire is clamped completely.



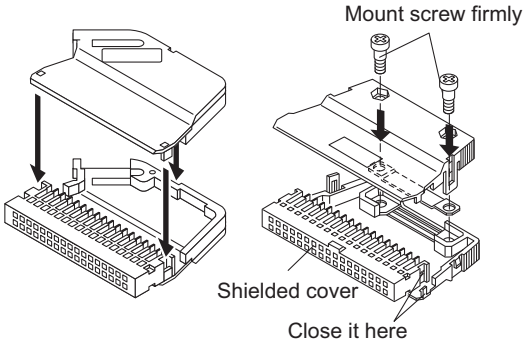
5 Pull the wire lightly, take the contact of the crimp connection wire out of special crimping tool.



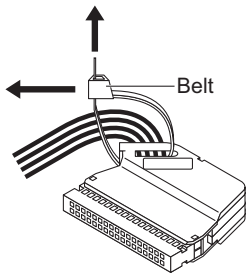
6 **Straight insert along the slot of shield sleeve without contact.**



7 **Install shield sleeve to contact, fit cover.**



8 **Fix the wire with band as shown.**

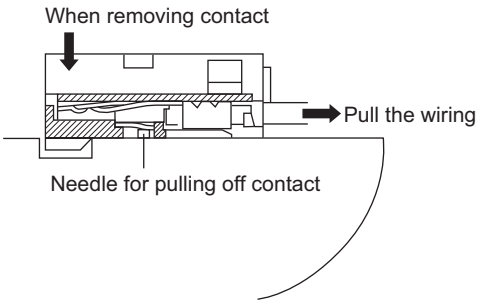


● When removing contact

Remove the contact from connector as follows.

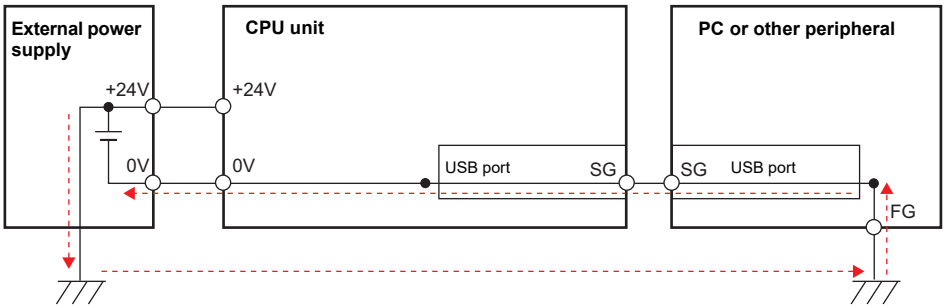
1 **Align the terminal to be removed with the arrow on box of special crimping tool.**

2 **Press the contact, and pull out the wire.**



Precautions When Using the USB Port to Communicate with a PC or Other Peripheral

0 V of the power supply of CPU units is electrically connected to the SGs of the USB port. An electric potential difference between the port and SG at the destination could damage the CPU unit and devices connected at the destination. For example, if you positively ground (perform grounding on the 24 V side) the products, a potential difference will occur, and current will flow along the path shown in the following figure. This may cause damage to the products. Do not wire the products in the manner shown in the following figure.



Reference

You may be able to prevent the problem described above by:

- Not grounding the products on the 24 V side (not performing positive grounding).
- Use KV-PU1
- Use the isolated power supply for power connected to CPU unit.
- Not grounding the FG of the PC or other peripheral.
- Using an isolator to electrically isolate the USB signal wires.

3-4 Maintenance

This section describes how to inspect and maintain the units as well as how to replace batteries and relays.

Inspection and Maintenance

■ Inspection

When the CPU unit and other units are used for a long time, problems such as the connecting section of the connectors becoming loose occur. Continued use of devices in this state may cause operating difficulties.
For this reason, periodically inspect the unit body, wired parts and other sections of CPU units and other units.
The main items to inspect are listed below.

- Are the connecting sections of the connectors of each unit disconnected or loose?
- Are the connecting sections of the connectors disconnected or loose?
- Are the terminal block screws loose?
- Are the wiring cables between the unit and the other devices damaged?

■ Maintenance

Dirt adheres to the CPU unit and other units when they are used over a prolonged period of time.
Clean off any dirt from the units using a clean, dry cloth.
Dust and dirt on fine components such as connectors can be removed with a cotton swab or similar item after first removing the connector.

About Maintenance

■ Battery

Data such as ladder programs, device comments, and setting information is recorded on flash ROM. Data such as device values, and the error log is recorded on nonvolatile RAM.
However, backup capacitors are used to save the clock data, so this data may be cleared. Use a battery (KV-B1) if you need to back up the clock data.

"How to set clock data" (page 3-18)

■ When the battery voltage is low

When the "CPU Error E83" or "CPUエラー E83" or "CPU错误 E83" appears, replace the lithium battery as soon as possible.
The CR2306 turns on when a battery error occurs.
Data can be stored for at least a week (at room temperature) after the battery error message has occurred.
Error checks are performed every 24 hours after the unit is started up.
When no battery is used, the message "CPU Error E85" or "CPUエラー E85" or "CPU错误 E85" appears when the voltage in the backup capacitors drops.

■ When the battery is depleted

When the battery is fully depleted, the clock data will be lost, but memory data and ladder programs will remain. As long as the ladder programs do not require clock data, you will be able to run the programs for temporarily by performing an error clear in the access window.
Replace the battery with the following product.
Replacement lithium battery Product No.: KV-B1

Reference When no battery is used, a battery depleted error will not be generated.

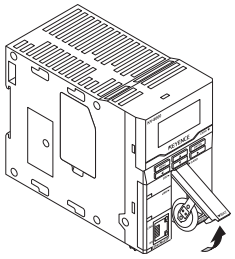
■ Operation with no battery

The CPU unit can be operated without a battery.
The message "CPU Error E85" or "CPU エラー E85" or "CPU 错误 E85" appears when the voltage in the backup capacitors drops and the clock data is cleared. This message is automatically displayed and need not be preset.

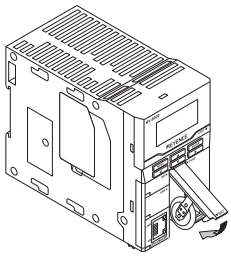
Moreover, the error can be avoided in the following methods.
- By continuing (ignoring) the execution of CPU operation of time abnormality with CPU system configuration (operation configuration during error occurrence).

■ Battery replacement procedure

1 Remove the backup battery cover and remove the old battery.
- When exchanging during power OFF, flow the current for 10 minutes or longer prior to exchange and complete the exchange task in 10 minutes or shorter.



2 Insert a new battery.



3 Turn the power ON and make sure that the access window is not displaying an error.

NOTICE	• **Battery life is as follows:** **25°C: 5 years** **40°C: 2 years** **75°C: 1 year** • **The battery can be replaced while the power is on.** • **The system must be on for 10 minutes or more before replacing the battery or the data may be lost. Battery replacement must be completed within 10 minutes.** • **Never throw a CPU unit battery into fire or disassemble it. The battery is not rechargeable. Do not attempt to recharge it. Attempts to recharge it could cause an explosion.** • **Be sure to use only the KV-B1 battery as use of any other battery could result in a fire or explosion.**

About the Restoration Method

■ How to set clock data

With the CPU unit, built-in backup capacitors are used to hold the clock data. If you hold the clock data for some time without turning the CPU unit on, the clock data will be cleared. Install a battery (KV-B1) to permanently hold the clock data.

* Let electricity flow through the backup capacitor for more than 30 minutes to fully charge it.

Hold time of the backup capacitors (when fully charged)

Temperature	TYP	MIN
25°C	60 days	15 days
40°C	40 days	10 days
75°C	12 days	3 days

If necessary, use one of the following methods to set the clock data.

- While using KV STUDIO to perform monitoring, on the "Debug" menu, click "Setup calendar timer."
- Use the WTIME instruction in a ladder program.
- Set the data using a clock from the access window.
- If you are using a touch panel (VT5/VT3 Series), use the PLC time and date synchronization function. (If you are using the restoration function, the time will be adjusted automatically.)

"VT STUDIO Reference Manual"

Reference

If the clock data is cleared, the following functions may not operate correctly.

- Logging/trace function (execution time of logging/trace differs)
- Error monitor function (the error occurrence times will be incorrect)
- Weekly contact instructions

Restoration Method

■ How to clear errors

With the CPU unit, you can use the following methods to clear errors when they occur.

- Execute error clear from the access window.
  "Error Clear" (page 5-127)
- While using KV STUDIO to perform monitoring, you can clear the error from the dialog box that appears when the error occurs.
- Use a ladder program to set the error clear request relay (CR3909). (This method can only be used for minor errors).
- Restart the product.

Point

- **When a BOOL array type variable is assigned to CR, the element number of the array should be specified with a value.**
  When specified with device or variable, the SET instruction cannot operate properly.
- **Some major errors related to the system may require power restart.**

■ How to clear the PLC status

You can set the CPU unit to the factory default setting and clear the device values that are being used in programs.

Resetting the base unit status to the factory default setting

You can clear data such as ladder programs, device comments, setting information, device values, error log, the event log and CPU memory.

Use one of the following methods to clear the data.

- In KV STUDIO, on the "Tool" menu, click "Clear PLC (1)" to display the Clear PLC dialog box. Select the "Clear All" check box, and then click "Execute."
- Execute all clear from the access window.

Clearing the device values

You can clear data such as device values and the error log.

Use one of the following methods to clear the data.

- In KV STUDIO, on the "Tool" menu, click "Clear PLC (1)" to display the Clear PLC dialog box. Select the "Clear RAM" check box, and then click "Execute."
- Execute RAM clear from the access window.

Clearing CPU memory (user memory)

Clear the user document, file register configuration, logging data, project save data etc. inside the CPU memory.

Use one of the following methods to clear the data.

- In KV STUDIO, on the "Tool" menu, click "Clear PLC" to display the Clear PLC dialog box. Select the "Clear user memory " check box, and then click "Execute."
- Execute "Clear user memory " from the access window.

When clearing all log

Clear all error log of CPU unit, power ON/OFF log and PLC event log.

Please clear them with the following method.

- Set the event record clear relay (CR3910) in the ladder program.

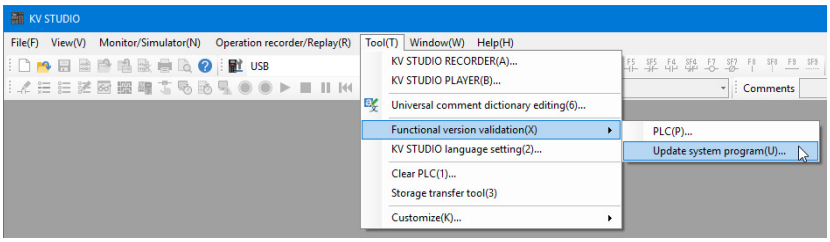
Point

- **When a BOOL array type variable is assigned to CR, the element number of the array should be specified with a value.**
  When specified with device or variable, the SET instruction cannot operate properly.

System Program Update

CPU function version can be upgraded by updating the system program from the KV STUDIO or memory card. The update procedure is as follows.

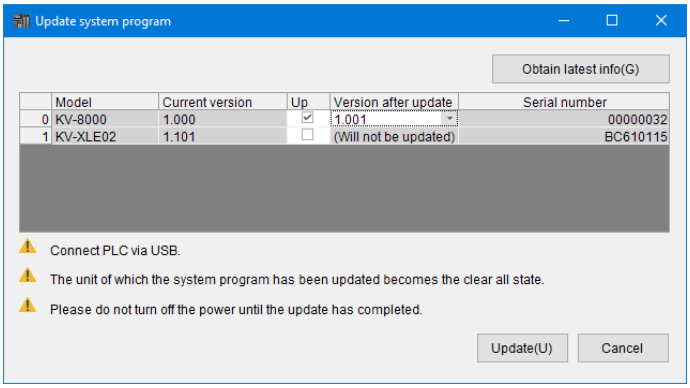
1 **The version can be confirmed through "Tool (T)" -> "Functional version validation (X)" ->"Update system program (U)" on the KV STUDIO menu.**



Point

- **System program updates can be carried out when connected via USB.**

2 **Whether or not the connected KV-8000 series is the latest version or not should be confirmed, and if it is not, the (update) button should be clicked to begin the system version upgrade.**

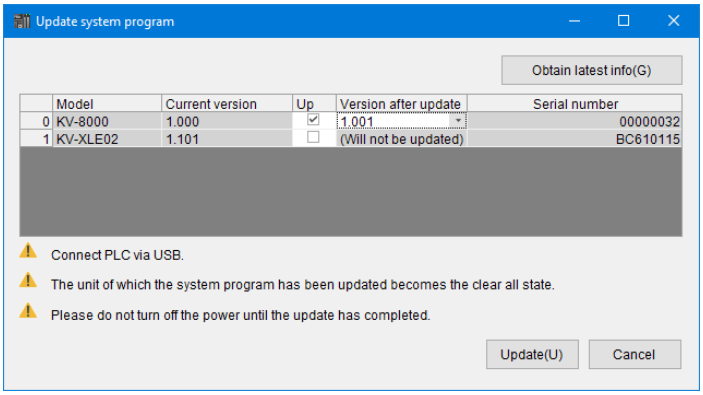


Point

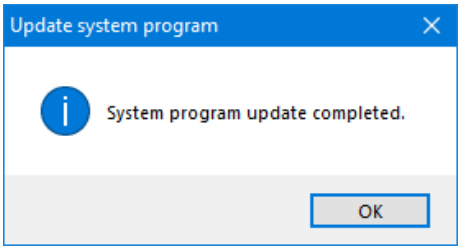
- **Do not turn the unit off during system updates.**
- **When a system program update is performed, the programs in the unit are deleted. Create a backup of the programs before performing a system program update.**
- **To update the system program to the latest version, update KV STUDIO to the latest version beforehand.**

Point

- **If the system program is the latest version, the dialog below will be shown.**

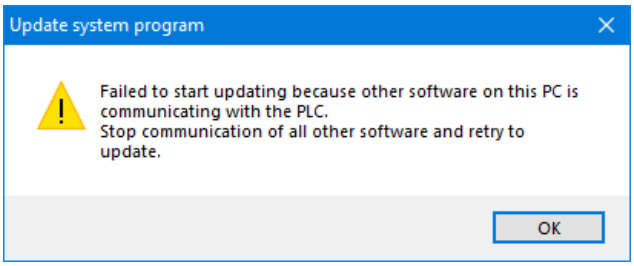


3 **The dialog below is displayed when the update is complete.**



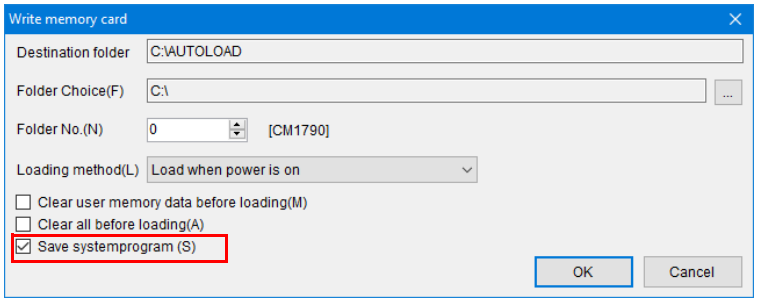
Point

- **If there is other communication between the CPU unit and computer when the system program update is executed, the following dialog is displayed and the system update is not performed.**



If updating from the memory card

1 **Memory card data will be created with【File (F)】→【Memory Card (M)】→【Save (W)】from the KV STUDIO menu.**



2 **The saved memory card data should be stored in the memory card, and the loading process should begin.**

Point

- **For information about loading from the memory card (including auto load/run load), please see "Load" (page 5-140) and "Auto Load Function" (page 5-45).**
- **If loading is performed when the memory card data is stored in the CPU memory, loading will fail.**

4

LADDER PROGRAMMING

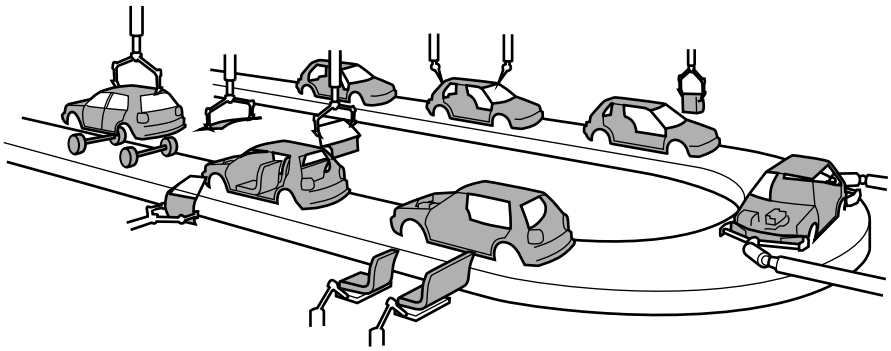
This chapter describes the specifications and setup methods of the functions that you have to use when creating ladder programs. You can improve programming efficiency by using modules, function blocks, macros and local devices.

4-1 What Is Sequence Control? 4-2
4-2 Programming Language..... 4-5
4-3 Projects 4-11
4-4 Devices 4-20
4-5 Variable 4-55
4-6 Constants 4-73
4-7 How Data Is Processed..... 4-75
4-8 Program Structure and Operation 4-82
4-9 Modules..... 4-98
4-10 Function Blocks.....4-117
4-11 Macros 4-178
4-12 Local Devices..... 4-200
4-13 Interrupts 4-206
4-14 File Register 4-226
4-15 Programming Skills 4-236

4-1 What Is Sequence Control?

In vehicle assembly lines such as the one shown below, the various operations, sequences, and conditions of the equipment and devices are recorded in control units in advance. When operations are started, the target operations are executed in sequence according to the status and conditions of the equipment and devices in order to assemble the vehicle. This is called sequence control.
PLCs (Programmable Logic Controllers) such as the KV-8000 Series are used as the devices for sequence control.

Body assembly line at a vehicle factory

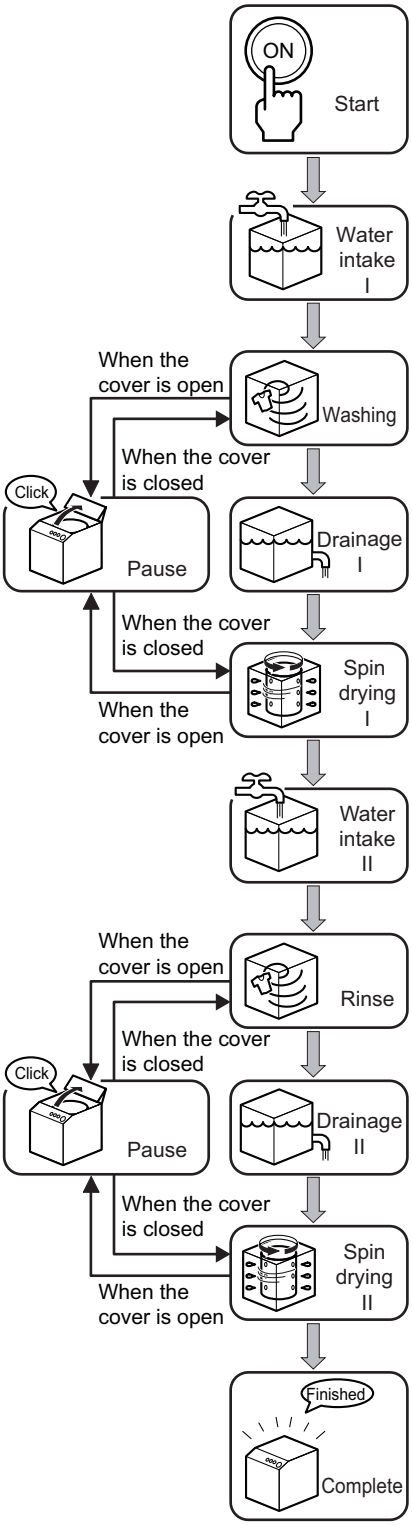


Reference

- In JIS (Japan Industrial Standards), sequence control is defined as "control for successively advancing each step of a control procedure according to a predetermined order or an order determined according to a fixed logic."
- In PLC control, in addition to sequence control, feedback control is also available. Feedback control is the process of detecting the current value of the object and comparing this with the target value and then repeatedly performing control until the values match.
  (Examples) Temperature control, pressure control, and speed control

■ Sequence control example

Let's take a close look at a fully automatic washing machine as an example of sequence control. When you press the start button, the fully automatic washing machine follows the predetermined order to automatically carry out the washing sequence. The washing sequence control is shown below.



(1) Insert the clothes to be washed, and press the start button.

(2) After the start button has been pressed, insert the detergent, and open the water intake valve.

(3) When it is detected that the specified water level has been reached, the water intake valve is closed, and the washing motor is started.

(4) When the specified time has elapsed, the washing motor is stopped, and the drainage valve is opened.

(5) When it is detected that drainage is complete (the water level is 0), the drainage valve is closed, and the spin dry motor is started.

(6) When it is detected that the specified time has elapsed, the spin dry motor is stopped, and the water intake valve is opened.

(7) When it is detected that the specified water level has been reached, the water intake valve is closed, and the washing motor is started.

(8) When it is detected that the specified time has elapsed, the washing motor is stopped, and the drainage valve is opened.

(9) When it is detected that drainage is complete (the water level is 0), the drainage valve is closed, and the spin dry motor is started.

(10) When it is detected that the specified time has elapsed, the spin dry motor is stopped.
The completion of the washing process is indicated, and a sound is generated as a notification of this.

Types of Sequence Control

Basic sequence control is divided into three main groups: sequence control, condition control, and time limit control. Target control circuits are designed by combining these types of control.

● Sequence control

In sequence control, the next operation in a predetermined sequence starts when an external signal, such as from a detector, indicates that the previous operation has finished.
Sequence control of the fully automatic washing machine
(2) Water intake I → Water intake finishes → (3) Washing (washing motor starts)
(4) Drainage I → Drainage finishes → (5) Spin drying I (spin dry motor starts)

Reference Typical ladder program instructions used in sequence control include STG and JMP.

● Condition control

In condition control, operations are performed when, in combination with the status signals of the object being controlled, predetermined conditions are met.
Condition control of the fully automatic washing machine
(3) Washing → The washing machine door is opened. → Pause → The washing machine door is closed. → (3) Washing
(5) Spin drying I → The washing machine door is opened. → Pause → The washing machine door is closed. → (5) Spin drying I

Reference Typical ladder program instructions used in condition control include STP, STE, CJ, NCJ, and LABEL.

● Time limit control

In time limit control, the operation instruction given to the object being controlled is stopped when the specified time elapses, as determined by a timer or similar device, or when the specified date and time is reached. The next operation is then started according to the predetermined order.
Time limit control of the fully automatic washing machine
(3) Washing → Detection of the specified time elapsing → (4) Drainage I
(5) Spin drying I → Detection of the specified time elapsing → (6) Water intake II

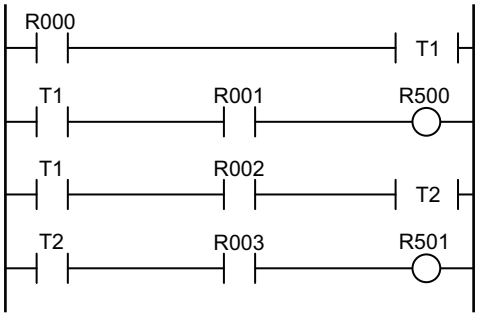
Reference Typical ladder program instructions used in time limit control include TMR and LDWK.

4-2 Programming Language

With KV STUDIO, you can use "ladder programs," "ST Language" and "KV Script" to create programs. "Mnemonics" are used as the language for operating the programs that you have created on the CPU unit. Also, the separation of 16-bit and 32-bit values and signed and unsigned values is determined not by the instructions, but by the suffixes used.

Ladder Program

Ladder circuits are schematics of actual wiring diagrams. The ladder circuits that you write in KV STUDIO are called ladder programs.
In KV STUDIO, in addition to the instructions for general switches and lamps that are available, you have access to instructions such as useful ones for controlling ON time and OFF time and for PID control. You can easily write sequence control programs by combining these instructions.

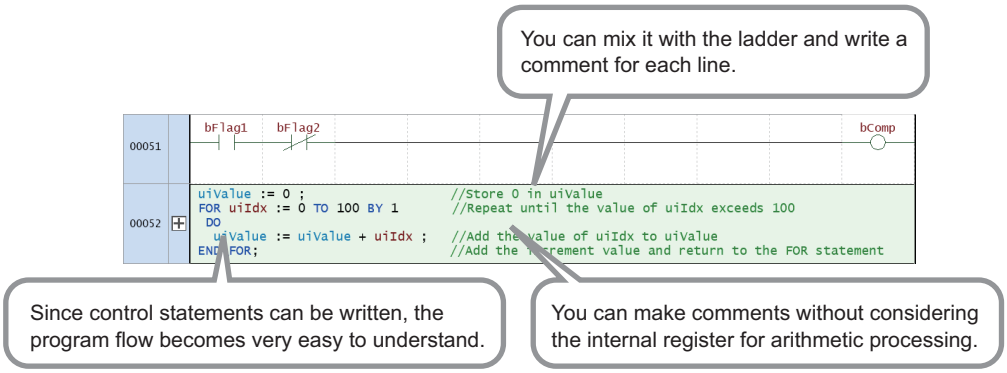


In addition, you can use expanded ladder mode with KV STUDIO to reduce the number of ladder rungs and make processing sequences clearer, both of which improve the readability of your programs.

"Expanded ladder mode" (page 4-258)

ST Language

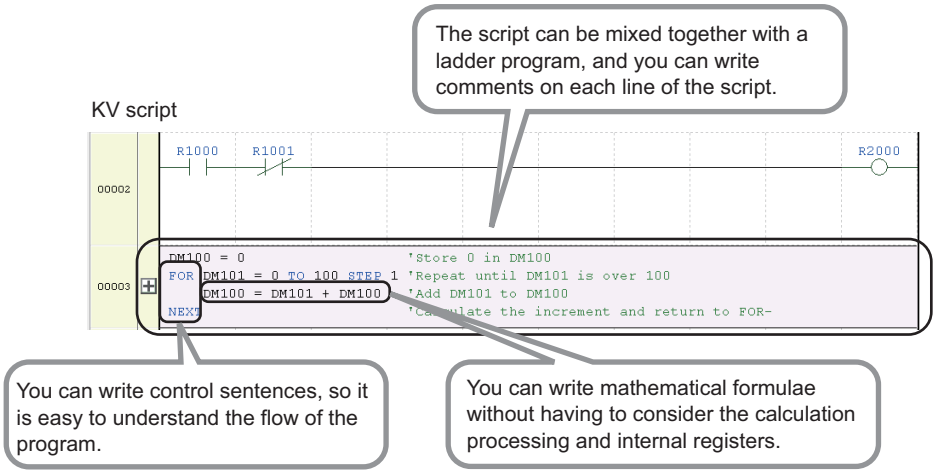
While Ladder Program is suitable for describing Sequence Control, it is often cumbersome to describe numerical processing such as the complex calculation and character string, and difficult to debug and maintain.
You can use ST Language instead of Ladder Program to describe complex parts, thereby effectively creating programs.



Point **To use the ST language, CPU function version 2.1 or later is necessary.**

KV Script

Although ladder programs are suitable for sequence control programs, using ladder programs for complicated calculation processing that handles values and processing that handles character strings leads to complicated programs, which makes debugging and maintenance difficult.
KV script is a programming language that has been developed to compensate for these ladder program problems. By using scripts as substitutes for the parts of a program that become too complicated when written as a ladder program, you can create programs efficiently.



Reference

Script programs are converted into ladder programs before they are executed.
You can monitor the operations of script programs after they have been converted into ladder programs.

KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Script Programming Manuals"

Mnemonics

Mnemonics are the language that is used to make the ladder programs that you have created operate on the CPU unit.
By simplifying mnemonics, expanded ladder programs are expressed in a manner that is easy to understand visually.
Expanded ladder programs are converted to mnemonics lists when they are transmitted to the CPU unit.
You can edit the mnemonics list in KV STUDIO, but this is not required when you create normal ladder programs.

LD	R000
OUT	R500
LD	R002
OUT	R501
LD	R003
OUT	R502

Reference

- To edit the mnemonics list, on the "Edit" menu, click "Edit list."
  Alternative procedure Ctrl + D
- Mnemonics lists are included in this manual for CPU built-in functions and instruction sample programs. You can use these mnemonics lists to easily enter sample programs.
  "How to Use Mnemonic Lists" (page 16)
- When you create expanded ladder programs, you can reduce the production costs of programming by entering instructions as mnemonics.
  "Mnemonic entry" (page 4-238)

Suffixes

Suffixes are appended after an instruction. You can use them to change the way that device data is handled. There are the following seven types of suffixes.

□.U □.S □.D □.L □.F □.DF □.A

* The device with corresponding suffix is input in □.

Example

```
MOV.(U)
MOV.S
MOV.D
MOV.L
MOV.F
MOV.DF
MOV.A
```

□.U Data is processed as 16-bit, unsigned BIN data (0 to 65535).

□.S Data is processed as 16-bit, signed BIN data (-32768 to +32767).

□.D Data is processed as 32-bit, unsigned BIN data (0 to 4294967295).

□.L Data is processed as 32-bit, signed BIN data (-2147483648 to 2147483647).

□.F Data is processed as single precision floating-point real number data.

-3.4E + 38 ≤ N ≤ -1.4E - 45
N = 0
+1.4E - 45 ≤ N ≤ +3.4E + 38
(Approximately 7 significant digits)

□.DF Data is processed as double precision floating-point real number data.

-1.79E + 308 ≤ N ≤ -2.23E - 308
N = 0
+2.23E - 308 ≤ N ≤ +1.79E + 308
(Approximately 16 significant digits)

□.A The data type processed by the instruction varies depending on the data type of the variable used by the operand. The .A suffix is only supported by some instructions. For details, refer to the "Instruction Reference Manual".

In KV STUDIO, □.U is not displayed. Also, instructions that do not have a suffix are processed as 16-bit, unsigned BIN data.

Instructions to which suffixes are applied are executed with BIN data treated as decimal values according to the suffix.

Therefore, if the suffix is different, the results of executing an instruction may differ even if the BIN data is the same.

Point

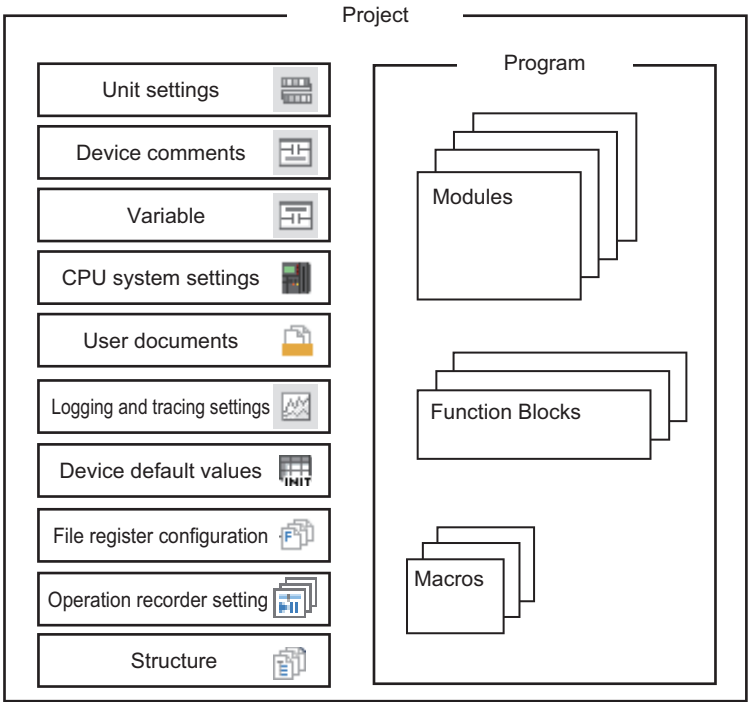
The .A suffix can be used on KV-8000 Series CPU units with CPU function version 2.3 or later.

MEMO

Project Configuration

■ What are projects?

All the information that is required to operate the KV-8000 Series, such as the program and unit settings, is managed in projects.
When you create a new project in KV STUDIO, a folder with the same name is generated automatically.
The information contained in each project is shown below.



Point **You can use one project to manage the information of only one KV-8000 Series.**

● Program

A program is constructed from modules, function blocks and macros.
By repeatedly executing this program, you can perform various types of control.

"4-8 Program Structure and Operation" (page 4-82)
"4-9 Modules" (page 4-98)
"4-10 Function Blocks" (page 4-117)
"4-11 Macros" (page 4-178)

● Unit settings

Use this to manage the unit configuration and the detailed settings of each unit. You can use KV STUDIO to easily configure the settings.

See the "User's Manuals of each expansion unit".
"Chapter 3 UNIT EDITOR" in the "KV STUDIO User's Manual"

● Device comment

By assigning a comment to each device, you can make it easier to see what functions are assigned to each device.

"Device comments" (page 4-252)

● Variable

Set the variable name and data type. When using variables, any names defined can be used for programming, which makes the program more readable.

"4-5 Variable" (page 4-55)

● Structure

Set the structure. When using structures, variables with the same structure can be registered / monitored in batch.

"Structure" (page 4-59)

● CPU system settings

Set the following items.

Classification	Item	Description	Reference Page
Program setting	Execute sequence of modules	Set the execution sequence of the modules within the scan.	"Module Execution Sequence" (page 4-109)
	Fixed-period execution and user interruption configuration	It configures the fixed-period module and priority of interruption of user interruption.	"Scan Time" (page 4-90) "Interrupt Priority" (page 4-209)
	Scantime related	Set the fixed scantime operation and the END processing time	"Scan Time" (page 4-90)
	Protection setting	Set the project's read/write protection.	"Project Protection Settings" (page 4-16)
	Other settings	Defining the language used for a project.	"Defining the project language" in the "KV-STUDIO User's Manual"
System setting	Entire assignment setting of local devices	Assign all local devices and configure the settings of each program.	"Local Device Assignment" (page 4-202)
	CPU unit buffer capacity	It configures buffer capacity when logging trace function is used.	"Option" (page 5-67)
	Device recorder buffer capacity setting	Set the device recorder buffer capacity when using the operation recorder function	"Option" (page 5-67)
	Power off holding	Set the devices whose status and value will be maintained when the power turns off.	"Configuring Power Off Holding Settings (Latching)" (page 4-26)
	Event/error setting	Perform the settings related to the operation / stop status when an error occurs or the recording of device value change events.	"Event/error settings" (page 4-86)
	Setting the upper limit of the digital trimmer	Disable/Enable the upper/lower limit of the digital trimmer.	"Digital Trimmer" (page 5-128)
	Access Window Setup	Specify the language, the comment number and other settings you want to display in the access window.	"Access Window" (page 5-111)
	Other settings	Set the data transfer settings used by the ladder monitor function of VT5-WX and whether to display the dialog for assigning variables to the EtherNet / IP connection.	"KV STUDIO User's Manual" "Other Settings"

● Logging and tracing settings

Set the logging trace function of device data.

"5-5 Logging/Tracing" (page 5-57)

● Device default values

Set the device default values that the specified devices will be automatically initialized to when operation starts.

"Device defaults" (page 4-257)

Program Capacity

The maximum program capacity is as follows.

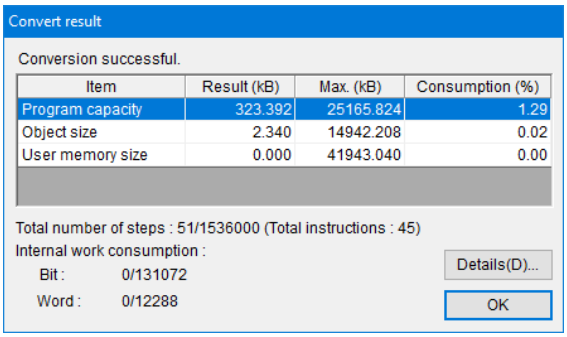
Item		KV-8000(A)
Program capacity		64 MB*1
Object size		14.25 MB
Size of one module		16 MB*2
Number of modules		4,000
Number of function blocks + Number of macros		4,000
Number of module objects		4,000
Number of steps		1,500 k steps
Internal work consumption	Bits	131,072 bits
	Words	12,288 words
Maximum global device comment and local device comment size		No limitation
Total maximum size of setting data and logged data for individual units		3,136 KB
Size of transferred data during circuit block transfer		384 kB
Structure definition size		2 MB

*1 You can use the CPU memory capacity setting to change size.
*2 For CPU function version 2.0 or later. 4MB for CPU function version 1.0.

The percentage of the program capacity that is used is calculated automatically by KV STUDIO when you convert the program and is displayed in the "Convert result" dialog box, so you do not have to perform the calculations and manage this information by yourself.

Maximum program capacity can be set by CPU memory capacity setting.

"CPU memory" (page 5-29)



Point

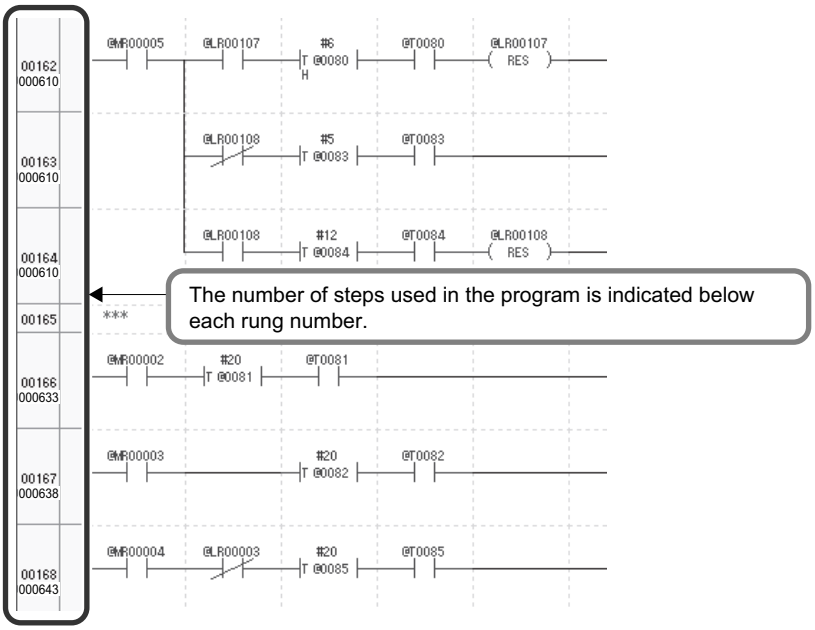
You will not be able to transmit or run the program if the number of steps, program size, object size or internal work consumption is exceeded.

Reference

Program capacity

The byte size of each instruction is fixed, but the amount of memory required varies depending on the types of operands, the instruction suffixes, and the presence of index modification.
In the program capacity calculation, the number of bytes used by row comments, device comments, variable, device default values, unit setup information, operation recorder setting, recording and tracking setting information and CPU system settings are added to the total number of bytes used by the instructions. The size of device comments, rung comments and variable varies according to the number of characters used.
The above results assume that the units are in their default settings. Changing the CPU memory capacity setting will also change the number of device comments and variable that can be managed.

<Total steps>
The total number of steps is the total number of instructions and rung comments used in the project.
The number of steps used is displayed on the left side of the ladder diagram after the program is converted by KV STUDIO.



<Internal work consumption>
This is the total data that is used internally, such as the previous status of the devices specified by the operands of each instruction.

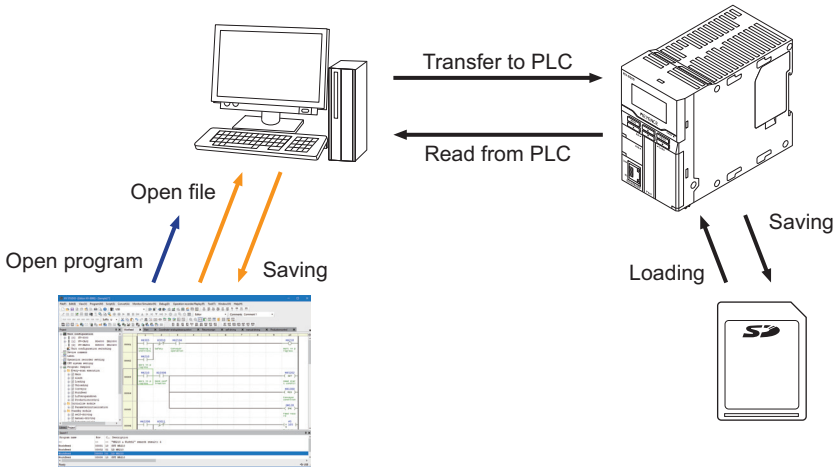
Project Protection Settings

■ Overview of protection settings

Protection can be set for the project created by KV STUDIO.

To configure protection settings, click "CPU system setting" in KV STUDIO.

Item	Description
Write protection	The ladder programs stored in the CPU unit are write protected (a confirmation dialog will be displayed when an attempt is made to write to these programs). You can use this to prevent the programs from being overwritten by mistake.
Read protection	The ladder programs stored in the CPU unit are read protected (it will not be possible to read these programs). You can use this to prevent your valuable programs from being leaked.
Project password	To read the ladder programs stored in the CPU unit, you will be prompted to enter the password in the dialog box that appears). * When you have set read protection, the password entry dialog box will not be displayed, and you will not be able to read the ladder programs.
Program passwords	When you are editing a project, if you attempt to display the program, a dialog box will be displayed asking you to enter the password.



● Editing projects from a PC

- Opening a file

  Protection cannot be set on this operation.
- Saving a file

  Protection cannot be set on this operation.
- Opening a program

  You can set a password on the program so that a dialog box asking the user to enter the password is displayed when they attempt to open the program.

● Reading and writing projects between a PC and the CPU unit

- Reading projects from the base unit (1)

  You can set read protection to prevent projects from being read.

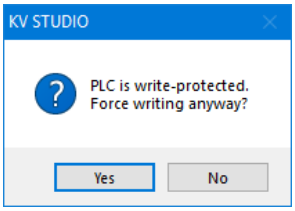
  In this situation, even when the project password has been set, the confirmation dialog box will not be displayed.

- Reading projects from the base unit (2)

  You can set a project password so that a dialog box asking the user to enter the password will be displayed.

- Writing projects to the base unit

  You can set write protection so that a confirmation dialog box is displayed when an attempt is made to overwrite the project.



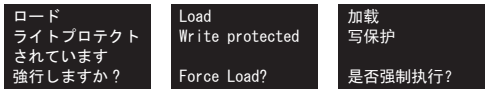
Reference

When you have set a project password, if the password authentication fails, you will have to wait a set amount of time until you can read the project again. Also, the wait time gets longer each time that the authentication fails. The project password authentication disabled status relay (CR2314) turns ON when you can no longer perform read authentication due to repeated password authentication failures.
The number of consecutive password authentication failures and the total number of password authentication failures are stored in CM1658, the number of consecutive project password authentication failures, and CM1659, the total number of project password authentication failures.

● Using a memory card to load and save projects

- Loading

  When you have set write protection, confirmation is performed before a project is loaded.



- Saving

  When read protect and project password is configured, save can be executed, but it only saves value of the device and ladder program will not be saved.

■ Project passwords

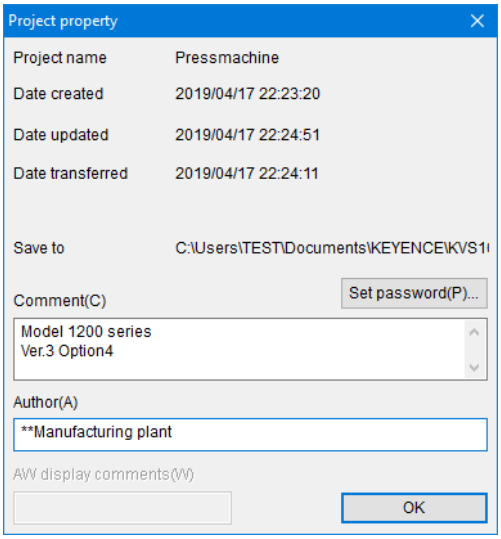
You can set passwords on projects to place restrictions on what can be read from the CPU unit.
Follow the procedure below to set a password.

1 Click "Workspace."

2 Right-click to display a menu, and then click "Project Properties."

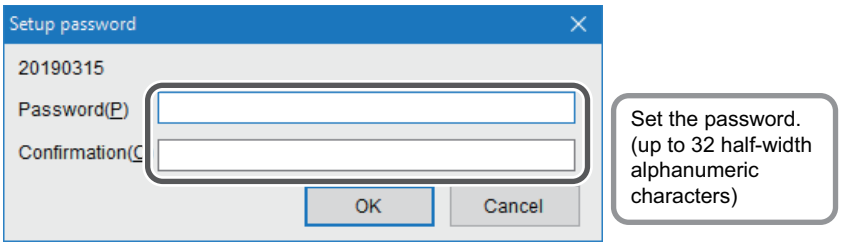
Alternative procedure On the "File" menu, click "Project Properties."

The "Project property" dialog box is displayed.



3 Click "Set password."

The "Setup password" dialog box is displayed.

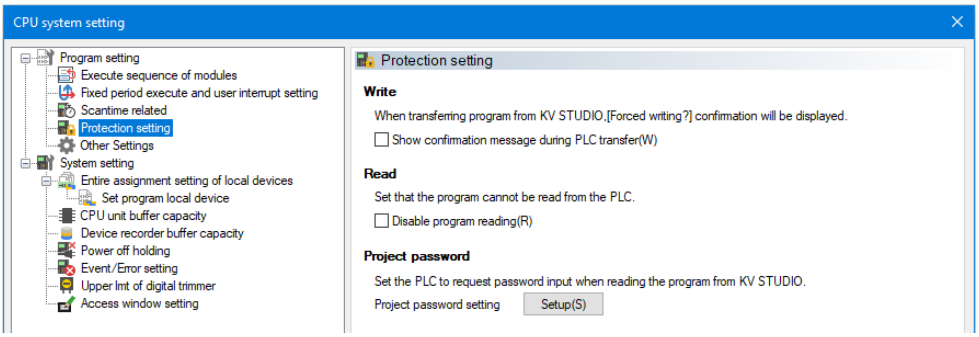


4 Enter the password in both text boxes (the second is for confirmation), and then click "OK."

After you set the password, when an attempt is made to read this project from the CPU unit, a dialog box will be displayed asking the user to enter the password.
The user has to enter the password to read the project from the CPU unit.
Also, when you transfer a project that has a different password to the PLC, to protect the projects on the PLC, you have to enter the password, or transfer all items.

■ Project write and read protection

You can set protection on projects to place restrictions on writing to and reading from the CPU unit. To configure protection settings, in the workspace, click "CPU System Setting" and then "Protection Setting."



Select the checkboxes, and then transfer programs to the CPU unit to enable the protection settings.

Point

If you transfer programs to the CPU unit when you have set read protection, you will not be able to monitor the execution status of the ladder program from that point.
After you finish debugging, set read protection, and then transfer programs to the PLC.
If you are using a password and read protection at the same time, the password confirmation dialog box will not be displayed, and you will not be able to read projects.

Reference

For details on program passwords, see "Module Passwords" (page 4-114), "Setting a Function Block Password" (page 4-141) and "Macro Passwords" (page 4-191).

4-4 Devices

As elements used in the program, KV-8000 (A) provides variables that can be freely set by the user and devices that can be specified with dedicated symbols and numbers (such as Relay, Timer, Counter, and Data memory). Both are written after the instruction and used to give the data used in the instruction or to store the execution result of the instruction. This section describes the Device.

Lists of Devices

■ Bit devices

Device Name	Range*1	Number of Points	Main Function	Start of Operation		Local Device (@)*1	Index Modification (:Z/:#)	Indirect Specification (*)*7	
				Power On	PROG → RUN			Reference	Storage
Input/output/ internal auxiliary relay*9	R00000 to R199915*9	32000	Devices used by the CPU and by expansion units	Clear/hold	Clear/hold	Yes	Yes	Yes	No
Link relay	B0000 to B7FFF*8	32768	Devices that can only be used internally by the CPU	Clear/hold	Clear/hold	No	Yes	Yes	No
Internal auxiliary relay	MR000000 to MR399915	64000		Clear	Clear	Yes	Yes	Yes	No
Latch relay	LR00000 to LR99915	16000		Hold	Hold	Yes	Yes	Yes	No
Timer (contact)	T0000 to T3999	4000	Contact device for the 10 µs, 1 ms, 10 ms, and 100 ms down timers and for the 10 ms up/ down timer	Clear	Clear	Yes	Yes	Yes	No
Counter (contact)	C0000 to C3999	4000	Contact device for the up counter and the up/down counter	Clear/hold*6	Hold	Yes	Yes	Yes	No
Control relay*3	CR0000 to CR7915*3	1280	Devices used to control the PLC functions and to capture the PLC status	Clear	Clear	No	No	No	No

■ Word devices

Device Name	Range	Number of Points	Main Function	Start of Operation		Local Device (@)	Index Modification (:Z/:#)	Indirect Specification (*)	
				Power On	PROG → RUN			Reference	Storage
Data Memory	DM00000 to DM65534	65535	Devices that can store numerical data (16 bit)	Clear and hold*2	Hold	Yes	Yes	Yes	Yes
Expansion data memory	EM00000 to EM65534	65535		Clear and hold*2	Hold	Yes	Yes	Yes	Yes
File register	Consecutive number mode: ZF000000 to ZF524287	524288		Clear and hold*2	Hold	No	Yes	Yes	Yes
	Current bank: FM00000 to FM32767	(32768 ×16 banks)				Yes*10			
Link register*8	W0000 to W7FFF*8	32768		Clear and hold*2	Hold	No	Yes	Yes	Yes
Temporary memory	TM000 to TM511	512	Devices that can store numerical data (16 bit)	Clear	Clear	Yes	Yes	Yes	Yes
Timer (current value and set value)	T0 to T3999	4000	Current value and set value of the 10 µs, 1 ms, 10 ms, and 100 ms down timers and of the 10 ms up/down timer (32-bit)	Current value: Clear*4 Set value: Hold	Hold	Yes	Yes	Yes	No
Counter (current value and set value)	C0 to C3999	4000	Current value and set value of the down counter and up/down counter (32-bit)	Clear/hold*6	Hold	Yes	Yes	Yes	No
Index register	Z1 to Z23	23	Devices that are used with index modification (32-bit)	Clear	Clear	No	No	No	No
Control memory	CM0000 to CM7599	7600	Devices used to control the PLC functions and to capture the PLC status (16 bit)	Hold	Hold	No	No	No	No

*1 Because this is used as a local device, the applicable range of global devices is limited by the number of reserved local devices.
"List of Local Devices" (page 4-201)
*2 You can change the clear/hold setting by using "Power off holding" in the CPU system settings.
*3 "Lists of CR and CM Devices" (page A-10)
*4 Up/down counters (UDT) hold the current value.
*5 Z11~Z23 are reserved for the system, so cannot be used.
*6 You can change the hold/clear setting by using "Power off holding" in the CPU system settings. (This only applies to contacts and current values.)
7 The devices that have "Yes" in the Storage column can be written as " (device)". The devices that have "Yes" in the Reference column can be specified indirectly Z11 and Z12 are reserved for the system, so you cannot use these devices.
"Indirect Specification" (page 4-46)
*8 Use hexadecimal numbers to specify the device numbers.
*9 "DR" is the label for direct processing. When the label is "DR", local devices cannot use indirect specification.
"Direct processing" (page 4-83)
*10 Local devices cannot perform bank switching.

■ Function block argument

The arguments used in the function block only are as shown below.

Argument type		Argument name		Description	Indirect Specifying (*)	Local Device (@)	Index Modify(:)
IN		Any argument name		Device values or constants are used as argument for processing.	No	No	No
OUT		Any argument name		Written to the device that specified the operation results.	No	No	No
IN-OUT		Any argument name		Devices are used as argument.	No	No	Yes*1
Unit	Leading R	UR	UR0 to UR9	Leading relay (R) assigned to the unit is processed as an argument.	No	No	Yes
	Leading DM	UM	UM0 to UM9	Leading data memory (DM) assigned to the unit is processed as an argument.	No	No	Yes
	No.	UV	UV0 to UV9	Unit number is processed as an argument.	No	No	No

*1 Arguments that specify arrays or structures cannot use index modification.
"Setting Arguments" (page 4-131)

■ Macro argument devices

The macro argument devices, which can only be used in macros, are listed below.

Attribute		Symbol	Range	Description	Indirect Specifying (*)	Local Device (@)	Index Modify(:)
Device		P	P0 to P9	The device is received as the argument.	No	No	Yes
Value		V	V0 to V9	The device's value or constant is received as the argument.	No	No	No
Unit device	Leading R	UR	UR0 to UR9	The leading relay (R) assigned to the expansion unit is received as the argument.	No	No	Yes
	Leading DM	UM	UM0 to UM9	The leading data memory (DM) entry assigned to the expansion unit is received as the argument.	No	No	Yes
	No.	UV	UV0 to UV9	The unit number of the expansion unit is received as the argument.	No	No	No

"Setting Arguments" (page 4-186)

■ Constants

Constant	Range	Main Function
Decimal (#)	#0 to #65535 (unsigned 16 bit) -32768 to +32767 (signed 16 bit) #0 to #4294967295 (unsigned 32 bit) -2147483648 to +2147483647 (signed 32 bit)	Used to specify decimal constants. Reference When you select XYM marking mode, this is displayed as "K."
Single precision floating-point real number	-3.4E + 38 ≤ N ≤ -1.4E - 45 N = 0 1.4E - 45 ≤ N ≤ +3.4E + 38 (Approximately 7 significant digits)	Used to specify single precision floating-point constants.
Double precision floating-point real number	-1.79E + 308 ≤ N ≤ -2.23E - 308 N = 0 2.23E - 308 ≤ N ≤ 1.79E + 308 (Approximately 16 significant digits)	Used to specify double precision floating-point constants.
Hexadecimal ($)	$0000 to $FFFF (16 bit) $00000000 to $FFFFFFFF (32 bit)	Used to specify hexadecimal constants. Reference When you select XYM marking mode, this is displayed as "H."
Fixed character string (" ")	Examples) "Model number ABC", "MON/08:30:15"	Used to specify fixed character strings.

■ Internal registers

Device Name	Range	Number of Points	Main Function	Start of Operation (Power On, PRG → RUN)
Internal register	-	1	A device used to temporarily store data (16 bit, 32 bit, 64 bit)	Clear

Device notation method

● Writing index modification (modification according to index registers and according to index constants)

• Modification according to index registers

After the device (R (DR) / MR / LR / B / T / C / DM / EM / FM / ZF / W / TM / *DM / *EM / *FM / *ZF / *W / *TM), append ":Zn" (where "n" is a number from 1 to 10).

Example

DM0:Z01	**(If the value of Z1 is 100, this means DM100.)**
R1000:Z01	**(If the value of Z1 is 100, this means R1604.)**

• Modification according to index constants

After the device (R (DR) / MR / LR / B / T / C / DM / EM / FM / ZF / W / TM / *DM / *EM / *FM / *ZF / *W / *TM), append ":#n" (where "n" is a number from -524287 to 524287).

Example

DM0:#1	**(This means DM1.)**
R1000:#1	**(This means R1001.)**

• Index modified using mathematic formula

It successively enters the computation formula after the device (R(DR) / MR / LR / B / T / C / DM / EM / FM / ZF / W / TM / *DM / *EM / *FM / *ZF / *W / *TM).

Example

DM0: (DM100+20)	**(This means DM70 if DM100 is 50.)**
R1000: (DM0*4)	**(This means R1008 if the value of DM0 is 2.)**

Point

Indices modified using mathematic formula can be used only in KV scripts.

KV-8000/7000/5000/3000/1000 and KV Nano Series Script Programming User's Manual

● Writing indirect specification (*)

Enter "*" before the device (DM/EM/FM/ZF/W/TM). Use this together with the indirect specification instruction.

Example

***DM0 *TM100**

"Indirect Specification" (page 4-46)

● Writing local devices

Enter "@" before the device (R/MR/LR/T/C/DM/EM/FM/TM).

These devices are only valid in programs. You can also perform indirect specification and index modification.

Example

@DM0 @R10000

"4-12 Local Devices" (page 4-200)

● XYM marking

You can use the following XYM marking to specify devices (R (DR)/MR/LR/ DM/EM/FM).

Device Name	KEYENCE Marking	XYM Marking	Device Number during XYM Marking
Relay	R (DR)	X (DX)	0000 to 1999F*1
		Y (DY)	0000 to 1999F*1
Internal auxiliary relay	MR	M	00000 to 63999
Latch relay	LR	L	0000 to 15999
Data memory	DM	D	00000 to 65534
Extended data memory	EM	E	00000 to 65534
File register	FM	F	00000 to 32767

*1 The device numbers of 0000 to 199915 are also displayed during XYM marking for R to which the input/output is not assigned.
To switch to XYM marking, in KV STUDIO, on the "Tools" menu, click "Options," "Display mode setting," and then "XYM marking."

● Entry specifying bits of a word device

Devices (DM / EM / FM / ZF / W / TM / CM) Enter.□ (□: 0 - 15 values) here.

Example

DM0.0	**(Indicates the first bit of DM0)**
EM10.4	**(Indicates the fifth bit of EM10)**

"Bit device processing of word devices" (page 4-52)

● Suffix entry

Devices:(R (DR) / MR / LR / B / T / C / DM / EM / FM / ZF / W / TM / CM / *DM / *EM / *FM / *ZF / *W / *TM) Enter .□ (□:U, S, D, L, F, DF, B, T) here.

Example

DM0.F	**(Values DM0 - DM1 are calculated as floating point data)**
R1000.U	**(calculated as 16-bit data between R1000 - R1015)**

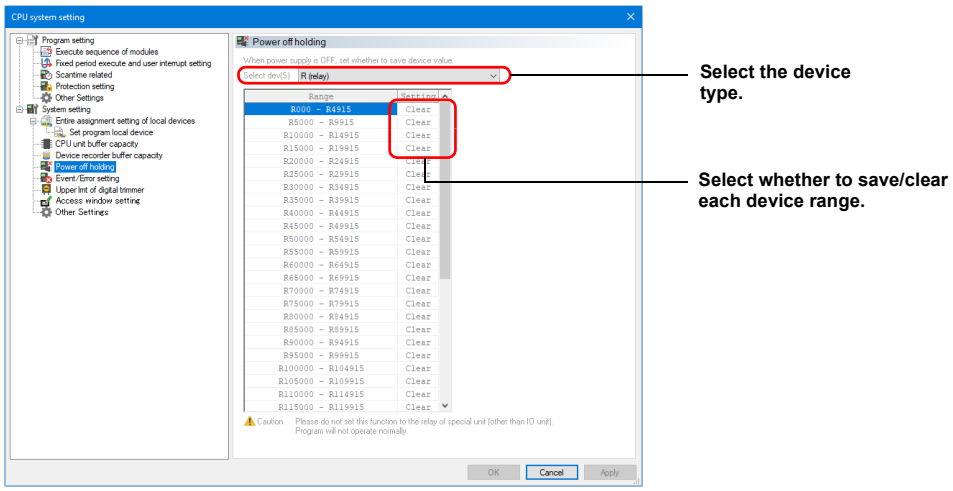
Point

Suffix writing can be used only in ST/Script.
KV-8000 Series ST Language Programming Manual
KV-8000/7000/5000/3000/1000 and KV Nano Series Script Programming User's Manual

Configuring Power Off Holding Settings (Latching)

Power off holding is a function that can be set to retain or clear device data in the event of a power outage.

To configure this setting, open the CPU system setting from the <<KV STUDIO>> workspace and select "System settings" "Power off holding."



Bit Devices

I/O relay R

Overview

The input relays are the devices that are used to capture the ON/OFF information from the peripherals and to transfer this information to the PLC.
The output relays are the devices that are used to transfer the ON/OFF instructions and the calculation results from the PLC to the PLC peripherals.

Notation

Device numbers are structured as shown below.
(Example)

Channel number (000 to 1999) Contact number (00 to 15)

R01012

Device name Device number (00000 to 199915)

Reference

- "R" can be omitted when you write the device. (However, you cannot omit "R" for operands in which you can enter a constant.)
- You can use zero suppression when you write your programs
  (Example) R12
- This can be used as a word device.
  "Processing Bit Devices in Units of Words" (page 4-50)
- If you include "D" in the device name, the relay will operate as a direct I/O relay.
  (Devices that do not support direct I/O will operate as standard relays even when "D" is included in the device name.)
  (Example) DR5000 R5000 will operate as a direct output relay.
  "Direct I/O relay (DR)" (page 4-83)
- When you select XYM display mode, input relays are displayed as "X" and output relays are displayed as "Y."
  "XYM marking" (page 4-25)

Range

R00000 to R199915 (can be assigned as expansion I/O unit numbers)

Application

Input relay
- Used for connected peripherals such as buttons and switches (selector switches, limit switches, photoelectric switches, and digital switches).
- Can be used as the high-speed inputs of the interrupts, high-speed counters, and the positioning function when assigned to an expansion unit.

Output relay
- Used for connected peripherals such as solenoids, electromagnetic switches, and displays (signal lamps and digital displays).
- Can be used as the high-speed outputs of the high-speed counter positioning function when assigned to an expansion unit.

Details

- One bit of the input relay is assigned per one input terminal.
- One bit of the output relay is assigned per one output terminal.
- You can specify a contact (normally open) or b contact (normally closed).

Default setting When the power is turned on and when the mode switches from PRG to RUN, the status is cleared and the devices are turned OFF.

Reference

- Device notation for relay (R) units
  In KV STUDIO, aside from the normal notation, you can use a relay notation in which the unit numbers are the reference. To use the device notation for units, click "Display unit device." on the "View" menu, or press Shift + Space on the ladder edit window. When using the unit device notation, the leading device number becomes to 00000, so you do not have to consider the leading device number of the unit when referring to the manual.

Unit number — Unit relay number

R05_00010

Channel number — Contact number

You can use zero suppression when you write your programs.
(Example) R5_10

Unit number: Starting with the CPU unit, numbers are assigned in the order (00, 01, 02, 03, 04 and so on) that the units are connected.

Unit relay number: Starting from 00000, the number of relay points that will be occupied are assigned as the relay number.

Unit number	00	01		02	03	04	05
KV-PU1	KV-8000 CPU	KV-SIR32XT 32 input points 32 output points	Bus connection	KV-C16XTD 16 input points 16 output points	KV-C32XC 32 input points	KV-C64TD 64 output points	KV-B8XTD 8 input points 8 output points
Unit relay number:		00000 to 00115		00000 to 00015	00000 to 00115		00000 to 00015*1
		00200 to 00315		00100 to 00115		00200 to 00315	00100 to 000115*1

(Example) 5th KV-SIR32XT input point.....01_0004
8th KV-C32XC input point.....03_0007
18th KV-C64TD output point.....04_0101
4th input of KV-B8XTD.....05_0003
4th output of KV-B8XTD .....05_0103

*1 In the case of KV-B8XTD, each input / output relay occupies 1 CH, and only the first 8 bits are valid.

Link relay B

Overview

These relays are used to perform data linking in simple PLC links etc. Hexadecimal notation is used to specify device numbers, which is different from other relays.

Notation

Device numbers are structured as shown below.
(Example)

Channel number (000 to 7FF) Contact number (0 to F)

B1A6F

Device name Device number (0000 to 7FFF)

Reference

- You can use zero suppression when you write your programs.
  (Example) B7A
- This can be used as a word device.
  "Processing Bit Devices in Units of Words" (page 4-50)

Point

- **Use hexadecimal numbers (0 to F) to specify device numbers.**
- **You cannot use local devices (@).**

Range

B0000 to B7FFF

Application

These relays can be used to perform data linking in EtherNet/IP communication etc.

Details

These cannot be assigned as unit devices of expansion units.

Default setting

When the power is turned on and when the mode switches from PRG to RUN, the status is cleared and the devices are turned OFF.

Internal auxiliary relay R, B, MR, and LR

■ R

Overview You can use R devices that are not assigned to an expansion unit relay as internal auxiliary relays.

Notation Device numbers are structured as shown below.
(Example)

Channel number (000 to 1999) Contact number (00 to 15)

R01012

Device name Device number (00000 to 199915)

Reference
- "R" can be omitted when you write the device.
- You can use zero suppression when you write your programs.
  (Example) R12
- This can be used as a word device.
  "Processing Bit Devices in Units of Words" (page 4-50)

Range 00000 to 199915 (excluding the numbers that are assigned as expansion input unit numbers)

Default When the power is turned on and when the mode switches PRG ⇔ RUN, the status is cleared and the devices are turned OFF.

■ B

Overview You can use B devices that are not assigned to the link relays of EtherNet/IP communication etc. as internal auxiliary relays.

Notation Device numbers are structured as shown below.
(Example)

Channel number (000 to 7FF) Contact number (0 to F)

B1EE7

Device name Device number (0000 to 7FFF)

Reference
- You can use zero suppression when you write your programs.
  (Example) B7A
- This can be used as a word device.
  "Processing Bit Devices in Units of Words" (page 4-50)

Point
- **Use hexadecimal numbers (0 to F) to specify device numbers.**
- **You cannot use local devices (@).**

Range B0000 to B7FFF

Details These cannot be assigned as unit devices of expansion units.

Default setting When the power is turned on and when the mode switches PRG ⇔ RUN, the status is cleared and the devices are turned OFF.

■ MR

Overview These devices can only be used with CPU internal functions.

Notation Device numbers are structured as shown below.
(Example)

Channel number (0000 to 3999) Contact number (00 to 15)

MR002809

Device name Device number (000000 to 399915)

Details These cannot be assigned as unit devices of expansion units.

Default setting When the power is turned on and when the mode switches from PROG to RUN or from RUN to PRG, the status is cleared and the devices are turned OFF.

Reference
- You can omit "R" when writing "MR."
- You can use zero suppression when you write your programs.
  (Example) M2809
- When you select XYM marking mode, this is displayed as "M."
  "XYM marking" (page 4-25)

■ LR (Latch relay)

Overview These devices can only be used with CPU internal functions. These devices are used to hold the status of devices.

Notation Device numbers are structured as shown below.
(Example)

Channel number (000 to 999) Contact number (00 to 15)

LR02308

Device name Device number (00000 to 99915)

Range LR000 to LR99915

Details These cannot be assigned as unit devices of expansion units.

Default setting The status of another device is held in this device.

Reference
- You can omit "R" when writing "LR."
- You can use zero suppression when you write your programs.
  (Example) L2308
- When you select XYM marking mode, this is displayed as "L."
  "XYM marking" (page 4-25)

Timer (contact) T

Overview

Timers have a contact (1 bit), a set value (32 bits), and a current value (32 bits). These devices function as follows: when the current value of a count-down timer is 0 (time is up) or when the current value of a count-up or count-down timer matches the set value or 0, the contact turns ON.

Notation

Device numbers are structured as shown below.
(Example)

Device number (0000 to 3999)

T0501

Device name

The contact, current value, and set value are determined by the operands of the instruction used.

Range

T0000 to T3999

Details

There are five types of timers: the 1 ms timer, the 10 ms timer, the 100 ms timer, the 10 μs timer, and the up/down timer.

Default setting

When the power is turned on and when the mode switches from PRG to RUN, the status is cleared and the devices are turned OFF.
When the power is turned on and when the mode switches from PRG to RUN, the current value is set to the same value as the set value.
The set value is held.

Reference

You can use zero suppression when you write your programs.
(Example) T501

■ How to program Timer operations

To program timer operations, specify in the timer instruction operands the timer number and settings (the length of time until time is up) of the timer that you want to use.
(Example)

Contact number (0000 to 3999) Set value (0 to 4294967295)

TMR 0001 #1000
UDT 0002 #1000

Timer instruction

Reference

- You can also specify a device in the set value.
  For details, see "Timer and Counter Instructions" in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."
- You can use a device with the same number with both a timer and counter.

Operation differences between timer types

Timer Type (Instruction)	Device Number	Set Value	Time until Contact Turns ON
100 ms count-down timer (TMR)	0000 to 3999	0 to 4294967295	n × 100 ms
10 ms count-down counter (TMH)			n × 10 ms
1 ms count-down counter (TMS)			n × 1 ms
10 μs count-down timer (TMU)			n × 10 μs
10 ms up/down timer (UDT)			n × 10 ms

■ Timer error

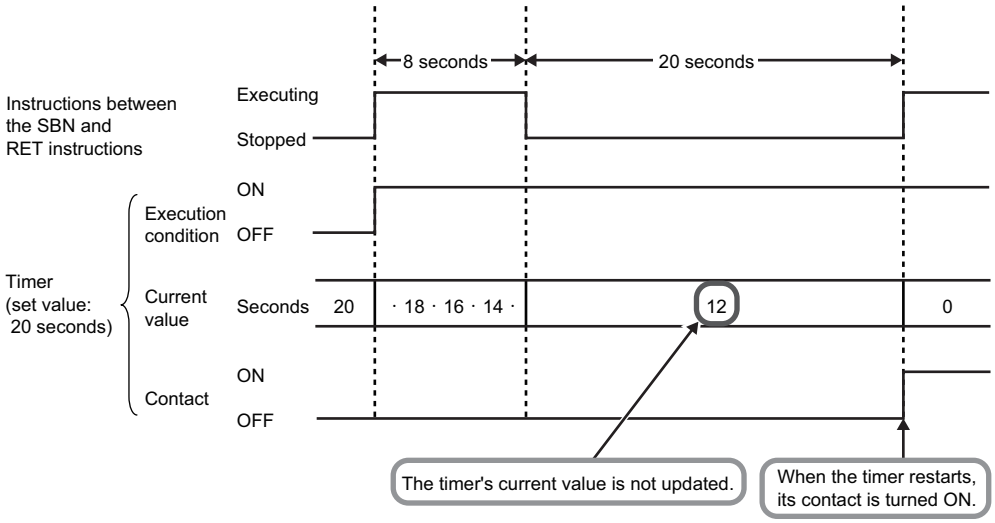
Timer error is shown below:

- TMR (100 ms timer): Within + (±100 ms + 1 scan time)
- TMH (10 ms timer): Within + (±10 ms + 1 scan time)
- TMS (1 ms timer): Within + (±1 ms + 1 scan time)
- TMU (10 μs timer): Within + (±10 μs + 1 scan time)
- UDT (up/down timer): Within + (±10 ms + 1 scan time)

■ Precautions when using Timer instructions

When you use a timer instruction between subroutine instructions (SBN to RET), between step instructions (STP to STE), or between condition branch instructions (CJ or NCJ to LABEL), if, while the timer instruction is being executed, the execution conditions are turned OFF between the above opening and closing instructions, the timer's current value will not be updated.
The time calculation continues, and the current value of the timer reflects the elapsed time when the instructions are executed again. If the set time has elapsed, the timer contact will normally be turned ON immediately after the execution begins again. However, even if the set time has elapsed, if there is a long time before execution starts again, the timer contact may not be turned ON. Therefore, execute timer instructions under the condition that the execution of the timer instruction and its surrounding instructions will not be stopped.

(Example) Programming a timer instruction with a set value of 20 seconds between the SBN and RET instructions



Reference

- Note that the ONDL, OFDL, SHOT, and FLIK instructions also operate as explained above.
- If a program timer instruction with a checked "Perform OFF Process When Stopping Execution" checkbox in its properties is used, the phenomenon above does not occur as the timer instruction is reset by the OFF processing when each program stops executing.

"Standby module operation" (page 4-101)
"Function Blocks" (page 4-117)
"Macro Type" (page 4-181)

■ How to handle the set value and current value of a timer

The instructions to use when reading and changing the set value and current value of a timer are shown in the following table.

	Reading	Changing
Current value	MOV and LDA instructions	MOV and DW instructions
Set value	(This value cannot be read.)	STA instruction*

* When the power is turned on and when the mode switches from RUN to PRG, the set value that you changed with the STA instruction is cleared and returned to its original value.
If a device is used to specify the set value of the timer instruction, the set value of the timer is updated each scan to the value of the specified device.

Point

When you change the value, if the current value is greater than the set value, the current value will be updated to match the set value.

Counter (contact) C

Overview

Counters have a contact (1 bit), a set value (32 bits), and a current value (32 bits). These devices function as follows: when the counter's current value is the same as the set value (counting is finished) the contact is turned ON.

Notation

Device numbers are structured as shown below.
(Example)

Device number (0000 to 3999)

C0001

Device name

The contact, current value, and set value are determined by the operands of the instruction used.

Range

C0000 to C3999

Details

There are three types of counters: the counter, the output counter, and the up/down counter.

Default setting

With the default settings, the contact status, current value, and set value are all held.
By clicking "Power off holding" in the CPU system settings, you can configure the settings so that the contact status, current value, and set value are cleared when the power is turned on and when the mode switches from PRG to RUN.

Reference

- You can use zero suppression when you write your programs.
  (Example) C15
- You can use a device with the same number with both a timer and counter.

■ How to program counter operations

To program counter operations, specify in the counter instruction operands the number, set value, and count input relay of the counter that you want to use.

Device number (0000 to 3999) Set value (0 to 4294967295)

C 0011 #01000 R03000

Count input relay

OUTC 0012 #01000

UDC 0013 #01000

Counter instruction

Reference You can also specify a device in the set value.

Operation differences between counter types

Counter Type (Instruction)	Counter Number Assignment	Set Value	Reset Mode
Counter (C)	0000 to 3999	0 to 4294967295	Execution condition OFF
Output counter (OUTC)			RES instruction
Up/down counter (UDC)			Reset input ON

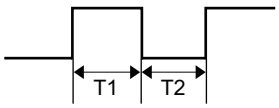
For details, see "Timer and Counter Instructions" in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

■ Counter maximum counting speed

Counting is performed only when both the counter input ON time and OFF time are longer than the scanning time.
The formulae are shown below. If a counting speed greater than that shown below is necessary, use a high-speed counter or the interrupt function.

$$\text{Maximum counting speed} = \frac{\text{Duty cycle (\%)}}{100} \times \frac{1}{\text{Scan time (sec)}} \quad \text{(Unit: times/second)}$$

* The ratio of the time that the signal is ON or the time that the signal is OFF to the total time is called the duty cycle.



$$T1 \le T2\text{: Duty cycle} = \frac{T1}{T1+T2} \times 100 \text{ (Unit: \%)}$$

$$T1 > T2\text{: Duty cycle} = \frac{T2}{T1+T2} \times 100 \text{ (Unit: \%)}$$

■ How to handle the set value and current value of a counter

The instructions to use when reading and changing the set value and current value of a counter are shown in the following table.

	Reading	Changing
Current value	MOV and LDA instructions	MOV and DW instructions
Set value	(This value cannot be read.)	STA instruction*

* If a device is used to specify the set value of the counter instruction, the set value of the counter is updated each scan to the value of the specified device.

Point **When you change the value, if the current value is greater than the set value, the current value will be updated to match the set value.**

Control relay (contact) CR

Overview These are special internal auxiliary relays that are used to control the PLC functions and to capture the PLC status.

Range CR0000 to CR7915

Details "App-2 Lists of CR and CM Devices" (page A-10)

Word Devices

Data memory/expansion data memory DM/EM

Overview

Data memory and expansion data memory are devices that are used to save the reference data during function operations and to store the results of calculations. Each entry has 16 bits.

Notation

Device numbers are structured as shown below.
(Example)

Device name — Device number

DM06143

Reference

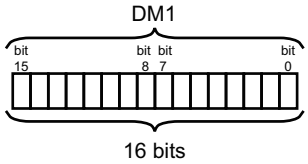
- You can omit "M" when writing "DM/EM"
- You can use zero suppression when you write your programs.
  (Example) D6143
- You can use this as a bit device.
  (Example) DM1100 .....The lowest bit of DM1100 is processed.
- You can use this as a bit device by specifying the bit position.
  (Example) DM1200.12 .....The 13th bit of DM1200 is processed.
  "Bit device processing of word devices" (page 4-52)
- When you select XYM marking mode, this is displayed as "D."
  "XYM marking" (page 4-25)

Range

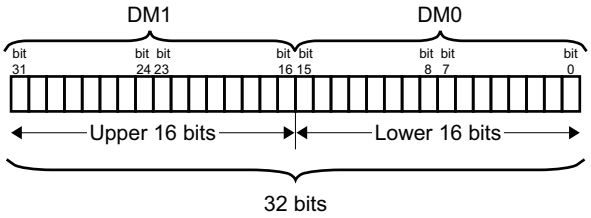
DM00000 to DM65534
EM00000 to EM65534

Details

- You can only assign data memory (DM) entries to the unit devices of expansion units.
- The bit width of one device is 16 bits. When the device is being used as an unsigned, decimal value, it takes on a value in the range of 0 to 65535. When the device is being used as a signed, decimal value, it takes on a value in the range of -32768 to +32767. When the device is being used as a hexadecimal value, it takes on a value in the range of $0 to $FFFF.



- You can also use two devices to process 32-bit data. For 32-bit data, the lower 16 bits will be stored in the data memory entry of the specified device number and the upper 16 bits will be stored in the data memory entry of the next device number. When the devices are being used as an unsigned, decimal value, they take on a value in the range of 0 to 4294967295. When the devices are being used as a signed, decimal value, they take on a value in the range of -2147483648 to +2147483647. When the devices are being used as a hexadecimal value, they take on a value in the range of $0 to $FFFFFFFF.



- When you are using data memory to specify the operands of an instruction that handles 32-bit data, specify only the number of the device in which the lower 16 bits are stored.

Default setting

You can change the range to hold by using "Power off holding" in the CPU system settings.

Reference

Device notation for data memory (DM) units
In KV STUDIO, aside from the normal notation, you can use a notation in which the unit numbers are the reference.
To use the device notation for units, on the "View" menu, click "Display unit device."

Unit number Device number

DM05_0010

You can use zero suppression when you write your programs.
(Example) DM5_10

Unit number: Starting with the CPU, numbers are assigned in the order (00, 01, 02, 03, 04, and so on) that the units are connected.

Unit DM number: Starting from 0000, the amount of data memory that will be occupied is assigned as the number.

Unit number	00	01	02		03	04
KV-PU1	KV-8000 CPU	KV-SAD04 60 words	KV-SDA04 40 words	Bus connection	KV-TP40 90 words	KV-ML16V 222 words
Unit DM number		0000 to 0059	0000 to 0039		0000 to 0089	0000 to 0221

(Example) The 6th KV-SAD04..... 01_0005
The 15th KV-TP40 ..... 03_0014

* When using the CPU built-in EtherNet/IP/Ethernet function, devices are labeled as unit devices with "00_" as the leading number.

File Register ZF/FM

Overview

File register can be classified into ZF whereby all areas are processed sequently and FM whereby the area is divided into sixteen memory banks which are switched over in processing.

Notation

The device No. is structured as follows:
(Example)

Device name Device No.

ZF000372

Reference

- The letter M in FM can be omitted.
- Zero suppression can be used in programming.
  (Example) ZF372
- Can be used a bit device.
  (Example) FM1100.....The lowest bit of FM1100 is processed.
- Can be used as a bit device by specifying the bit position.
  (Example) DM1200.12 .....The 13th bit of ZF1200 is processed.
  "Bit device processing of word devices" (page 4-52)
- “FM" is displayed as ”F" when the XYM Marking option is selected.
  "XYM marking" (page 4-25)

Point

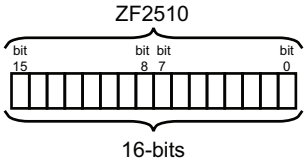
- **ZF local devices (@zfxxx) cannot be used.**
- **FM local devices (@FM0 to @FM32767) cannot perform memory bank switching.**
- **FM local devices use the work area (@FM0 to @FM32767).**

Range

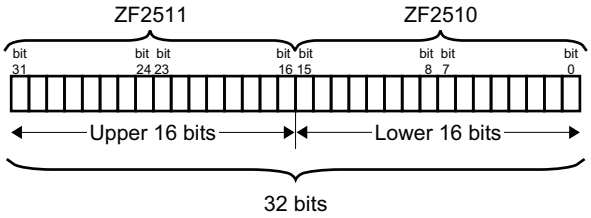
ZF000000 to ZF524287
FM00000 to FM32767

Details

- These relays cannot be assigned as the unit device of expansion unit.
- The device category and range varies depending on consecutive numbering and memory bank switching modes, .
- A device's bit width is 16 bit. If unsigned decimal is used, the value will be 0 to 65535; if signed decimal, it will be -32768 to +32767; if hexadecimal, it will be $0- $FFFF.

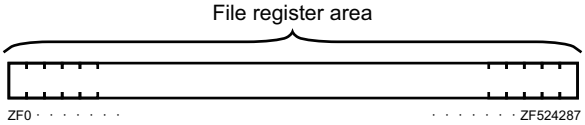


- 2 devices can also be used to process 32-bit data. For 32-bit data, the low 16 bits will be stored in the data memory of the specified device No. and the high 16 bits will be stored in the data memory of the next device No.. If used as unsigned decimal, the value will be 0 to 4294967295; if used as signed decimal, the value will be -2147483648 to +2147483647; if used as hexadecimal, the value will be $0 - $FFFF.



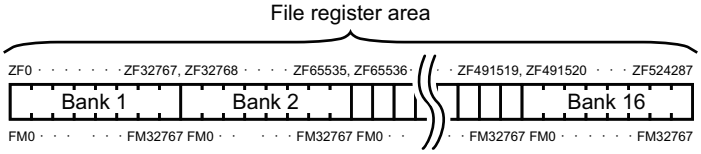
● ZF: used in the consecutive numbering mode

The whole file register area is used as a single device range.



● FM: used in the memory bank switching mode

The file register is divided into 16 areas which are switched over in use.



Through switching memory banks, 16 different device ranges are available for a single device.
Since it can be mixed with ZF devices in use, attention should be made in data processing.
For more information about memory bank switching, please refer to "4-14 File Register" (page 4-226).

Default setting Under default status, the value is held.
You can also select the "Clear" option by clicking "CPU system settings" and " Power-off hold settings ".

Link register W

Overview

These registers are used to perform data linking. Hexadecimal notation is used to specify device numbers, which is different from other word devices.

Notation

Device numbers are structured as shown below.
(Example)

Device name Device number (0 to 7FFF)

W3A6F

Reference

- You can use zero suppression when you write your programs.
  (Example) W7A
- You can use this as a bit device. You can also specify the bit position.
  (Example) W1200.12 .....The 13th bit of W1200 is processed.
  "Bit device processing of word devices" (page 4-52)

Point

- **Use hexadecimal numbers (0 to F) to specify device numbers.**
- **You cannot use local devices (@).**

Range

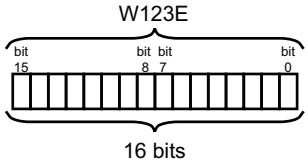
W0000 to W7FFF

Application

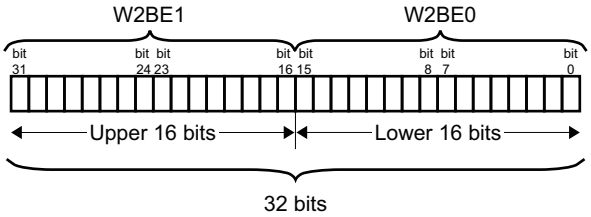
These registers can be used to perform data linking.

Details

- These cannot be assigned as unit devices of expansion units.
- The bit width of one device is 16 bits. When the device is being used as an unsigned, decimal value, it takes on a value in the range of 0 to 65535. When the device is being used as a signed, decimal value, it takes on a value in the range of -32768 to +32767. When the device is being used as a hexadecimal value, it takes on a value in the range of $0 to $FFFF.



- You can also use two devices to process 32-bit data. For 32-bit data, the lower 16 bits will be stored in the data memory entry of the specified device number and the upper 16 bits will be stored in the data memory entry of the next device number. When the devices are being used as an unsigned, decimal value, they take on a value in the range of 0 to 4294967295. When the devices are being used as a signed, decimal value, they take on a value in the range of -2147483648 to +2147483647. When the devices are being used as a hexadecimal value, they take on a value in the range of $0 to $FFFFFFFF.



Default setting

You can configure the settings so that the values are held by using "Power off holding" in the CPU system settings.

Temporary data memory TM

Overview Temporary data memory entries are word devices that can be processed in the same manner as data memory. Each device has 16 bits. When you use calculation instructions, a part of the device area can also be used to temporarily store the data for the calculation and the result data.

Notation Device numbers are structured as shown below.
(Example)

Device name Device number

TM010

Range TM000 to TM003 (used with calculation instructions)
TM004 to TM511 (used with simple indirect specification and similar functions)

TM Number	Application
TM000	Calculations (DIV, MUL, and ROOT instructions)
TM001	
TM002	Calculations (DIV instruction)
TM003	
TM004 to TM511	Simple indirect specification and general use

For details on instructions such as DIV and MUL, see the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."
For details on simple indirect specification, see "Indirect Specification" (page 4-46).

Default setting When the power is turned on and when the mode switches from PRG to RUN, values are cleared and set to 0.

Reference

- TM000 to TM003 are designed for usage with calculation instructions such as DIV and MUL. We recommend that you do not use these devices for any other purpose.
- You can use zero suppression when you write your programs.
  (Example) TM10
- TM devices are stored in the internal cache, so you can access them at high speeds.

Timer (current value and set value) T

Use timers as word devices when you set the value of the timer (32 bits) and when you read the current value of the timer (32 bits).

"Timer (contact) T" (page 4-32)

Counter (current value and set value) C

Use counters as word devices when you set the value of the counter (32 bits) and when you read the current value of the counter (32 bits).

"Counter (contact) C" (page 4-35)

Index register Z

Overview These are 32-bit word devices that are used to specify the value to add to the target device number during index modification. You can use these devices in the same manner as data memory to save data and to store calculation results.

Range Z01 to Z23

Reference You can use zero suppression when you write your programs.
(Example) Z3

Point

- **Z11 to Z23 are reserved for the system, so you cannot use these devices.**
- **You cannot specify the bit for word devices.**
- **There are no local devices. Use the ZPUSH and ZPOP instructions.**

For details, see the ZPUSH instruction in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."
For details, see the ZPOP instruction in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

Default setting When the power is turned on and when the mode switches from PRG to RUN, the values are cleared and set to 0.

"Index Modification" (page 4-44)

Control memory CM

Overview Special word device used for PLC function control and PLC status confirmation.

Range CM0000 to CM7599

Details "App-2 Lists of CR and CM Devices" (page A-10)

Index Modification

Index modification is a method for specifying devices by the number that is calculated by adding the value of the index register (Z) or a constant to devices.
Values in the signed, 32-bit range can be added.
The types of devices that you can use index modification with are R (DR), B, MR, LR, T, C, DM, EM,FM,ZF, W, and TM. Index modification can be performed over all device ranges.
Index modification can be used together with indirect specification (*) and local devices (@).

Point

- **Index modification cannot be performed in the following cases.**
  - **CR/CM/Z**
  - **Array type variable / label**
  - **Variables not assigned to global devices**
  - **When specifying the bit position of word device**

  "Variable" (page 4-55)
  "Bit device processing of word devices" (page 4-52)
- **If you specify in a parameter a device that is outside of the usable range after index modification, control relay CR2012 is turned ON and the instruction is not executed. Operation will continue.**

■ Modification according to index registers

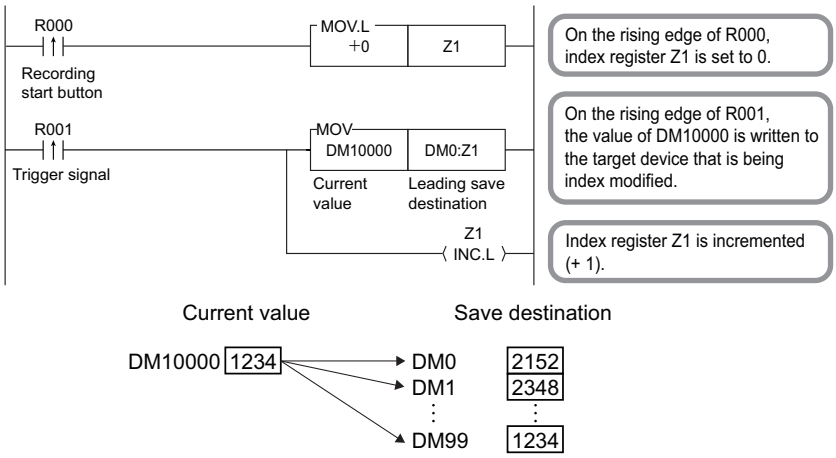
Use this to change the device specified with the instruction's operand during operation.
To program device modification according to index registers,
(Device number): Z(1 to 10)
write instructions as shown above.
(Example) DM1000: Z1..... DM990 is referenced when the value of Z1 is -10.
R000: Z5..... R100 is referenced when the value of Z5 is +16.

Point

Z11~Z23 are reserved for system, so you cannot use these devices. Series.

Example

The current value, which will be stored in DM10000, is saved when a trigger signal (the rising edge of R001) is received. Recording starts on the rising edge of R000. Data is saved in the data memory entries in order, starting from DM0.

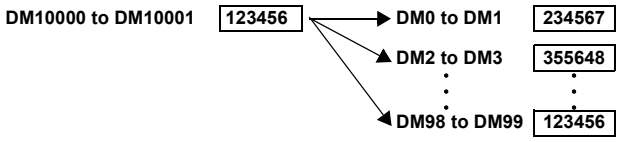


Point

Because the value to change through index modification is processed as a signed, 32-bit value, write ".L" as the suffix of instructions that operate the index registers.

Reference

- If you need to use 10 or more index registers (Z) in a project, use the index register instructions (ZPUSH and ZPOP) to store and restore all the index registers of a program at once.
  For details, see "Index Instructions" in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."
- In the above example, if the suffix of the MOV instruction was ".L," the value of DM10000 to DM10001 would be saved as data in the following data memory entries: DM0 to DM1, DM2 to DM3, ... DM98 to DM99.



■ Modification according to index constants

To program index modification by using constants,
(Device number):#(a constant in decimal notation)
write instructions as shown above.

(Example) DM1000:#-10.....DM1000- 10 words = DM990
R000:#16:.....R000+ 16 bits (the amount for one channel) = R100

Reference

If you use modification according to index constants in a macro that uses unit devices, it is easy to tell what number device is specified from the assigned leading device. This is also effective when you use UR and UM as argument devices during macro creation.

(Example)
UR0:#2
UM1:#32

"Usage example" (page 4-187)

Indirect Specification

■ Indirect specification (*)

Indirect specification (*) is used to modify during program execution the devices that you specify with instruction operands. You can also use index modification to change the reference devices, but you can use indirect specification (*) to change the device type.

● What is indirect specification?

Each device is managed with the CPU unit internal memory, and each device has a unique address. Because addresses are managed as 32-bit values, 2 words worth of devices are required to store an address.
In indirect specification (*), the address is manipulated by means of special instructions to change the reference device.
Index modification (:Z/:#) can be used together with local devices (@).

Point

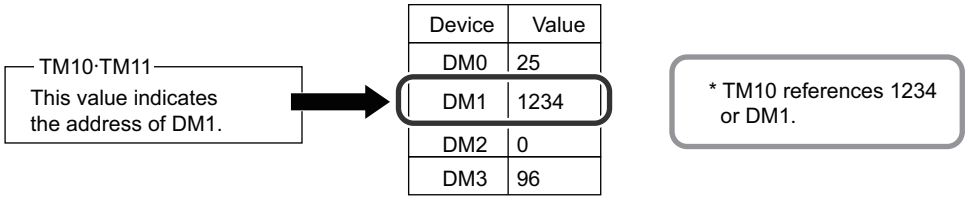
- **Indirect specification (*) cannot be used together with simple indirect specification (#TM).**
- **If you specify in an operand a device that is outside of the usable range after indirect specification, control relay CR2012 is turned ON and the instruction is not executed. Operation will continue.**
- **Indirect specification cannot be performed for array elements and structure members.**

● How to program indirect specification

If you write "*(device)" and the instruction operands, the device or device value that corresponds to the address stored in the device will be referenced. What is referenced is determined by the operands of the instruction that you use.

(Example)

When you write "*TM10"



Point

- **The devices that you can write in "*(device)" (that is, the addresses that can be stored) are DM, EM, FM, ZF, TM, and W.**
- **The devices that you can specify indirectly are R, MR, LR, B, T, C, DM, EM, FM, ZF, TM, and W.**
- **Only variables set to UDINT type can be written as "(*Variable)" (address can be stored).**
- **Array elements, array subscripts and structure members cannot be indirectly specified with "(*Variable)".**

Reference

- If the device that you want to specify indirectly is a timer (T) or counter (C), the contact, current value, and set value are determined by the operands of the instruction that you use.
- When you are using local devices, write "*@(device)."

● How to use indirect specification

Follow the procedure below to perform indirect specification.

1 Use the address set instruction (ADRSET) to write the address of the device that you want to reference.

(Example)

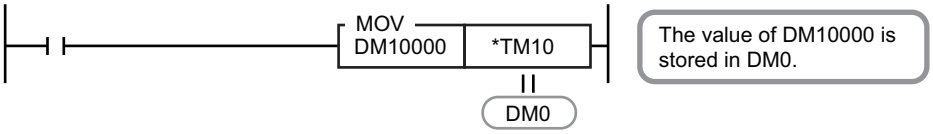
Point

- **The devices that can store addresses are TM, DM, EM, FM, ZF, and W.**
- **Only UDINT type variables can store an address. However, it cannot be stored in UDINT type structure members or UDINT type array elements.**

For details, see the ADRSET instruction in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

2 To use the operands of the instruction to indirectly specify the device, add the prefix "*" before the device in which the address is stored.

(Example)

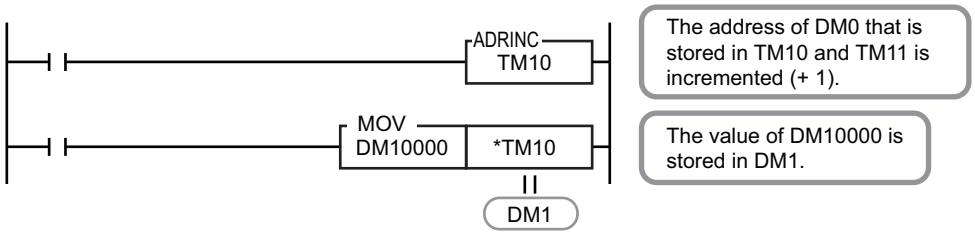


Point

Before you add the prefix "*" in front of the device in which the address is stored, be sure to use the ADRSET instruction to store the address of the device that you want to specify indirectly.

3 To change the device that you want to reference, use the specialized indirect specification commands (ADRINC, ADRDEC, ADRADD, and ADRSUB) to change the stored address.

(Example)



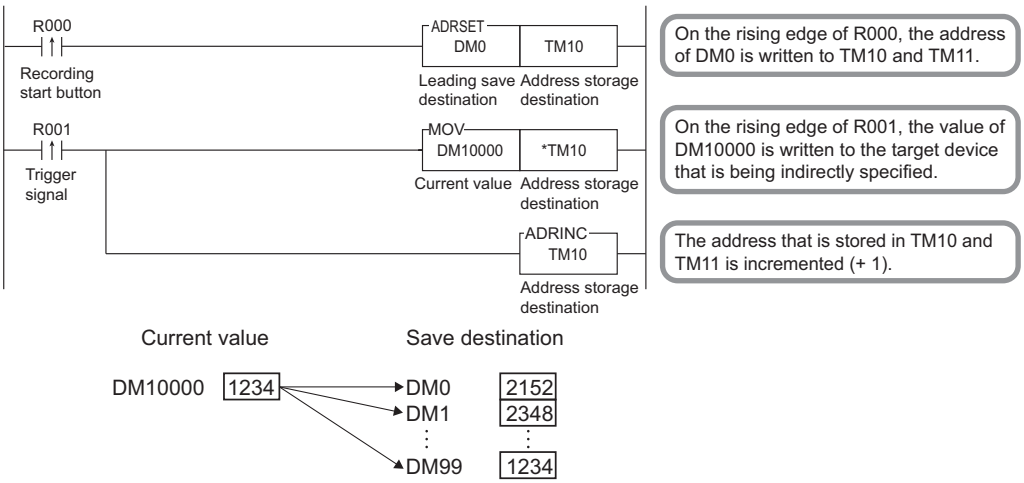
Point

Do not use normal calculation instructions (such as ADD, INC, SUB, and DEC) to change the address that you want to reference. The ladder program will not operate correctly.

For details, see "Indirect Specification Instructions" in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

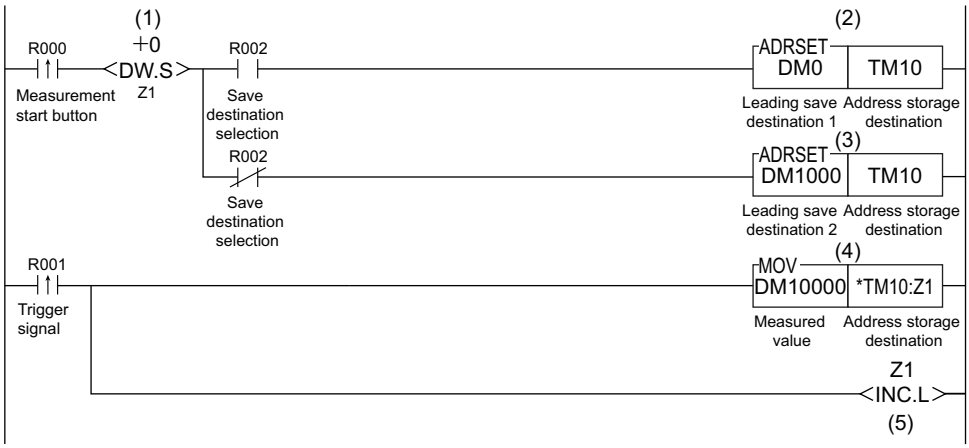
● Usage example of indirect specification

The rising edge of R001 is used as the trigger signal, which causes the current value stored in DM10000 to be saved.
Recording starts on the rising edge of R000. Data is saved in the data memory entries in order, starting from DM0.



● Usage example together with index modification

The rising edge of R001 is used as the trigger signal, which causes the current value stored in DM10000 to be saved.
Measurement starts on the rising edge of R000. Data is saved in the data memory entries in order, starting from DM0 if R002 is ON when measurement starts and starting from DM1000 if R002 is OFF when measurement starts.



[When measurement starts]
(1) 0 is written in index register Z1.
(2) If R002 is ON, the address of DM0 is written in TM10 and TM11.
(3) If R002 is OFF, the address of DM1000 is written in TM10 and TM11.
[When a trigger occurs]
(4) Z1 is used to index modify the device at the address stored in TM10 and TM11 (the value of Z1 is added to this device), and the value of DM10000 is written to the resultant device.
(Example)
*TM10 → DM1000
Z1 = 23
→ *TM10:Z1 → DM1023
(5) Z1 is incremented.

■ Simple indirect specification #TM

You can use indirect specification that uses temporary data memory (TM) as "simple indirect specification."

Point

- **You cannot use simple indirect specification together with index modification, indirect specification (*), or local devices (@).**
- **Simple indirect specification is supported so that existing programs may be reused. If you are creating new programs, we recommend that you use index modification or indirect specification.**

● How to specify simple indirect specification

To use simple indirect specification, write #TMnnn in the instruction operands, and change the value stored in TMnnn.
The relationship between the value stored in TMnnn and the reference value of simple indirect specification is shown in the following table.

Value Stored in TMnnn	Reference Value of Simple Indirect Specification	
0	DM0	DM device indirectly specified
1	DM1	
2	DM2	
⋮	⋮	
9997	DM9997	
9998	DM9998	
9999	DM9999	
10000	R000	Relay (R) device indirectly specified
10001	R001	
10002	R002	
⋮	⋮	
27913	R17913	
27914	R17914	
27915	R17915	

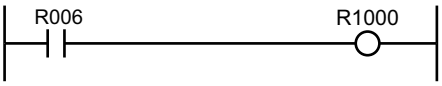
The range of temporary data memory entries that can be used is TM0 to TM511. However, take care to not overlap over TM0 to TM3, as these are used for calculations.

Point

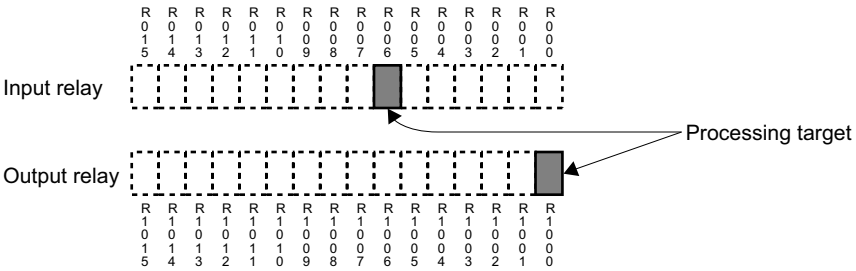
- **With simple indirect specification, you can specify only DM and R devices and only the ranges of DM10000 to DM65534 and R18000 to R199915.**
- **If you attempt to perform simple indirect specification when the value of temporary data memory is greater than 27915, control relay CR2012, the calculation flag, will be turned ON when the attempt to execute the instruction is made, and the instruction will not be executed. Operation will continue.**

Processing Bit Devices in Units of Words

Normally (in sequence control), a bit device is processed as a single-bit device.

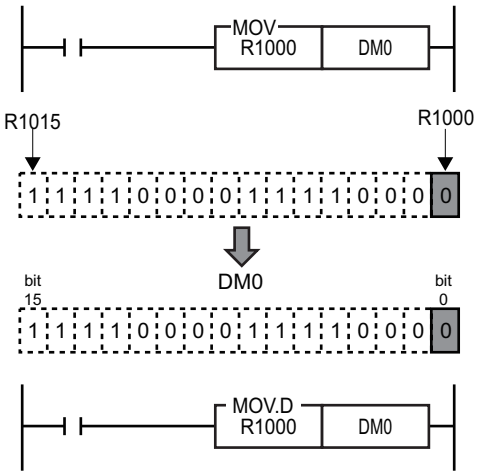


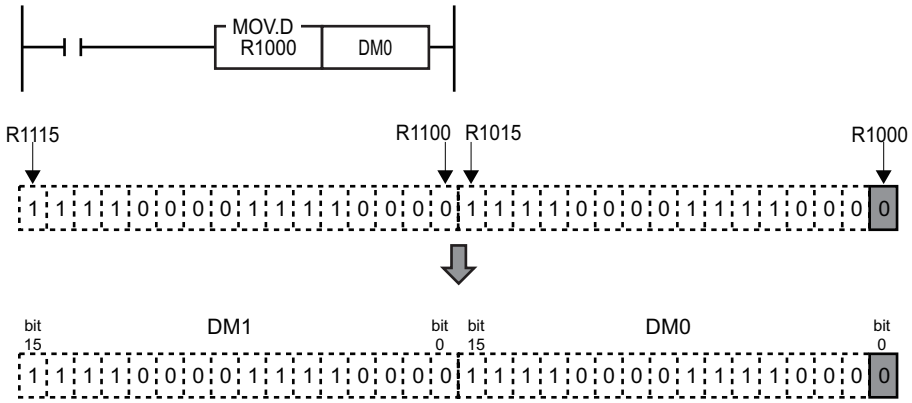
With an instruction like this one,
the processing target of both R006 and R1000 is 1 bit.



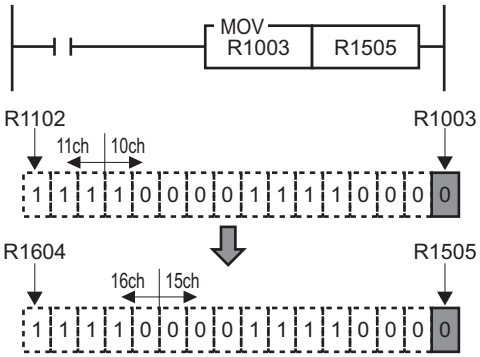
In calculation instructions, bit devices can be grouped into 16 bits or 32 bits for processing.
If you specify a bit device as an operand of an instruction that processes data in units of words, the specified bit device will be treated as the leading bit of the word data for processing.

Example





If the bit device specified as an operand is not the leading channel, the next channel will also be used to provide the word unit of data for processing



Point

When the word unit of data is taken across multiple channels for processing, the processing speed decreases.

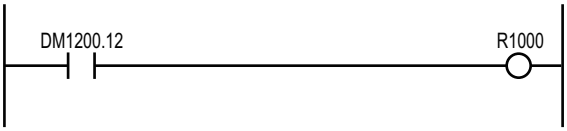
Bit device processing of word devices

Normally, word devices are processed as 16-bit or 32-bit devices.
To process a specific bit of a word device, use the bit contact instruction and the bit output instruction.
You can specify the bit position of a word device in the operands of the following instructions.

LD, LDB, AND, ANB, OR, ORB, LDP, ANP, ORP, LDF, ANF, ORF, LDPB, ANPB, ORPB, LDFB, ANFB, ORFB, OUT, OUB, SET, RST, KEEP, DIFU, and PIFD

Example

DM1200.12.....The 12th bit of DM1200 is processed.



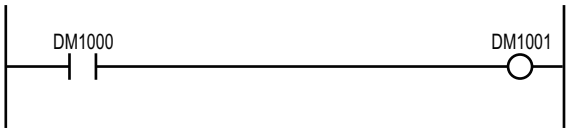
DM1200

0	1	1	1	0	1	1	0	0	1	1	0	1	1	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

If you specify a word device with an instruction other than those listed above, only the lowest bit of the specified word device is processed as a bit device.

Example

When the value of DM1000 is #46563 ($B5E3)



DM1000=#46563($B5E3)

1	0	1	1	0	1	0	1	1	1	1	0	0	0	1	1

Lowest bit
1: The contact turns ON.
0: The contact turns OFF.

In the above example, the contact is turned ON, so the lowest bit of DM1001 is set to 1.

DM1001=#14($E)

0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

DM1001=#15($F)

0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Reference Bits other than the lowest bit are not changed.

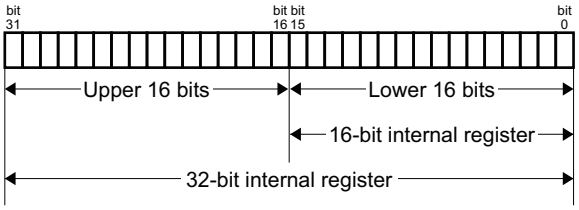
Internal Registers

An internal register is a device that is used to store the calculation results of application instructions, calculation instructions, and expansion instructions and to temporarily store data during calculations. The data stored in an internal register is not changed until the next time that new data is stored in the internal register.
When the power is turned on and when the mode switches from PRG to RUN, the values are cleared and set to 0.

The usage of the internal registers is automatically separated into the 16-bit, 32-bit, and 64-bit internal registers according to the instruction and suffixes used.
"Suffixes" (page 4-9)

● 16-bit and 32-bit internal registers

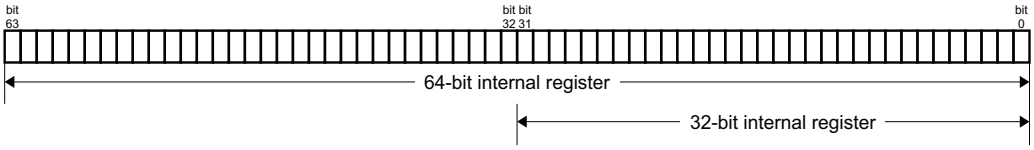
The 16-bit internal register and the 32-bit internal register are shown below. The 32-bit register consists of 16 upper bits and 16 lower bits.



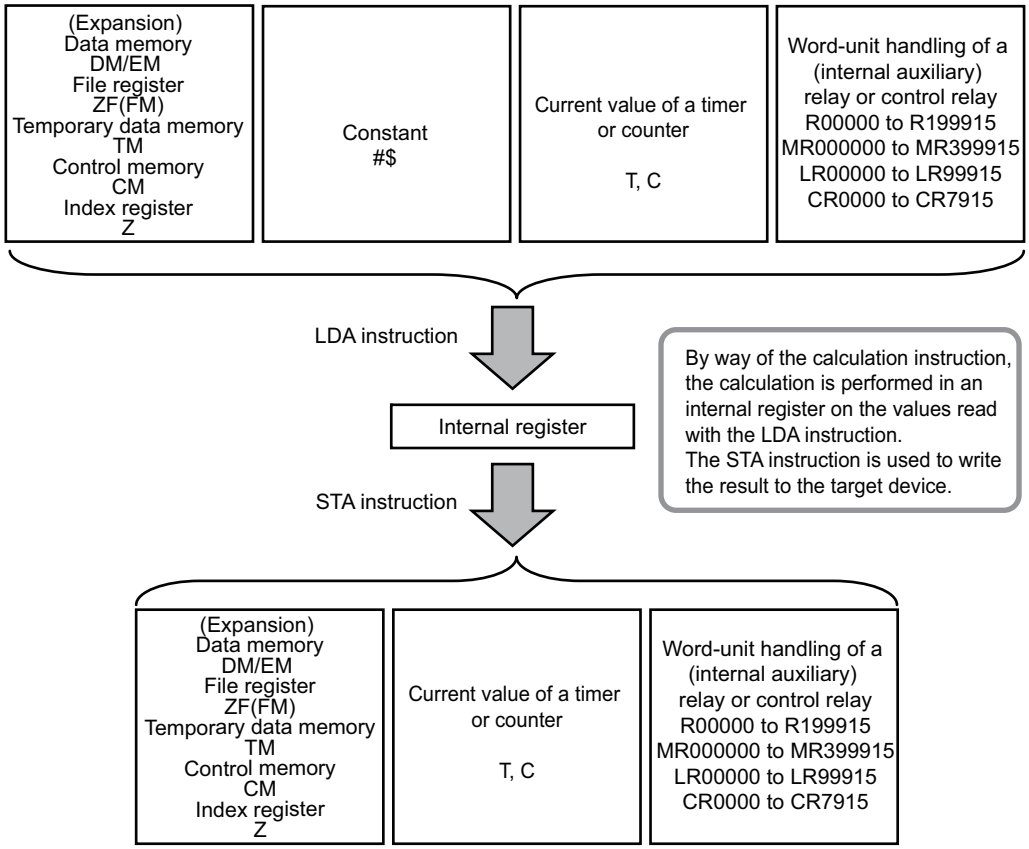
The lower 16 bits of the 32-bit internal register are known as the 16-bit internal register. The 32-bit internal register consisting of the lower 16 bits and the upper 16 bits is known as the 32-bit internal register.

● 64-bit internal registers

The 64-bit internal register is shown below. The 64-bit register consists of 64 bits, and is only used when double precision floating-point number instructions and suffixes (.DF) are being processed. The lower 32 bits of the 64-bit internal register are shared with the 32-bit internal register.

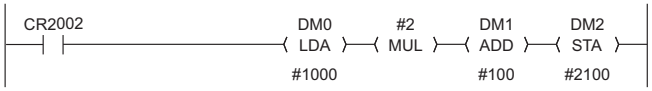


● Internal registers and calculation instructions



Example

The value of DM0 is read into the internal registers. This value is then multiplied by 2, the value of DM1 is added to the product, and the result is then stored in DM2.

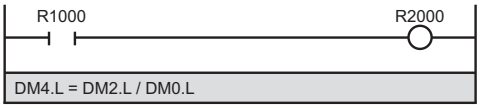


4-5 Variable

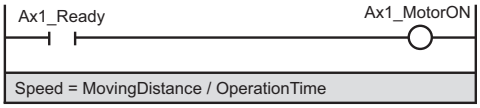
As elements used in the program, KV-8000 (A) provides variables that can be freely set by the user and devices that can be specified with dedicated symbols and numbers (such as Relay, Timer, Counter, and Data memory). Both are written after the instruction and used to give the data used in the instruction or to store the execution result of the instruction. This section describes the Variable.

- Since the predefined names can be used for programming, the program readability is improved.

When programming with devices



When programming with variables



- The required size is automatically assigned to the work area of the CPU unit according to the variable settings. Therefore, there is no need to care about the assignment.
- Variables are automatically assigned even when a program is copied between different projects. Unlike the devices, the device map need not be changed before copying, which makes the program more reusable.
- Since the touch panel display of VT5 series also supports the variables of the KV-8000 series, there is no need to change the device number, etc. on the touch panel display side when changing a device number of PLC like the devices.
- Unregistered variables used in the program can be imported in batch later, so it is not necessary to register all variables before programming.
- Devices and variables can be used together.

Point

- **Variables can be used only with KV-8000 with CPU function version 2.0 or later. When using variables, the CPU function version should be upgraded from KV STUDIO Ver.11.0 or later.**
- **Variables can be used on the touch panel display only with VT STUDIO Ver.8.0 or later, and the compatible model is only the VT5 series.**
- **When using variables in VT STUDIO, it is necessary to obtain the variable information of KV STUDIO. For details, refer to "VT STUDIO Reference Manual".**
- **Unit variables settable with the KV-XD02 cannot be used in ladder programs.**

Variable Types

There are two types of variables.

■ Global variables

Variables that are processed in common throughout the project. Variables defined as global variables can be referenced from multiple programs or the VT5 series touch panel display. For example, if the global variable "Var_X" is defined, the same entity will be referenced when "Var_X" is used in module A and when "Var_X" is used in module B.
Global variables can also be assigned manually to CPU unit devices.

■ Local variables

Variables processed by program. Variables defined as local variables can be referenced only in each program. For example, if the local variable "Var_Y" is defined in module A and the variable "Var_Y" is also defined in module B, "Var_Y" used in module A and "Var_Y" used in module B refer to different entities. Since local variables cannot be referenced from the VT5 series, the variables referenced from the VT5 series should be registered as global variables.
In addition, local variables cannot be assigned manually to CPU unit devices.

List of Variables and Data Types

When using variables, it is necessary to clearly define the data type. The CPU unit automatically assigns the required data size to the work area according to the defined data type.
The work area sizes and variable specifications that can be used with KV-8000(A) are as follows.

Work area size

Item	Content
Bit	64000 points
Word	589824 words
Timer	4000 points
Counter	4000 points

Variable specifications

Item	Content
Maximum number of definitions	100000*1
Maximum length of variable name	128 characters (128 characters in regardless of full-width / half-width)
Maximum number of hold settings	30000*2*3

*1 Sum of global and local variables. Depending on the number of characters in the variable name, 100000 may not be set.
*2 Sum of bits, words and counters. Their own number is set in the CPU system settings. Depending on the number of characters in the variable name, 30000 may not be set.
*3 One variable name registered in the variable edit window is counted as one setting. For example, when an array type variable with 10000 elements is set, the number of settings will be 1.

Reference
The entire variable area can also be held by setting variables that occupy multiple bits or words with 1 variable name such as array or structure.

The data types and data sizes of variables that can be used with KV-8000(A) are as follows. The CPU unit automatically assigns the following data sizes to the work area according to the data type of the variables*1*2.

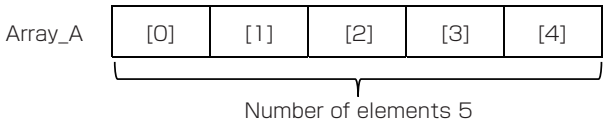
Data type		Data size (in bits)	Alignment size*3 (in bits)
Type name	Description		
BOOL	Bit	1	1
UINT	1-word unsigned integer	16	16
INT	1-word signed integer	16	16
UDINT	2-word unsigned integer	32	32
DINT	2-word signed integer	32	32
REAL	Single-precision floating point type	32	32
LREAL	Double-precision floating point type	64	32
STRING※4	String	(Number of characters*5 + 1) × 8 rounded up to an integral multiple of 16	16
TIMER	Timer	—	—
COUNTER	Counter	—	—
ARRAY	Array	Data type × Array size	According to the data type
Structure	—	By definition※6	Maximum member alignment size
Function block	Function block	By definition※6	32

*1 It is also possible to manually assign to CPU unit devices. In that case, no work area is consumed.
*2 When a variable is set as a constant, the work area is not consumed.
*3 Unit for securing the memory.
*4 The character code processed in STRING type follows the language set in the project language settings. For details on project language settings, refer to "KV STUDIO User's Manual".
*5 The number of characters is the number of bytes up to NULL. For double-byte characters such as Shift-JIS, the number of characters is calculated as 2 with 1 character.
*6 Structures and function blocks are data types used by setting multiple data types. Therefore, the data size is determined by the settings.

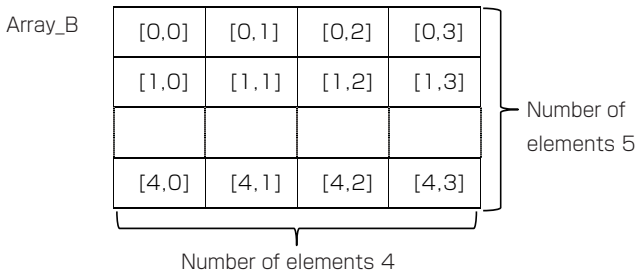
Array

Array is a data type that defines a collection of data of the same data type with 1 name. Each data that constitutes an array is called an element. When using an array, the data type, number of elements and number of dimensions should be set in addition to the variable name. An array with n dimensions is called an n-dimensional array. An example of 1-dimensional array to 3-dimensional array is shown below.

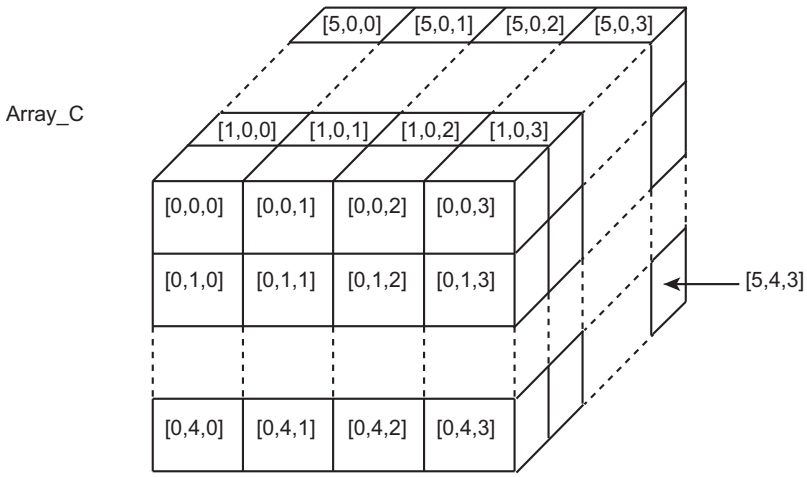
- 1-dimensional array　Variable name: Array_A　Number of elements: 5



- 2-dimensional array　Variable name: Array_B　Number of elements: 5×4



- 3-dimensional array　Variable name: Array_C　Number of elements: 6×5×4

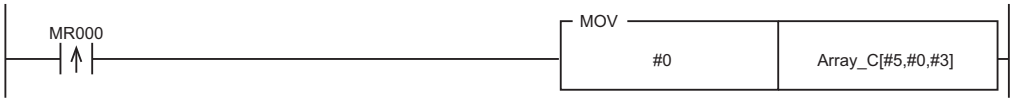


Point　**Array type variables have the same data type for all elements.**

■ Writing of array type variables

● Specification of element of array type variable

When an array element is specified for use in a program, the array name and the subscript of [ ] are combined. For an array with 2 or more dimensions, the element numbers should be separated by "comma (,)". For example, when using the elements [5,0,3] of the 3-dimensional array Array_C, "Array_C[#5,#0,#3]" is written in the program as follows. (# can be omitted)



● Specification of entire array type variable

All elements that constitute an array are called "the entire array". When an entire array is specified in a program, only the variable name is written. For example, when specifying an entire array called Array_A, write "Array_A".

Reference: Arrays can be processed with ladder programs in KV STUDIO Ver.11.0.

Point: **Only the ACNT/WSIZE/BSIZE instruction writes the variable name only and processes it as an entire array.**
When only the variable name is written by other instructions, only the instruction processing range from the first element is processed.
For example, when "MOV.D 0 Array_A" is written for the UINT array type variable Array_A, 0 is stored only in the 2 words Array_A[0] and Array_A[1].

Array specifications

Item	Content
Maximum number of elements per dimension	589823
Maximum number of dimensions	8
Maximum data size that can be occupied per array	589823 words
Data types that can be specified	BOOL / UINT / INT / UDINT / DINT / REAL / LREAL / STRING / Structure / Function block
Subscripts that can be specified*1	Device / constant / variable / expression*2
Data types that can be used for subscript	UINT / INT / UDINT / DINT

*1 Specify 0 or a positive integer for the subscript. If any other number is specified, a conversion error or calculation error will occur.
*2 Only KV scripts can use expressions as subscripts. Expressions cannot be used as subscripts in ladder programs.

Point:
- **When an element is specified beyond the specifiable range, such as specifying Array_A[10] (11th element) for an array called Array_A set with the number of elements "10", the value in an unintended area may be referenced. Especially when using variables or expressions as subscripts, the ACNT instruction or WSIZE instruction should be executed before referring to the array to obtain the number of elements in the array or the data size of the entire array, and avoid referring to elements beyond the specifiable range. A conversion error or calculation error will occur when specified out of the array range.**
  "ACNT" (Page 4-66), "WSIZE" (Page 4-68), "BSIZE" (Page 4-70)
- **When a device is specified as the subscript of an array, it will be processed as a 16-bit unsigned integer.**
- **Bit data such as bit devices (R, MR, etc.), BOOL type variables and word data bits cannot be specified as array subscripts.**
- **Depending on the KV STUDIO version, array data types are expressed as follows:**
  Example: UINT type array with three elements
  Ver. 11.20 or later: ARRAY[0..2] OF UINT
  Ver. 11.10 or earlier: UINT[3]

Structure

Structure is a data type that defines a collection of multiple variables of different data types with 1 name. Each data that constitutes a structure is called a member. When using a structure, the names of the members that constitute the structure and the data types of the members should be set in addition to the variable name. The structure has the following advantages.

- Registering the structure members and data types in advance makes it easier to create variables with the same structure.
- All members can be monitored in batch by registering a structure variable when monitoring in the registration monitor window, etc.
- Visual recognition is improved since the contents can be confirmed with the member names.

The following is an example of using the following information of a motor axis in a program.

Current coordinate (CurCoordinate)
Current speed (CurSpeed)
Axis control (Control)
Axis error (Error)
Axis error code (ErrorCode)

If multiple axes are used without a structure, variables with the same content except for the axis number should be registered as shown below.
When using a structure, the data type should be set to the same structure only when using multiple axes after defining the data structure of axis information as a structure in advance.

<When not using a structure>

Variable name	Data type
Ax1_CurCoordinate	DINT
Ax1_CurSpeed	DINT
Ax1_Control	BOOL
Ax1_Error	BOOL
Ax1_ErrorCode	UINT
Ax2_CurCoordinate	DINT
Ax2_CurSpeed	DINT
Ax2_Control	BOOL
Ax2_Error	BOOL
Ax2_ErrorCode	UINT

<When using a structure>

Variable name	Data type
Ax1	AxisInfo
Ax2	AxisInfo

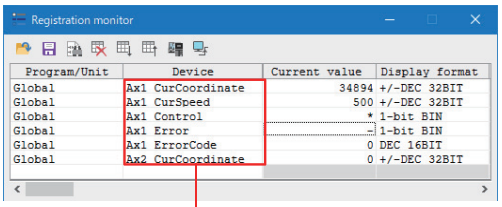
Definition of "AxisInfo" type data

Member name	Data type
CurCoordinate	DINT
CurSpeed	DINT
Control	BOOL
Error	BOOL
ErrorCode	UINT

Reuse the pre-set structure

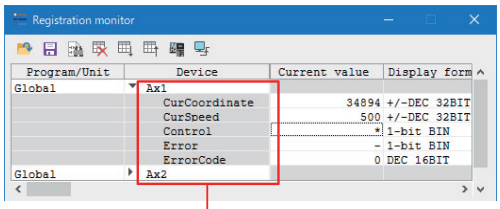
Also, if no structure is used when monitoring in KV STUDIO, it is necessary to manually enter all variables, but if a structure is used, the members in the structure can be monitored in batch simply by entering the variable name.

<When not using a structure>



All variables to be monitored must be entered.

<When using a structure>



When using a structure, its members can be monitored in batch after entering the variable name.

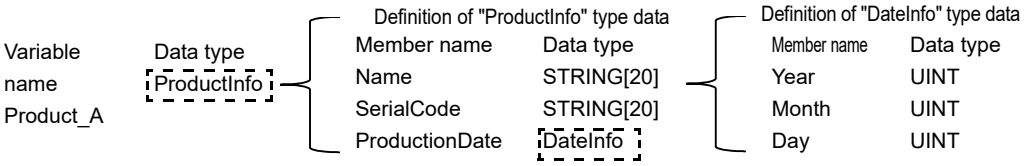
■ Writing of structure type variables

This section describes how to write structure type variables when using them in a program.

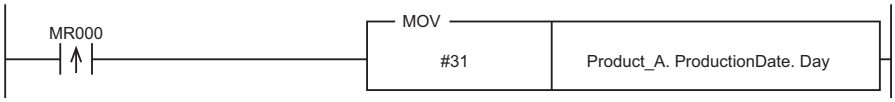
● Specification of member of structure type variable

When using a member of a structure type variable in a program, write "Structure type variable name + Period (.) + Member name". For example, "Ax1.CurSpeed" is written to refer to a member called CurSpeed of an AxisInfo type variable called Ax1.

Structure can also be specified for the data type of structure type variable members. In the following example, a structure type variable called "ProductionDate" is nested in the structure type variable Product_A of "ProductInfo".



When Day of Product_A is specified with a program, write "Product_A.ProductionDate.Day".



Point **Structures can be nested up to 8 stages.**

● Specification of entire structure type variable

All members that constitute a structure type variable are called "the entire structure". When an entire structure is specified in a program, only the variable name is written. For example, when specifying the entire structure of a structure type variable called Product_A, write "Product_A".

Point

- **When a value is written to an entire structure using the FMOV instruction, etc., the value in an unintended area may be changed if a value larger than the number of data in the entire structure is specified as the number of data transferred. When an entire structure is referenced with the FMOV instruction, etc., the WSIZE instruction should be executed first to obtain the data size of the entire structure.**
  "WSIZE" (Page 4-68)
- **Only the WSIZE/BSIZE instruction writes the variable name only and processes it as an entire structure. When only the variable name is written by other instructions, only the instruction processing range from the start of the structure is processed. For example, if the wisdom "MOV.D 0 Ax1" is written for the structure type variable Ax1, 0 is stored only in the first 2 words of Ax1.**
  "BSIZE" (Page 4-70)

● Structure member offset specification

After the offset is specified for the structure members, the memory configuration in the work area of each member can be set. By setting the memory configuration, when a data link is established with external equipment, the memory configurations of the structure type members of both companies can be matched. Offset of the structure members can be set from "Auto" and "Manual".

● When the offset is automatic

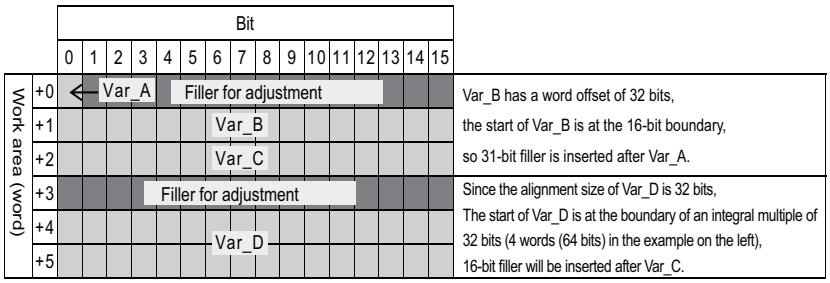
When the offset is automatic, the memory configuration is performed automatically according to the following rules. Filler for boundary adjustment will be automatically inserted.

Item	Content
Start of structure	Assigns the start of the structure to the boundary of the maximum alignment size among the structure members. However, when assigning to CPU unit devices, assignment in words is possible regardless of the alignment size.
Start of each member	Assigns to the boundary of the alignment size of each member. However, BOOL type arrays are assigned to the 16-bit boundary.

Example) Structure with the following members

Member name	Data type
Var_A	BOOL
Var_B	INT
Var_C	INT
Var_D	DINT

Var_D has the maximum alignment size of 32 bits among the structure members, so the start of the structure is configured in an even-numbered area counted from the start of the work area. The configuration of each member is as follows.



* Offset to the boundary position of the next alignment size is called filling.

● When the offset is manual

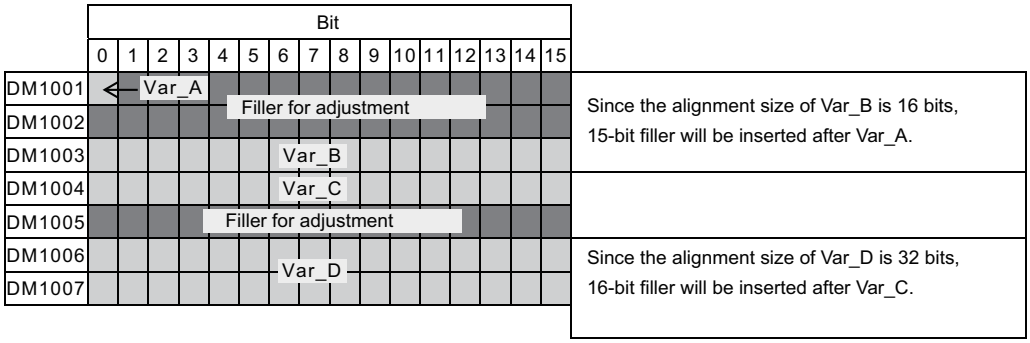
When the offset is manual, the memory configuration can be set arbitrarily according to the following rules.

Item	Content
Start of structure	Assigns the start of the structure to the boundary of the maximum alignment size among the structure members. However, it can be assigned to CPU unit devices in words.
Start of each member	BOOL type members are in bits, other data type members are in words, and the offset from the start of the structure can be specified. However, the maximum offset is up to the end of the work area for word offset and 15 bits for bit offset.

Example) When a structure with the following members and offset specifications is assigned to DM1001

Member name	Data type	Word offset	Bit offset
Var_A	BOOL	0	0
Var_B	INT	2	0
Var_C	INT	3	0
Var_D	DINT	5	0

The structure can be assigned to CPU unit devices in words, so the start of the structure can be configured in an odd-numbered device like DM1001.



Structure specifications

Item	Content
Maximum number of structures defined	800
Maximum number of members per structure	500
Maximum number of stages for nesting of structures	8*1
Data types that can be specified	BOOL / UINT / INT / UDINT / DINT / REAL / LREAL / STRING/ Array / Structure

*1 When stAAA.member is written, the number of nesting stages is 1. When stAAA.stBBB.member is written, the number of nesting stages is 2.

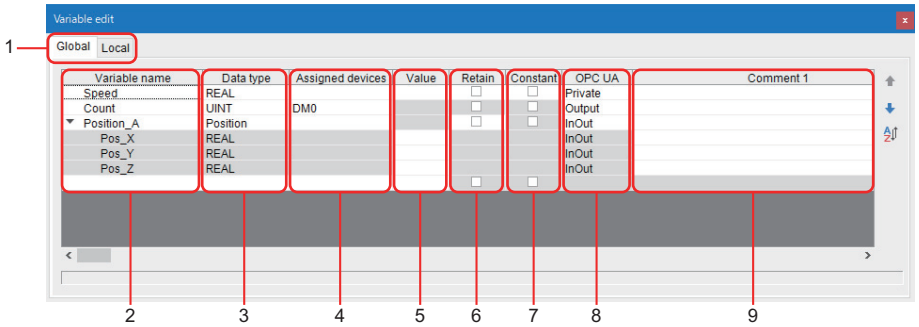
Variable Setting

This section describes the procedure from defining variables to using them in a ladder program.

■ Setting method

1 Select "Display (V)" ▶ "Variable Edit Window (L)" to open the "Variable Edit" dialog.

Other Double-click "Variable" in the workspace

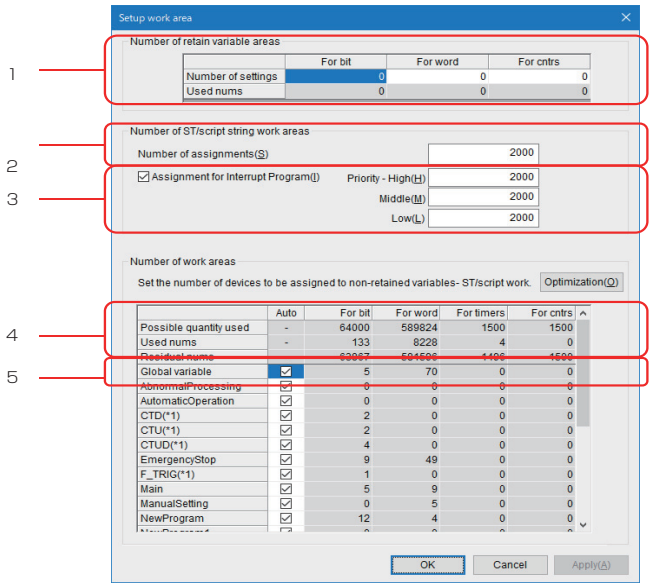


No.	Item	Content
1	Global / Local tab	Switch between the global variable edit window and local variable edit window. "Variable Types" (Page 4-55)
2	Variable name	Enter the variable name (128 half-width characters / 128 full-width characters).
3	Data type	Input the data type of the variable. BOOL/UINT/INT/UDINT/DINT/REAL/LREAL/STRING/ TIMER/COUNTER/Array / Structure / Function block
4	Assigned device	Set when manually assigning variables to devices of the CPU unit. Assignment is not mandatory. This item does not exist in the local variable edit window.
5	Value	When the constant is checked in item 7: Specify the initial value of the variable. When the constant is checked, the value is fixed and cannot be changed from the program. When the constant is not checked in item 7: Sets the initial value of the variable. It can be changed to a value other than the initial value in the program. When the assigned device is set, the initial value cannot be set in the variable edit window. Set the initial value of the assigned device in the device initial value setting. "Device default values" (Page 4-13)
6	Retain	Checked variables keep their values even when the power is turned off. The maximum number of variables that can be kept is 30000*1 including bits, words and counters. To hold the variable values, set the number of areas to hold variables in the [Work Area Settings] dialog. If the total size of the variables with Hold checked is larger than the number of areas to hold variables, a conversion error will occur.*3 "Variable Assignment" (Page 4-64) When the assigned device is set, since it is not included in the above 30000 devices the keep-alive setting should be performed for the set device. "Configuring Power Off Holding Settings (Latching)" (Page 4-26) Variables without Keep checked will be initialized to the value set in "Value" when switching from PROGRAM mode to RUN mode. If not set, they are initialized to 0.
7	Constant	Checked variables are processed as constants. included in the maximum of 1 billion definitions, Variables set as constants do not consume the work area.
8	OPC UA	Set the variables that can be read/written with the OPC UA server function. When this is set to Private, variables cannot be read/written. When this is set to InOut, variables can be read/written. When this is set to Output, variables can only be read. This item does not exist in the local variable edit window.
9	Comment	Sets the comment of the variable (128 half-width characters / 128 full-width characters)*2. The set comment is displayed in the KV STUDIO ladder edit window. Comments 1 to 8 can be set for one variable.

*1 Depending on the number of characters in the variable name, 30000 may not be set.
*2 Up to 32 characters for versions earlier than 2.1.
*3 The value will not be held when the FB instruction is used. Use the FBCALL or FBSRT instruction.

■ Variable Assignment

Variables are assigned to the work area when defined. "Work Area Setting" is used to check the number of used work areas and change the number of assigned work areas.
Select "Display (V)" ▶ "CPU System Setting (P)" in the KV STUDIO menu, and click "Local Overall Assignment Setting". Click "Work Area Setting" to display the "Work Area Setting" dialog.



No.	Item	Description
1	Number set	Set the size of the area that holds variables. Ensure that the "number set" is more than the "number used" when setting. A conversion error will occur if the "number set" is less than the "number used". Set a multiple of 16 for the "number set" for bits. Writing during RUN cannot be performed when the "number set" is changed.
	Number used	The total size of variables for which "Hold" is checked in the variable edit window is displayed.
2	Number of work areas for ST/Script string	Set the number of work areas when processing String Function in ST/Script. The work area for words is consumed.
3	Assignment for interrupt program	When processing String Function by an interrupt program or the ST/Script in periodic module and inter-unit sync module, set the number of work areas according to the priority of the interrupt. The work area for words is consumed.
4	Usable number	Displays the total number of usable work areas or function blocks / macro arguments, number of areas to hold variables and number of @FMs used.
	Number used	Displays the number of work areas used in the item.
	Remaining number	Displays the remaining number of work areas that can be used. [Remaining number] = [Usable number] – [Number used]
5	Assignment status display by program	Displays the number of work areas assigned to each program. The assigned number is calculated automatically according to the set variable size. The number used can be set manually when the "Auto" checkbox is unchecked.

Point

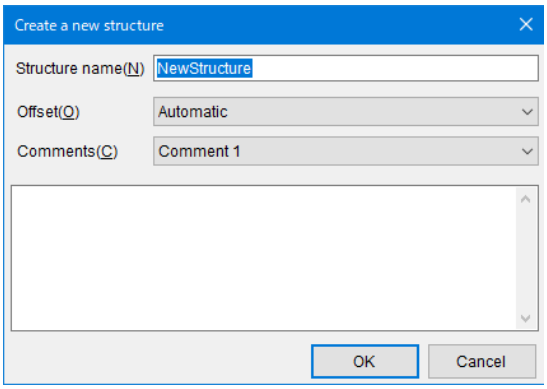
- **When using the ST/Script string function both in the interrupt program (or periodic module, inter-unit sync module) and in the normal module other than interrupt, seperetly assign the number of ST/Script string work areas for interrupt/normal modules. Alternatively, the DI / EI instructions are used during programming so that interrupt is not input during string processing in a normal module.**
  If the string function of the interrupt program is executed during the string function processing in a normal module without assignment, the same work area will be used, so that the string processing cannot be performed normally.

Structure Definition

■ Setting method

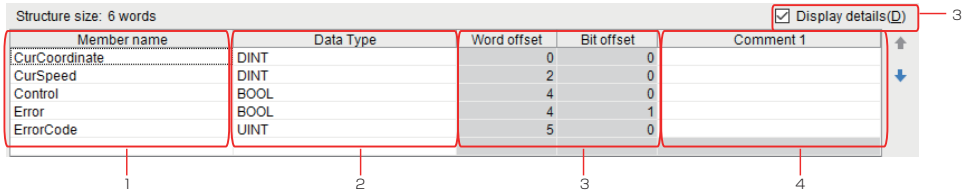
1 **Select "Display (V)" ▶ "Structure (J)" ▶ "New (N)" to create a new structure.**

Other Double-click "Structure" in the workspace



Item	Content
Structure name	Specifies the name of the structure. Up to 128 characters in regardless of full-width / half-width.
Offset	Selects the offset of each member of the structure from "Auto / Manual". "Structure member offset specification" (Page 4-61)
Comment	The string entered in the input field is displayed on the right of the structure name in the KV STUDIO workspace. Up to 8 types of comments 1 to 8 can be set.

2 **Set the member name and data type, etc. of each member.**



No.	Item	Description
1	Member name	Specifies the member name of the structure. Up to 128 characters in regardless of full-width / half-width.
2	Data type	Select the data type of the member.
3	Detail display	Check to display word offset and bit offset.
	Offset	Displays the offset information of each member of the structure. When the offset is set to "Auto", the offset automatically set is displayed, and the background is grayed out and cannot be set. When the offset is set to "Manual", the word offset and bit offset can be entered for each member.
4	Comment	Sets the comment for each member. Up to 8 types of comments 1 to 8 can be set.

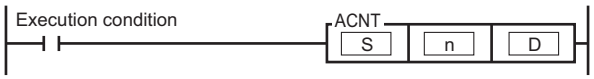
ACNT

ACNT [ ] [ ]

Get number of array elements

Gets the number of array elements in the specified dimension.

Ladder program



Input method

Operand	Description	Occupied area
S	Specifies the entire array.*1*2	—
n	Specifies the number of dimensions (1～8) to get the number of elements.	16 bits
D	Specifies the result storage location.	32 bits

*1 Only variable names of the array type can be specified for S. Variables other than arrays or elements other than variables cannot be specified.

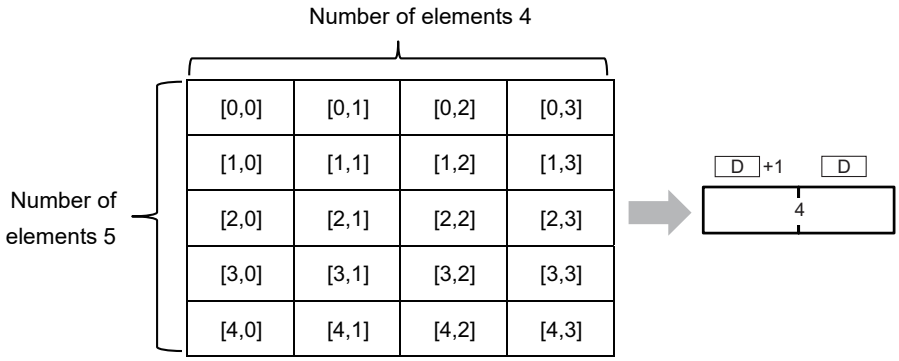
*2 Variable-length array of FB argument cannot be specified.

Description of Operation

ACNT When the execution condition is ON, the number of elements in the n th dimension of the array specified by S is stored in [D · D +1].

Example When S =Array_A (2-dimensional array of 5×4) and n =2

Array_A



Point

- **The ACNT instruction obtains the number of elements instead of the word size. For example, when UDINT (2-word unsigned integer) specifies 4 dimensions, 4 is stored in [D · D +1]. Word size 8 is not stored.**

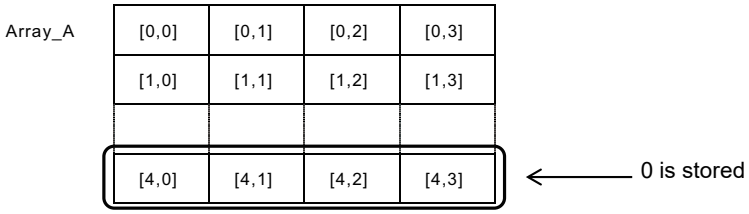
Calculation flag

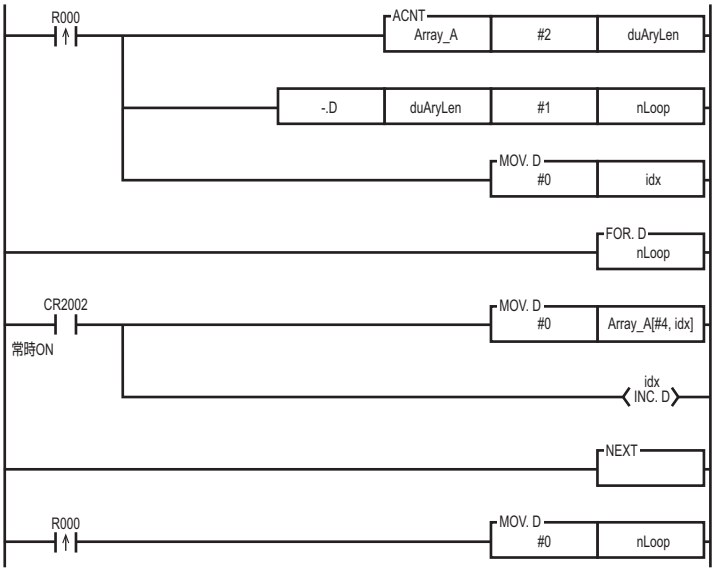
CR2009	No change
CR2010	No change
CR2011	No change

CR2012	• ON if any of the following conditions is met, and OFF otherwise
	  • When [n] is not specified as a constant or [n] is 0 or a number of dimensions that does not exist
	  • When indirect reference or index modification is inappropriate
	• No change if all of the following conditions are met.
	  • [n] is specified as a constant
	  • No indirect specification or index modification is used for [D] and when the out-of-range access of operands is not checked

Sample program

The number of elements in the second dimension of 5×4 two-dimensional array Array_A is stored in DM0 at the rising edge of input relay R000. Then, 0 is stored in Array_A[4,0], Array_A[4,1]…Array_A[4,3].





<Mnemonic list>

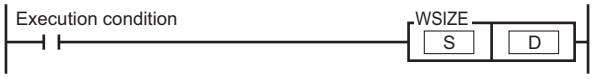
```
LDP R000
ACNT Array_A #2 duAryLen
CAL-.D duAryLen #1 nLoop
MOV.D #0 idx
FOR.D nLoop
LD CR2002
MOV #0 Array_A[#4,idx]
INC.D idx
NEXT
LDP R000
MOV.D #0 nLoop
```

WSIZE

WSIZE [ ] Get word size

Gets the word data size occupied by the specified variable.

Ladder program



Input method

Operand	Description	Occupied area
S	Specifies a variable.*1*2	—
D	Specifies the result storage location.	32 bits

*1 Only variables can be specified for S. Device cannot be specified.

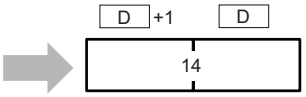
*2 Variable-length array and variable-length string of FB argument cannot be specified.

Description of Operation

WSIZE When the execution condition is ON, the word size occupied by S is stored in [D · D+1].

Example S =When the following structure type variable is specified

Member name	Data type
member1	UINT
member2	STRING[20]
member3	REAL



→ The number of words in STRING[20] is 11 after (20+1)/2 is rounded up, so 1 + 11 + 2 = 14.

Point

- **The WSIZE instruction obtains the word size. The WSIZE instruction obtains the word size. For example, when an array type variable or structure type variable with 4 UDINT data (2-word unsigned integers) is specified, since it is 2 words × 4 = 8 words, 8 is stored in[D · D+1]. It does not store the number of elements or the number of members.**
- **[D · D+1] stores the word size including the offset. For details on the offset, refer to "Structure member offset specification" (Page 4-61).**
- **[D · D+1] stores the result rounded to word size. For example, 3 is stored for BOOL[33].**

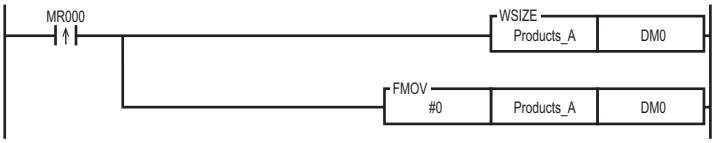
Calculation flag

CR2009	No change
CR2010	No change
CR2011	No change

CR2012	• ON if any of the following conditions is met, and OFF otherwise
	  • When indirect reference or index modification is inappropriate
	• No change if all of the following conditions are met.
	  • No indirect specification or index modification is used for the operand
	  • No device or variable is used for the array subscript

Sample program

The word size occupied by the variable "Products_A" is calculated at the rising edge of input relay MR000 and stored in DM0. Then, 0 is stored in the area occupied by "Products_A".



<Mnemonic list>

```
LDP MR000
WSIZE Products_A DM0
FMOV #0 Products_A DM0
```

Point

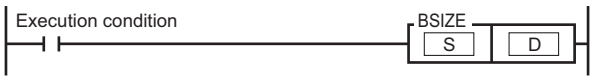
The above reference program can be used only when the word size of the variable is 65535 or less.
When the word size of the variable is larger than 65535, 0 should be stored using the recipe load function.
"Recipe Load/Save Function" (Page 5-50)

BSIZE

BSIZE [ ] [ ] Gets bit size

Gets the bit size occupied by the specified variable.

Ladder program



Input method

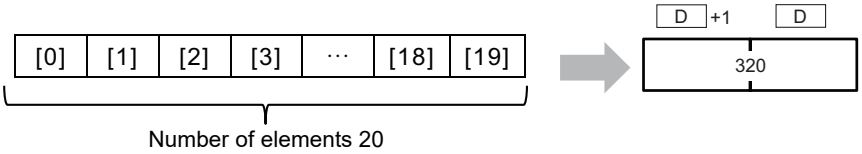
Operand	Description	Occupied area
S	Specifies a variable.*1*2	—
D	Specifies the result storage location.	32 bits

*1 Only variables can be specified for S. Device cannot be specified.
*2 Variable-length array and variable-length string of FB argument cannot be specified.

Description of Operation

BSIZE When the execution condition is ON, the bit size occupied by S is stored in [D · D+1].

Example When an INT type array variable is specified in S



→ Since the number of elements is 20 for the INT type (16 bits), 16 bits × 20 = 320 bits

Point **[D · D+1] stores the bit size including the offset. For details on the offset, refer to "Structure member offset specification" (Page 4-61).**

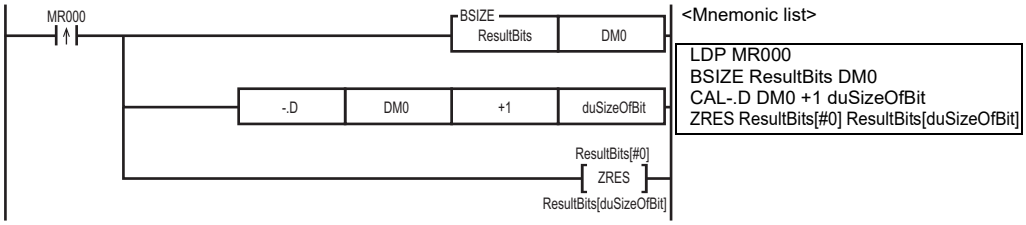
Calculation flag

CR2009	No change
CR2010	No change
CR2011	No change

CR2012	• ON if any of the following conditions is met, and OFF otherwise
	  • When indirect reference or index modification is inappropriate
	• No change if all of the following conditions are met.
	  • No indirect specification or index modification is used for the operand
	  • No device or variable is used for the array subscript

Sample program

The bit size occupied by the BOOL array type variable "ResultBits" is calculated at the rising edge of input relay MR000 and stored in DM0. The end number of the array is stored in UDINT type variable duSizeOfBit, and all the bits of "ResultBits" are reset.



Point

- **When a device such as DM is specified as the array subscript, it will be processed as a 1-word unsigned integer, so only arrays up to 65535 bits can be processed. When the number of array elements is more than 65535, a UDINT type variable will be used as the subscript as shown in the sample program.**
- **The BSIZE and BBCMP instructions are combined to compare structures. For details on the BBCMP instruction, refer to "KV-8000/7000/5000/3000/1000 Series/KV Nano Series Instruction Reference Manual".**
- **The BSIZE and BBMOV instructions are combined to transfer the entire structure to another structure. For details on the BBMOV instruction, refer to "KV-8000/7000/5000/3000/1000 Series/KV Nano Series Instruction Reference Manual".**

MEMO

4-6 Constants

The constant value and range varies according to the instruction and suffix used.

#Decimal constants

Overview Values that are displayed with a bit width of 16 or 32 bits are represented in decimal notation.

Notation

- Unsigned (when the suffix is ".U" or ".D")
  A decimal constant is represented by the prefix "#," which is followed by decimal numbers.
  (Example)

  Symbol that indicates this is a decimal value — Value

  #06143

- Signed (when the suffix is ".S" or ".L")
  A decimal constant is represented by a sign, which is followed by decimal numbers.

Range

- 16-bit, unsigned constant: #0 to #65535
- 32-bit, unsigned constant: #0 to #4294967295
- 16-bit, signed constant: -32768 to +32767
- 32-bit, signed constant: -2147483648 to +2147483647

Reference

- You can use zero suppression when you write your programs.
- You can also prefix signed constants with "#." If you include "#," only "+" may be omitted.
- When you select XYM marking mode, you can append "K" when you write these constants.
  "XYM marking", (page 4-24)

$Hexadecimal constants

Overview Values that are displayed with two bit widths,16 bits and 32 bits, are represented in hexadecimal notation.

Notation A hexadecimal constant is represented by the prefix "$," which is followed by hexadecimal numbers.
(Example)

Symbol that indicates this is a hexadecimal value — Value

$1AF7

Range

- 16-bit constant: $0000 to $FFFF
- 32-bit constant: $00000000 to $FFFFFFFF

Point

If you enter a hexadecimal value that contains letters A to F without the prefix "$," an error will occur. If you enter a hexadecimal value that does not contain letters A to F without the prefix "$," the hexadecimal value will be recognized as a relay (R).

Reference

- You can use zero suppression when you write your programs.
- When you select XYM marking mode, you can append "H" when you write these constants.
  "XYM marking", (page 4-24)

Single precision floating-point constants

Overview These constants can be written as the operands of floating-point instructions.

Notation
- Signed decimal values can be written as-is for these constants.
- For eight digits or more, exponential notation is used.

Range -3.4E + 38 ≤ N ≤ -1.4E - 45
N = 0
+1.4E - 45 ≤ N ≤ +3.4E + 38 (approximately 7 significant digits)

Reference The prefix "#" can be used. Only "+" can be omitted.

Double precision floating-point constants

Overview These constants can be written as the operands of double precision floating-point instructions.

Notation
- Signed decimal values can be written as-is for these constants.
- For 15 digits or more, exponential notation is used.

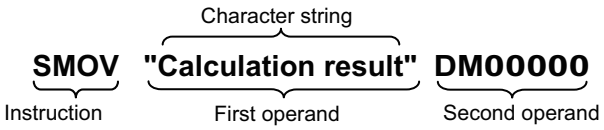
Range -1.79E + 308 ≤ N ≤ -2.23E - 308
N = 0
+2.23E - 308 ≤ N ≤ +1.79E + 308 (approximately 16 significant digits)

Reference
- The prefix "#" can be used. Only "+" can be omitted.
- If a sub-normal double precision floating-point number (-2.23E - 308 < N ≤ -4.9E - 324 or +4.9E - 324 ≤ N < +2.23E - 308) that is outside of the normal number range is calculated, a calculation error will occur.
  If the calculation result is a sub-normal number, the value will be set to 0.

Fixed character strings

Overview These can be written as the operands of character string processing instructions and calendar contact instructions.

Notation Surround the character string with quotation marks.
(Example)



Point **If you do not include the opening and closing quotation marks, the string will be processed as a variable.**

Reference To process the quotation mark as a character, escape it by entering two quotation marks.

Range Up to 499 characters (half width)

Point **Up to 31 characters when the KV-8000 CPU function version is lower than 2.0.**

Point **In ST language, the string is written between " ' ' ".**

4-7 How Data Is Processed

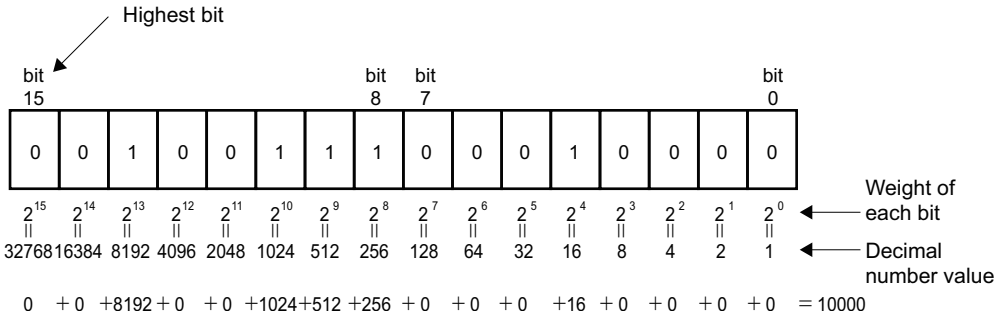
The values that are processed by ladder programs are binary values, decimal (unsigned and signed) values, hexadecimal values, binary coded decimal (BCD) values, single precision floating-point real numbers, and double precision floating-point real numbers.
The character codes that can be processed are ASCII code and shift JIS code.

Binary Data

On the KV-8000 Series, all data is processed by devices and internal registers as binary data.
One digit of a binary value is processed as a bit unit. A bit indicates two states: 0 (OFF) and 1 (ON).

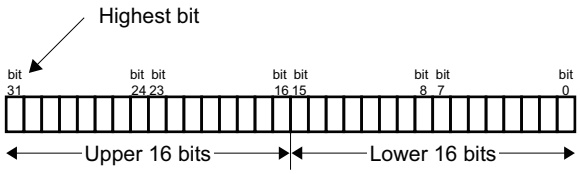
■ 16-bit binary data

16 bits (16 binary digits) of data are normally processed in units of words. For an unsigned decimal value, one word indicates a value in the range of 0 to 65535.
(Example)



■ 32-bit binary data

For an unsigned decimal value, 32 bits of data (two words) indicates a value in the range of 0 to 4294967295.



If you are using a DM, EM, ZF(FM), TM, or other 16-bit word device to process a 32-bit device, 2 consecutive words will be used. The lower 16 bits of data are processed by the word device with the smaller number, and the upper 16 bits of data are processed by the word device with the larger number.

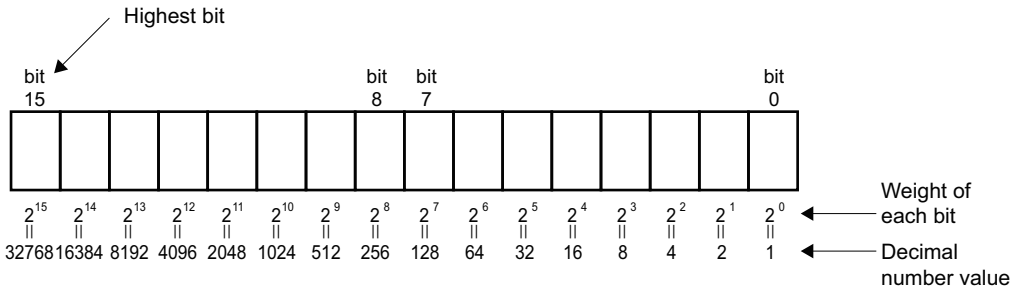
Reference

Whether a device is processed as 16-bit data or 32-bit data is determined by the suffix.
"Suffixes" (page 4-9)

Value Notation

■ Binary BIN

The internal binary data of the CPU unit is indicated as 0s and 1s without any changes. When a digit is greater than 1, the digit is carried.



(Example) 0110110100110001

Reference These values cannot be entered as instruction operands.

■ Decimal DEC

The internal binary data of the CPU unit is indicated as decimal values.

● Unsigned

For 16-bit data, the range of values that can be processed is 0 to 65535. For 32-bit data, the range of values that can be processed is 0 to 4294967295.

● Signed

Whether a value is positive or negative is determined by the highest bit (0: positive; 1: negative).
For 16-bit data, the range of values that can be processed is -32768 to +32767. For 32-bit data, the range of values that can be processed is -2147483648 to +2147483647.

Reference These values can be entered as instruction operands. Prefix the value with "#" or a sign.
(Examples) #1000, +12, and -25489

■ Hexadecimal HEX

The internal binary data of the CPU unit is indicated as hexadecimal values.
For hexadecimal values, four bits are processed as one digit. Because the values that can be expressed in a single digit are 0 to 15 in decimal notation, A, B, C, D, E, and F are used for values greater than 9. When a digit is greater than F, the digit is carried.
For 16-bit data, the range of values that can be processed is $0000 to $FFFF. For 32-bit data, the range of values that can be processed is $00000000 to $FFFFFFFF.

Reference These values can be entered as instruction operands. Prefix the value with "$."
(Example) $00FF

■ Binary Coded Decimal BCD

With binary coded decimal notation, 4 bits are processed as 1 digit in the same manner as hexadecimal notation. In the same manner as decimal notation, when a digit is 9, the digit is carried, so the letters A to F are not used.
For 16-bit data, the range of values that can be processed is 0000 to 9999. For 32-bit data, the range of values that can be processed is 00000000 to 99999999.

Example

bit 15							bit 8	bit 7							bit 0
0	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1

When 16-bit data is stored as shown above, while the actual value is $0987 (#2439 in decimal notation), in binary coded decimal, this value is handled in decimal notation as "0987."

0	0	0	0	1	0	0	1	1	0	0	0	0	1	1	1

Thousands position	Hundreds position	Tens position	Ones position
0	9	8	7

Binary coded decimal is a useful notation to use when processing data that comes from an external source (such as a digital switch).
However, this data must be processed as binary data in calculations performed by the CPU unit. When reading data from an external source, use the TBIN instruction to convert the data to binary data. When outputting data to an external device (such as a digital display), use the TBCD instruction to convert the data to binary coded decimal data.

Reference To enter binary coded data values as instruction operands, prefix the value with "$" to enter the value in hexadecimal notation.

■ Single precision floating-point real numbers FLOAT

Floating-point real numbers are processed as values in which 32-bit data has a decimal point.
You can use values in the following range (N).

$$-3.4 \times 10^{+38} \leq N \leq -1.4 \times 10^{-45}$$
$$N = 0$$
$$+1.4 \times 10^{-45} \leq N \leq +3.4 \times 10^{+38}$$

Reference
- These values can be entered as floating-point instruction operands.
  In addition to the normal notation using the decimal point (example: -1.2345), you can also use an exponential notation using "E."
  (Examples) $-2.743 \times 10^{12} \rightarrow$ -2.743E + 12
  $+8.9025 \times 10^{-4} \rightarrow$ +8.9025E - 4
- The precision of floating-point real numbers is 32-bit single precision. There are approximately 7 significant digits.
- Floating-point real numbers stored in 32-bit data are represented internally in the IEEE standard format.

■ Double precision floating-point real numbers DOUBLE-FLOAT

Double precision floating-point real numbers are processed as values in which 64-bit data has a decimal point.
You can use values in the following range (N).

$$-1.79 \times 10^{+308} \leq N \leq -2.23 \times 10^{-308}$$
$$N = 0$$
$$+2.33 \times 10^{-308} \leq N \leq +1.79 \times 10^{+308}$$

Reference

- These values can be entered as double precision floating-point instruction operands. In addition to the normal notation using the decimal point (example: -1.2345), you can also use an exponential notation using "E."
  (Examples) $-2.1437956 \times 10^{12} \rightarrow -2.1437956E + 12$
  $+7.63259856 \times 10^{-4} \rightarrow +7.63259856E - 4$
- The precision of double precision floating-point real numbers is 64-bit double precision. There are approximately 16 significant digits.
- Double precision floating-point real numbers stored in 64-bit data are represented internally in the IEEE standard format.
- Whether a device is processed as single precision floating-point data or as double precision floating-point data is determined by the suffix.
  "Suffixes" (page 4-9)

Correlations between Value Notations

The correlations between binary values, decimal (unsigned and signed) values, hexadecimal values, and binary coded decimal values are shown below.

■ 16-bit data

Decimal		Hexa-decimal (HEX)	Binary (BIN)			
Unsigned	Signed					
0	+0	0	0000	0000	0000	0000
1	+1	1	0000	0000	0000	0001
2	+2	2	0000	0000	0000	0010
3	+3	3	0000	0000	0000	0011
4	+4	4	0000	0000	0000	0100
5	+5	5	0000	0000	0000	0101
6	+6	6	0000	0000	0000	0110
7	+7	7	0000	0000	0000	0111
8	+8	8	0000	0000	0000	1000
9	+9	9	0000	0000	0000	1001
10	+10	A	0000	0000	0000	1010
11	+11	B	0000	0000	0000	1011
12	+12	C	0000	0000	0000	1100
13	+13	D	0000	0000	0000	1101
14	+14	E	0000	0000	0000	1110
15	+15	F	0000	0000	0000	1111
16	+16	10	0000	0000	0001	0000
⋮	⋮	⋮			⋮	
99	+99	63	0000	0000	0110	0011
100	+100	64	0000	0000	0110	0100
⋮	⋮	⋮			⋮	
32767	+32767	7FFF	0111	1111	1111	1111
32768	-32768	8000	1000	0000	0000	0000
⋮	⋮	⋮			⋮	
65534	-2	FFFE	1111	1111	1111	1110
65535	-1	FFFF	1111	1111	1111	1111

Binary Coded Decimal (BCD)	Binary (BIN)			
0000	0000	0000	0000	0000
0001	0000	0000	0000	0001
0002	0000	0000	0000	0010
0003	0000	0000	0000	0011
0004	0000	0000	0000	0100
0005	0000	0000	0000	0101
0006	0000	0000	0000	0110
0007	0000	0000	0000	0111
0008	0000	0000	0000	1000
0009	0000	0000	0000	1001
0010	0000	0000	0001	0000
0011	0000	0000	0001	0001
0012	0000	0000	0001	0010
0013	0000	0000	0001	0011
0014	0000	0000	0001	0100
0015	0000	0000	0001	0101
0016	0000	0000	0001	0110
⋮			⋮	
0099	0000	0000	1001	1001
0100	0000	0001	0000	0000
⋮			⋮	
0999	0000	1001	1001	1001
1000	0001	0000	0000	0000
⋮			⋮	
9998	1001	1001	1001	1000
9999	1001	1001	1001	1001

■ 32-bit data

Decimal		Hexadecimal	Binary (BIN)							
Unsigned	Signed	(HEX)								
0	+0	0	0000	0000	0000	0000	0000	0000	0000	0000
1	+1	1	0000	0000	0000	0000	0000	0000	0000	0001
2	+2	2	0000	0000	0000	0000	0000	0000	0000	0010
3	+3	3	0000	0000	0000	0000	0000	0000	0000	0011
4	+4	4	0000	0000	0000	0000	0000	0000	0000	0100
5	+5	5	0000	0000	0000	0000	0000	0000	0000	0101
6	+6	6	0000	0000	0000	0000	0000	0000	0000	0110
7	+7	7	0000	0000	0000	0000	0000	0000	0000	0111
8	+8	8	0000	0000	0000	0000	0000	0000	0000	1000
9	+9	9	0000	0000	0000	0000	0000	0000	0000	1001
10	+10	A	0000	0000	0000	0000	0000	0000	0000	1010
11	+11	B	0000	0000	0000	0000	0000	0000	0000	1011
12	+12	C	0000	0000	0000	0000	0000	0000	0000	1100
13	+13	D	0000	0000	0000	0000	0000	0000	0000	1101
14	+14	E	0000	0000	0000	0000	0000	0000	0000	1110
15	+15	F	0000	0000	0000	0000	0000	0000	0000	1111
16	+16	10	0000	0000	0000	0000	0000	0000	0001	0000
⋮	⋮	⋮	⋮							
99	+99	63	0000	0000	0000	0000	0000	0000	0110	0011
100	+100	64	0000	0000	0000	0000	0000	0000	0110	0100
⋮	⋮	⋮	⋮							
2147483647	+2147483647	7FFFFFFF	0111	1111	1111	1111	1111	1111	1111	1111
2147483648	-2147483648	8000000	1000	0000	0000	0000	0000	0000	0000	0000
⋮	⋮	⋮	⋮							
4294967294	-2	FFFFFFFE	1111	1111	1111	1111	1111	1111	1111	1110
4294967295	-1	FFFFFFFF	1111	1111	1111	1111	1111	1111	1111	1111

Binary Coded Decimal (BCD)	Binary (BIN)							
00000000	0000	0000	0000	0000	0000	0000	0000	0000
00000001	0000	0000	0000	0000	0000	0000	0000	0001
00000002	0000	0000	0000	0000	0000	0000	0000	0010
00000003	0000	0000	0000	0000	0000	0000	0000	0011
00000004	0000	0000	0000	0000	0000	0000	0000	0100
00000005	0000	0000	0000	0000	0000	0000	0000	0101
00000006	0000	0000	0000	0000	0000	0000	0000	0110
00000007	0000	0000	0000	0000	0000	0000	0000	0111
00000008	0000	0000	0000	0000	0000	0000	0000	1000
00000009	0000	0000	0000	0000	0000	0000	0000	1001
00000010	0000	0000	0000	0000	0000	0000	0001	0000
00000011	0000	0000	0000	0000	0000	0000	0001	0001
00000012	0000	0000	0000	0000	0000	0000	0001	0010
00000013	0000	0000	0000	0000	0000	0000	0001	0011
00000014	0000	0000	0000	0000	0000	0000	0001	0100
00000015	0000	0000	0000	0000	0000	0000	0001	0101
00000016	0000	0000	0000	0000	0000	0000	0001	0110
⋮	⋮							
00000099	0000	0000	0000	0000	0000	0000	1001	1001
00000100	0000	0000	0000	0000	0000	0001	0000	0000
⋮	⋮							
00000999	0000	0000	0000	0000	0000	1001	1001	1001
00001000	1000	0000	0000	0000	0001	0000	0000	0000
⋮	⋮							
99999998	1001	1001	1001	1001	1001	1001	1001	1000
99999999	1001	1001	1001	1001	1001	1001	1001	1001

■ Changes to values when the value is greater than the maximum value or smaller than the minimum value

When the result of an arithmetic operation is greater than the maximum value or smaller than the minimum value, the value is changed according to the data format as shown below. Whether data is processed as unsigned values or signed values is determined by the suffix.

"Suffixes" (page 4-9)

■ 16-bit data

Changes to values when 65535 is exceeded

Unsigned decimal notation: 3, 2, 1, 0, 65535, 65534, 65533, 65532, ⋮, 32771, 32770, 32769, 32768, 32767, 32766, 32765, 32764, 32763

Hexadecimal notation: 0003, 0002, 0001, 0000, FFFF, FFFE, FFFD, FFFC, ⋮, 8003, 8002, 8001, 8000, 7FFF, 7FFE, 7FFD, 7FFC, 7FFB

Changes to values when the value is greater than the maximum value or smaller than the minimum value

Unsigned decimal notation: 3, 2, 1, 0, -1, -2, -3, -4, ⋮, -32765, -32766, -32767, -32768, 32767, 32766, 32765, 32764, 32763

Signed decimal notation: 0003, 0002, 0001, 0000, FFFF, FFFE, FFFD, FFFC, ⋮, 8003, 8002, 8001, 8000, 7FFF, 7FFE, 7FFD, 7FFC, 7FFB

Reference

Whether a device is processed as 16-bit data or 32-bit data is determined by the suffix.

"Suffixes" (page 4-9)

■ 32-bit data

Changes to values when 4294967295 is exceeded

Unsigned decimal notation: 3, 2, 1, 0, 4294967295, 4294967294, 4294967293, 4294967292, ⋮, 2147483651, 2147483650, 2147483649, 2147483648, 2147483647, 2147483646, 2147483645, 2147483644, 2147483643

Hexadecimal notation: 00000003, 00000002, 00000001, 00000000, FFFFFFFF, FFFFFFFE, FFFFFFFD, FFFFFFFC, ⋮, 80000003, 80000002, 80000001, 80000000, 7FFFFFFF, 7FFFFFFE, 7FFFFFFD, 7FFFFFFC, 7FFFFFFB

Changes to values when the value is greater than the maximum value or smaller than the minimum value

Unsigned decimal notation: 3, 2, 1, 0, -1, -2, -3, -4, ⋮, -2147483645, -2147483646, -2147483647, -2147483648, 2147483647, 2147483646, 2147483645, 2147483644, 2147483643

Signed decimal notation: 00000003, 00000002, 00000001, 00000000, FFFFFFFF, FFFFFFFE, FFFFFFFD, FFFFFFFC, ⋮, 80000003, 80000002, 80000001, 80000000, 7FFFFFFF, 7FFFFFFE, 7FFFFFFD, 7FFFFFFC, 7FFFFFFB

Reference

- Single precision floating-point data
  Overflow:If the absolute value is larger than 3.4×10^{38}, the value is set to infinity (∞).
  Underflow:If the absolute value is less than 1.4×10^{-45}, the value is set to 0.

- Double precision floating-point data
  Overflow:If the absolute value is larger than 1.79×10^{308}, the value is set to infinity (∞).
  Underflow:If the absolute value is less than 2.23×10^{-308}, the value is set to 0.

Character Codes

The numbers that are assigned to the characters that are used to process ladder programs in terms of character strings are called character codes.
The device can handle the character codes used by the KV-8000 Series.

■ ASCII code

This character code uses 8 bits to represent 1 half-width character.
(Examples) A ... $41 1 ... $31

"ASCII Code Table" (page A-18)

■ UNICODE

This character code uses 16bits to represent 1double-byte character.

Reference

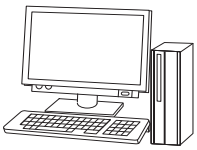
It will be entered as the operand of character processing instruction, calendar point of contact instruction. Character strings are placed inside quotation marks (").
(Examples) "ABC format" "MON/08:30:15"

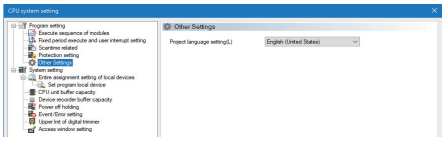
UNICODE supports the alphabet, symbols, Japanese, Chinese, Korean and other languages. Please note that some character codes are not available.

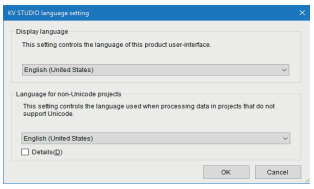
Point

- **UNICODE integrates Chinese character of Japanese, Simplified Chinese, Traditional Chinese and Korean. Therefore characters deemed to have the same shape are assigned the same code. For that reason, Japanese fonts may differ from Chinese fonts.**
  For example, in the Chinese variant of the character "骨," the right angle element in the top box is on the left side.
- **To prevent the intrusion of character variants from other languages, the language code in the PLC has to be synchronized to the language code used by KV STUDIO.**
  The KV STUDIO language settings can only be changed when the project is not open.

PLC setting　　KV STUDIO setting





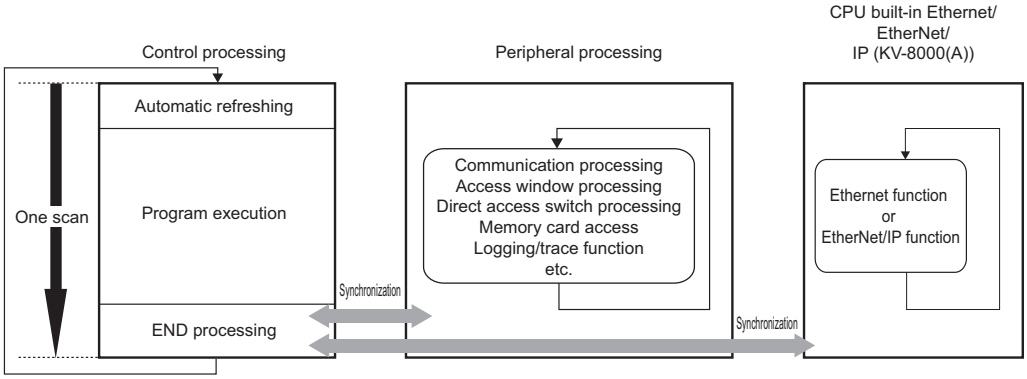


CPU system settings →other settings

"Tools (T)" → "KV STUDIO language settings (2)"

CPU Unit Operation

The CPU unit follows the procedure shown below to execute programs repeatedly.



High speed processing is achieved through parallel execution of control processing, peripheral processing and CPU built-in Ethernet/EtherNet/IP processing.

Reference

In addition to the three processes mentioned above, there are also unit-interrupting process, interrupting processes by fixed period module execution/inter-unit synchronizing modules as well.

"4-13 Interrupts" (page 4-206)

■ Control processing

One control processing stage is called a scan. Scan time is the execution time of one scan.

● Automatic refreshing

The status and values of devices assigned to the CPU built-in functions and expansion units are updated. First, the device status and values to be generated are transferred to each unit. Next, the status and values of each unit are transferred to the input devices.

● Program execution

Calculation processing of the stored program method is performed. The ladder program instructions are executed one instruction at a time in order starting from the first step.

● END processing

Device values and other data used in peripheral processing and CPU built-in Ethernet/EtherNet/IP processing executed in parallel are applied to control processing. Furthermore, an error check of the whole system is performed.

Reference

Normally, the END processing time is determined automatically according to the program execution time. However, you can set the END processing time in situations such as when there is a great deal of data communications with the peripherals.

"Setting the END processing time" (page 4-92)

■ Peripheral processing

In this process, information other than that used in program processing, such as communications and data management information, is processed. In addition, instructions related to memory cards, calendar timers and similar are executed.
The process is designed to minimize the effect of processing on scan time by operating in parallel with control processing.

Direct processing

As written above, automatic refreshing processing, in which the status and values of devices assigned to the CPU built-in functions and expansion units are updated, is normally performed only once per scan. To update information during program execution (direct processing), use one of the following methods.

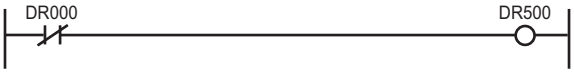
Point **The following direct processing can only be executed when a real time compatible expansion unit is connected.**

■ Expansion I/O unit

● Direct I/O relay (DR)

When you specify an I/O relay as the operand of an instruction, if you specify "DR" instead of "R," I/O will be updated when the instruction is executed. For details on the instructions with which you can use "DR," see the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

Example)



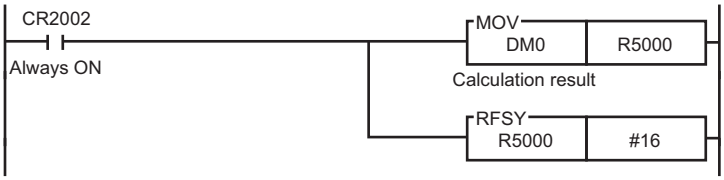
Reference If you are using XYM marking, write "DX" and "DY."

"XYM marking" (page 4-25)

● RFSX and RFSY instructions

When the instructions are executed, all the I/O relays specified with the operands are updated at once.

Example)



For details, see the RFSX and RFSY instructions in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

■ Expansion special units

Each unit has a buffer memory that makes it possible to retrieve unit information and write commands any time during program execution. EtherNet/IP and Ethernet built into the KV-8000 CPU have similar functionality.

● Dedicated unit instructions

The unit-specific instructions make it possible to exchange data with units without having to remember backup memory addresses.

See "Unit-specific instructions" in "Expansion Special Unit User's Manual"

● Buffer Memory Access (UREAD/UWRIT/UFILL) Instruction

The buffer memory read instruction (UREAD) and the buffer memory write instruction (UWRIT) can be used for directly read/write data from and to buffer memory.

See "UREAD/UWRIT/UFILL instructions" in the "KV-8000/7000/5000/3000/1000 Series Instruction Reference Manual"

See "List of Buffer Memories" in the "Expansion Unit User's Manual"

■ CPU BUILT-IN FUNCTIONS

Like the expansion special units, the CPU unit can be processed directly.
In addition, there is an exclusive instruction term that can update the current value of free run counter to the latest value.
Refreshing free run counter current value...RFSFRC instruction

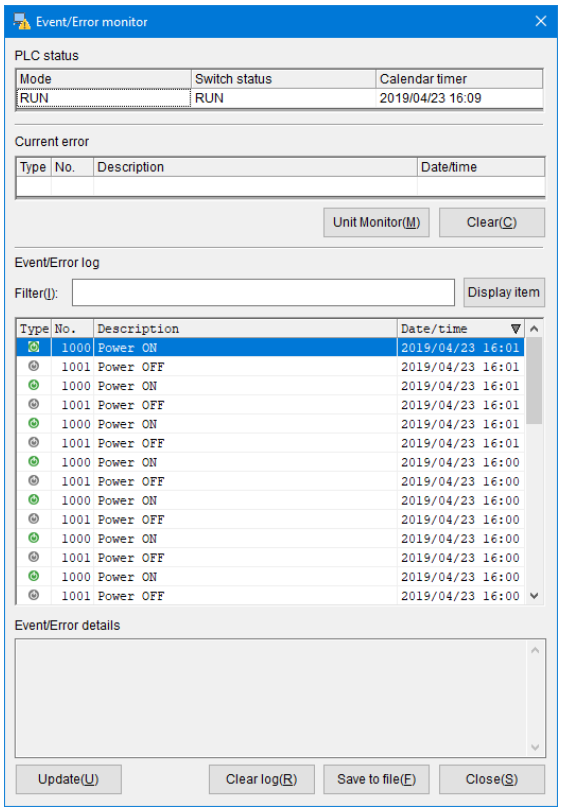
Operations when power is turned on and off

The following items are checked when the power is turned on. If everything is OK, operations are started.

- Hardware check and initialization
- Device check and initialization "Configuring Power Off Holding Settings (Latching)" (page 4-26)
- Expansion bus check and initialization
- Ladder program check and execution preparation

■ Power on/off log

You can retain up to 20 power on log entries and 20 power off log entries. Log entries are saved in volaltile RAM. The newer the log entry, the smaller the log entry number. If an attempt is made to store more than 20 log entries, existing log entries are deleted starting with the oldest one. In addition, while using KV STUDIO to perform monitoring, you can view the log in the Event/error monitor.



Event/error settings

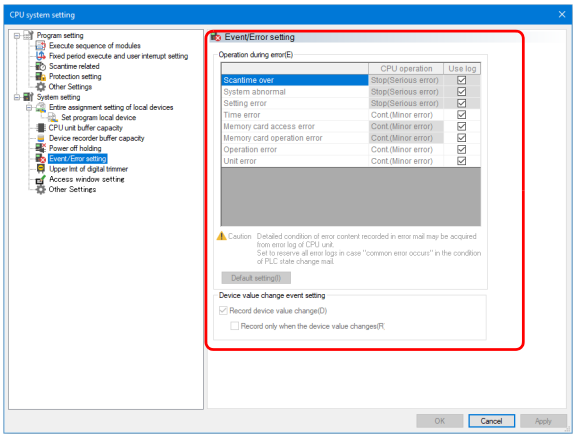
There are two types of errors: serious errors that stop operation and minor errors in which operation continues.

For details on the CPU unit errors, see "A-1 List of PLC Unit Errors" (page A-2).

For details on the expansion unit errors, see the "User's Manual" of each expansion unit.

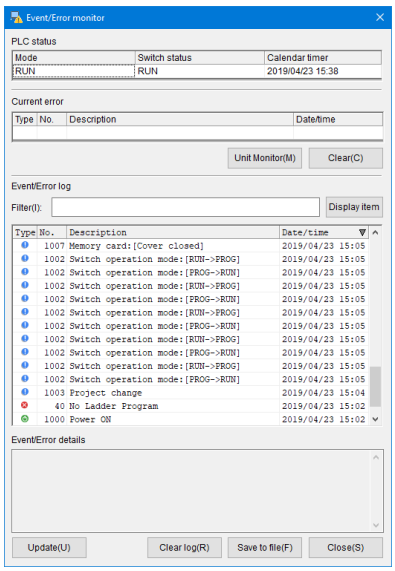
■ Event/error setting

To set the severity (serious or minor) of CPU unit errors, open "CPU system settings" in the KV STUDIO workspace and select "Event/error setting".



■ Error log

Up to 32 serious error logs or 200 minor error logs could be kept. When it exceeds the maximum number, it will be deleted from the old one. In addition, while using KV STUDIO to perform monitoring, you can view the log in the "Event/error monitor".



● Latest error details (CM)

The detailed information of the latest error is recorded in control memory entries.

Control Memory	Description	
CM5100	Detailed information of the latest serious error, 1 unit	Year and month represented as a decimal number (the last two digits represent the month, and the digits starting from the hundreds digit represent the year) Example: 1910 -> October 2019
CM5101		Day
CM5102		Hour
CM5103		Minute
CM5104		Second
CM5105		Error number
CM5106		Detailed information, number of valid pieces of data*
CM5107		Detailed information, piece #1*
CM5108		Detailed information, piece #2*
: :		: :
CM5126		Detailed information, piece #20*
CM5150	Detailed information of the latest minor error, 1 unit	Year and month represented as a decimal number (the last two digits represent the month, and the digits starting from the hundreds digit represent the year) Example: 1910 -> October 2019
CM5151		Day
CM5152		Hour
CM5153		Minute
CM5154		Second
CM5155		Error number
CM5156		Detailed information, number of valid pieces of data*
CM5157		Detailed information, piece #1*
CM5158		Detailed information, piece #2*
: :		: :
CM5156		Detailed information, piece #20*

* The detailed information varies as shown on the next page according to the number of the error that occurred. Please refer to the detailed information up to the number stored in Detailed information, number of valid pieces of data.

- Error number 128 (calculation error)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5157)	Error details 0 (Not used.) 1 Incorrect indirect specification. 2 The indirectly specified target is T/C. 3 Incorrect simple indirect specification. 4 Outside of operand range. 5 Floating-point real number overflow. 6 Incorrect device specification. 7 The ROOT instruction operand is negative. 8 No bits set to ON with the ENCO instruction. 9 Floating-point real number format error. 10 Incorrect trigonometric function instruction operand. 11 Data could not be converted. 12 Incorrect table specification. 13 Incorrect time comparison instruction operand. 14 Incorrect cam switch operand. 15 Incorrect frequency counter instruction operand. 16 Divide by zero. 17 Illegal logging ID. 18 Direct processing error. 19 Unit on which direct processing cannot be performed. 20 Jump destination subroutine does not exist. 21 Data being written during RUN mode, so instruction cannot be executed. 22 Error for expansion unit-dedicated instructions (unit number). 23 Reserved for the system. 24 PIDAT, outside of operation setting range. 25 PIDAT, outside of sampling period range. 26 PIDAT, outside of comparison constant range. 27 PIDAT, outside of integral constant range. 28 PIDAT, outside of differential constant range. 29 PIDAT comparison constant, two-degree freedom parameters outside of range. 30 PIDAT integral constant, two-degree freedom parameters outside of range. 31 PIDAT differential constant, two-degree freedom parameters outside of range. 32 PIDAT, outside of operation quantity upper limit or lower limit range. 33 PIDAT, outside of operation quantity change width limit range. 34 PIDAT, outside of measured value change width range. 35 PIDAT, AT state transition error. 36 PIDAT, outside of operation control flag range. 37 Operation period exceeded. 38 PIDAT, outside of AT adjustment parameter range. 39 PIDAT, outside of AT hysteresis range. 40 PIDAT, outside of AT timeout range. 41 PIDAT, outside of AT calculation control mode setting range. 42 PIDAT, AT timeout. 43 PIDAT, AT comparison constant exceeded. 44 PIDAT, AT integral constant exceeded. 45 PIDAT, AT differential constant exceeded. 46 Root of negative number cannot be calculated. 47 Incorrect ABS instruction operand. 48 Calculation overflow. 49 UR, UM, or UV unit not connected. 50 Unit device (R) not assigned. 51 Unit device (DM) not assigned. 52 Unit categories disagree. 53 operation recorder ID invalid 54 Stack overflow
Detailed information, piece 2 (CM5158)	Fixed to 00H.
Detailed information, piece 3 (CM5159)	Upper bits of the step number at which the error occurred.
Detailed information, piece 4 (CM5160)	Lower bits of the step number at which the error occurred.
Detailed information, pieces 5 to 20 (CM5161 to CM5176)	The module name is saved according to settings made in the project language. (Data other than the module name is 00H.)

- Error numbers 50, 53, 55, 56, 58, 59, and 63 (unit ID errors)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5107)	Unit number (the unit number will not be stored if it cannot be determined according to the occurrence conditions of the error)
Detailed information, pieces 2 to 20 (CM5108 to CM5126)	Fixed to 00H

- Error number 54 (number of units error)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5107)	The number of units as specified in the unit setup information
Detailed information, piece 2 (CM5108)	The number of units that are actually connected
Detailed information, pieces 3 to 20 (CM5109 to CM5126)	Fixed to 00H

- Error number 127 (Auto loading failure error)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5157)	Detailed content of the error Please refer to " The error code during the execution of load, save" (page 5-143) for details.
Detailed information, pieces 2 to 20 (CM5158 to CM5176)	Fixed to 00H

- Error number 129 (unit error)

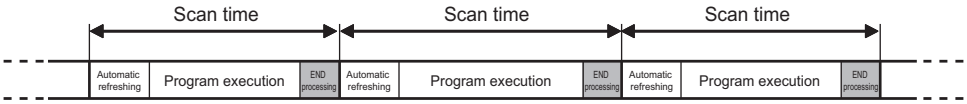
Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5157)	Unit number
Detailed information, piece 2 (CM5158)	Error number of each unit
Detailed information, pieces 3 to 20 (CM5159 to CM5176)	Fixed to 00H

- Error numbers other than those listed above

Detailed Information, Piece n	Description
Detailed information, pieces 1 to 20	Fixed to 00H

Scan Time

The time required to execute one control processing sequence (automatic refreshing → program execution → END processing) is called the scan time. The scan time varies according to the program size and the type of instructions that are used.

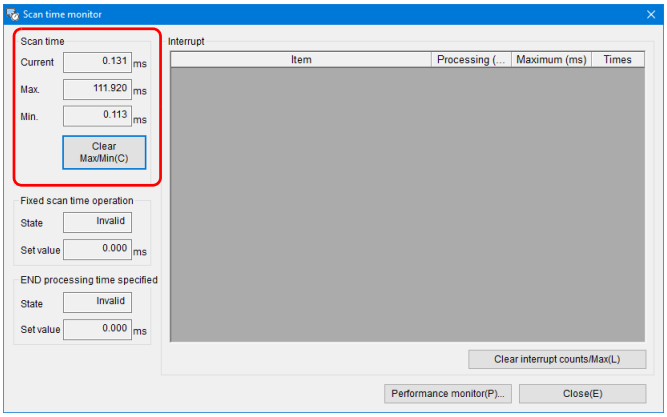


Point

If the scan time exceeds 300 ms, a scan time over error (CPU error number 30) will occur, and operations will stop (the mode will be switched to PRG mode).

Reference

- To view the actual measured scan time, while using KV STUDIO to perform monitoring, on the "Debug" menu, click "Scan time monitor."



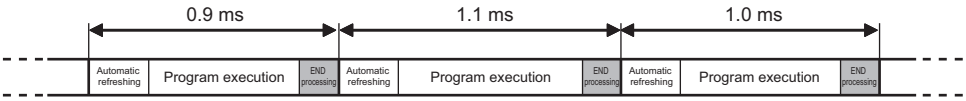
- The actual measured value of scan time can be confirmed at CM1000 to 1001 (1μs unit).

"A-2 Lists of CR and CM Devices" (page A-10)

Fixed scan time operation

During normal program execution, the scan time varies according to factors such as (1) the execution status and processing details of each program and (2) interrupt processing.
By using the fixed scan time processing function, you can maintain a fixed scan time.

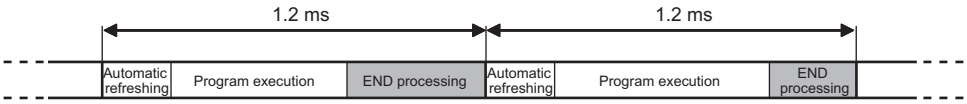
● Not using the fixed scan time operation function (normal operation)



The scan time varies according to the execution status of each instruction of each program.

● Using the fixed scan time operation function

Example) When the value is set to 1.2 ms:



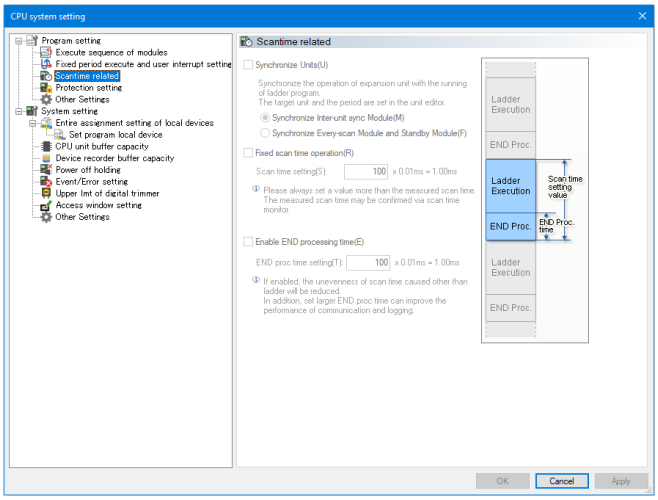
In one scan, after all processing is completed, operations do not proceed to the next step until the set amount of time elapses, so it is possible to fix the scan time.

Point

- **If the scan time is longer than the set time, the scan time will be extended. Be sure to confirm the actual measured scan time of the target program before you set the time.**
- **If an interrupt occurs during the wait time before the next scan, the interrupt program will be executed (only when interrupts are enabled). Even if the set time of fixed scan time operation elapses during the execution of the interrupt program, operations will not proceed to the next scan, and instead the scan time will be extended until the interrupt program execution is completed.**

■ Configuring fixed scan time operation settings

To configure the fixed scan time operation settings, from the workspace, open the "CPU system setting" dialog box, and then click "Program setting" and then "Scantime related."



The range that can be configured is “0.10ms to 200.00ms”.

Point

Set a value that is larger than the actual scan time.

Reference

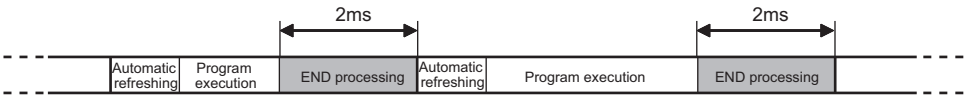
- The actual measured value of scan time can be confirmed at CM1000 to 1001 (1μs unit).
  "A-2 Lists of CR and CM Devices" (page A-10)
- If the scan time value is exceeded, CR2304 is set to ON for one scan only.

Setting the END processing time

During normal program execution, the scan time varies according to factors such as (1) the execution status and processing details of each program and (2) interrupt processing.
If you set the END processing time, END processing is executed for a fixed length of time in each scan. Because the maximum scan time is large regardless of whether there is a great deal of data communications with the peripherals, use this function in situations such as when you cannot set the fixed scan time operation.

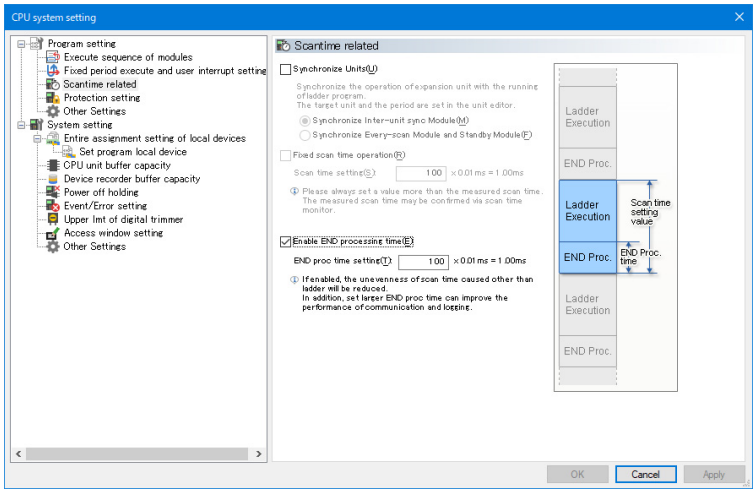
● Setting the END processing time

Example) When the value is set to 2 ms:



■ Configuring the END processing time settings

To configure the END processing time settings, from the workspace, open the "CPU system setting" dialog box, and then click "Program setting" and then "Scantime related."



The range that can be configured is “0.02ms to 200.00ms”.

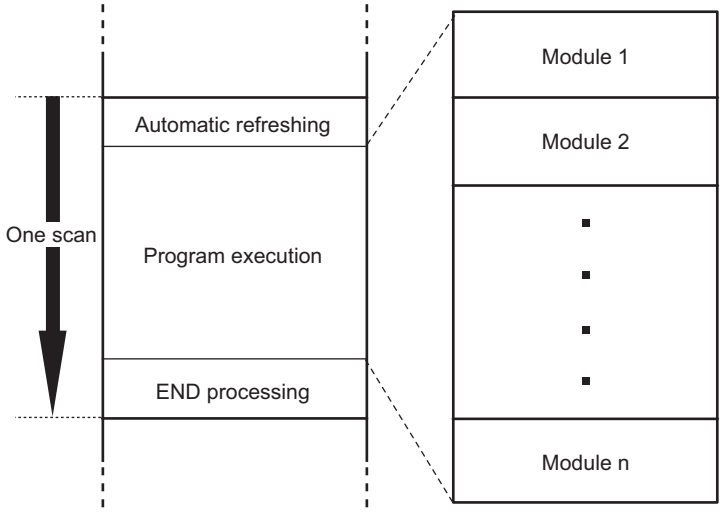
Reference

The actual measured value of END processing time can be confirmed at CM1002 to 1003 (1μs unit).

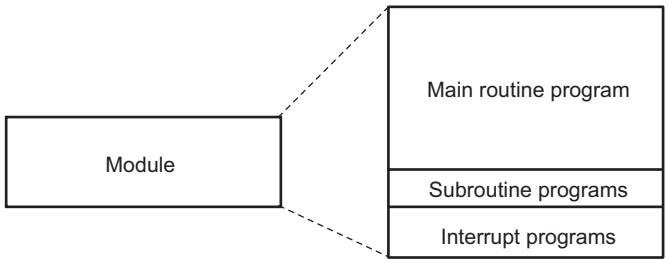
"A-2 Lists of CR and CM Devices" (page A-10)

Program Configuration

In a single scan, program execution consists of the modules being executed in the predetermined order, and within each module, the main routine program being executed step by step from its first step.



In each module, you can create the main routine program, subroutine programs, and interrupt programs.



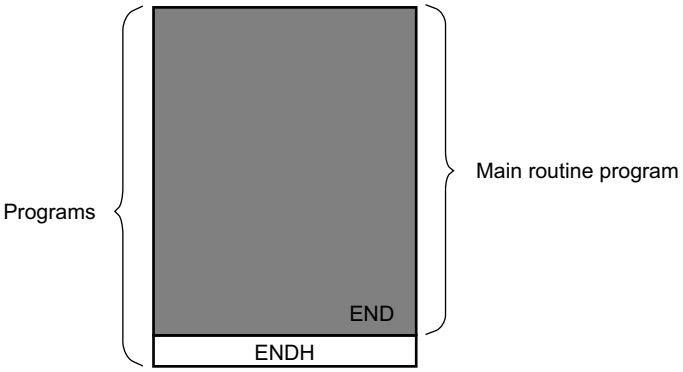
Reference

Function blocks and macros can be called from within the module, but the function blocks and macros themselves are constructed from main routine and subroutine programs. Interrupt programs cannot be created.

"4-10 Function Blocks" (page 4-117)
"4-11 Macros" (page 4-178)

■ Main routine program

The program from the start of the program to the END instruction is called the main routine program. If you will not write any subroutine or interrupt programs, write the ENDH instruction immediately after the END instruction.



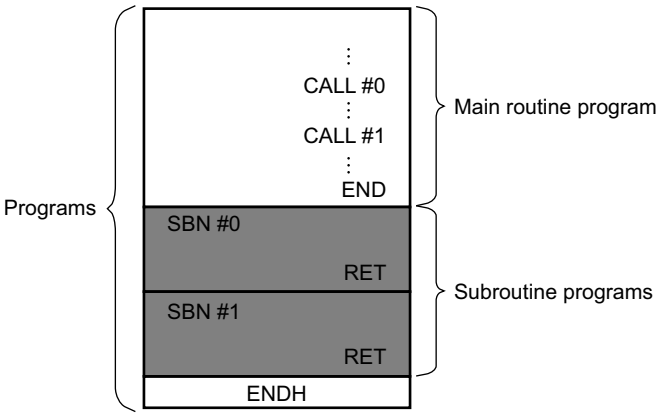
Point **If the END or ENDH instruction is not present, a conversion error will occur.**

Reference The END and ENDH instructions are inserted automatically when you create a new program, so do not delete these instructions.

■ Subroutine programs

Subroutine programs are programs that are executed from the main routine program by using the CALL (call subroutine) or ECALL (call subroutine between modules) instruction. When the execution condition of the CALL instruction turns ON, the subroutine program is executed.
If you want to perform the same processing in multiple locations of the main routine or if you want to process a program only when the set conditions are met, you can execute the program only when necessary by writing it as a subroutine program.
You can write up to 100 subprograms (numbered from 0 to 99) between the END and ENDH instructions.
Each subroutine program starts with the SBN (subroutine entry) instruction and ends with the RET (subroutine return) instruction.
In one program, you cannot give multiple subroutine programs the same number. In the main routine program, you can make multiple CALL instructions to the same subroutine program number.

For details, see the CALL, SBN, and RET instructions in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."



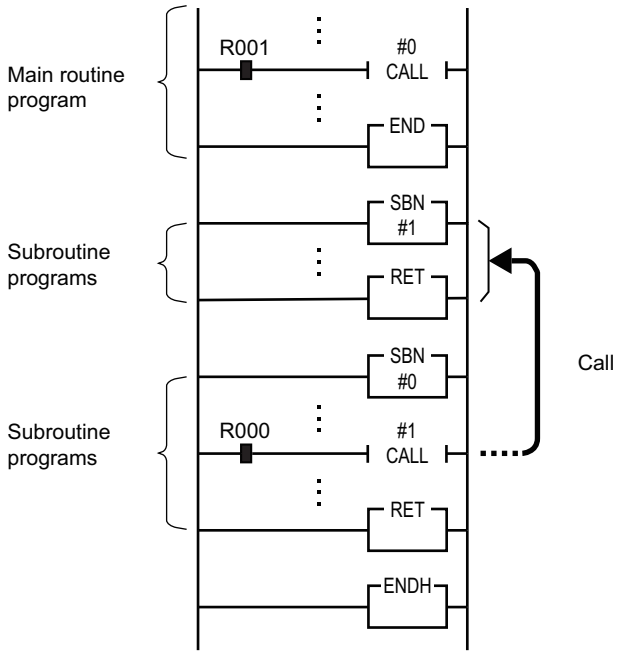
Point

- **If the SBN instruction that corresponds to the CALL instruction does not exist, a conversion error will occur.**
- **If the SBN instruction that corresponds to the ECALL instruction does not exist, a conversion error or a calculation error will occur.**
- **The INT, RETI, STG, JMP, and ENDS instructions cannot be used in subroutine programs.**
- **Be careful when you use differential execution type instructions, timer instructions, function block instructions, and macro instructions in subroutine programs.**

For details, see "Precautions for Differential Execution Type Instruction" in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."
"Precautions when using Timer instructions" (page 4-34)
"Function block instructions" (page 4-141)
"Macro instructions" (page 4-192)

Reference

- From within a subroutine program, you can use the CALL and ECALL instructions to call up to 16 layers of other subroutine programs (nesting).
- Although a subroutine program can call itself (a recursive call), if more than 16 layers of nesting occur during operations, a CALL nesting error (CPU error number 10) will occur.



- To execute a subroutine in another program, use the ECALL instruction.

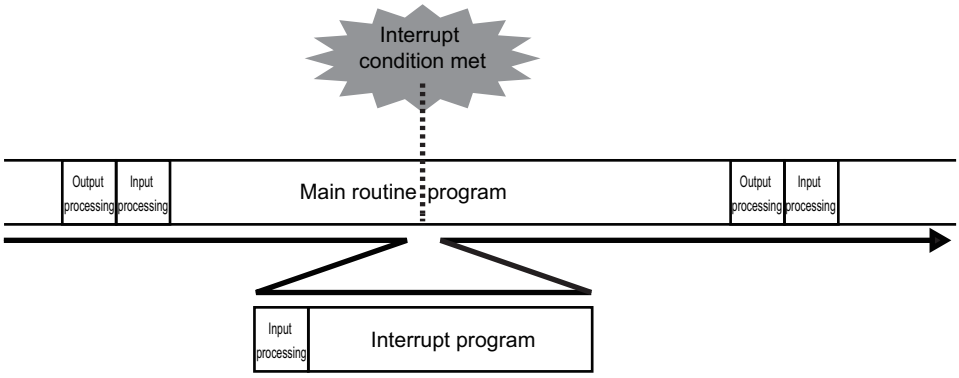
For details, see the ECALL instruction in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

- To call while specifying the target device to process and the value, use function blocks and macros.

"Differences between Macros and Subroutines" (page 4-195)

■ Interrupt programs

Use the interrupt function when you are acquiring short signals, during input response, and when you are performing highly accurate time measurements. When the interrupt conditions are met, the processing of the main routine program is interrupted, and the interrupt program is executed immediately. After interrupt processing is completed, the processing of the main routine program begins again from the point where it was interrupted.
The only interrupt condition is unit interrupt.

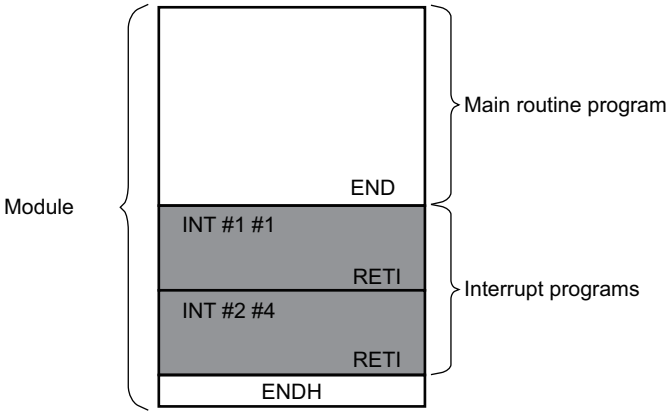


Write interrupt programs between the END and ENDH instructions.

Within one project, you can create one interrupt program per interrupt condition.

You can write interrupt programs in any module.

Start interrupt programs with the INT instruction and end them with the RETI instruction.

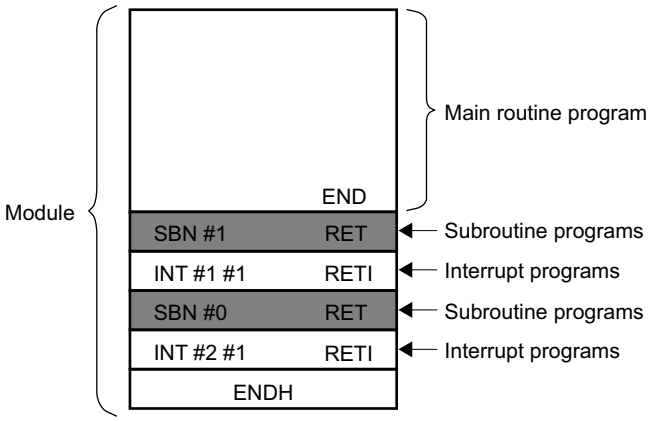


Point

- **Interrupt programs cannot be written in function blocks or macros.**
- **To execute interrupt programs, use the EI instruction to enable the execution of interrupt programs (when operations start, the execution of interrupt programs is disabled).**
- **When an interrupt program starts, the values of the internal registers, the operation flags (CR2009 to CR2012), TM0 to TM3 and Z11 to Z23, are automatically saved. When the interrupt program finishes, these values are restored.***
- **Some instructions cannot be used in interrupt programs.**
  "Instructions That Cannot Be Used in Interrupt Programs" (page 4-210)

Reference

- There are no restrictions on the order that you write subroutines and interrupt programs.

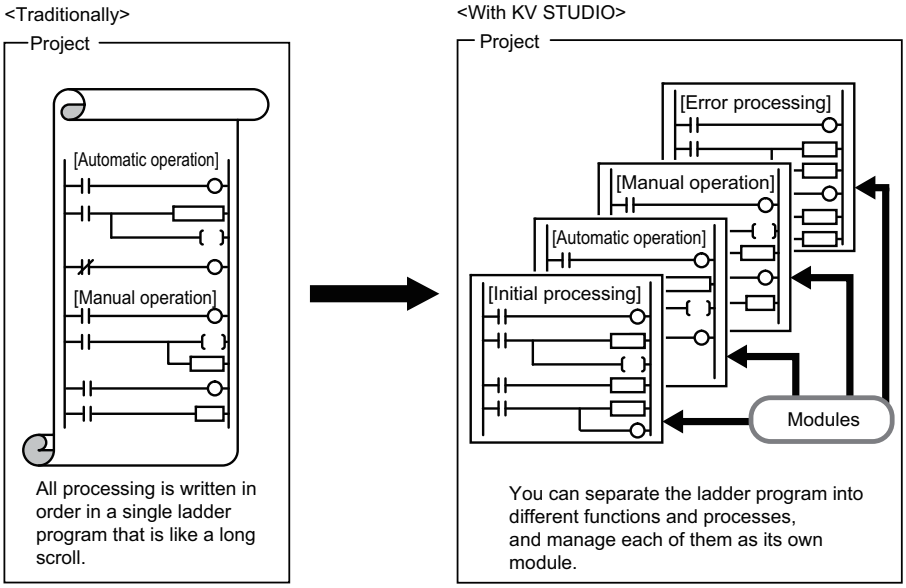


- Interrupt programs are executed even when their modules are not being executed.
- Interrupt programs are executed regardless of the execution status of the module, so you can make it easier to understand the project structure by managing a single stand-by module as the one that contains all the interrupt programs.

"4-13 Interrupts" (page 4-206)

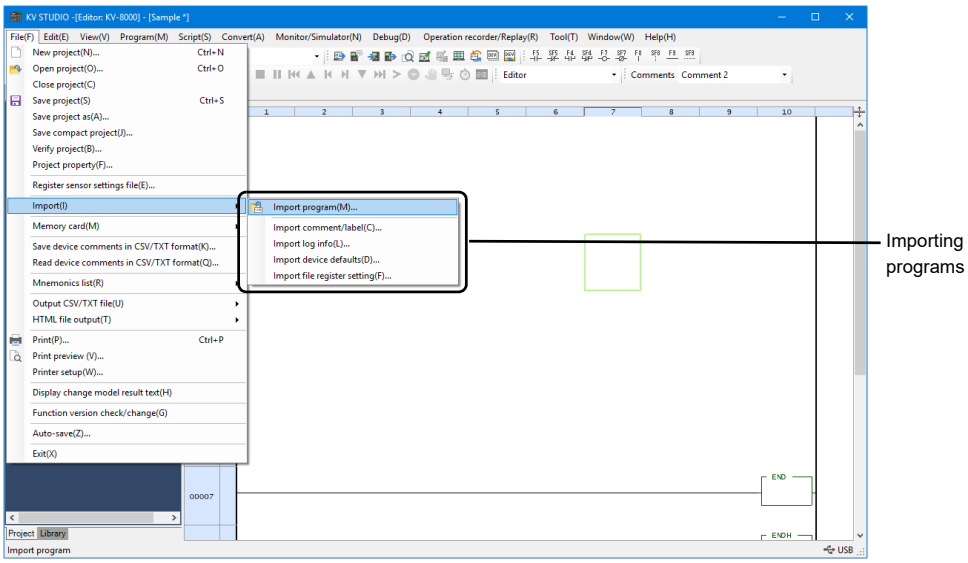
Overview

A normal ladder program that is used to control a device involves multiple processes such as initialization, automatic operation, manual operation, and error handling. Traditionally, these multiple processes have been managed consecutively in a single ladder program. With KV STUDIO, you can create programs within a project for the different aspects such as processes and functions, and then manage these programs as independent components (modules).



■ Module management

In KV STUDIO, modules are managed from the workspace of each project.
You can easily reuse programs by importing programs.



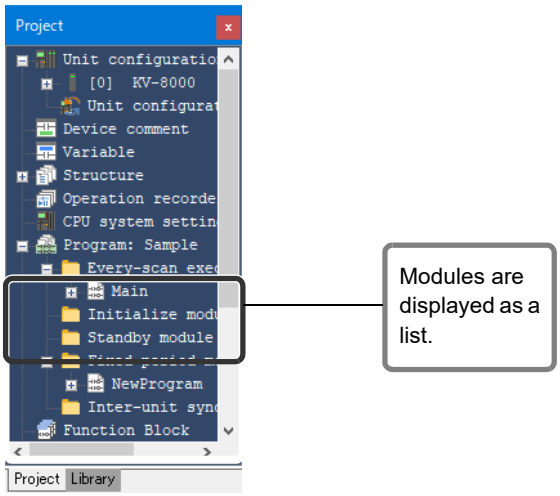
To display the workspace, click the corresponding item on the "View" menu. You can also use the icons on the toolbar to show and hide this element.
You can change the display position of this element.

Reference You can also reuse modules by opening multiple instances of KV STUDIO and copying the modules between applications.

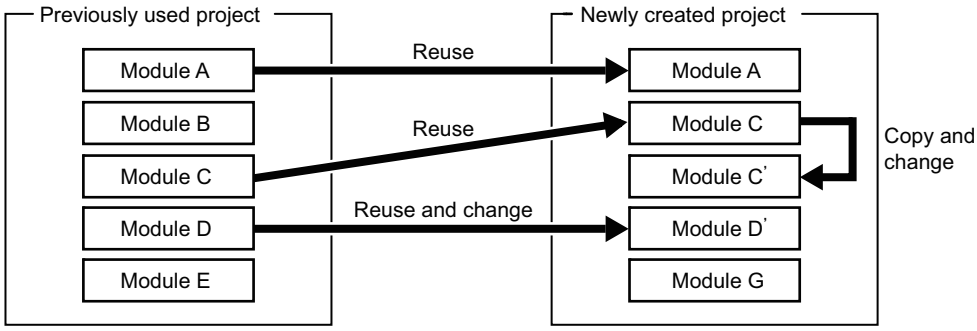
■ Module features

● It's easy to understand the structure of an entire program

When you use the workspace of KV STUDIO, each module is displayed as a tree, so even individuals who did not create the program can understand its entire structure at a glance.



● It's easy to reuse just the necessary parts of a program



● Program creation, editing, and debugging can be done efficiently

By splitting modules up according to functions and processes, you can reduce the scope that must be considered during program creation and debugging. Additionally, by using local devices, you do not have to assign devices to each module, which makes it easier to have multiple people create programs.

"4-12 Local Devices" (page 4-200)

● Password protection is available

To prevent the contents of modules from being changed easily, you can set a password.
When you set a password, the module program cannot be displayed or edited without first entering the password.
Use the "Program properties" dialog box to set the password.

"Module Passwords" (page 4-114)
"Setting a Function Block Password" (page 4-141)
"Macro Passwords" (page 4-191)

Module Type

There are "every-scan execution modules", "initialization modules", "standby modules", "fixed period modules" and "inter-unit synchronization modules" as modules.
Use the KV STUDIO workspace to manage these different types of modules that you create. You can also change the module type after you create modules.

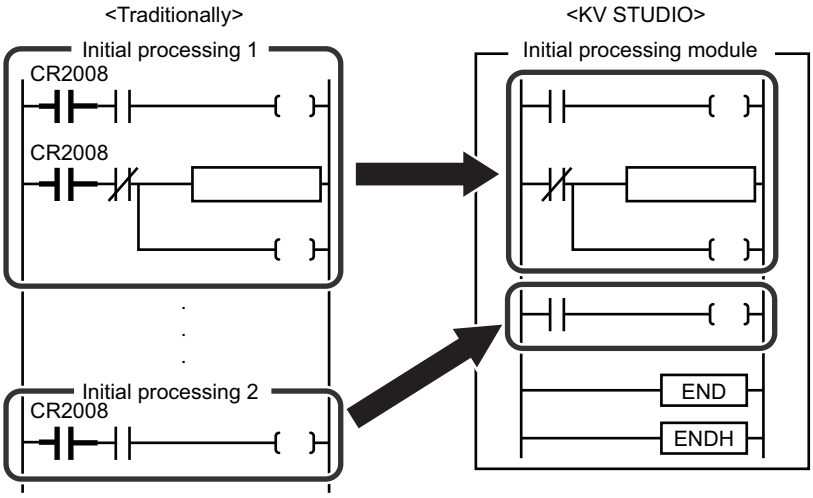
Every-scan execution type

These modules are executed from the first scan when operations start (when power is turned on and when the mode is switched from PRG to RUN).
You can use the module instructions, module stop (MDSTOP) and module start (MDSTRT), to control the execution status of the module during operation.
Operations when you use the module instructions are the same as the standby module operations.
"Standby module operation" (page 4-101)

Reference
- If you use a module instruction but do not control the execution status, the module will be executed during each scan.
- When you create a new project in KV STUDIO, one every-scan execution type module with the same name as the project will be created automatically.

Initialize module

These modules are executed only during the first scan when operations start. Use these modules to perform initializations such as the initial processing of devices and restoring the default settings of KV-8000 Series functions. Even if there are no initialize modules in the project, the KV-8000 Series can operate. Create these modules as necessary.



CR2008: This is turned ON for only one scan when operation starts.

By registering modules as initialize modules, you can reduce the scan time by the time that is required to process the initialize module programs because these modules will not be executed from the second scan.

Point
- **Some instructions cannot be used in initialize modules.**
  "List of Unusable Instructions" (page A-21)
- **You cannot use the module instructions (MDSTRT and MDSTOP) to control the execution status of initialize modules.**
- **You cannot operate the KV-8000 Series with only initialize modules. Be sure to use these modules together with at least one every-scan execution type module or standby module.**

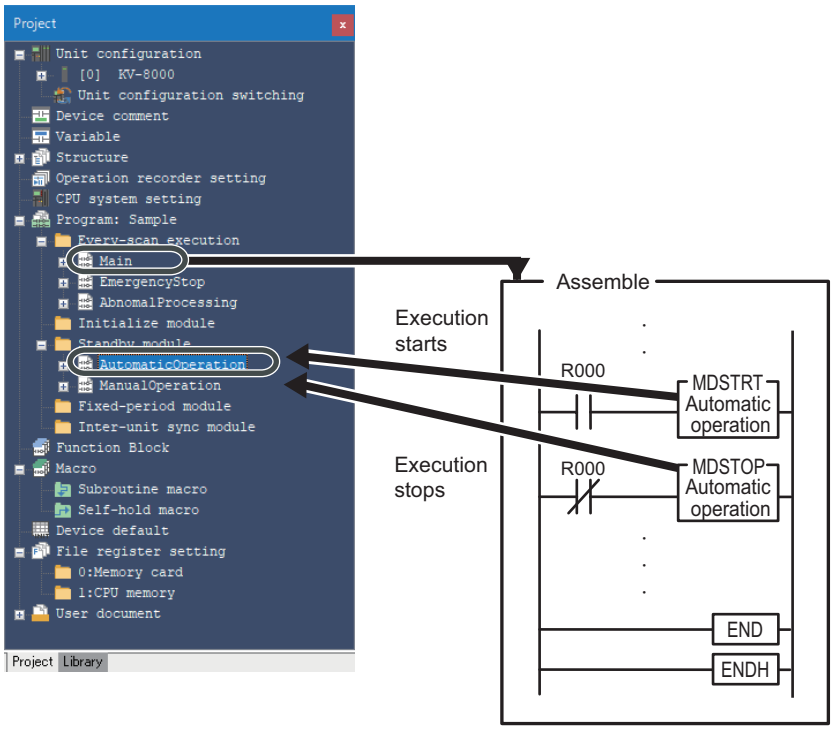
Standby module

These modules are stopped when operations start. You can use the module instructions (MDSTRT and MDSTOP) to control the execution status of these modules during operation.

"Module instructions" (page 4-115)

Example of calling standby modules

In this example, the execution status of the standby module "AutomaticOperation" is controlled from the every-scan execution type module "Main."



Point

You cannot operate the KV-8000 Series with only standby modules. Be sure to use these modules together with at least one every-scan execution type module or initialize module.

Reference

You can control the execution status of every-scan execution type modules in the same way.

■ Standby module operation

● Starting the execution of standby modules

In the scan in which the module execution start instruction is called, standby modules are not executed. They will be executed from the next scan.

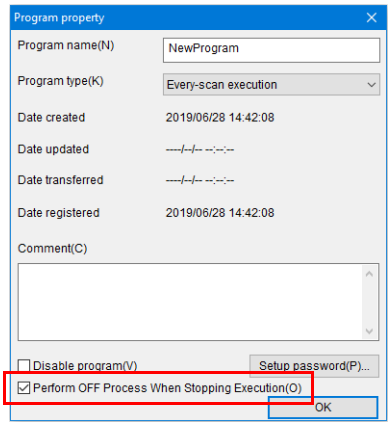
Reference

The initial status of the differential execution instruction when the execution of a standby module starts is ON when a rising edge is detected and OFF when a falling edge is detected. For example, if this is turned ON from the first scan when execution starts, rising edge differentials will not be detected.

● Stopping the execution of standby modules

The operation when standby modules stop executing (when an MDSTP instruction is executed) differs according to the "Program property" -> "Perform OFF Process When Stopping Execution" setting of each program.

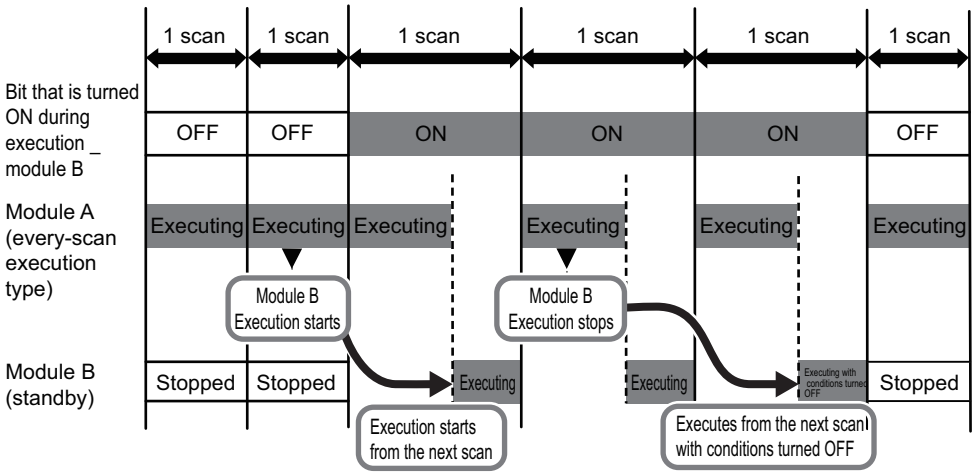
Point **In the case of KV-8000A, the setting is not displayed, and the operation is the same as when "OFF processing when execution stopped" item is always checked.**



If "Perform OFF Process When Stopping Execution" is not checked
The standby module stops in the next scan after the stop module execution instruction (MDSTP) is executed.

If "Perform OFF Process When Stopping Execution" is checked
When the stop instruction is called during program execution, these modules are executed in the scan that the stop instruction was called.
In the subsequent scan, these modules are executed when all execution conditions are turned OFF.*
Operations are stopped in the following scan (the second scan after the stop instruction was called).

The operation of standby modules is shown in the following figure.



* The following instructions are exceptions.

Instruction	Status
OUB	Turned OFF.
OFDL	Reset.
SHOT	
C	The status is held.
CJ, NCJ, and GOTO	Execution does not jump.

Reference The ladder program within the module operates the same as when the execution condition of the MC instruction is turned OFF for just one scan before operations are stopped.
For details, see the MC and MCR instructions in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

Fixed-Period Module

Routines in Fixed-Period Module can be separated from ordinary scanning and executed according to pre-defined period. Each project can use 4 fixed period modules.

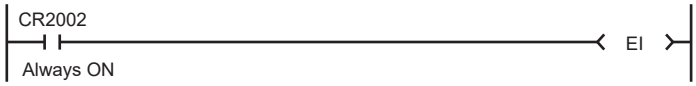
■ Fixed-Period Module Execution

To execute Fixed-Period Module, in addition to writing the Fixed-Period Module, the following also needs to be done.

● Interrupt enable (EI) instruction

Like the interrupt routine, Enable/Inhibited of the fixed period execution is done through the EI/DI instruction. The EI instruction is written in each scanned module and executed.

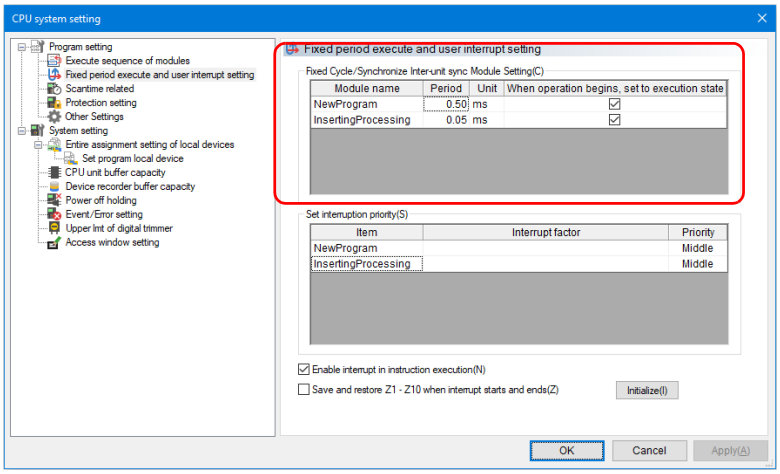
Example



"EI/DI Instruction" (page 4-218)

● CPU System settings

Open the "CPU system settings" window from the KV STUDIO workspace, and set up the execution period and priority in the "Fixed-period execution and user interrupt" settings option of the "Program settings"



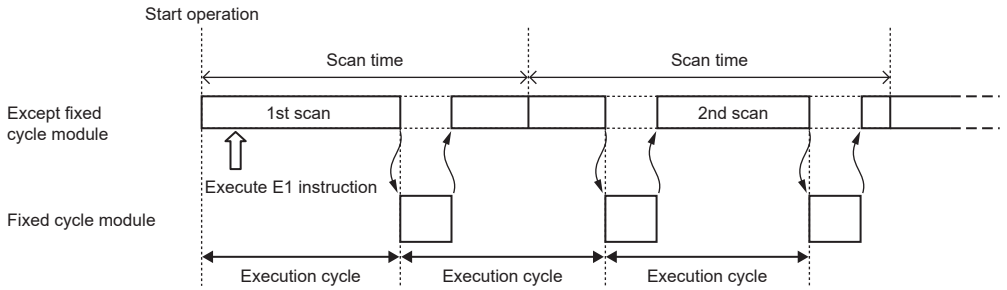
"Fixed-Period Module settings "
The period range is 0.01 s to 60.00 s (in 0.01 s increments) or 0.05 ms to 6000.00 ms (in 0.01 ms increments).
If you want to execute Fixed-Period Module right from the program is started, select the "Execute right from the start" check box. If you want to start/stop it during operation, use the module instruction (MDSTRT/MDSTOP).
"interrupt priority settings"
When the priority of a fixed period module is set to "medium," only interrupt programs and fixed period modules with a "High" priority setting can perform multiple interrupts.
Interrupt programs set to "medium" or "low" cannot interrupt the execution of fixed period module programs.
Interrupt programs set to "medium" priority can interrupt the processing of interrupt programs set to "low priority. However, during the processing of interrupt programs with "high" or "medium" priority and fixed period modules, they are in standby mode.

Point **The execution cycle of fixed period modules can be set only from KV STUDIO. It can't be changed from the ladder program.**

■ Fixed-Period Module Operation

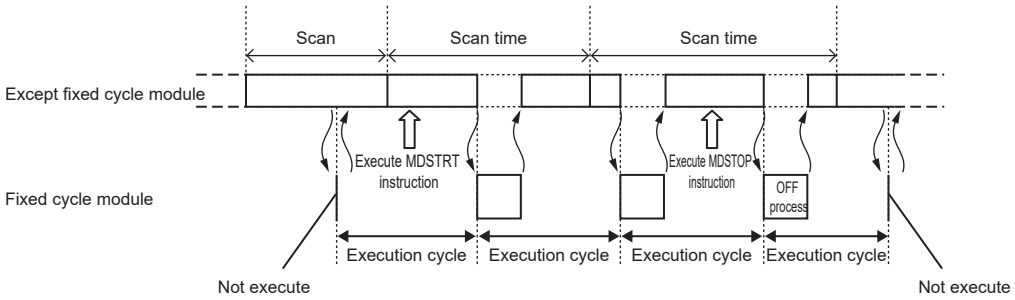
● When the "When operation begins, set to execution state" check box is selected (default)

If EI instruction is executed, begin to execute Fixed-Period Module.



● When the "When operation begins, set to execution state" check box is not selected

In EI instruction execution state, the module instruction is used to control execution state of Fixed Period Module.



For OFF processing, please see "Standby module operation" (page 4-101).

Point

Execution period is irrelevant with execution state of Fixed-Period Module, counting is started when starting to run. In case of executing EI command while driving in fixed period module mode, it might performed immediately after EI command execution without waiting for the cycle of fixed period execution.

Reference

- When all of the following conditions are met, the period fluctuation is within 15 μs.
  - When selecting the "Enable interrupt during instruction execution" checkbox in "Fixed period execution and user interrupt" setting of "CPU system setting"
  - User interrupts with a priority higher than Fixed Period Module are not used
  - High priority interrupt conditions have not been executed.
  - Low priority interrupt conditions have not been executed beforehand (4 μs maximum when they have been executed)
  - The KV-5000/3000 Series expansion units are not used (14 μs maximum when they have been used)
  - DI instructions/DIC instructions are not being used to inhibit interrupts.
  - 1 ID is not registered with a device of 513 words or more during the recording/tracking setting.
- When fixed-period execution timing occurs in interrupt inhibited state (when executing DI instruction) within the same execution period, Fixed-Period Module is executed in the interrupt enable state (executing EI instruction), .

Fixed-Period Module is separated with the ordinary scanning in interrupt processing. For details, please refer to "Interrupts".

■ Device processing in the start/end

Device processing at start/end is the same as ordinary interrupt processing, at start/end of Fixed-Period Module execution, the following projects are saved/reset automatically.

- Internal register
- Calculation flag (CR2009 to CR2012)
- Index register(Z11 to Z23)
- TM0 to TM3

Relay To programmed in just executed STG instruction operand

Reference

- Index register Z01 to Z10 can be set to execute save/reset processing in "CPU system setting" ➔ "fixed period execution and user interrupt" settings.

Inter-unit Synchronous Module

The program within the inter-unit synchronous module is run in a preset cycle, separate from regular scans. One Inter-unit Synchronous Module project can be used per project.

■ Inter-unit Synchronous Module Operation Method

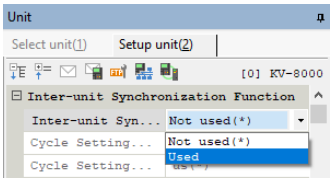
The requirements for running the inter-unit synchronous module other than creating the inter-unit synchronous module are as follows.

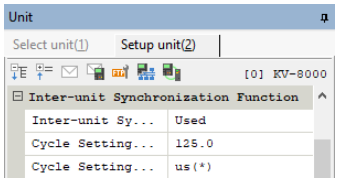
● Settings for inter-unit synchronization

The contents below must be configured using the unit editor.

Settings for the CPU unit

The inter-unit synchronization of the unit setting for the CPU unit should be changed to “use”.
After that, the parameter of the cycle must be changed.





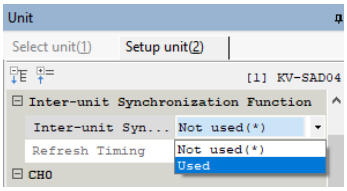
Point

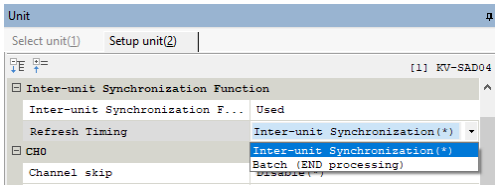
The value that can be configured for the cycle will differ depending on the expansion unit performing the inter-unit synchronization. Please check the manual for each expansion unit.

Settings for the expansion unit

The inter-unit synchronization function of the expansion unit to be run should be set to use.
The setting for refresh timing can be changed as necessary.

"Inter-Unit Synchronization Refresh Devices" (page 5-169)





● Settings for inter-unit synchronization authorization

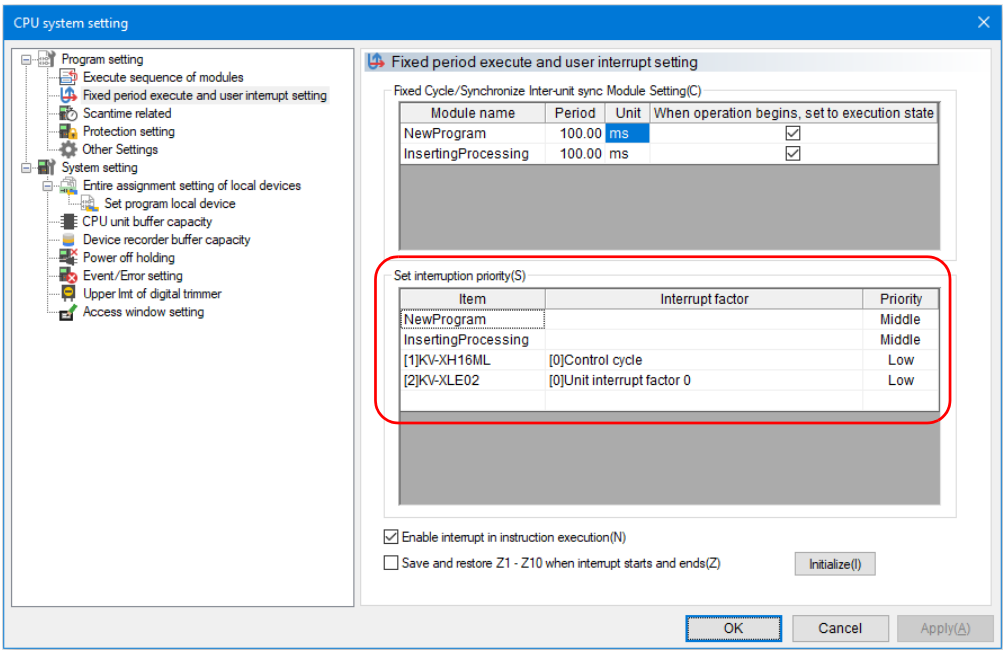
Authorization/prohibition of inter-unit synchronization will be conducted by EI/DI command, in the same way as interruption programs. It should be performed after interrupting every scan module or standby module and stating an authorization (EI) command.

Example)

"EI/DI Instruction" (page 4-218)

● Settings for CPU system

The priority should be configured from [Fixed cycle operation and user interruption setting] in [Program settings] after opening “CPU system settings” from the 《KV STUDIO》 workspace.



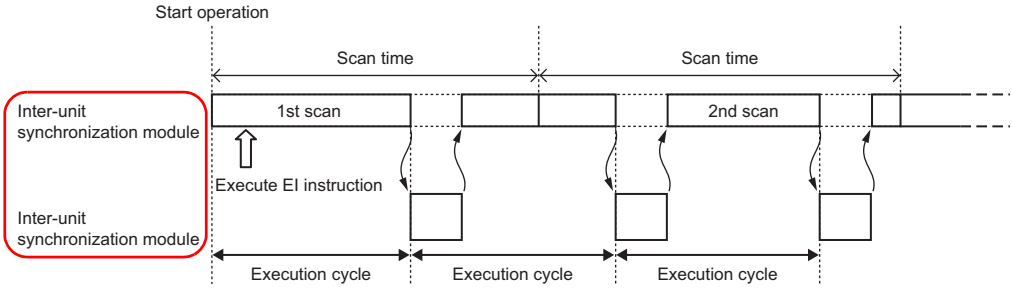
[Settings for interruption priority]

The inter-unit synchronous module with the priority set to “high” cannot interrupt other interruption programs or fixed cycle modules. Operation will have to wait while running the inter-unit synchronous module.

■ Inter-unit Synchronous Module Performance

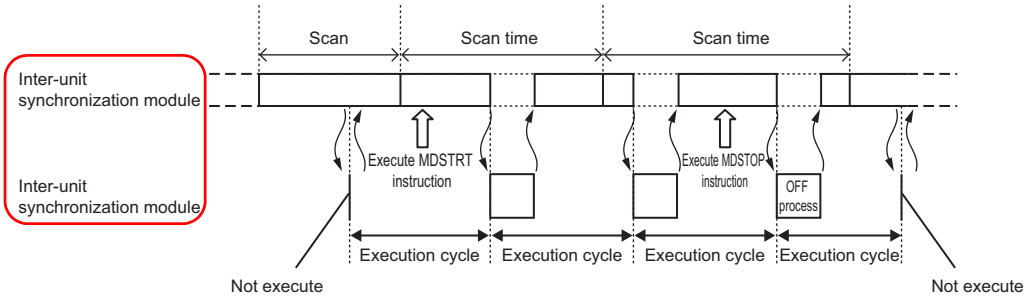
● If "Be in a performing state when starting operation" is checked (default)

If an EI command is performed, operation of the inter-unit synchronous module will begin.



● If "Be in a performing state when starting operation" is not checked

If an EI command is in a state of performance, the operational state of the inter-unit synchronous module should be regulated.



Please see "Stopping the execution of standby modules" (page 4-102) for information about OFF processing.

Point

The performance cycle will be counted from when operation begins, regardless of the performance state or the inter-unit synchronous module.In case of executing EI command while driving in inter-unit synchronization module mode, it might performed immediately after EI command execution without waiting for the cycle of inter-unit synchronization execution.

Reference

- If all the conditions below are to be met, the variation between cycles will be within 15μs.
  - If “Authorize interruption while performing a command” is being checked with the〔Fixed cycle performance and user interruption settings〕of the〔CPU settings〕
  - A user interruption with a higher priority than the inter-unit synchronous module is not being used
  - A factor with a high interruption priority is not being performed
  - A factor with a low interruption priority is not being performed immediately before (4μs at most will be added when being performed)
  - An expansion unit for the KV-5000/3000 series is not being used (14μs at most will be added when being performed)
  - A command from DI command/DIC command is not being prohibited
- If the inter-unit synchronizing execution timing has come while it is in interruption disable mode (DI command execution), the inter-unit synchronization module execution will be started immediately after turns to interruption enabled mode (EI command execution).

The inter-unit synchronous module should be run as an interruption process, separate from regular scans. See "4-13 Interrupts" for further details on the interruption process.

■ Regarding processing of the device when beginning and finishing

Saving/restoring of the items below will be conducted automatically when beginning/finishing the inter-unit synchronous module performance, in the same way as a regular interruption process.

- Internal register
- Operation flag (CR2009-CR2012)
- Index register (Z11-Z23)
- TM0-TM3
- Relay stated in the operand of the STG command performed directly before

Reference

The index register Z01-Z10 can be configured to conduct a saving/restoring process with "Fixed cycle performance and user interruption settings" ➔ "CPU system settings".

Module Execution Sequence

You can set the execution sequence within a scan of the modules registered in a project.
Normally, this is the order in which the modules are created. Only when the execution sequence within a scan of the ladder programs has an effect on operations should you change the sequence.
From the workspace, open the "CPU system setting" dialog box, and then click "Program setting" and then "Execute sequence of modules."

Order	Module Name	Module Type
1	PosInit	Initialize
2	HSPInit	Initialize
3	Pos	Every-scan execution type
4	LogInfoUpdate	Standby
5	HSP	Every-scan execution type
6	CommProc	Standby
7	MainOpeProc	Every-scan execution type

Range within which the order can be changed

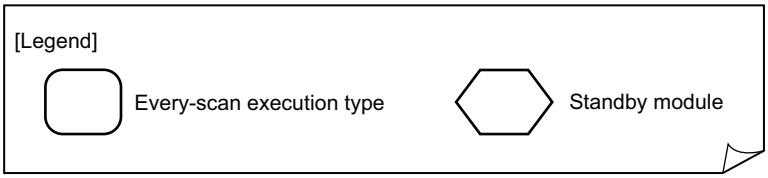
Range within which the order can be changed

The following rules apply to setting the execution sequence of modules within a scan.

- You can change the order of initialize modules, but you cannot switch the order of initialize modules with those of other types of modules.
- You can change the order of every-scan execution type and standby modules, but you cannot move these modules ahead of the initialize modules in the sequence.

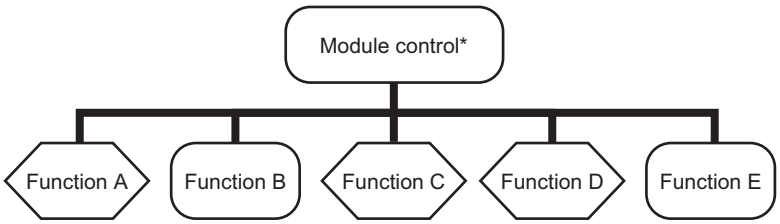
Module Configuration Examples

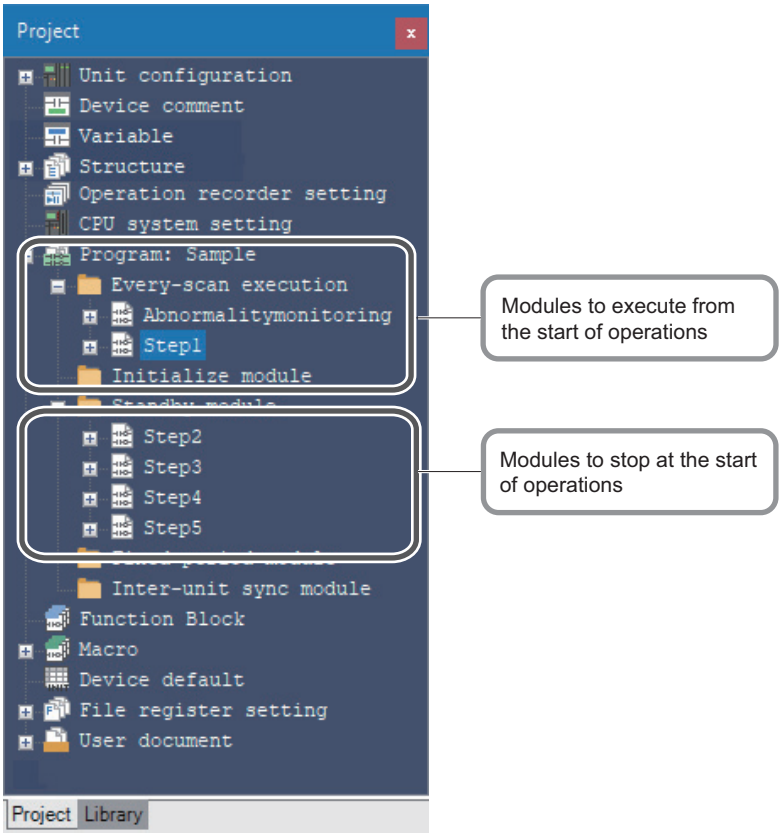
This section shows module configuration examples.
To simplify the explanations, the initialize modules are omitted.



● Configuration by functions

Modules are created according to process details, connected devices, and expansion units.
Use the every-scan execution type and standby modules differently depending on the execution conditions when operations start.
Stopping the unnecessary modules makes debugging easier and reduces scan time.

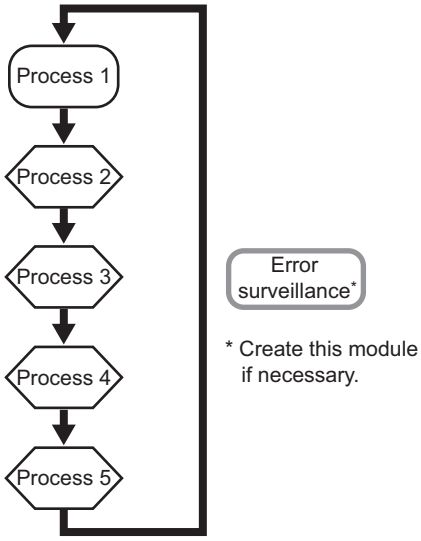


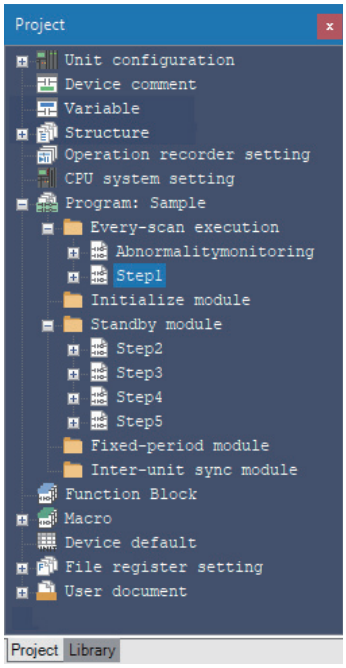


* The execution status of each function is controlled by the "Project" module. Create the "Project" module if necessary.

● Configuration by processes

Modules are created according to processes and operations.
In this example, the modules to be executed when operations start are created as every-scan execution type modules and all other modules are created as standby modules.



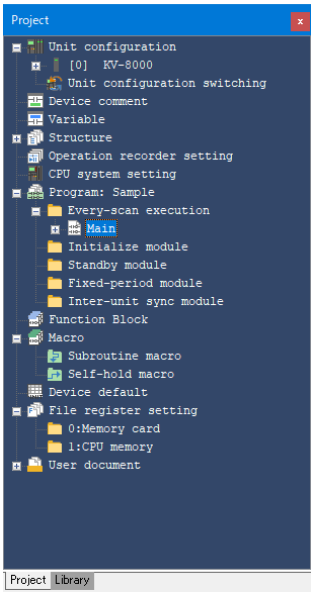


Reference

Because the module execution sequence is the execution sequence within a scan, there is also no need to consider the module execution sequence when creating programs by processes.

● Programming without considering modules

When you create a new project, one every-scan execution type module with the name "Main" is created automatically, so you can create programs without considering modules.



Module System Devices

A module system device is a device that is used to capture the operation status of a module.
Only the bit that is turned ON during execution can be referenced by other programs.

Point **You cannot attach device comments to module system devices.**

Reference KV STUDIO automatically assigns module system devices to the CPU unit work area.

● Bit that is turned ON for only one scan when module execution starts, "@CR2008"

This bit is turned ON for only the first scan after module execution starts.
From the second scan, it is turned OFF.
This bit can only be referenced from its own module. It cannot be referenced from the macros executed in its own module.
In KV STUDIO, this device is represented by "@CR2008."

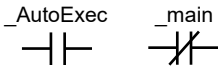
● Bit that is turned OFF for only one scan when module execution starts, "@CR2007"

This bit is turned OFF for only the first scan after module execution starts.
From the second scan, it is turned ON.
This bit can only be referenced from its own module. It cannot be referenced from the macros executed in its own module.
In KV STUDIO, this device is represented by "@CR2007."

● Bit that is turned ON during execution, "_(module name)"

This bit is turned ON when the module is being executed.
Because this bit can be referenced not only from its own module but from other programs as well, it is useful in situations such as when you want to control the execution of standby modules.
In KV STUDIO, modules are indicated by "_(module name)" (underscore + module name).

Example



Module Passwords

■ Module passwords

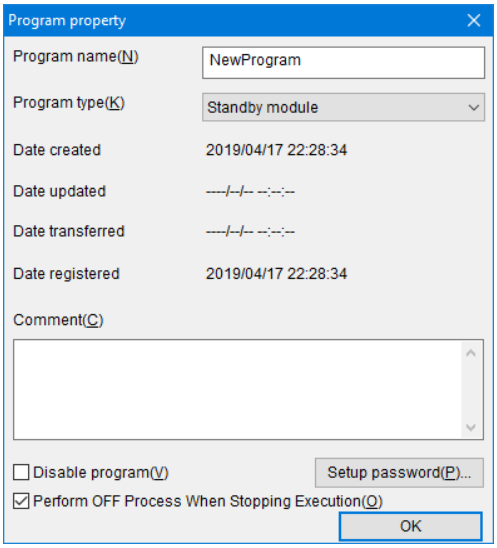
You can limit who can display modules in KV STUDIO by setting passwords on the modules.
Follow the procedure below to set a password.

1 Select the module to set a password for in the workspace.

2 Right-click, and then select "Property" in the menu that is displayed.

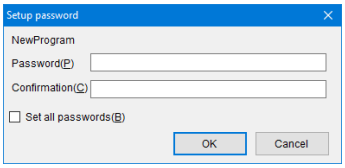
The "Program property" dialog box appears.

Alternative procedure
- Select the module in the workspace, and select "Program" -> "Properties" from the menu.
- Select and right click the module in the workspace and select "Setup password" from the menu.



3 Click "Set password."

The "Set password" dialog box is displayed.

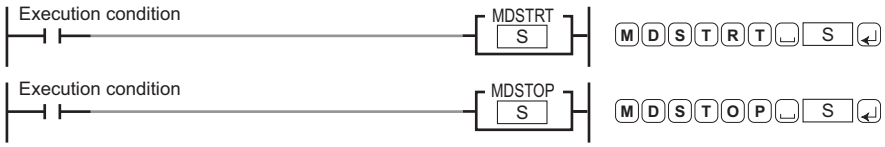


4 Enter the password in both text boxes (the second is for confirmation), and then click "OK."

After you set the password, when an attempt is made to display the module in KV STUDIO, a dialog box will be displayed asking the user to enter the password.
If the user does not enter the password, they will not be able to display the module in KV STUDIO.

MDSTRT	MDSTRT	Module execution start	Starts the execution of the specified module
MDSTOP	MDSTOP	Module execution stop	Stops the execution of the specified module

Ladder program



Entry mode

Operation	Explanation	Occupied size
S	Specifies the module name.*1	—

*1 Use of " ", # , $ is not allowed. Directly enter the name of the module set on KV STUDIO.
The number of characters that can be entered is as follows.

- KV-8000(A)、KV-7000 (CPU function version 2.0 or later)
  Up to 64 characters regardless of half-width or full-width
- KV-7000 (CPU function version lower than 2.0)
  Up to 24 characters regardless of half-width or full-width
- KV-5500/5000/3000/1000、KV Nano Series
  Up to 24 half-width alphanumerics, or up to 12 full-width alphanumerics

Operation Description

MDSTRT When the execution condition is ON, the module specified by S is started.
The specified module starts operating from the next scan.
Nothing happens on modules that are already started up.

MDSTOP When the execution condition is ON, the module specified by S is started.
Nothing happens on modules that are already stopped.
This instruction can also stop the self-module.

The OFF process when stopped differs according to the CPU unit. Refer to the instruction manual foreach CPU unit for details.

Point

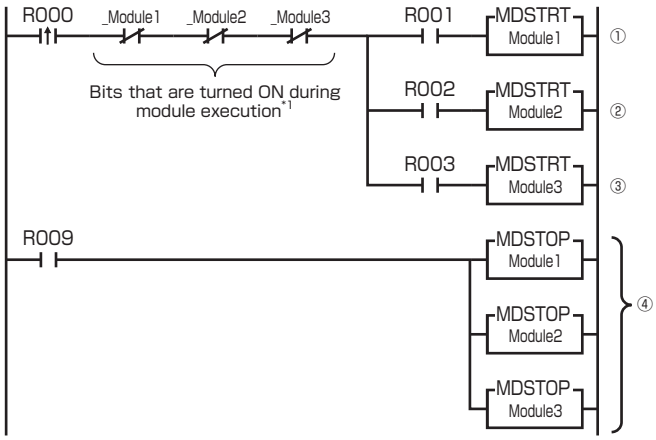
- **The initial state of differential execution type instructions at start of module execution is ON when the rising edge is detected, and OFF when the falling edge is detected. Differentiation is never detected immediately when execution is started.**
- **When the MDSTRT and MDSTOP instructions have been executed multiple times in the samescan on the same module, execution follows the module instruction that was last executed.**
- **Do not stop all modules. All programs will not be executed when all modules have stopped. So, restore this state by one of the following methods. (Execution cannot be restored unless program execution is temporarily stopped.)**
  - **◆Turn the CPU unit OFF then back ON again.**
  - **◆Switch the RUN-PROG selector switch on the CPU unit to PROG then backto RUN again.**
  - **◆Turn the RUN mode of the program OFF then back ON again in KV STUDIO.**
- **Execution of the initialization module cannot be started or stopped.**

Operation flags

CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	No change in status

Sample Program

When the rising edge of input relay R000 is detected, the module currently selected by mode switching (R001 to R003) is executed, and execution of all modules is stopped when input relay R009 turns ON. During execution of one of the modules, execution of other modules cannot be started.



(Mnemonics list)

```
LDP R000
ANB _Module1
ANB _Module2
ANB _Module3
MPS
AND R001
MDSTRT Module1
MRD
AND R002
MDSTRT Module2
MPP
AND R003
MDSTRT Module3
LD R009
MDSTOP Module1
MDSTOP Module2
MDSTOP Module3
```

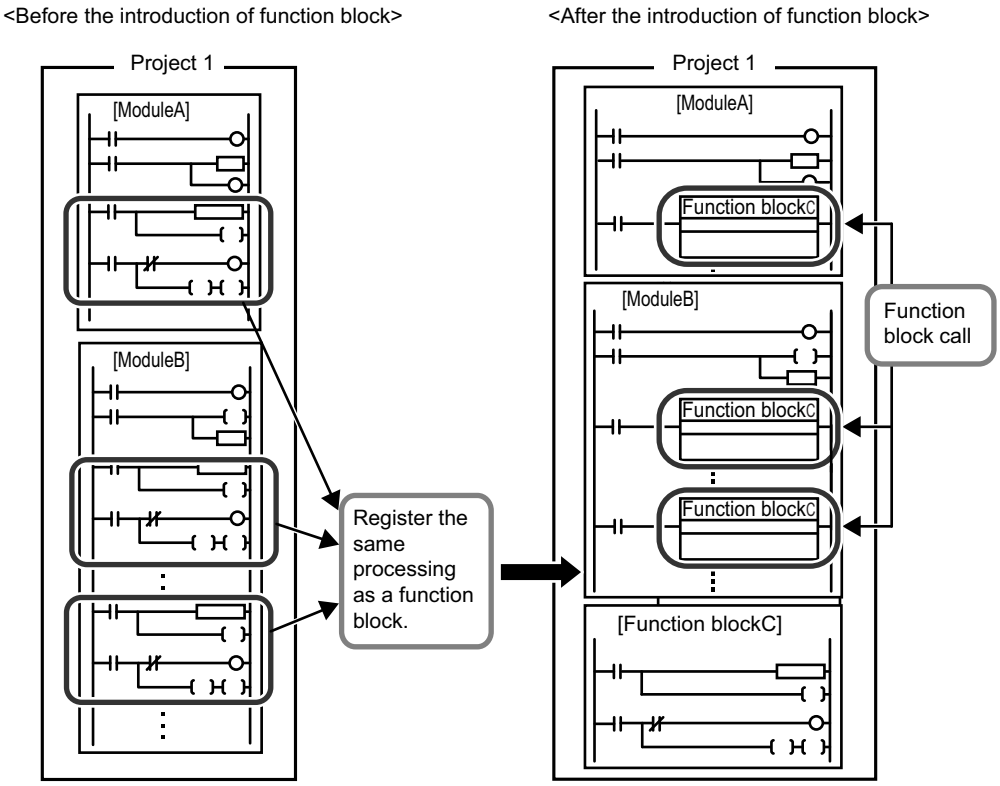
① "Module 1" is executed when R001 is ON.
(2) "Module 2" is executed when R002 is ON.
(3) "Module 3" is executed when R003 is ON.
(4) Module 1 to 3 is stopped when R009 is ON.

For details on the module execution ON bit, see "Modules" in the "Module System Devices" (page 4-113).

4-10 Function Blocks

Overview

Function blocks are circuit blocks used repeatedly in a project that are registered as one component under a name of your choosing. By creating function blocks, you can use the sections of ladder programs in the same way that ladder instructions are used, which enables you to reduce the amount of time you spend writing programs. When you make changes to the program, just by modifying the data in the location that is registered as a function block, all the process details executed when the corresponding function block is called in the project are changed. This reduces the amount of time that you spend changing programs and also prevents mistakes that occur during program modifications.



■ Function block management

In KV STUDIO, the workspace is used to manage macros for each project.

■ Function block features

● Function blocks can be used as original instructions

If you register a ladder program segment that occurs multiple times in the program as a function block, you can reduce the amount of time you spend writing programs and you can also reduce input mistakes.
As a result, the number of rungs in the ladder program decreases, which makes it easier to understand the program structure.
You can assign any name to the function block that you want to register.

● You can use arguments

You can use arguments to create highly general function blocks because you can specify the operands of instructions in the function block (devices and values) at the time that the function block is called. Furthermore, arguments can be set to any name, so the contents of the argument are easy to see and the readability of the program can be improved.

"Setting Arguments" (page 4-131)

● Password protection is available

To prevent the contents of function blocks from being changed easily, you can set passwords.
When you set a password, people who do not know the password cannot display or edit the function block program.
Use the "Program properties" dialog box to set the password.

● You can create two types of function blocks

There are two types of function block: "function blocks" and "functions".
The differences between "function blocks" and "functions" are shown below.

	Function block	Function
Execution condition "Function Block Type" (page 4-120)	So when the execution condition turns ON, the operation will be executed. The operation stop condition varies depending on the instruction used to start the function block. "Function Block Type" (page 4-120)	Executed only while the execution condition is ON. Operation ends when the execution condition turns OFF.
Program objects	Ladder program entities are created by the number of instances, increasing the program capacity. "Function block instance" (page 4-121)	The same area is used, so the program volume does not increase.
When using local variables/ devices "4-12 Local Devices" (page 4-200)	Different areas are occupied by the number of instances. so there is no need to care about the usage status of the device during use. "Function Block Type" (page 4-120)	A shared area is used. When it is executed multiple times in the same scan, initialize it each time in the function.
Nest structure "Function Block Nests" (page 4-140)	Compatible up to 8 layers.	Compatible up to 8 layers. (However, function blocks cannot be called from functions)

Reference [Reference] For the differences between function block and macro, see "Differences between Function Blocks and Macros" (page 4-175).

Usage Procedure

The macro usage procedure is shown below.

(1) Create a new function block.

Click "Program" and then "New."

Select "Function block"/"Function" as the program type.

"Function Block Type" (page 4-120)

(2) Set the arguments.

Set arguments for the function block.

"Setting Arguments" (page 4-131)

(3) Create the ladder program.

Use the set arguments to write a ladder program in the function block.

(4) Place the function block.

Place the function block.

"Function block instructions" (page 4-141)

Function Block Type

There are two types of function blocks: “function blocks” and “functions”.

Function blocks

● Overview

The function block executes the program in the function block when the execution condition is ON.
There are three instructions for executing function blocks, which have the following characteristics.

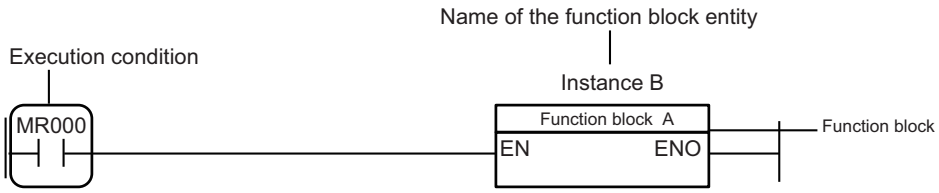
Item	FBCALL Function block call	FBSTRT Function block start	FB Function block call
Start method	Execution condition of the function block ON		
Stop method	• Stop the program (module, function block, etc.) that executes the function block • The execution condition of the MC instruction is turned OFF when executed between the MC and MCR instructions		
	• FBCALL instruction is turned OFF	• The FEND instruction is executed in the function block	
Specify by instance name	○	○	–

Point

- **When the function block is started from the program written between END and ENDH of the module, it does not stop even if the module is stopped.**
- **If the execution condition of the MC instruction is turned OFF when executed between the MC and MCR instructions, the execution result will not be reflected in the argument of argument type OUT.**
- **The FBCALL/FBSTRT instruction can be used only with KV-8000 with CPU function version 2.0 or later. When programming with KV-8000 with CPU function version 2.0 or later, it is recommended to execute the function block using the FBCALL/FBSTRT instruction.**
- **If the instance name is specified in ST language and executed, it can be stopped by turning off the execution condition.**

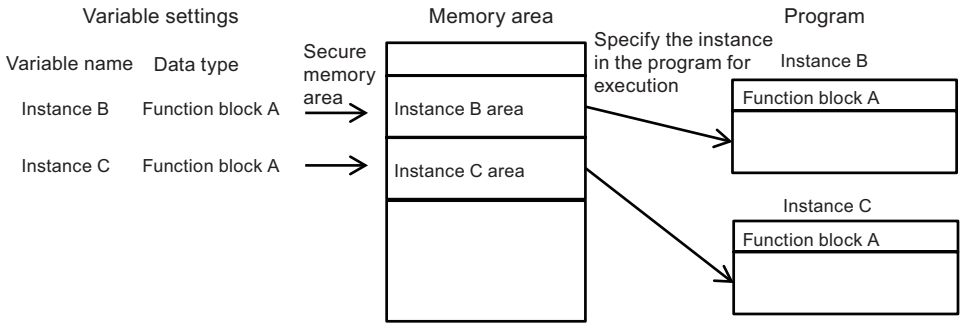
● How to use

Configured in a ladder program for use.



● Function block instance

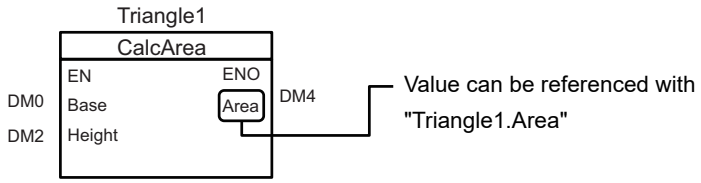
When a function block is configured, the function block type variable is registered. When a function block type variable is registered, an instance for entity execution will be generated. When variables are registered with different names for the same function block, different instances will be generated accordingly and different memory areas will be occupied. When a function block is executed by specifying an instance, the FBCALL/FBSTRT instruction should be used.



Point

- **Instance can be specified only when executing a function block with the FBCALL/FBSTRT instruction. No instance can be specified in case of the FB instruction, and an entity is always generated by the number of calls.**
- **The same instance can be specified for multiple times in the same program.**
- **When using the FBCALL/FBSTRT instruction, the KV-8000 CPU function version should be upgraded to 2.0 or below using KV STUDIO Ver.11.0 or later.**

When an instance of the function block is generated, the function block arguments of the specified instance can be referenced from outside the program in the function block. Argument name" is written for referencing.
For example, when the function block is called by specifying an instance as shown below, the value of the OUT argument "Area" of the instance Triangle1 can be referenced from outside the function block by writing "Triangle1.Area".



However, the following restrictions exist depending on the setting of the argument type.

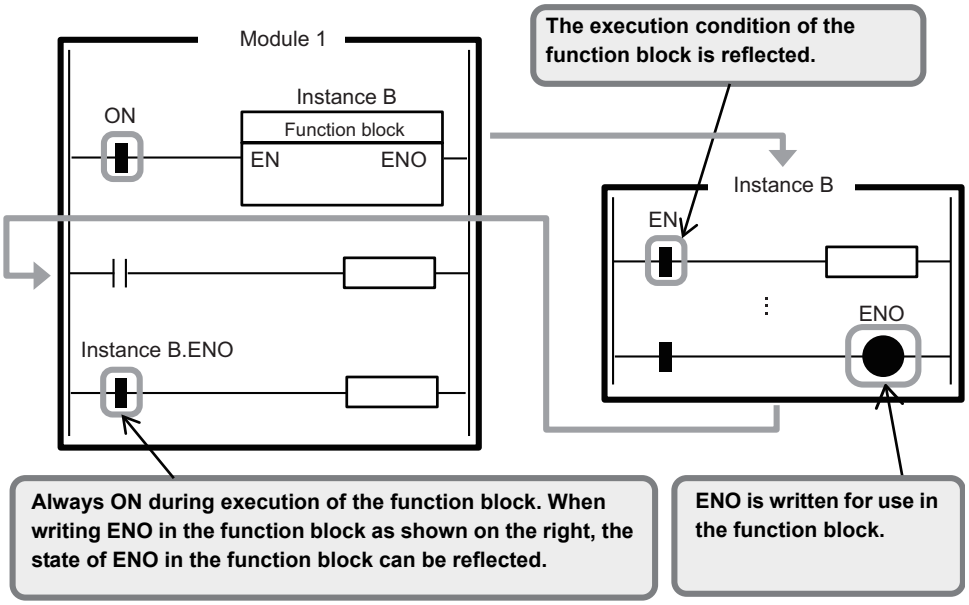
Argument type	Content
IN	The value of the argument can be referenced and changed with "Instance name.Argument name".
OUT	The value of the argument can be referenced with "Instance name.Argument name". The value cannot be changed.
IN-OUT	The value of the argument cannot be referenced or changed with "Instance name.Argument name".
Unit	The value of the argument cannot be referenced or changed with "Instance name.Argument name".

● Function block enable input / output

Enable input and enable output refer to the function block symbols "EN" and "ENO".

Argument type	Content
EN	The ON/OFF status of the execution condition of the function block is transferred to the inside of the function block. It can be used by writing "EN" in the program inside the function block. However, values cannot be referenced or changed by writing "Instance name.EN" from outside the program in the function block.
ENO	It is always ON during execution of the function block, and is OFF when execution of the function block is stopped. However, when a value is written by the program inside the function block, the written value will be reflected. Values can also be referenced by writing "Instance name.ENO" from outside the program in the function block.

The following shows the operation diagram of EN and ENO when a function block is executed.



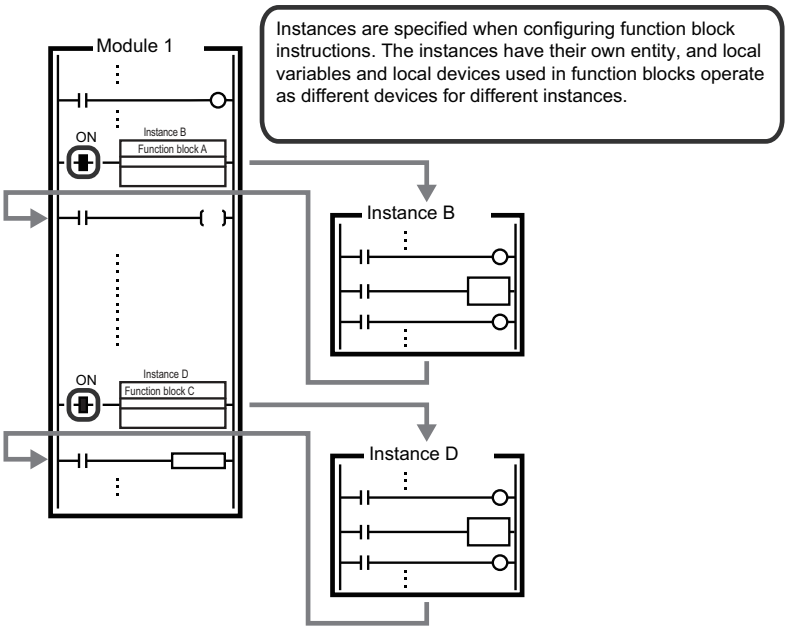
Point

- **When the ENO value in the function block is referenced with "Instance name.ENO", it is necessary with the FBCALL/FBSTRT instruction. The FB instruction does not support EN/ENO.**
- **When using the FBCALL/FBSTRT instruction, the KV-8000 CPU function version should be upgraded to 2.0 or higher using KV STUDIO Ver.11.0 or later.**
- **EN/ENO need not be set as an argument, and will not be counted as the number of arguments.**

● Operation description

An explanation of the function blocks is shown in the following figure.
Here is an example of specifying and executing Instance B of Function Block A and Instance D of Function Block C.
For details on instances, refer to "Function block instance" (page 4-121).

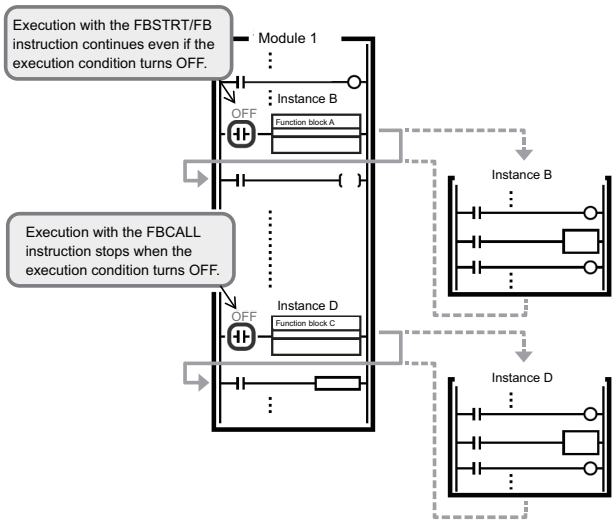
1 When the execution condition is turned ON, function block A starts.



Point

- **Function blocks started by the FBCALL/FBSTRT instruction are distinguished by the specified instance name and processed as different entities.**
- **Function blocks started by the FB instruction are distinguished by adding a control number after the function block name, such as function block A#2_1. This control number is displayed only when KV STUDIO is set to monitor / simulator.**

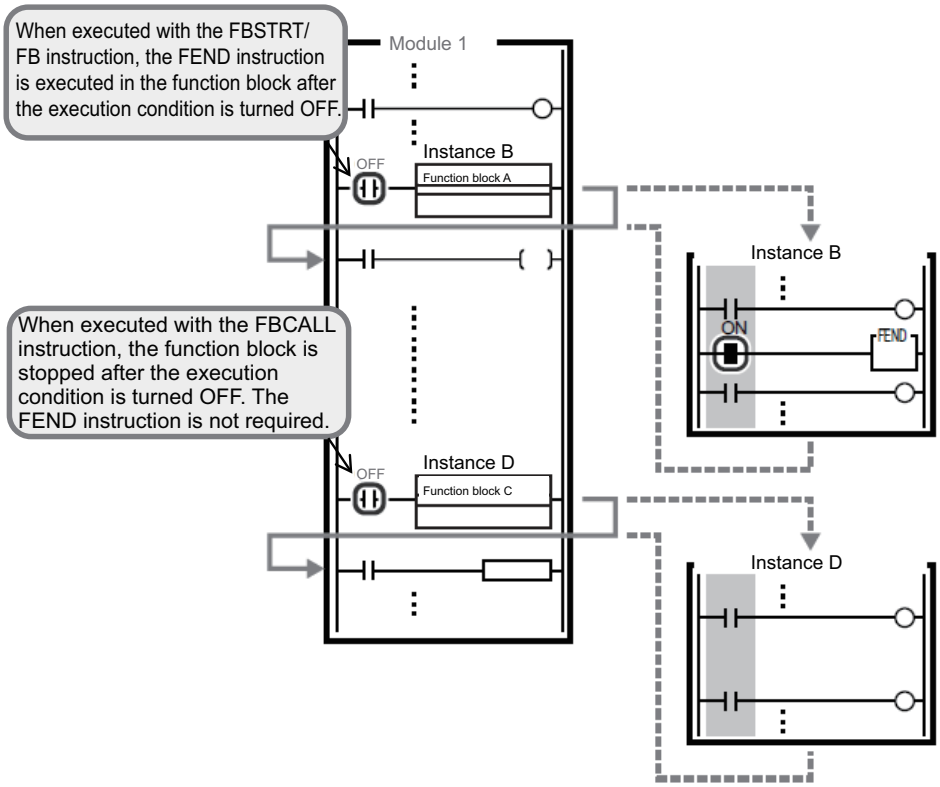
2 The execution conditions and operation states are as follows depending on the instruction that starts the function block.



● Stop conditions

Function blocks stop in the following conditions.

- When the read source program stops
- When an FEND instruction is executed in a function block (For FBSTRT/FB instruction)
- When the execution condition of the function block is turned OFF (When executed by instance name from FBCALL instruction or ST language)



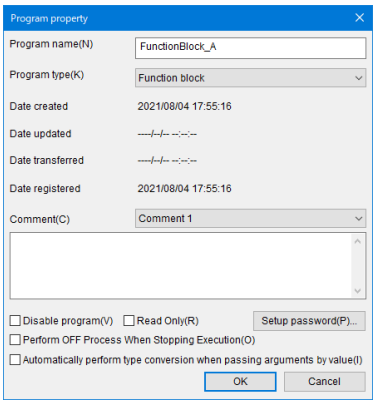
Point

- **When a function block is executed with the FB/FBSTRT instruction, the FEND instruction should be executed with the execution condition of the function block set to OFF. When the FEND instruction is executed while the execution condition of the function block is ON, the function block will not stop.**
- **If you execute a function block with an instance name from the FBCALL instruction or ST language, turn off the execution condition of the function block and stop the function block.**
- **When using function block instances with array types, only the instance specified with the subscript can be executed/stopped by the execution condition turning ON/OFF.**
  If the subscript specifies something other than a constant, note that changing this subscript before execution/stopping will lead to the instance specified with the subscript prior to the change not being executed/stopped.

● Operations when stops

The operation when stopped of the function block differs according to the "Program property" -> "Perform OFF Process When Stopping Execution" setting.

Point **In the case of KV-8000A, the setting is not displayed, and the operation is the same as when "OFF processing when execution stopped" item is always checked.**



If "Perform OFF Process When Stopping Execution" is not checked

The function block stops at the following timing.

- Next scan after the FEND instruction is executed in the function block (For FB/FBSTRT instruction)
- When the execution condition of the function block is turned OFF (For FBCALL instruction)

Point **If the function block is described in END to ENDH, the execution of the function block will not stop even if the destination module is stopped.** **"Function block instructions" (Page 141)**

If "Perform OFF Process When Stopping Execution" is checked

The function block executes 1 scan only at the following timing with all execution conditions in the function block set to OFF.*

- Next scan after the FEND instruction is executed in the function block (For FBSTRT/FB)
- When the execution condition of the function block is turned OFF (For FBCALL)

They will be stopped from the following scan.

* The following instructions are exceptions.

Instruction	Status
OUB	Turned OFF.
OFDL	Reset.
SHOT	
C	The status is held.
CJ, NCJ, and GOTO	Execution does not jump.

[Operation diagram of function block executed with FBSTRT/FB instruction]

	1 scan	1 scan	1 scan	1 scan	1 scan	1 scan
Function block A execution conditions	OFF	ON	*1	*1	OFF*2	OFF*3
Execution condition of the FEND instruction within Function block A	-	OFF	OFF	ON	*1	-
Function block A status	Stopped	Executing	Executing	Executing	Executing with conditions turned OFF	Stopped

*1: This being ON or OFF has no effect on operations.
*2: Be sure to turn this OFF. Until this is turned OFF, the FEND instruction will be ignored, and execution will continue.
*3: If this has been turned ON, the function block will start executing.

[Operation diagram of function block executed with FBCALL instruction]

	1 scan	1 scan	1 scan	1 scan	1 scan
Function block A execution conditions	OFF	ON	ON	OFF	OFF*1
Function block A status	Stopped	Executing	Executing	Executing with conditions turned OFF	Stopped

*1: If this has been turned ON, the function block will start executing.

Reference

Off processing when function block execution stops performs the same processing as when the MC instruction execution condition is OFF, for one scan only.

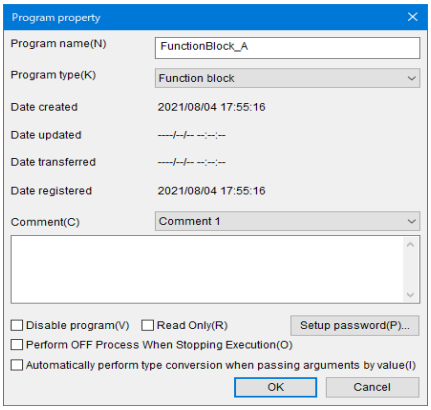
For details, see the MC and MCR instructions in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series Instruction Reference Manuals.

● Automatic data type conversion

The storage of values for function block arguments differs according to the "Program property" -> "Automatically Convert Types When Passing Arguments by Value" setting of each program.

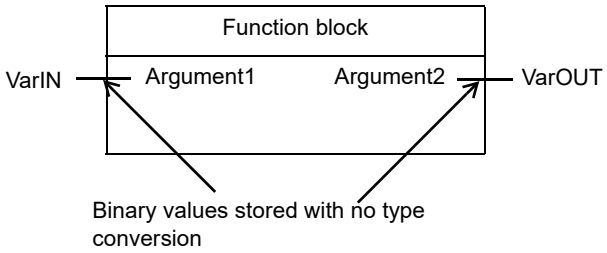
Point

- **When using the Logix format, the "Automatically perform type conversion when passing arguments by value" check box is selected by default. Otherwise, it is cleared by default.**
- **The "Automatically perform type conversion when passing arguments by value" setting is only possible on KV-8000 Series units with CPU function version 2.3 or later.**



When the "Automatically perform type conversion when passing arguments by value" check box is cleared

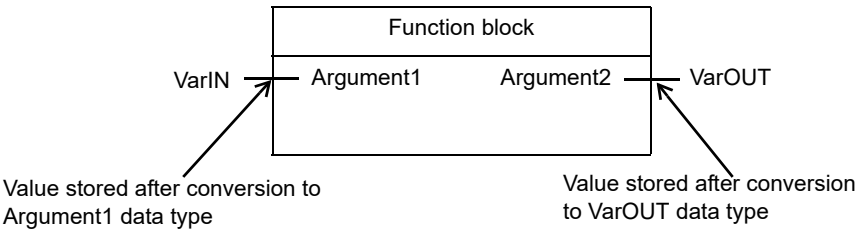
When storing values from operands to IN arguments or from OUT arguments to operands, binary values are stored with no type conversion.



When the "Automatically perform type conversion when passing arguments by value" check box is selected

Before values are stored from operands to IN arguments, type conversion is performed to match the IN argument data type.

Before values are stored from OUT arguments to operands, type conversion is performed to match the operand data type.



Point

- **When the "Automatically perform type conversion when passing arguments by value" check box is selected, only variables and IN/OUT/IN-OUT function block arguments can be specified for operands. Devices and unit function block arguments cannot be specified.**
- **When the "Automatically perform type conversion when passing arguments by value" check box is selected, indirect references and index modified variables cannot be specified for operands.**

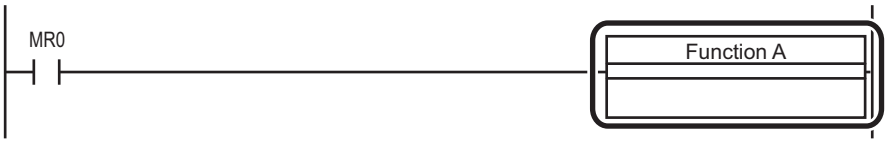
Functions

● Overview

Functions operate like subroutine programs, and execute the program in the function only when the execution condition is ON. After being executed, the execution proceeds to the next step. If the execution condition is OFF, the program in the function is not executed, and execution proceeds to the next step.

● Usage method

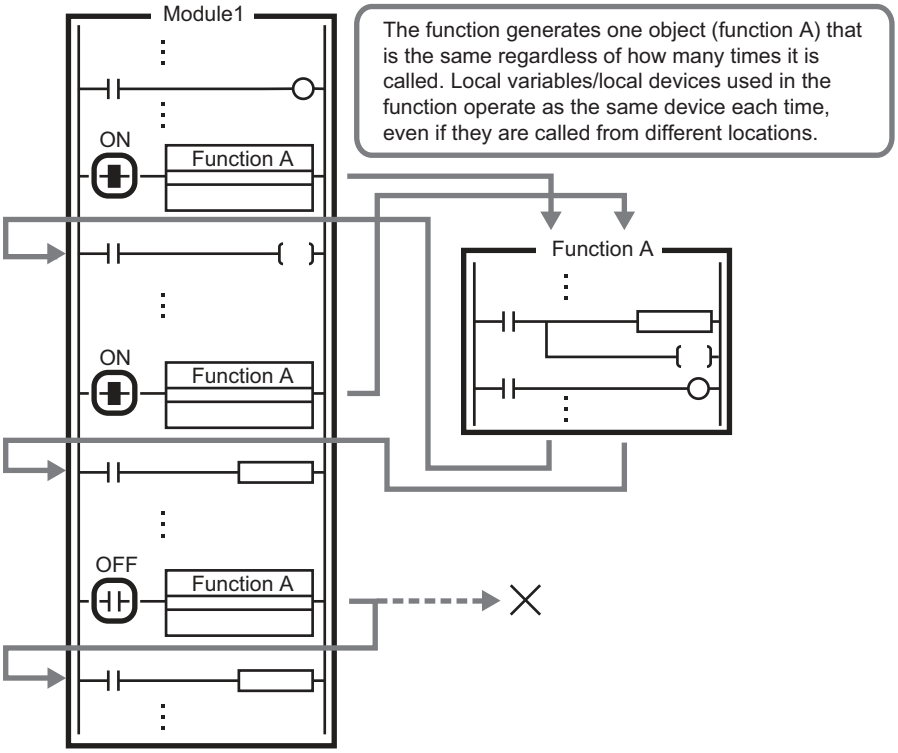
Function blocks are placed in the ladder program and used.



● Operation description

An explanation of the functions is shown in the following figure.
In this example, function block A is called twice from within module 1.

When the execution condition is ON, execution leaves module 1 and the ladder program in function A is executed.
After being executed, the execution returns to module 1 and proceeds to the next step.
If the execution condition is OFF, function A is not executed, and execution proceeds to the next step.



Point

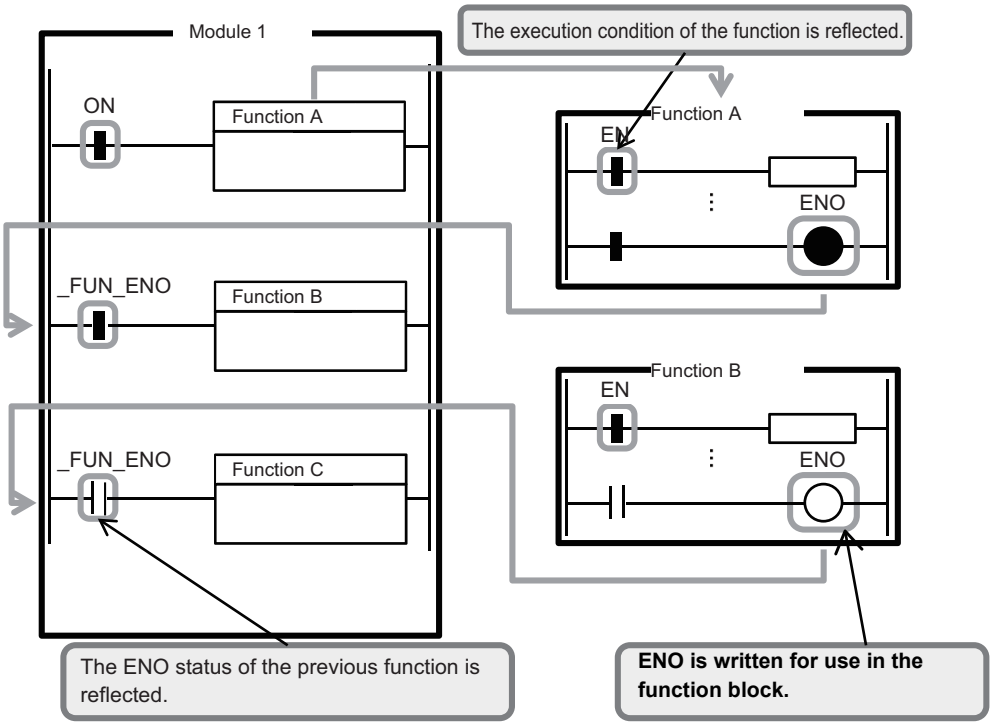
- **Local variables/local devices used in the function are shared. Take care when they are executed multiple times in the same scan, or if a function is executed using an interrupt program.**
- **When monitoring, the value of local devices used in the function display the result of the last time they were executed in the same scan.**
- **Functions do not create instances.**

● Function enable input / output

The arguments EN and ENO can be used program in the function in addition to the set arguments.

Argument type	Content
EN	The ON/OFF status of the function execution condition is transferred to the inside of the function. It can be used by writing "EN" in the program inside the function block. The status of "EN" cannot be referenced directly from outside the program in the function.
ENO	It is always ON during execution of the function, and is OFF when execution of the function is stopped. However, when a value is written the program in the function, the written value will be reflected. The value of "ENO" stored in the previous function can be referenced by writing "_FUN_ENO" outside the program in the function.

The following is the operation diagram of EN, ENO and _FUN_ENO when the function is executed. In the following example, Function A and Function B can be executed, but since ENO in Function B is OFF, _FUN_ENO as the execution condition of Function C also remains OFF, and Function C cannot be executed. In this way, it can be used during branch processing with the internal state of the previous function.



Point

- **When the ENO value in the previous function is referenced with "_FUN_ENO", the KV-8000 CPU function version should be upgraded to 2.0 or later using KV STUDIO Ver.11.0 or later.**
- **EN/ENO need not be set as an argument, and will not be counted as the number of arguments.**

● Function return value

The function return value is an argument with the same name as the function that can be used in the program inside the function. The return value can be referenced from a program outside the function. Since the FUN function is used in the script to directly process the function return value, it is not necessary to set an argument for storing the calculation result inside the function.

The following is the ST and KV script for storing the return value of Function1 in DM100.

ST program DM100: = Function1 (DM0, DM1);

Script program DM100 = FUN ("Function1", DM0, DM1)

* DM0 and DM1 are arguments of Function1.

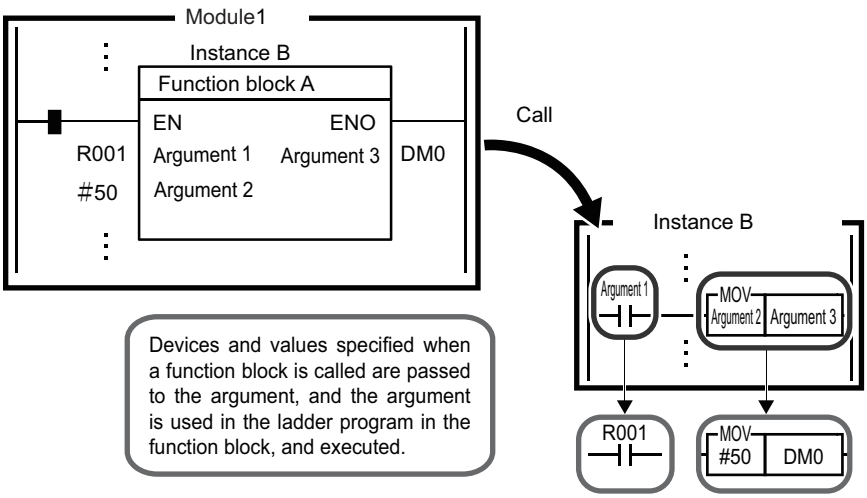
For example, if the program in Function1 is a program that calculates the area of a triangle with the first argument as the base and the second argument as the height, the calculation result of the area of the triangle is stored in DM100 by writing as above.

Setting Arguments

■ What are arguments?

Arguments are devices and values that are passed to the function block object when a function block is executed. They are specified as operands of function block inputs/outputs, or of call function block instructions (FBCALL, FBSTRT, FB, FUN). In function blocks, devices and values are passed to the argument, and the argument name is used to create a program in the ladder program in the function block (argument 1, 2 and 3 are argument name in the example below). Arguments are executed each scan if the function block is being executed.

The label used when passing arguments can be used in that function block only.

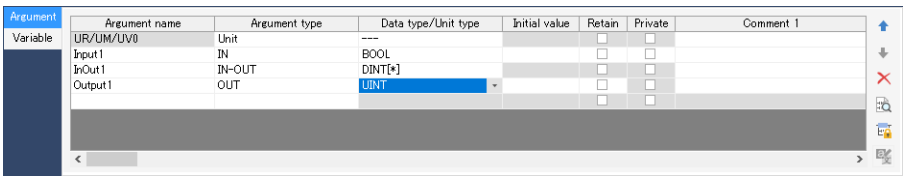


■ Setting method

Arguments are set in the argument setting area displayed in the top of the function block ladder edit window.

Alternatively, set them by right clicking the function block in the workspace, and selecting "Argument settings (Z)", and setting them in the "Set function block arguments" dialog box.

Alternative procedure Select the function block in the workspace, and select "Program (M)" -> "Argument settings (Z)" from the menu.



● Argument names

- **For IN, OUT and IN-OUT arguments**
  Set an argument names of up to 128 characters (either half-width or full-width characters).
  The set argument name can be used as a same as local variables in function blocks.
  "List of Characters That Cannot Be Used" (page A-39)
- **For Unit arguments**
  UR*, UM* and UV* arguments are assigned as fixed. * is a number from 0 that indicates the order from the top in argument settings.

■ Argument types

There are 4 types of argument: IN, OUT, IN-OUT, and Unit. The specifications for each type are shown below. Up to 64 arguments can be set in total. Up to 32 IN, OUT, and IN-OUT arguments can be set each, and up to 8 Unit arguments can be set.

Argument name		Argument type		Description	Display position*1	Indirect Specification (*)	Index Modification (:)
Any argument name		IN*1		The specified device or constant value is stored in the argument when function block execution starts.	Left	No	No
Any argument name		OUT*1		The argument value is stored in the specified device when function block execution ends.	Right	No	No
Any argument name		IN-OUT		The specified device reference is stored in the argument when function block execution starts. The processing carried out on the argument during function block execution is applied to the device specified each time.	Left/right*1	No	Yes*3
UR	UR0 to UR7	Unit*2	Leading R	The reference to the leading relay (R) assigned to the specified unit is stored in the argument when function block execution starts. The processing carried out on the argument during function block execution is applied to the device specified each time.	Left	No	Yes
UM	UM0 to UM7		Leading DM	The reference to the leading data memory (DM) assigned to the specified unit is stored in the argument when function block execution starts. The processing carried out on the argument during function block execution is applied to the device specified each time.		No	Yes
UV	UV0 to UV7		No.	The specified unit number is stored in the argument when function block execution starts.		No	No

*1 The position in which it is displayed above the function block symbol differs depending on the argument type. For IN-OUT arguments, the same input/output device is displayed on both the left and right, and can be edited from either.
*2 Specify only the target unit number in function block inputs/outputs.
Up to 8 units can be specified as arguments in one function block.
*3 When the data type is array / structure, index modification cannot be performed.

Reference When using variable-length strings and arrays as arguments, the argument type should be set to IN-OUT.

● Data type/unit type

Set the data type of the argument. The data types that can be set differ depending on the argument type.

- **For IN, OUT and IN-OUT arguments**

  Set the data type of the argument. The data types that can be set differ depending on the argument type.

Argument type	BOOL (Bit)	UINT (1 word unsigned integer)	INT (1 word signed integer)	2 word unsigned integer (UDINT)	DINT (2 word signed integer)	REAL (Single precision floating point)	LREAL (Double precision floating point)	STRING (Text string)
IN	Yes	Yes	Yes	Yes	Yes	Yes*1	Yes*1	Yes
OUT	Yes	Yes	Yes	Yes	Yes	Yes*1	Yes*1	Yes
IN-OUT	Yes	Yes	Yes	Yes	Yes	Yes*1	Yes*1	Yes

*1 When set to IN/OUT, the maximum number of characters and the number of elements should be set in advance. When set to IN-OUT, the maximum number of characters and the number of elements can be set to variable. However, if a STRING type array is set to IN-OUT, the number of elements can be set to variable but the maximum number of characters cannot.

For an argument with a large number of characters or array elements, it takes a long time to copy after setting it to IN/OUT, so it is recommended to set it to IN-OUT.

- **For Unit arguments**

  Sets the unit type. You can select from the units placed in the unit editor.

● Initial value

An initial value can be set for IN, OUT arguments. An initial value cannot be set for IN-OUT arguments. When an initial value is set, and the argument value is omitted, the initial value is used when executing the function block.

Point

- **Arguments will be initialized in the following cases in KV-8000 with CPU function version 2.0 or later.**
  - **When the program is transferred after the function block definition is changed for arguments with Hold set**
  - **When arguments without Hold set are switched from PROGRAM mode to RUN mode However, the arguments of the function block executed by the FB instruction are initialized when switching from PROGRAM mode to RUN mode, even if Hold is set.**

  Initialization is not performed when the function block starts and stops.
- **The arguments are also initialized when starting and stopping function blocks in KV-8000 with CPU function version 1.0.**

● Hold

Hold can be set when the argument type is IN or OUT.

When "Hold" is checked, the value of the argument will be held even when the power is turned off.

Point

- **When using the FB instruction without specifying the instance name, the value will not be held. To hold, use the FBCALL or FBSRT instruction to specify the instance name and execute it.**
- **When Hold is set for arguments, KV-8000 CPU function version should be upgraded to 2.0 or later using KV STUDIO Ver.11.0 or later.**

Private

Arguments that fulfill the following criteria can be set as private.

- IN arguments
- An initial value is set
- All arguments not set as private have a smaller No. than the arguments set as private.
  (Arguments set as private are automatically changed to the argument with the largest No.)

Arguments set as private are not displayed in the function block symbol. Instead, they can be displayed or edited from display details in the "Change operand" dialog box. Furthermore, a password can be set to restrict displaying/editing.

Comments

Explanatory comments that are displayed below the argument name can be set as necessary.

Point

- **Configure the comment section (below the function block name) in program properties.**
- **When the CPU function version is lower than 2.0, an argument uses a local work area of the following size.**
  - **When the argument type is IN or OUT**
    The size is 4 words when the display format is LREAL(double-precision floating point type), and 2 words otherwise
  - **The size is 2 words when the argument type is IN-OUT**
  - **The size is 6 words when the argument type is unit**
- **For CPU function version 2.0 or later, an argument will use the following local work area sizes.**
  - **When the argument type is IN or OUT**
    When the data type is structure, array or string, the data size is rounded up to an even number of words, LREAL type is 4 words, and other data types are 2 words
  - **The size is 4 words when the argument type is IN-OUT**
  - **The size is 6 words when the argument type is unit**
- **If the same "function" is executed in a main routine program and interrupt program simultaneously, the argument value is saved/restored when the interrupt program is executed.**
- **If the total number of local work words used by saved/restored arguments exceeds 6912 words, a calculation error [Ladder stack overflow] occurs.**
- **When the argument setting of the function block is different from the data type of the variable specified in the argument, the operation is as follows depending on the argument setting.**
  - **IN argument, OUT argument**

Structure and array (Specified number of elements)	**:A conversion error occurs when data types / numbers of elements are different**
String (Specified maximum number of characters)	**:A conversion error occurs when data types are different**
Other data types	**:A warning occurs during conversion when data types are different**

  - **IN-OUT argument**

Structure and array (Specified number of elements)	**:A conversion error occurs when data types or numbers of elements are different**
Array (Variable number of elements)	**:A conversion error occurs when data types are different**
String (Specified maximum number of characters)	**:A conversion error occurs when data types or maximum numbers of characters are different**
Other data types	**:A warning occurs during conversion when data types are different**

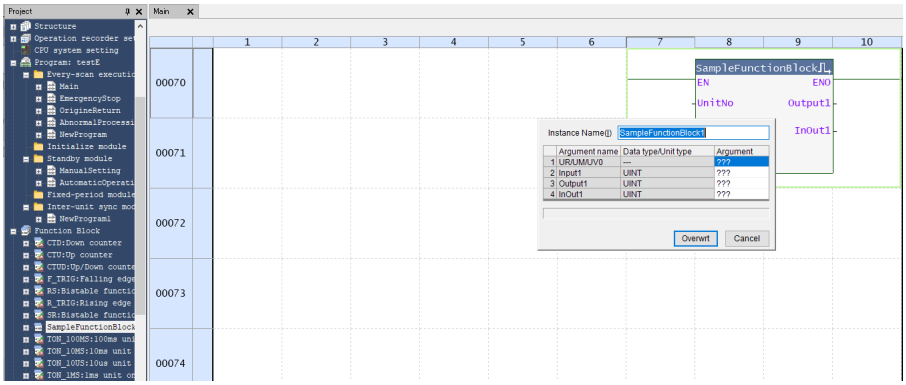
  - **Arguments set in IN and OUT are copied to the work area. Therefore, if the size of an argument is large, it is recommended to set the IN-OUT argument, which does not require copying time and consumes less work area.**

Placing Function Blocks

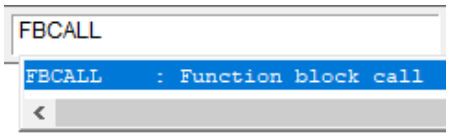
1 Drag the function block in the workspace and drop it in the ladder edit area.

The dialog for entering the variable name will be automatically displayed. Enter the variable name (instance name) and click "Overwrite".

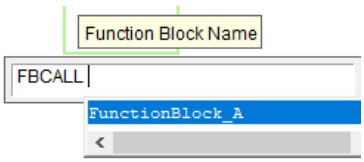
* When a function block is configured by drag and drop, the FBCALL instruction is used (The function block stops when its execution condition is turned OFF).



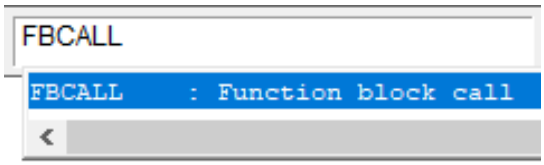
Other Enter "FBSTRT" or "FBCALL" with the keyboard.



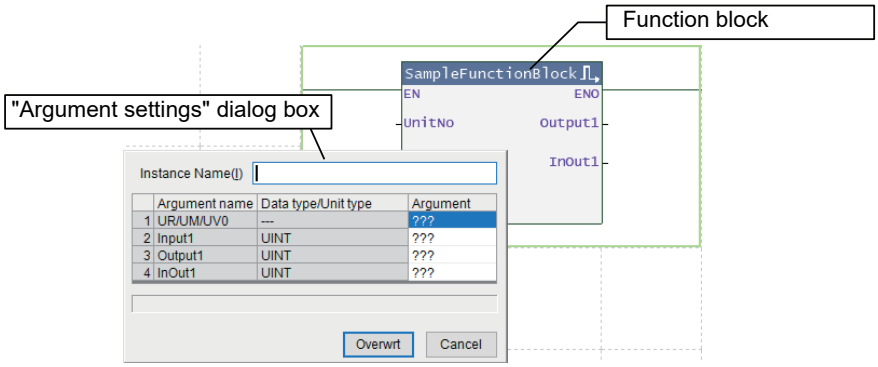
Press the space key to select a function block name from the candidates or enter it directly.



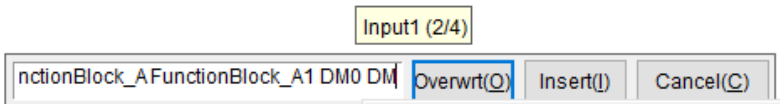
Press the space key again and enter the instance name. If the instance has been created beforehand,it will be displayed as a candidate.



2 **Specify the device to assign in the "Change operand" dialog box, and click the "Overwrite" button.**



Alternative procedure You can enter an argument directly in the direct entry dialog box.



Point

- **Function blocks cannot be placed in macros.**
- **Function blocks can be placed in function blocks.**
  However, it may not be possible to place them depending on the combination of program types (function block/function).
  "Function Block Nests" (page 4-140)

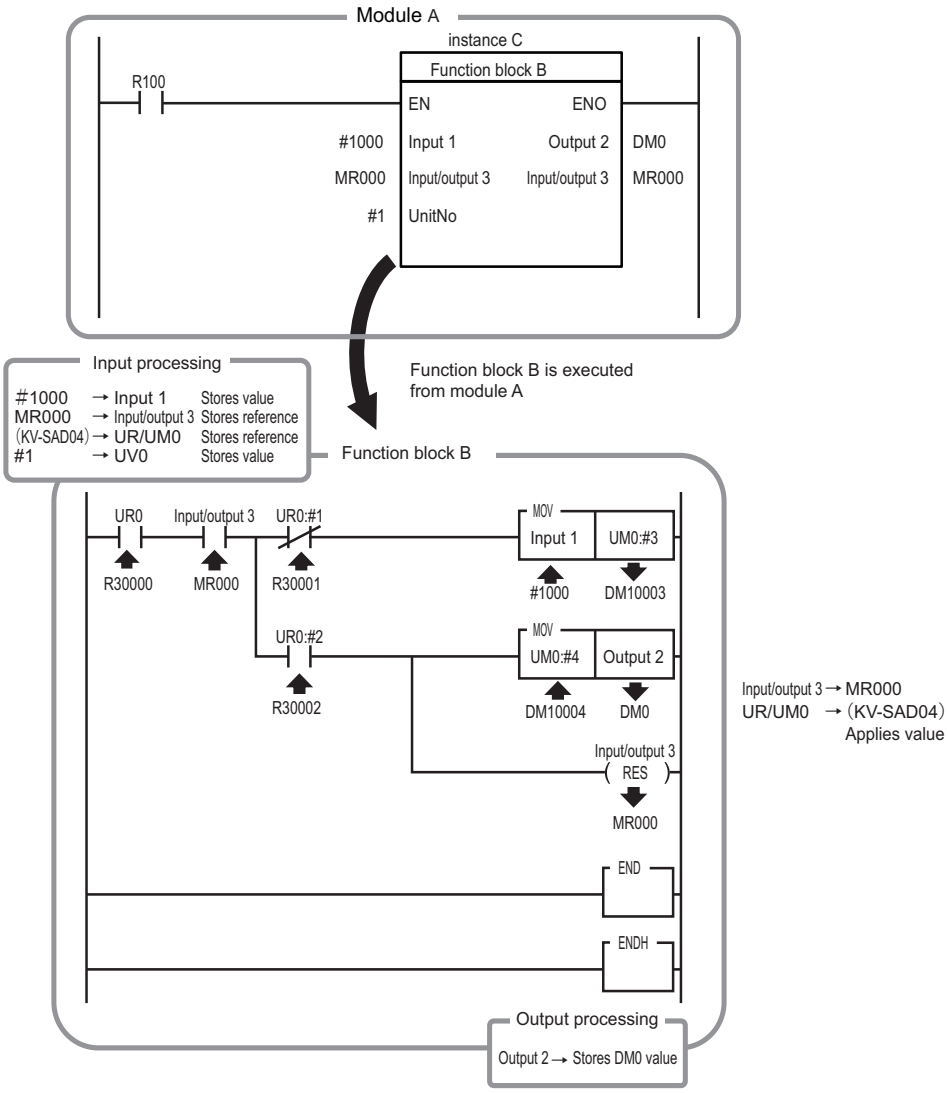
Function block usage example

The following is an example of how function block arguments are used in the ladder program.
The example shows function block B being executed from module A.

Argument usage example

No.	Argument types	Argument name/device
1	IN	Input 1
2	OUT	Output 2
3	IN-OUT	Input/output 3
4	Unit Device	UR/UM/UV0

Unit number 1 ... KV-SAD04 (R30000～R30115 / DM10000～DM10059)



Reference

To use unit device UR0 / UM0, specify the device number using indirect reference.
Example) To specify R36604 when the unit's leading relay number is R30000
R36604 = R30000 + 66ch (=66 × 16 = 1056) + 4
= R30000 + 1060
= UR0:#1060

Argument Abbreviation Settings

Function blocks can be executed even if the parameter values used and output devices are not set. You can create function blocks so detailed settings are provided, and use typical values to use them easily, etc.

Function blocks can be used for IN or OUT type arguments without specifying input / output device.

Point

- **For KV-8000 with CPU function version 1.0, when the argument type is IN, it can be omitted only when the initial value is set.**
- **For KV-8000 with CPU function version 1.0, all arguments that are not omitted must be listed with smaller No. than the omitted arguments (arguments of the middle No. cannot be omitted).**

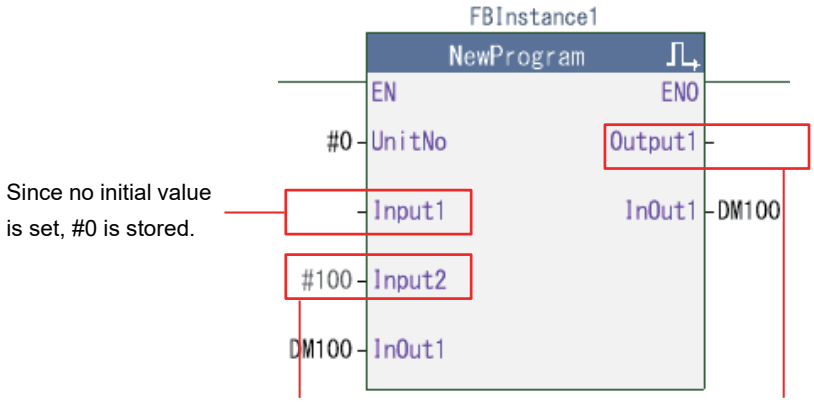
If IN arguments are abbreviated, when function block execution starts the value set as the initial value is automatically stored in the argument, and they are executed according to the initial value. "Initial value" (page 4-133)
If OUT arguments are abbreviated, the argument value is not output when function block execution ends.

Input1, Input2, Output1 arguments can be omitted for the following function block.

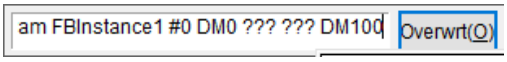
IN argument

OUT argument

Argument name	Argument type	Data type/Unit type	Initial value	Retain	Private
UR/UM/UV8	Unit			☐	☐
Input1	IN	UINT		☐	☐
Input2	IN	UINT	#100	☐	☐
Output1	OUT	UINT		☐	☐
InOut1	IN-OUT	UINT		☐	☐
				☐	☐



"???" is written to omit the argument during direct input. The following is an example of omitting the No.3 and No.4 arguments.



Point

- **Only KV-8000 with CPU function version 2.0 or later can enter "???" and omit the middle argument.**
- **When an argument is omitted with "???", enter all ? in half-width characters without spaces.**

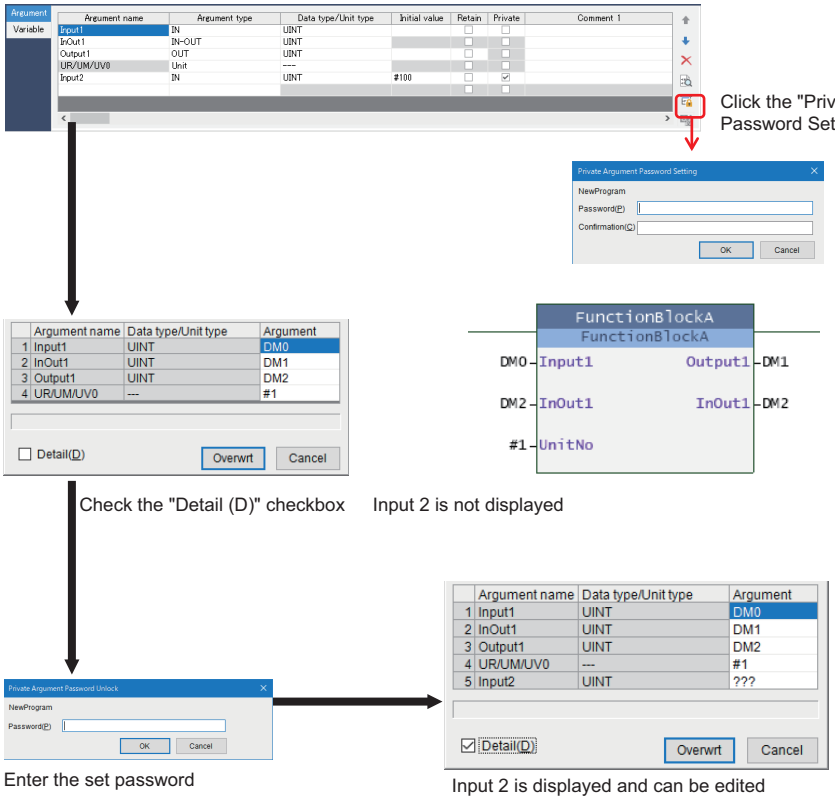
Argument Privacy Settings

In function blocks some of the arguments can be set as private. Function blocks with multiple adjustment parameters provided so they can be used even when carrying out complicated control can be created, and they can be made available with the setting parameters hidden so they can be used easily by the person in charge.

"Private" (page 4-134)

● Usage example

This example sets Input 2 (No. 5) to private.



Reference

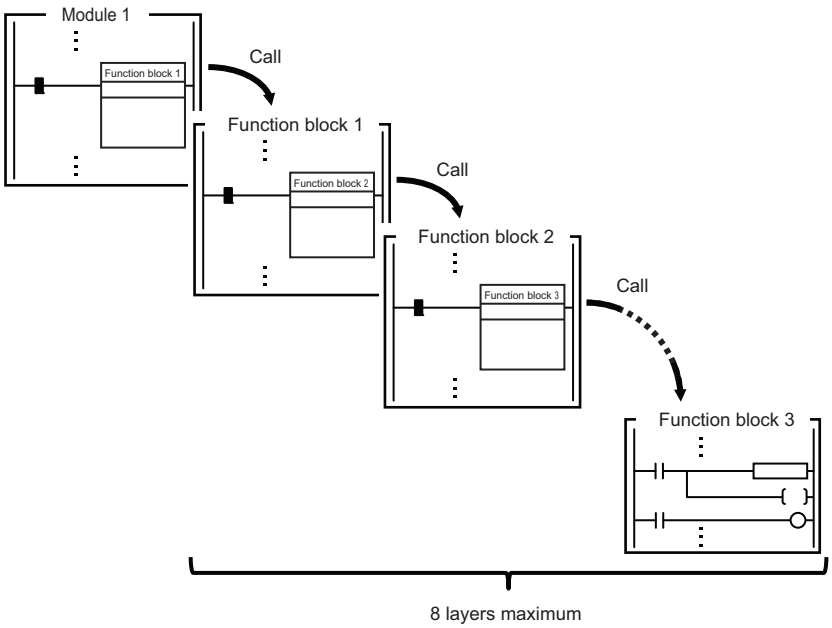
- The "Detail (D)" checkbox only appears when an argument is set to private.
- If the input device for arguments set to private is not edited (is omitted), the initial value is used.

Function Block Nests

Other function blocks can be called from function blocks. (Function block nests)
Function blocks where nesting is possible differ according to the program type (function block/function) as shown below.

Type of program used to call	Program types that can be called	
	Function block	Function
Function block	Yes	Yes
Function	No	Yes

Nests of up to 8 layers can be created.



Point

- **The same function block as the function block used to call cannot be nested.**
- **If the program type is function block, a separate object is created each time it is Instance.**
  When using nesting, pay attention to the volume of the program/local device.
  If the program type is function, only one object is generated regardless of the number of times it is called, so the volume of the program/local device does not increase even if nesting is used.
  "Function Block Type" (page 4-120)
- **Programs with more than 8 layers of nests cause a conversion error during conversion in KV STUDIO.**
- **When a program is transferred with a matching discrepancy, and the nest is more than 8 layers when executed, CPU error E33 "Program Exceeds Nesting Limit" occurs and operation stops.**

Setting a Function Block Password

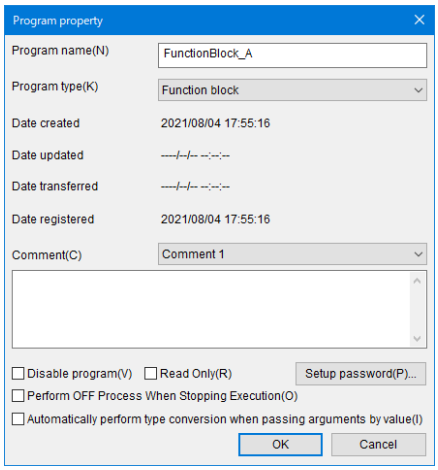
Passwords can be set to function blocks so the program display is limited in KV STUDIO.
Follow the procedure below to set a password.

1 Select the function block to set a password for in the workspace.

2 Right-click, and then select "Property" in the menu that is displayed.

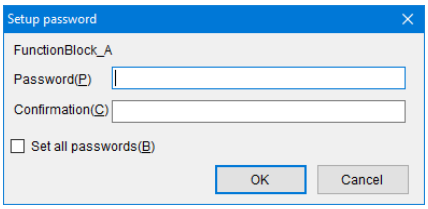
The "Program property" dialog box is displayed.

Alternative procedure
- Select the function block in the workspace, and select "Program" -> "Properties" from the menu.
- Select and right click the function block in the workspace and select "Setup Password" from the menu.



3 Click "Setup password."

The "Setup password" dialog box is displayed.



4 Enter the password in both text boxes (the second is for confirmation), and then click "OK."

After you set the password, when an attempt is made to display the function block in KV STUDIO, a dialog box will be displayed asking the user to enter the password.
If the user does not enter the password, they will not be able to display the function block in KV STUDIO.

Setting Function Blocks to Read Only

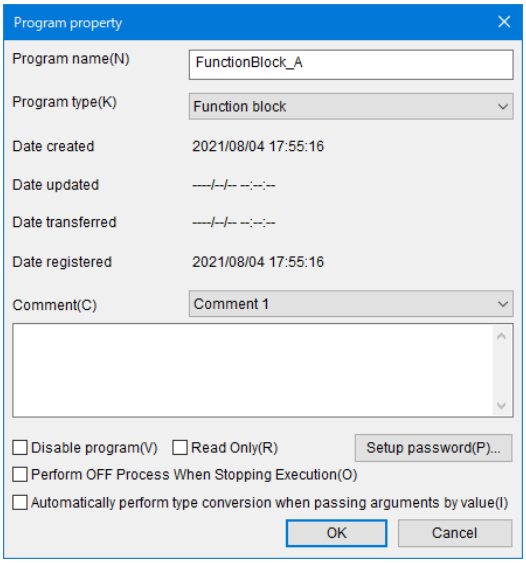
Function blocks can be set to read only so program editing is limited in KV STUDIO.
Follow the procedure below to set the function block to read only.

1 Select the function block to set to read only in the workspace.

2 Right-click, and then select "Property" in the menu that is displayed.

The "Program property" dialog box is displayed.

Alternative procedure
- Select the function block in the workspace, and select "Program" -> "Properties" from the menu.
- Select and right click the function block in the workspace and select "Setup Password" from the menu.



3 Check the "Read Only (R)" box.

Every time you click the box it changes as shown below.

Status	Description
☐	The program can be edited.
☑	The program cannot be edited.
■	The program can be edited. If the project is reopened, the program becomes uneditable again. Use this to temporarily enable editing.

Function Block System Devices

A function block system device is a device that is used to capture the operation status of a function block.

Point **You cannot attach device comments to function block system devices.**

Reference KV STUDIO automatically assigns function block system devices to the CPU unit internal work area.

● Bit that is turned ON for only one scan when function block execution starts, "@CR2008"

This bit is turned ON for only the first scan after function block execution starts.
From the second scan, it is turned OFF.
This bit can only be referenced from its own function block. It cannot be referenced from the program that called this macro.
In KV STUDIO, this device is represented by "@CR2008."

● Bit that is turned OFF for only one scan when function block execution starts, "@CR2007"

This bit is turned OFF for only the first scan after function block execution starts.
From the second scan, it is turned ON.
This bit can only be referenced from its own function block. It cannot be referenced from the program that called this macro.
In KV STUDIO, this device is represented by "@CR2007."

Point **System devices such as @CR2008 can be used in function blocks, but not in functions.**

Function Block Program Properties

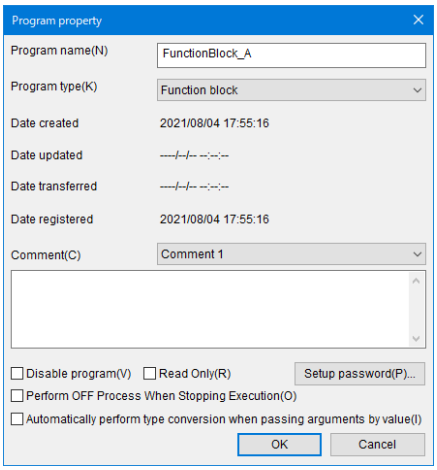
The function block program properties can be used to disable the function block and display the function block comments.
Follow the procedure below to open program properties.

1 Select the function block to set program properties for in the workspace.

2 Right-click, and then select "Property" in the menu that is displayed.

The "Program property" dialog box is displayed.

Alternative procedure
- Select the function block in the workspace, and select "Program" -> "Properties" from the menu.
- Select and right click the function block in the workspace and select "Setup password" from the menu.



● Disable program

Check this box to disable the function block. Function blocks can also be disabled by right clicking and selecting "Disable/Enable".

● Read Only

Check this box to set the function block to read only.
"Setting Function Blocks to Read Only" (page 4-142)

● Perform OFF Process When Stopping Execution

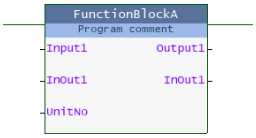
Check to perform OFF processing when execution stops.
Refer to "Operations when stops" (page 4-125) for information about OFF processing.

● Automatically perform type conversion when passing arguments by value

Check this box to convert data types if they differ between the operands and arguments when passing arguments by value from operands. "Automatic data type conversion" (page 4-127)

● Comment

Sets function block comments. If a comment is set it is displayed beneath the function block name. Comments can be up to 256 characters long.



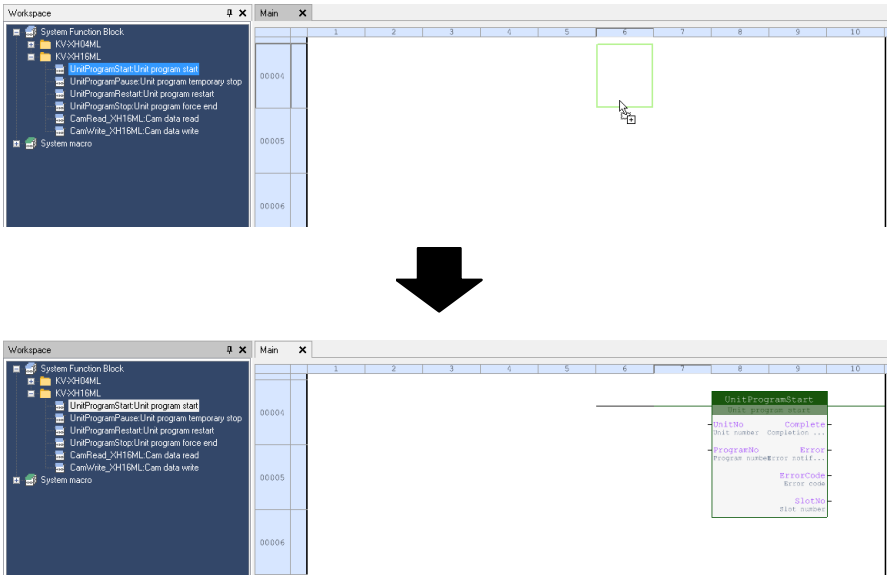
System Function Block

Specialized system function blocks designed for the uses of the unit are provided for some units.

1 Switch to the "Library" tab in the workspace.

2 Drag the system function block and drop it in the ladder edit area.

A function call instruction that calls the selected system function block is placed.
The selected system function block is also added to the project at the same time.



Reference Refer to the user manual for each unit for information about the system function block function.

Point

- **System function blocks are displayed in a different color to standard function blocks.**
- **System function blocks are initially set to read only, and cannot be edited. If read only is disabled in program properties, they become standard function blocks. These standard function blocks cannot be changed back to system function blocks.**
  "Setting Function Blocks to Read Only" (page 4-142)

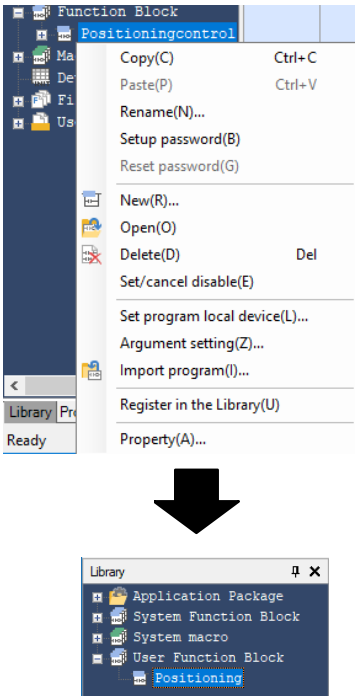
For the system function block for KV-8000 (A), refer to "System function block for "KV-8000"" (page 4-154).

Register a function block library

You can register the created function block in the library to divert it as materials. Since the function block registered in the library is registered beforehand as a user function block when creating a new project, you can save the trouble of importing the function block each time you create a project.

1 **Select the function block to be registered in the library on [Workspace].**

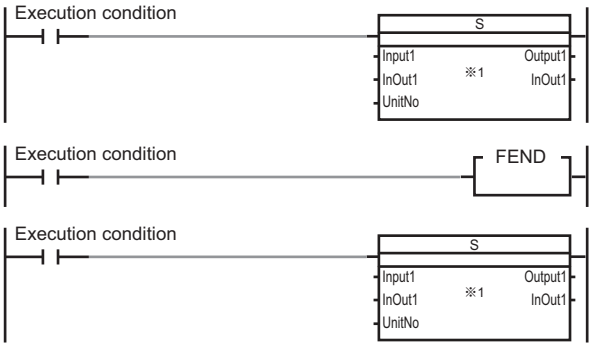
2 **Right-click and select【Add to Library】from the menu that is displayed.**



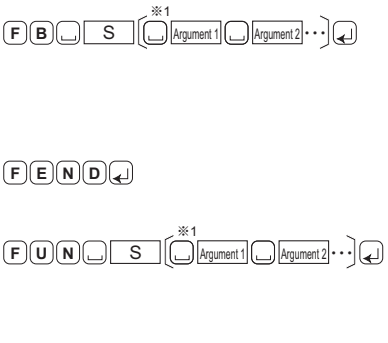
Point **KV STUDIO Ver.9.10 or later can register the function block in the library.**

FB		Function block call	Calls the specified function block.
FEND	FEND	Function block end	Stops execution of the function block. (Written in the function block)
FUN		Function call	Calls the specified function.

Ladder program



Entry mode



Operation	Explanation	Occupied size
S	Specifies the function block/function name.*3	—

*1 The number of operands other than the function block / function name and the data that can be specified differ as follows depending on the argument settings set in the function block / function to be executed.

- IN : Devices, variables, labels and constants can be specified.
- OUT : Devices, variables and labels can be specified. Constants cannot be specified.
- IN-OUT : Devices, variables and labels can be specified. Constants cannot be specified.
- Unit : Devices, variables, labels and constants can be specified.

*2 Handling of T, C, TIMER type variables and COUNTER type variables specified for the IN-OUT argument differs depending on the instruction used in the function block / function.
Example) For the MOV instruction, the current value is handled. For details, refer to the operation description of the respective instruction.

*3 "", # and $ cannot be used. Directly enter the name of the function block/function set in KV STUDIO. Up to 64 characters can be entered regardless of full-width or half-width.

Point

The FB instruction, FEND instruction and FUN instruction can be used with the KV-8000 series CPU units and KV-7000 Series CPU units with CPU function version 2.0 or later.

"KV-8000(A) Series CPU Function Version" (Page 2)

Description of Operation

FB The function block specified by [S] starts up when the execution condition is ON. Execution of the specified function block is started from the scan where the execution condition changes from OFF to ON.
Once execution starts, the function block continues to be executed even if the execution condition is OFF.

FEND Function blocks with a FEND instruction written to them stop when the execution condition is ON.

FUN The function specified by [S] is executed when the execution condition is ON.

The OFF process when stopped differs according to the program settings. Refer to the instruction manual for the CPU unit for details.

Point

- **Writing in RUN mode is not possible if the number or sequence of calls of the FB instruction is changed.**
- **Function block stops in the following cases.**
  - **When the read source program stops**
  - **When a FEND instruction is executed in a function block**
- **If the FBCALL/FBSTRT/FB instruction execution condition stays ON, the FEND instruction is ignored.**
- **The FEND instruction cannot be written in the module or function.**
- **The FEND instruction cannot be used to stop other function blocks.**
- **The previous state of differential execution type instructions in function blocks is initialized every time the execution of the function block is started. (rising edge differentiation: ON, falling edge differentiation: OFF)**
- **Differential execution type instructions and timer instructions cannot be used in functions.**
- **In the following cases the function block/function being executed stops when each module/function block stops or when the execution condition for each set of instructions turns OFF.**
  - **When an FBCALL/FBSTRT/FB/FUN instruction is written to a module/function block**
  - **When an FBCALL/FBSTRT/FB/FUN instruction is written to master control (between the MC~MCR instructions)**
  - **When an FUN instruction is written to stage (in the STG - JMP instructions)**
- **In the following cases the function block/function being executed is canceled when the execution condition for each set of instructions turns OFF. The function block execution resumes when each set of instructions is executed. The previous state of differential execution type instructions in function blocks is not initialized at this time. (If it is written in the initialization module only, 1 scan is executed when operation starts.)**
  - **When an FBCALL/FBSTRT/FB/FUN instruction is written to sub-routine (between the SBN~RET instructions)**
  - **When an FBCALL/FBSTRT/FB/FUN instruction is written to conditional branch (between the CJ/NCJ~LABEL instructions)**
  - **When an FBCALL/FBSTRT/FB/FUN instruction is written to step (between the STP~STE instructions)**
  - **When an FBCALL/FBSTRT/FB/FUN instruction is written to interrupt (between the INT~RETI instructions)**
  - **When an FBCALL/FBSTRT/FB instruction is written to stage (between the STG~JMP instructions)**
- **FB instructions cannot be used in functions.**
- **FB/FUN instructions cannot be used in macros.**
- **FBCALL/FBSTRT/FB/FUN instructions that call the function block/function itself cannot be used in that function block/function.**
- **Sub-routine programs can be used in function blocks/functions, but interrupt programs cannot be used.**
- **When performing a certain function block/function multiple times in a project, the global devices / global variables / global labels used for the OUT/OUB instruction in a function block/function will become double coil.**

Operation flag

CR2009	The state depends on the processing result in the function block/function to execute.
CR2010	The state depends on the processing result in the function block/function to execute.
CR2011	The state depends on the processing result in the function block/function to execute.
CR2012	ON in the following cases. Otherwise, OFF. • When the indirect specification or index modification range is inappropriate • When the unit specified in the "Unit" argument type is not connected • When the unit specified in the "Unit" argument type and the unit specified in the argument settings differ • When an access outside the operand range occurs

* If the argument is inappropriate, the function block/function will not be executed.

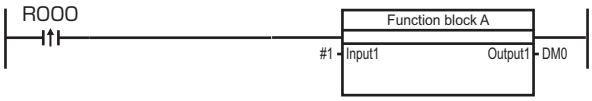
When CR2012 is ON, error details are stored in CM5150 to CM5176.

"Error log" (page 4-86)

Point **The state of the operation flag cannot be changed by the FEND instruction.**

Sample Program

Function block "Function block A" execution starts with variable 1 (unit number) as #1 and variable 2 as DM0 at the rising edge of input relay R000.



(Mnemonics list)

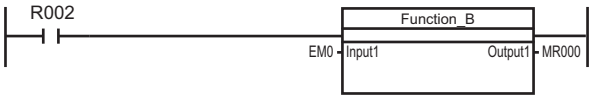
```
LDP R000
FB Function block A #1 DM0
```

Function block "Function block A" stops when input relay R001 is ON.

(Mnemonics list)

```
LD R001
FEND
```

Function "Function B" is executed with variable 1 as EM0 and variable 2 as MR000 when input relay R002 is on.

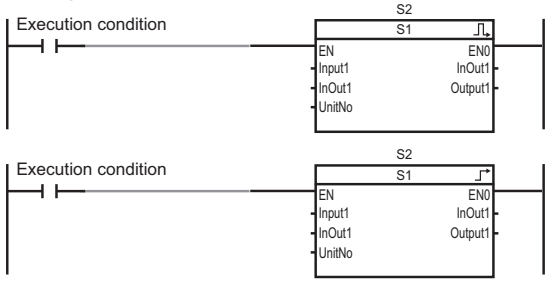


(Mnemonics list)

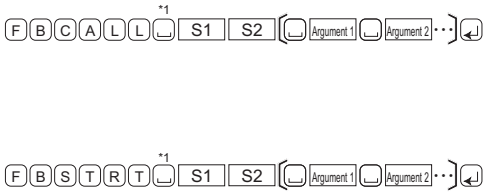
```
LD R002
FUN Function B EM0 MR000
```

FBCALL		Instance type function block start	Calls the specified function block and executes the instance.
FBSTRT		Instance type function block call	Calls the specified function block and executes the instance.

Ladder program



Input method



Operand	Description	Occupied area
S1	Specifies the function block name.*2	Number of characters
S2	Specifies the instance name.*2	Number of characters

*1 The number (0~64) and attributes of arguments after the third operand differ depending on the argument settings set in the function block to be executed.

*2 Only constants can be specified. Directly enter the function block name (Up to 64 characters regardless of half-width or full-width) or instance name (Up to 128 characters regardless of half-width or full-width).

Point

The FBCALL instruction and FBSTRT instruction can be used with KV-8000 series CPU units with CPU function version 2.0 or later.

"KV-8000(A) Series CPU Function Version" (Page 2)

Description of Operation

FBCALL When the execution condition is ON, the function block specified by S1 is activated and the instance specified by S2 is executed.
The specified function block is executed when the scan is turned from OFF to ON.
Once started, the execution of the function block continues while the execution condition is ON.
When the execution condition turns OFF, the execution of the function block stops.

FBSTRT When the execution condition is ON, the function block specified by S1 is activated and the instance specified by S2 is executed.
The specified function block is executed when the scan is turned from OFF to ON.
Once started, the execution of the function block continues even if the execution condition is OFF.
When the function block is stopped, the FEND instruction is executed in the function block.

The operation of the OFF process at stop varies depending on the program settings.

"Operations when stops" (page 4-125)

Point

- Function cannot be specified with the FBCALL/FBSTRT instruction. The FUN instruction should be used to execute a function.
- When an instance is added, deleted or renamed, or if the order is changed in the variable edit window, writing during RUN cannot be performed. When the FBCALL/FBSTRT instruction is deleted or their order is changed, writing during RUN can be performed.
- The function block stops in the following cases.
  - When the read source program stops
  - When the FEND instruction is executed in the function block (when called with the FBSTRT instruction)
  - When the execution condition of the FBCALL instruction is turned OFF (when called with the FBCALL instruction)
  - When the execution condition of the FBCALL/FBSTRT instruction remains ON, the FEND instruction will be ignored.
- The previous state of the differential execution type instruction in the function block is initialized each time the execution of the function block starts. (Rising edge differential: ON, falling edge differential: OFF)
- When each module / function block stops or the execution condition between the instructions turns OFF in the following cases, the function block being executed stops.
  - When the FBCALL/FBSTRT instruction is written in a module / function block
  - When the FBCALL/FBSTRT instruction is written in the master control (between MC and MCR instructions)
- When the execution condition between the instructions is turned OFF in the following cases, the function block being executed is interrupted. Next, the execution of the function block is resumed when the instructions are executed. At this time, the previous state of the differential execution type instruction in the function block is not initialized. (When written in the initialization module, only 1 scan is performed at the start of operation.)
  - When the FBCALL/FBSTRT instruction is written in a subprogram (between the SBN and RET instructions)
  - When the FBCALL/FBSTRT instruction is written in a conditional branch (between the CJ/NCJ and LABEL instructions)
  - When the FBCALL/FBSTRT instruction is written in a step (between the STP and STE instructions)
  - When the FBCALL/FBSTRT instruction is written in an interrupt (between the INT and RETI instructions)
  - When the FBCALL/FBSTRT instruction is written in a stage (between the STG and JMP instructions)
- The FBCALL/FBSTRT instruction cannot be used in a function.
- The FBCALL/FBSTRT instruction cannot be used in a macro.
- The FBCALL/FBSTRT instruction that calls the function block in itself cannot be used.
- Subprograms can be used in function blocks, but the interrupt program cannot be used.
- If the same function block is executed for multiple times in a project, using a global device / global variable for the OUT/OUB instruction in the function block will result in a double coil.

Calculation flag

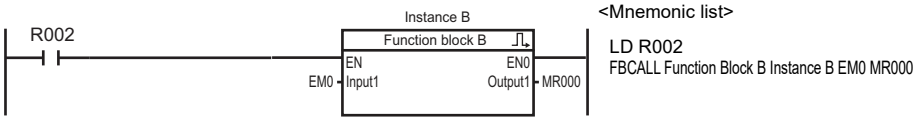
CR2009	Depends on the processing result in the function block executed.
CR2010	Depends on the processing result in the function block executed.
CR2011	Depends on the processing result in the function block executed.
CR2012	ON in the following cases, and OFF otherwise. • When the range of indirect specification or index modification is inappropriate • When the unit specified in the argument of the argument type "unit" is not connected • When the unit specified in the argument of the argument type "unit" is different from the unit type specified in the argument settings • When an out-of-range access of operands occurs

* When the argument is inappropriate, the function block cannot be executed.
When CR2012 is turned ON, the error details are stored in CM5150~CM5176.
"Error log" (page 4-86)

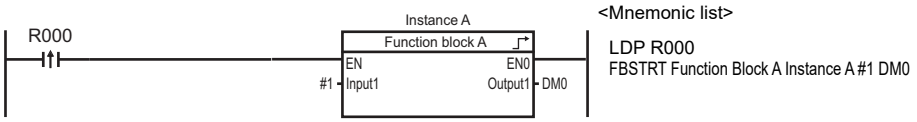
Point **The calculation flag is not changed by the FEND instruction.**

Sample Program

When the input relay R002 is ON, the instance named "Instance B" of the function block "Function Block B" is executed with argument 1 as EM0 and argument 2 as MR000. The execution stops when the input relay R002 turns OFF.



The instance named "Instance A" of the function block "Function Block A" is executed with argument 1 as #1 and argument 2 as DM0 at the rising edge of input relay R000.



When the input relay R001 is ON, the execution of "Instance A" of "Function Block A" is stopped.

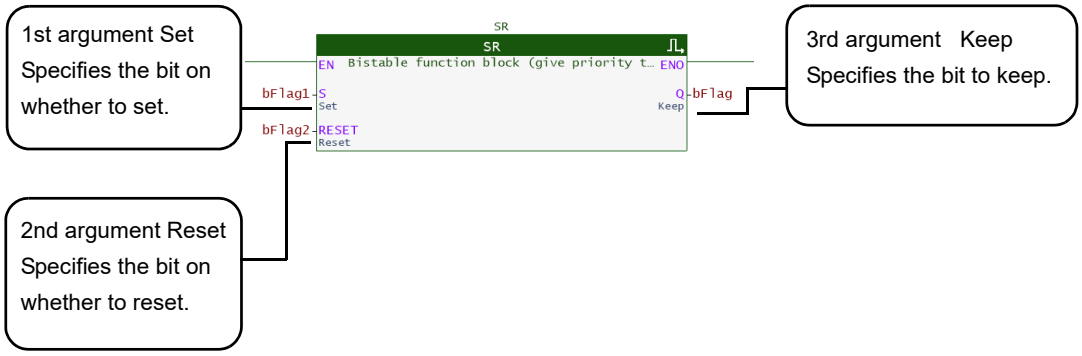
System function block for "KV-8000"

This section describes the system function block for << KV-8000 >> used in the ladder program.

Function	Function block program name	Overview	Page
Bistable function block (Set priority)	SR	The output signal is controlled by the Set/ Reset signal. (Set priority)	4-155
Bistable function block (Reset priority)	RS	The output signal is controlled by the Set/ Reset signal. (Reset priority)	4-157
Rising edge detector	R_TRIG	Detects the rising edge.	4-159
Falling edge detector	F_TRIG	Detects the falling edge.	4-160
Up counter	CTU	16-bit Up counter.	4-161
Down counter	CTD	16-bit Down counter.	4-163
Up/Down counter	CTUD	16-bit Up/Down counter.	4-165
100ms On-delay	TON_100MS	On-delay timer that is specified in units of 100ms.	4-167
10ms On-delay	TON_10MS	On-delay timer that is specified in units of 10ms.	4-169
1ms On-delay	TON_1MS	On-delay timer that is specified in units of 1ms.	4-171
10μs On-delay	TON_10US	On-delay timer that is specified in units of 10μs.	4-173

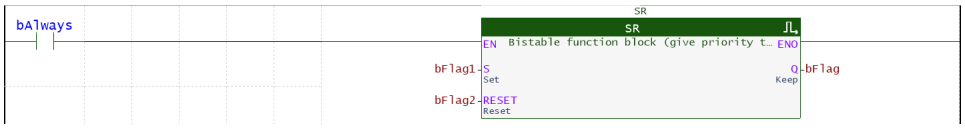
SR	Bistable function block (Set priority)	The bit state is controlled by the combination of Set/Reset signal. (Set priority)

Argument



Sample program

The bFlag state is controlled by the state combination of bFlag1 and bFlag2.

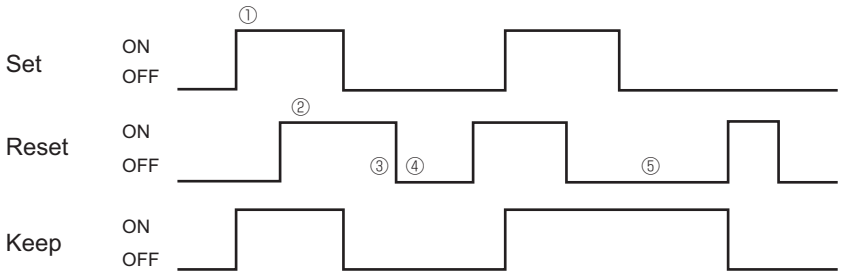


Description of Operation

The "Keep" state is controlled by the bit state combination of bits specified in "Set" and "Reset".

Combination table

		RESET	
		ON	OFF
S	ON	ON	ON
	OFF	OFF	(Keeps the last state)



① When Set is ON, Reset is OFF, Keep will be ON.

② Even if Reset is ON, but Set is ON, Keep will be ON.

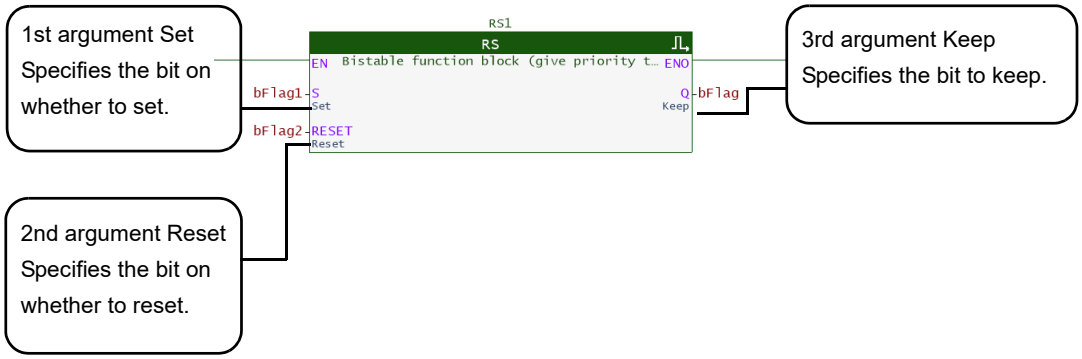
③ Set is OFF and Reset is ON, Keep will be OFF.

④ Keep is OFF, and both Set and Reset are OFF, Keep will remain OFF.

⑤ Keep is ON, and both Set and Reset are OFF, Keep will remain ON.

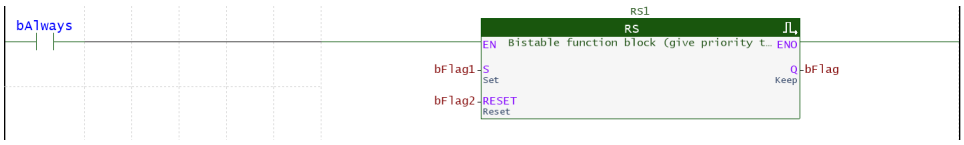
RS	Bistable function block (Reset priority)	The bit state is controlled by the combination of Set/Reset signal. (Reset priority)

Argument



Sample program

The bFlag state is controlled by the state combination of bFlag1 and bFlag2.

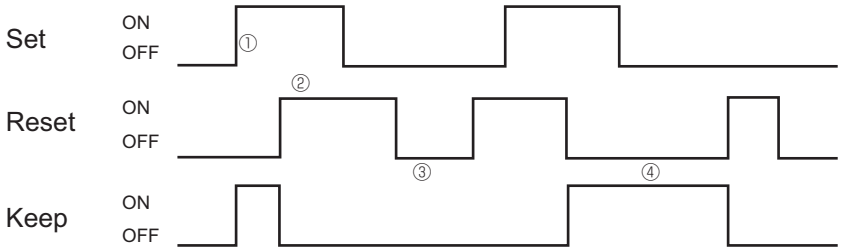


Description of Operation

The "Keep" state is controlled by the bit state combination of bits specified in "Set" and "Reset".

Combination table

		RESET	
		ON	OFF
S	ON	OFF	ON
	OFF	OFF	(Keeps the last state)



① When Set is ON, Reset is OFF, Keep will be ON.

② When Set is ON, Reset is ON, Keep will be OFF.

③ Keep is OFF, and both Set and Reset are OFF, Keep will remain OFF.

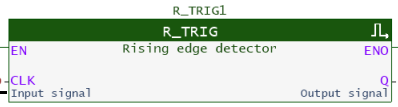
④ Keep is ON, and both Set and Reset are OFF, Keep will remain ON.

R_TRIG Rising edge detector

Turns on by just one scan at the rising edge.

Argument

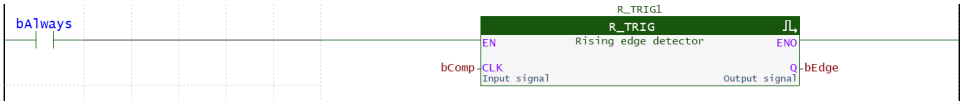
1st argument Input
Specifies the bit to be detected at the rising edge.



2nd argument Output
Specifies the bit that turns on by just one scan at the rising edge.

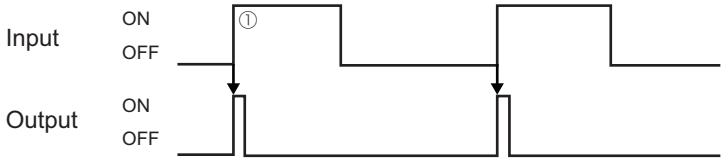
Sample program

Turns bStart on by just one scan at bRdy rising edge.



Description of Operation

At the rising edge of the bit specified in "Input", "Output" turns on by just one scan.



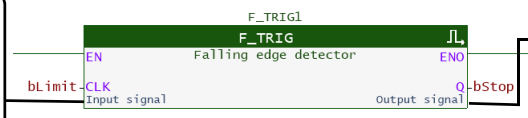
① Output turns on by just one scan at the rising edge of Input.

F_TRIG Falling edge detector

Turns on by just one scan at the falling edge.

Argument

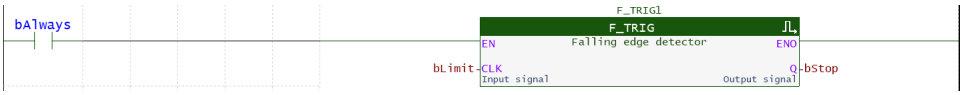
1st argument Input
Specifies the bit to be detected at the falling edge.



2nd argument Output
Specifies the bit that turns on by just one scan at the falling edge.

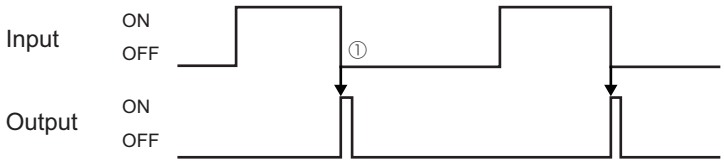
Sample program

Turns bStop on by just one scan at bLimit rising edge.



Description of Operation

At the rising edge of the bit specified in "Input", "Output" turns on by just one scan.



① Output turns on by just one scan at the falling edge of Input.

CTU

Up counter

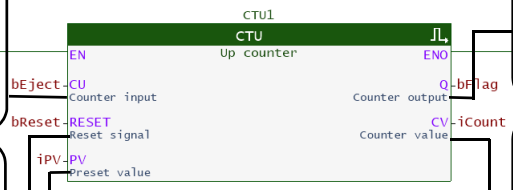
16-bit Up counter.

Argument

1st argument
Counter Input
Specifies the bit to be counted up at the rising edge.

2nd argument Reset
Specifies the bit on whether to reset.

3rd argument Preset Value
Specifies the upper limit that does not be counted up.
(0 to 32767)

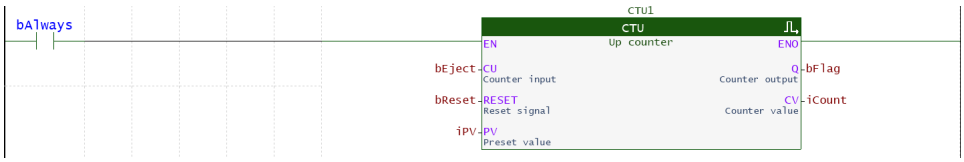


4th argument Output
Specifies the bit to be output when counting up.

5th argument
Counter Value
Specifies the data that stores the Counter Value to be counted up.

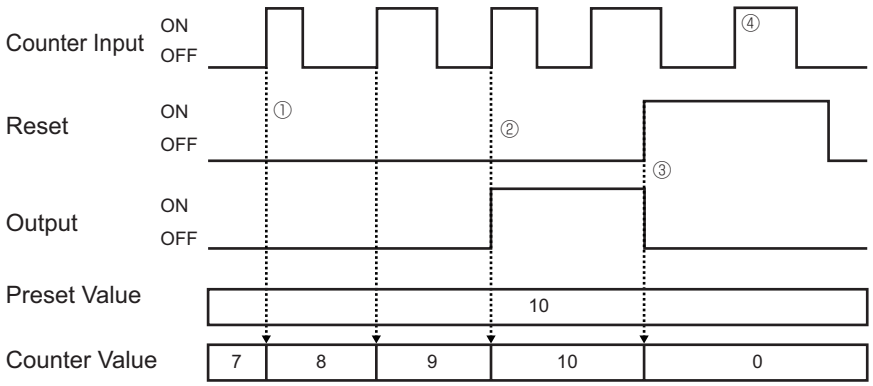
Sample program

Counts up the iCount by the number of bEject rises.



Description of Operation

At the rising edge of the bit specified in "Counter Input", counts up the "Counter Value". When "Counter Value" exceeds "Preset Value", "Output" turns on and "Counter Value" is no longer incremented. While "Reset" is ON, the "Counter Value" becomes 0.



① The Counter Value is incremented at the rising edge of Counter Input.

② When Counter Value exceeds Preset Value, the Output turns on.

③ When Reset turns on, Counter Value becomes 0 and Output also turns off.

④ While Reset is ON, Counter Value is not incremented even if Counter Input is ON.
(If Reset turns off while Counter Input is ON, the rising edge of Counter Input is detected and counted up.)

CTD Down counter

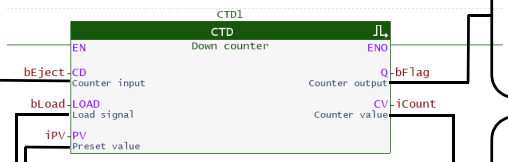
16-bit Down counter.

Argument

1st argument
Counter Input
Specifies the bit to be counted down at the rising edge.

2nd argument Load
Specifies the bit on whether to be loaded to Preset Value.

3rd argument Preset Value
Specifies the initial value to start the countdown.
(0 to 32767)

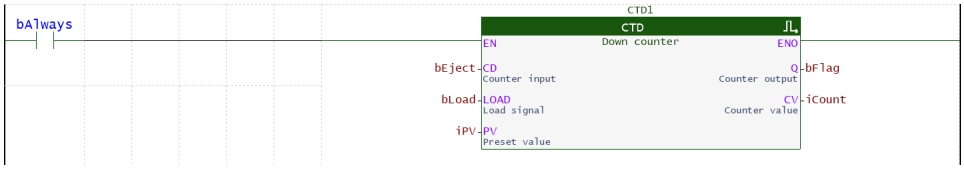


4th argument Output
Specifies the bit to be output when Counter Value is 0.

5th argument
Counter Value
Specifies the data that stores Counter Value to be counted down.

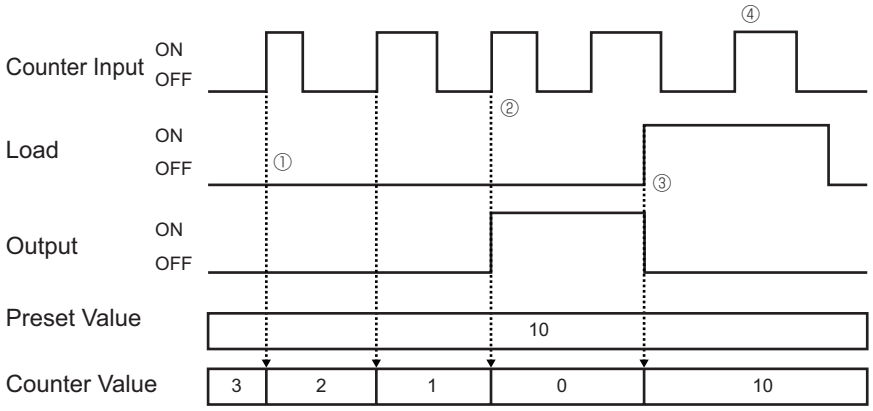
Sample program

Counts down the iCount by the number of bEject rises.



Description of Operation

At the rising edge of the bit specified in "Counter Input", counts down the "Counter Value". The "Counter Value" starts from "Preset Value", and when it reaches 0, "Output" turns on and "Preset Value" is no longer decremented. While "Load" is ON, "Counter Value" becomes "Preset value".



① The Counter Value is decremented at the rising edge of Counter Input.
② When Counter Value reaches 0, Output turns on.
③ When Load turns on, Counter Value becomes Preset Value and Output also turns off.
④ While Load is ON, Counter Value is not decremented even if Counter Input is ON.
④ (If Reset turns off while Counter Input is ON, the rising edge of Counter Input is detected and counted up.)

CTUD Up/Down counter

Refers to the Up/Down counter

Argument

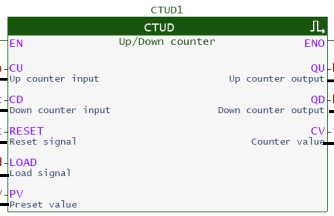
1st argument
Up counter input
Specifies the bit to be counted up at the rising edge.

2nd argument
Down counter input
Specifies the bit to be counted down at the rising edge.

3rd argument Reset signal
Specifies the bit on whether to reset Counter Value.

4th argument Load signal
Specifies the bit on whether to be loaded to Preset Value.

5th argument Preset value
The set Value for countdown, specifies the initial value to start the countdown.
(0 to 32767)



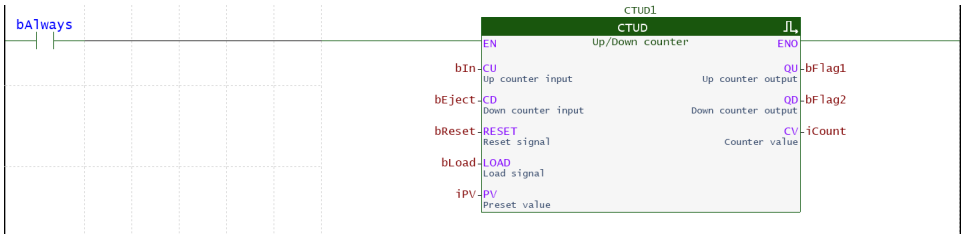
6th argument
Up counter output
Specifies the bit to be output when Counter value exceeds Preset value.

7th argument
Down counter output
Specifies the bit to be output when Counter value is 0.

8th argument Counter value
Specifies the data that stores Counter Value.

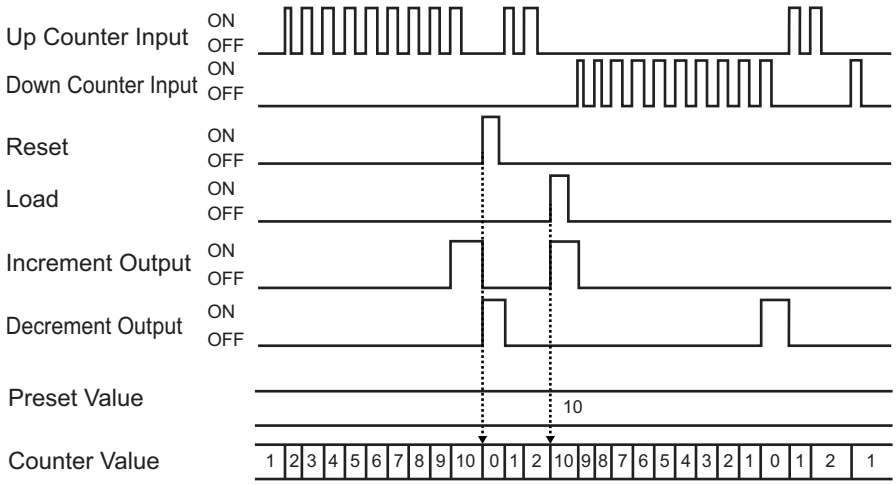
Sample program

Increments iCount by bIn and decrements iCount by bEject.



Description of Operation

The "Counter Value" is counted up at the rising edge of the bit specified by "Up Counter Input", and is counted down at the rising edge of the bit specified by "Down Counter Input". "Counter Value" starts from 0, and when it matches "Preset Value", "Increment Output" turns on; or starts from "Preset Value", and when it reaches 0, "Decrement Output" turns on. While "Increment Output" and "Decrement Output" are ON, "Counter Value" is no longer incremented or decremented. While "Reset" is ON, "Counter Value" becomes "0", and while "Load" is ON, "Counter Value" becomes "Preset Value". If "Reset" and "Load" are ON at the same time, "Counter value" will be "0".

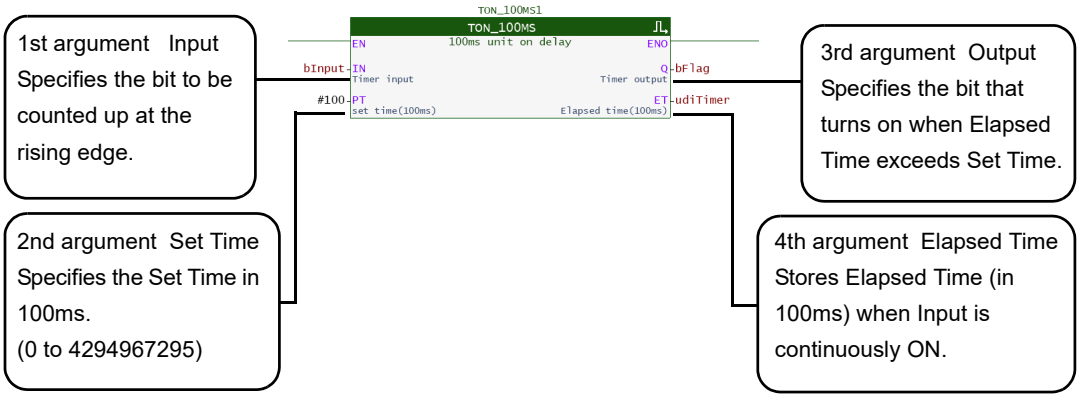


TON_100MS

100ms On-delay

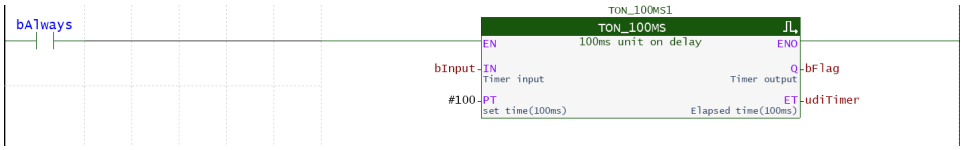
On-delay that in 100ms units

Argument



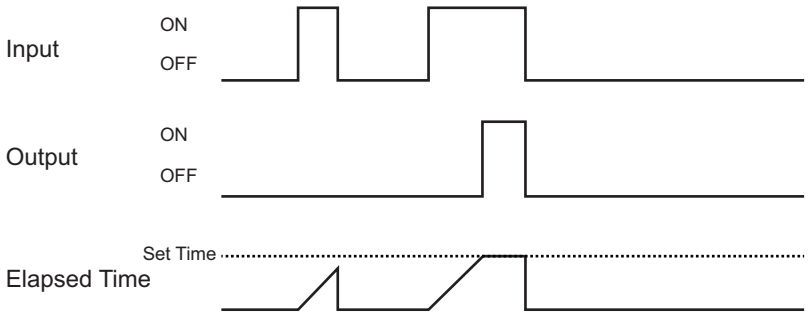
Sample program

Counts the time when bInput is ON.



Description of Operation

The "Eapsed Time" increases while the bit specified in "Input" is ON. When "Set Time" has elapsed, "Output" turns on and the "Elapsed Time" no longer increases. When "Input" turned off, "Elapsed Time" becomes 0 and "Output" also turns off.



Point

If the Execution Condition is skipped by CJ/NCJ instruction, etc., the value of "Elapsed Time" is not updated, but the timekeeping continues during that time. When it is executed next time, the Elapsed Time will be updated to the elapsed time instead of 0. When using it in the IF statement of ST/Script, please note that the Execution Condition may be jumped internally by CJ or NCJ.

TON_10MS

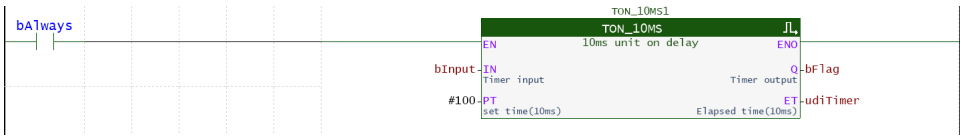
10ms On-delay

On-delay that in 10ms units

Argument

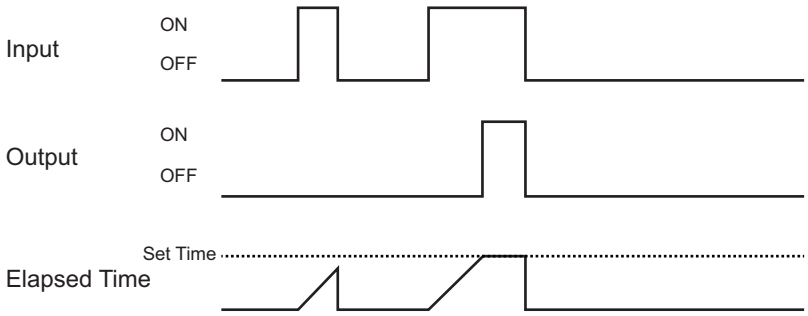
Sample program

Counts the time when bInput is ON.



Description of Operation

The "Eapsed Time" increases while the bit specified in "Input" is ON. When "Set Time" has elapsed, "Output" turns on and the "Elapsed Time" no longer increases. When "Input" turned off, "Elapsed Time" becomes 0 and "Output" also turns off.



Point

- **If the Execution Condition is skipped by CJ/NCJ instruction, etc., the value of "Elapsed Time" is not updated, but the timekeeping continues during that time. When it is executed next time, the Elapsed Time will be updated to the elapsed time instead of 0. When using it in the IF statement of ST/Script, please note that the Execution Condition may be jumped internally by CJ or NCJ.**
- **Do not change the "Set Time" while the Function Block is being executed. The "Elapsed Time" will also be offset.**

TON_1MS

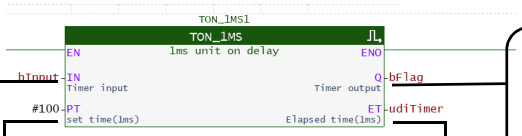
1ms On-delay

On-delay that in 1ms units

Argument

1st argument Input
Specifies the bit to be counted up at the rising edge.

2nd argument Set Time
Specifies the Set Time in 1ms.
(0 to 4294967295)

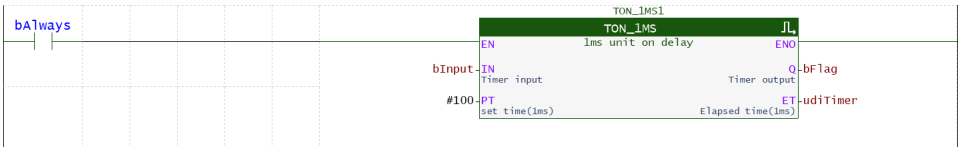


4th argument Output
Specifies the bit that turns on when Elapsed Time exceeds Set Time.

4th argument Elapsed Time
Stores Elapsed Time (in 1ms) when Input is continuously ON.

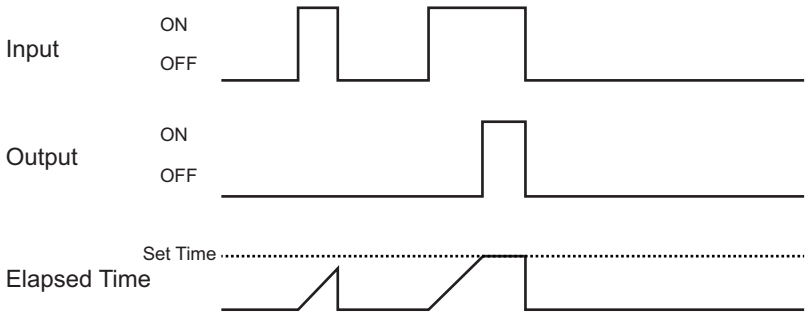
Sample program

Counts the time when bInput is ON.



Description of Operation

The "Eapsed Time" increases while the bit specified in "Input" is ON. When "Set Time" has elapsed, "Output" turns on and the "Elapsed Time" no longer increases. When "Input" turned off, "Elapsed Time" becomes 0 and "Output" also turns off.



Point

- **If the Execution Condition is skipped by CJ/NCJ instruction, etc., the value of "Elapsed Time" is not updated, but the timekeeping continues during that time. When it is executed next time, the Elapsed Time will be updated to the elapsed time instead of 0. When using it in the IF statement of ST/Script, please note that the Execution Condition may be jumped internally by CJ or NCJ.**
- **Do not change the "Set Time" while the Function Block is being executed. The "Elapsed Time" will also be offset.**

TON_10US

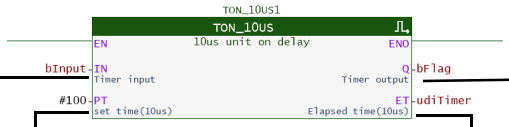
10µs On-delay

On-delay that in 10µs units

Argument

1st argument Input
Specifies the bit to be counted up at the rising edge.

2nd argument Set Time
Specifies the Set Time in 10µs.
(0 to 4294967295)

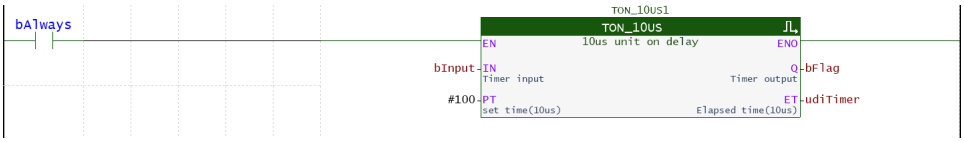


4th argument Output
Specifies the bit that turns on when Elapsed Time exceeds Set Time.

4th argument Elapsed Time
Stores Elapsed Time (in 10µs) when Input is continuously ON.

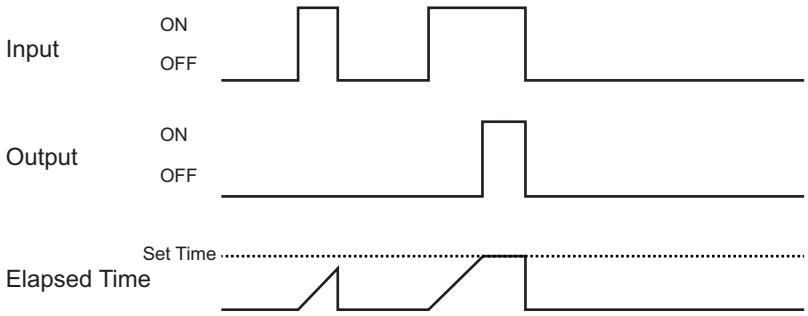
Sample program

Counts the time when bInput is ON.



Description of Operation

The "Elapsed Time" increases while the bit specified in "Input" is ON. When "Set Time" has elapsed, "Output" turns on and the "Elapsed Time" no longer increases. When "Input" turned off, "Elapsed Time" becomes 0 and "Output" also turns off.



Point

- **If the Execution Condition is skipped by CJ/NCJ instruction, etc., the value of "Elapsed Time" is not updated, but the timekeeping continues during that time. When it is executed next time, the Elapsed Time will be updated to the elapsed time instead of 0. When using it in the IF statement of ST/Script, please note that the Execution Condition may be jumped internally by CJ or NCJ.**
- **Do not change the "Set Time" while the Function Block is being executed. The "Elapsed Time" will also be offset.**

Differences between Function Blocks and Macros

Like macros, function blocks are a function that collect ladder circuits like a single ladder instruction.
This section compares function block operation to macro operation.
For details on macros, see "Modules" (page 4-98)

Object creation

[Function block]

Program type: Function block

Entities are generated by the number of instances.

Program type: Function

One object is generated, regardless of the number of times it is called in the program.

[Macro]

An object is generated every time it is called in the program.

Callable position

[Function block]

Program type: Function block

Can be called from modules and function blocks.

Program type: Function

Can be called from modules, function blocks, and functions.

Nests of up to 8 layers can be created.

"Function Block Nests" (page 4-140)

[Macro]

Can be called from modules. Macros cannot be called from macros.

Argument type

[Function block]

IN: The device value is stored in the argument when execution starts.
OUT: The argument value is stored in the device when execution ends.
IN-OUT: The device reference is stored when execution starts.

[Macro]

P: The device reference is stored when execution starts.
V: The device value is stored in the argument when execution starts.

Types of variables that can be specified as arguments

The following variables can be specified only for function blocks, not for macros.

- IN, OUT arguments: fixed-length arrays, structures, fixed-length strings
- IN-OUT arguments: fixed-length arrays, variable-length arrays, structures, fixed-length strings, Variable-length string

Reference

When creating a new program with the KV-8000 series, it is recommended to use function block instead of macro.

● Other Differences

Item		Function block	Macro
Program	Program name	Up to 64 characters	Up to 12 characters
	Read only setting	Yes	—
Argument	Number	IN, OUT, IN-OUT: Up to 32 each Unit: Up to 8 Total: Up to 64	Up to 10
	Argument name	IN, OUT, IN-OUT: Up to 128 characters long Unit: Local devices UR0 - 7, UM0 - 7, UV0 - 7	Local devices P0 - 9, V0 - 9, UR0 - 9, UM0 - 9, UV0 - 9
	Set initial value	Yes	—
	Set to private	Yes	—
	Argument omitting	Yes	—
System device	@CR2008	Yes	Yes
	@CR2007	Yes	Yes
	Bit turned ON during execution	—	Yes
Others*1	Instance	Yes	—
	EN/ENO	Yes	—
	Return value	Yes	—

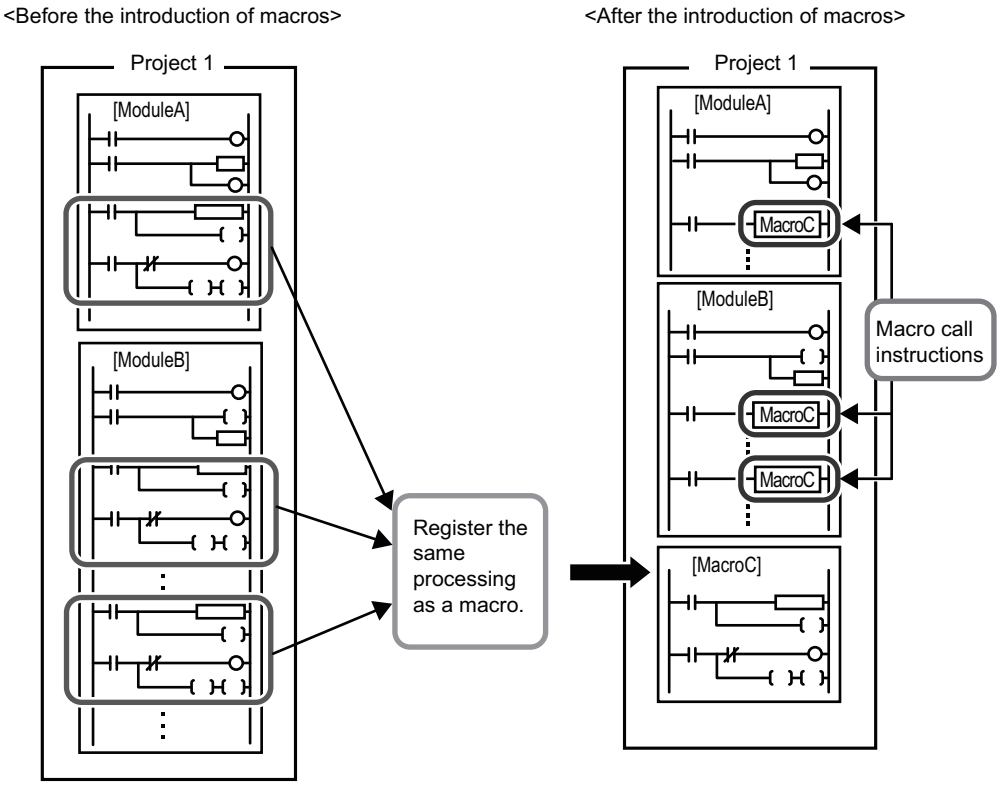
*1 Only supported by CPU function version 2.0 or later.

MEMO

4-11 Macros

Overview

A macro is a section of a ladder program that has been registered with a name so that it can be used repeatedly in a project. By creating macros, you can use the sections of ladder programs in the same way that ladder instructions are used, which enables you to reduce the amount of time you spend writing programs. When you make changes to the program, just by modifying the data in the location that is registered as a macro, all the process details executed when the corresponding macro instruction is called in the project are changed. This reduces the amount of time that you spend changing programs and also prevents mistakes that occur during program modifications.



■ Macro management

In KV STUDIO, the workspace is used to manage macros for each project.

■ Macro features

● Macros can be used as original instructions

If you register a ladder program segment that occurs multiple times in the program as a macro, you can reduce the amount of time you spend writing programs and you can also reduce input mistakes.
As a result, the number of rungs in the ladder program decreases, which makes it easier to understand the program structure.
You can assign any name to the macro that you want to register.

● You can use arguments

You can use arguments to create highly general macros because you can specify the operands of instructions in the macro (devices and values) at the time that the macro is called.

"Setting Arguments" (page 4-186)

● You can create two types of macros

There are two types of macros: subroutine macros and self-hold macros.
Subroutine macros are executed only when their execution condition is turned ON, the same as subroutine programs.
Once self-hold macros are executed, they will continue to operate until the macro execution stop instruction (MEND) is executed, regardless of the status of the execution condition.
This is useful in situations such as when you want to request processing from expansion units or peripherals and the processing will take time to complete (when the processing will take multiple scans).

"Macro Type" (page 4-181)

● You can use local devices

Even if you use a local device with the same number in a separate macro, these devices will be processed as different devices, so you can easily assign devices for usage only within the macro.

"4-12 Local Devices" (page 4-200)

● Password protection is available

To prevent the contents of macros from being changed easily, you can set passwords.
When you set a password, people who do not know the password cannot display or edit the macro program.
Use the "Program properties" dialog box to set the password.

● Objects are created when the macro is called

Ladder program objects are created each time that the macro is called from a module. When you want to execute the same macro repeatedly in the same scan or when there are multiple macros that must each be executed across multiple scans, by using macros together with local devices, you can use macros without having to consider conflicts between the devices that you are using and without having to consider the execution status of the different macros.

"Differences between Macros and Subroutines" (page 4-195)
"Differences between Function Blocks and Macros" (page 4-175)

Usage Procedure

The macro usage procedure is shown below.

(1) Create a new macro.

Click "Programs" and then "New macro."

Select macro as the program type.

"Macro Type" (page 4-181)

▼

(2) Create the laddrer program.

Write the ladder program in the created macro.

▼

(3) Set the arguments.

Right click in the workspace and select "Argument settings (Z)"

Register the arguments.

"Setting Arguments" (page 4-186)

▼

(4) Write the macro call instructions.

Use the dedicated instructions to call the created macro.

Use system devices if necessary.

"Macro instructions" (page 4-192)
"Macro System Devices" (page 4-188)

Macro Type

There are two types of macros: subroutine macros and self-hold macros.

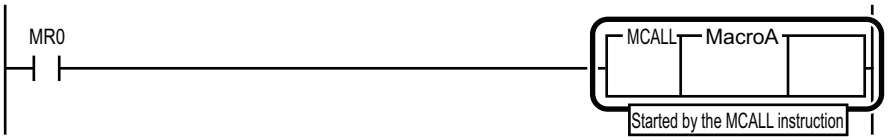
Subroutine macros

● Overview

Subroutine macros operate the same as subroutines in that the macro may only be executed when its execution condition is turned ON. The program within the macro is executed each time that the macro call instruction (MCALL) is executed. After the macro finishes executing, execution proceeds to the next step after the macro call instruction. If the execution condition is turned OFF, the program within the macro is not executed, and execution proceeds to the next step.

● Usage method

To start a subroutine macro, use the MCALL instruction from a module.

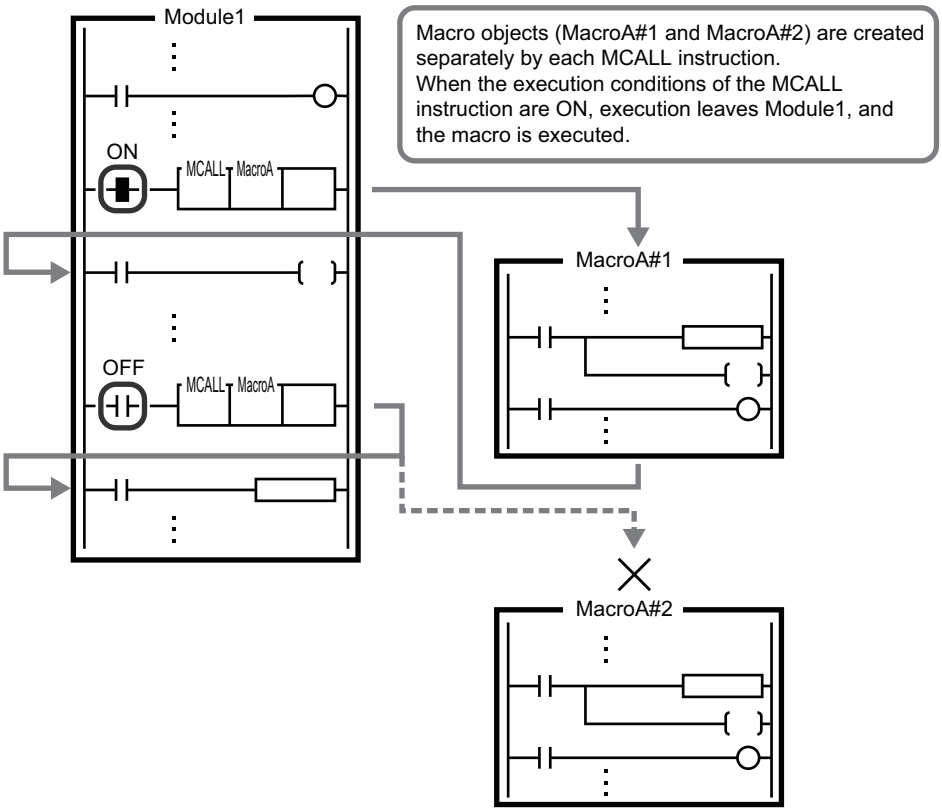


● Operation description

An explanation of the operation of subroutine macros is shown in the following figure.
In this example, macro A is called twice from within module 1.

Reference

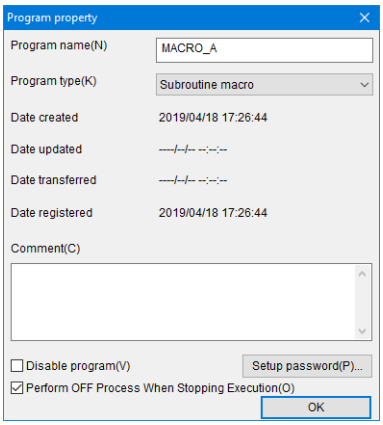
To differentiate between the objects of the created macro, "#(management number)" is appended after the macro name.
In KV STUDIO, this is only displayed during monitoring and simulation.



● Operations when execution stops

The operation when subroutine macros stop executing (when the execution condition turns from ON to OFF) differs according to the "Program property" -> "Perform OFF Process When Stopping Execution" setting of each program.

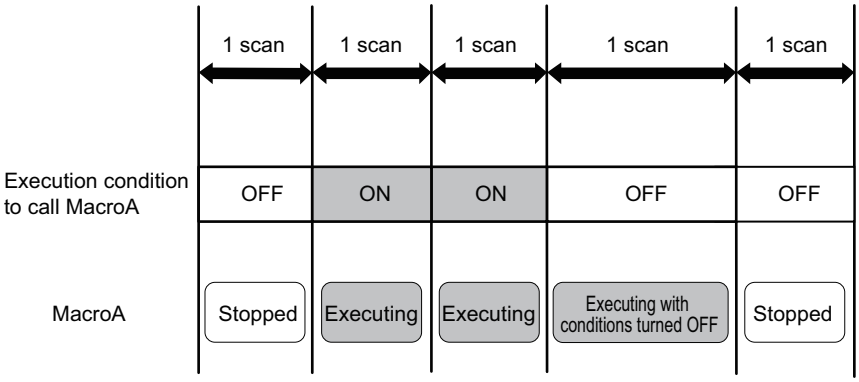
Point **In the case of KV-8000A, the setting is not displayed, and the operation is the same as when "OFF processing when execution stopped" item is always checked.**



If "Perform OFF Process When Stopping Execution" is not checked
Subroutine macros turn OFF in the scan in which the execution condition turned OFF.

If "Perform OFF Process When Stopping Execution" is checked
In only the one scan in which execution conditions are turned OFF, subroutine macros are executed with all execution conditions turned OFF.*
Operations are stopped in the following scan.

[Image of the operation of a subroutine macro]



* The following instructions are exceptions.

Instruction	Status
OUB	Turned OFF.
OFDL	Reset.
SHOT	
C	The status is held.
CJ, NCJ, and GOTO	Execution does not jump.

Reference For OFF processing when macro execution stops, the same processing is performed as when the MC instruction execution condition is OFF for one scan only.

For details, see the MC and MCR instructions in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series Instruction Reference Manual.

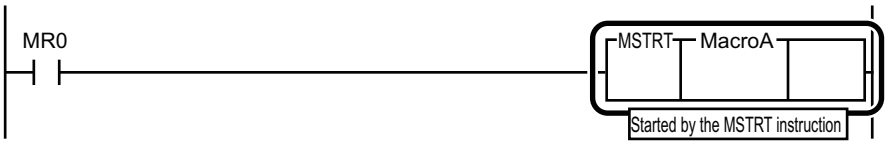
Self-hold macros

● Overview

Self-hold macros operate the same as subroutine macros in that the macro may only be executed when its execution condition is turned ON. The program within the macro is executed each time that the macro call instruction (MSTRT) is executed. Once the macro has started, it is continually executed during subsequent scans, regardless of the execution condition of the macro call instruction, until the macro execution stop instruction (MEND) is called from within the macro.

● Usage method

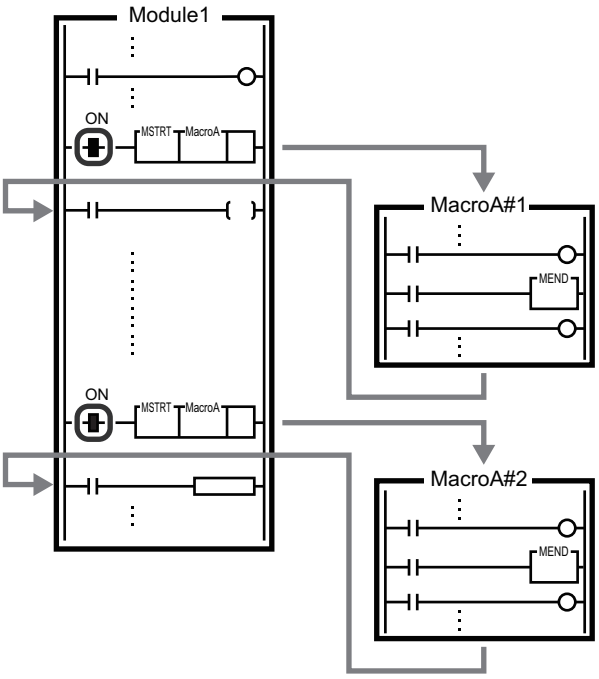
To start a self-hold macro, use the MSTRT instruction from a module.



● Operation description

An explanation of the operation of self-hold macros is shown in the following figure.
In this example, macro A is called twice from within module 1.

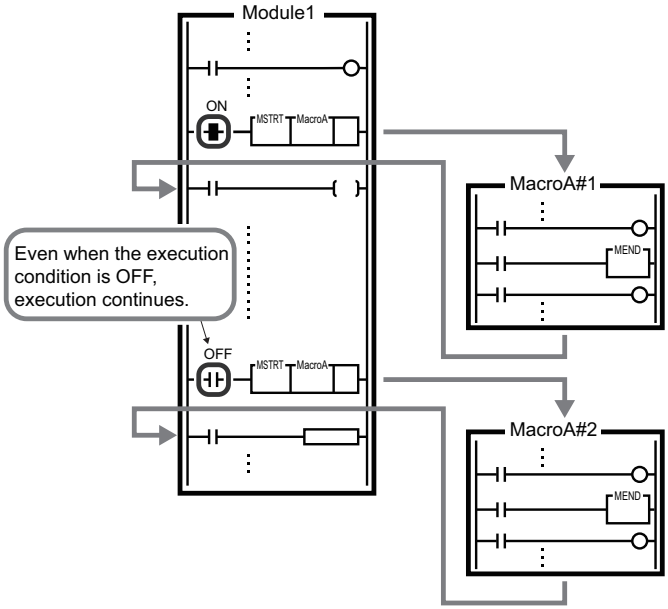
1 When the execution condition is turned ON, macro A starts.



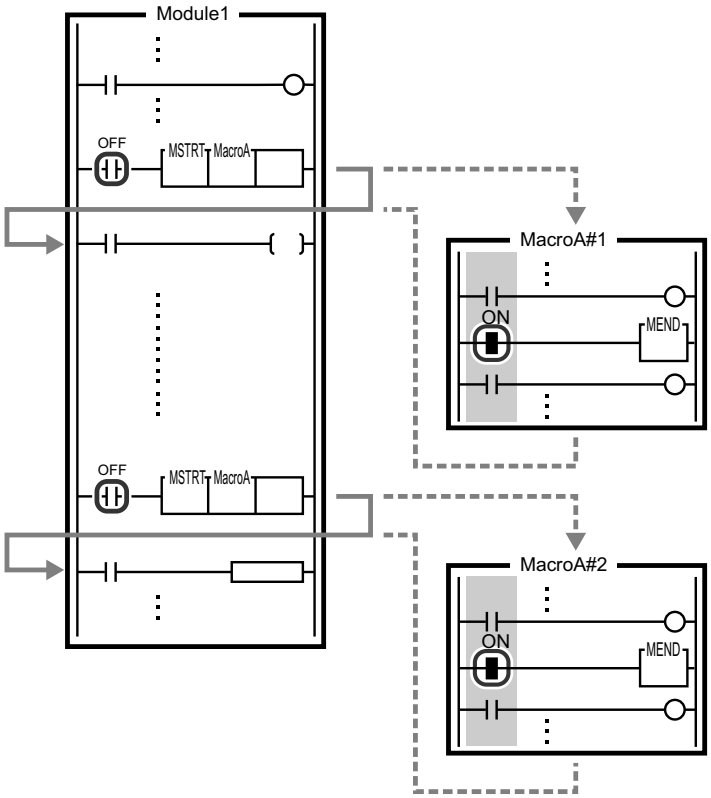
Reference

To differentiate between the objects of the created macro, "#(management number)" is appended after the macro name.
In KV STUDIO, this is only displayed during monitoring and simulation.

2 **Once the macro is started, it continues to be executed regardless of the status of the MSTRT instruction's execution condition.**



3 **It stops when the execution condition of the MEND instruction in the macro turns ON.**



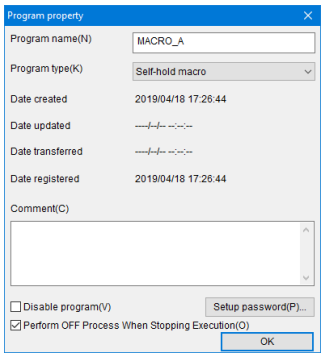
Point

- **You must include the MEND instruction in self-hold macros.**
- **To stop the execution of a macro, be sure to turn its execution conditions OFF. While the execution conditions of a macro are turned ON, the macro will not stop even if the execution conditions of the MEND instruction are turned ON.**

● Operations when execution stops

The operation when self-hold type macros stop executing (when an MEND instruction is executed) differs according to the "Program property" -> "Perform OFF Process When Stopping Execution" setting of each program.

Point **In the case of KV-8000A, the setting is not displayed, and the operation is the same as when "OFF processing when execution stopped" item is always checked.**



If "Perform OFF Process When Stopping Execution" is not checked

Self-hold type macros stop on the scan after the macro stop instruction (MEND) is executed.

If "Perform OFF Process When Stopping Execution" is checked

In only the one scan after the macro stop instruction (MEND) was executed, self-hold macros are executed with all execution conditions turned OFF. *
They will be stopped from the following scan.

[Image of the operation of a self-hold macro]

	1 scan	1 scan	1 scan	1 scan	1 scan	1 scan
Execution condition to call Macro A	OFF	ON	*1	*1	OFF*2	OFF*3
Execution condition of the MEND instruction within Macro A	-	OFF	OFF	ON	*1	-
Macro A	Stopped	Executing	Executing	Executing	Executing with conditions turned OFF	Stopped

*1 This being ON or OFF has no effect on operations.
*2 Be sure to turn this OFF. Until this is turned OFF, the MEND instruction will be ignored, and execution will continue.
*3 If this has been turned ON, the macro will start executing.

* The following instructions are exceptions.

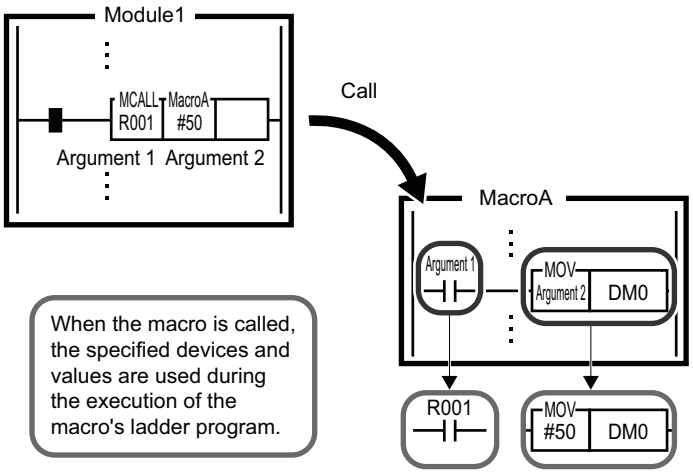
Instruction	Status
OUB	Turned OFF.
OFDL	Reset.
SHOT	
C	The status is held.
CJ, NCJ, and GOTO	Execution does not jump.

Reference For OFF processing when macro execution stops, the same processing is performed as when the MC instruction execution condition is OFF for one scan only.
For details, see the MC and MCR instructions in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series Instruction Reference Manual.

Setting Arguments

■ What are arguments?

Arguments are devices and values that are passed to the macro object when the macro is called. They are specified as the operands of macro call instructions. In the macro, the passed devices and values are used to execute the ladder program. If a macro is being executed, its arguments are updated each scan.



■ List of argument devices

An argument device is a device that is used to specify what instruction inside the macro the passed devices and values will be used as the operands of. You can only use argument devices inside macros. Right click the macro you wish to set arguments for in the workspace, and select "Argument settings (Z)" from the menu, then set them in the "Macro argument settings" dialog box. There are five types of argument devices: P, V, UR, UM, and UV. The specifications of each device are shown below. You can specify up to 10 macro arguments.

Type	Symbol	Range	Description	Indirect Specification (*)	Local Device (@)	Index Modification (:)
Device	P	P0 to P9	The device is received as the argument.	No	No	Yes
Value*2	V	V0 to V9	The device's value or constant is received as the argument.	No	No	No
Unit device (R)*1	UR	UR0 to UR9	The leading relay (R) assigned to the expansion unit is received as the argument.	No	No	Yes
Unit device (DM)*1	UM	UM0 to UM9	The leading data memory (DM) entry assigned to the expansion unit is received as the argument.	No	No	Yes
Unit number*1	UV	UV0 to UV9	The unit number of the expansion unit is received as the argument.	No	No	No

*1 In macro call instructions, only the target unit number is specified.
You can specify up to five units as arguments in a single macro.
*2 If you specify DR, direct processing is performed immediately before the macro is called, and the result is received as the argument.

● Usage example

A usage example of argument devices is shown in the following figure.
In this example, the macro call instruction of macro 1 is executed from module 1.

Argument device usage example

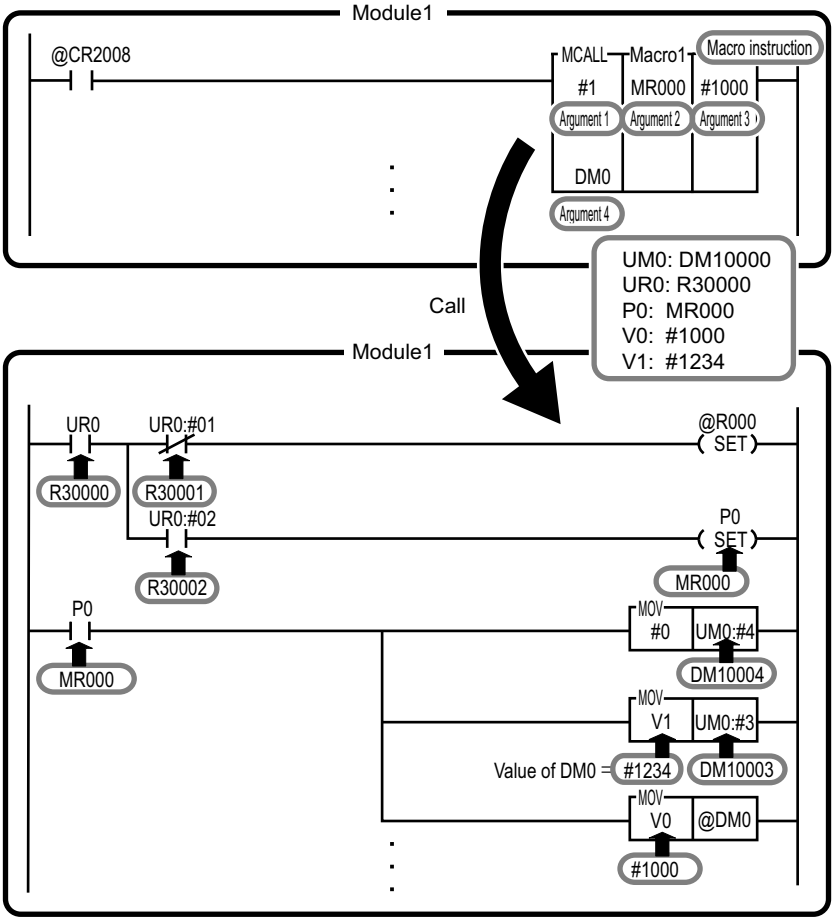
Argument 1: UR0/UM0 ← #1

Argument 2: P0 ← MR000

Argument 3: V0 ← #1000

Argument 4: V1 ← DM0

Unit number 1 ... KV-SAD04 (R30000～R30115, DM10000～DM10059)



Point

If you are using character strings as arguments, use argument device "P", and specify the leading device in which the character string is stored. You cannot use argument device "V" to designate character strings directly.

Macro System Devices

A macro system device is a device that is used to capture the operation status of a macro.
Only the bit that is turned ON during execution can be referenced by other programs.

Point **You cannot attach device comments to macro system devices.**

Reference KV STUDIO automatically assigns macro system devices to the CPU unit internal work area.

● Bit that is turned ON for only one scan when macro execution starts, "@CR2008"

This bit is turned ON for only the first scan after macro execution starts.
From the second scan, it is turned OFF.
This bit can only be referenced from its own macro. It cannot be referenced from the module that called this macro.
In KV STUDIO, this device is represented by "@CR2008."

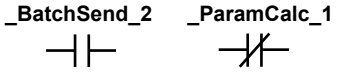
● Bit that is turned OFF for only one scan when macro execution starts, "@CR2007"

This bit is turned OFF for only the first scan after macro execution starts.
From the second scan, it is turned ON.
This bit can only be referenced from its own macro. It cannot be referenced from the module that called this macro.
In KV STUDIO, this device is represented by "@CR2007."

● Bit that is turned ON during execution, "_(macro name)_(ID number)"

This bit is turned ON when the macro is being executed.
Because this bit can be referenced from all modules, it is useful in situations such as when you want to reference the bit before calling the macro to control macro execution.
In KV STUDIO, this device is represented by "_(macro name)_(ID number)."

Example



Reference To use this bit that is turned ON during macro execution, you need an ID number for each macro that you want to call.
"Bit That Is Turned ON during Macro Execution" (page 4-189)

Bit That Is Turned ON during Macro Execution

■ Overview

This bit is turned ON when the macro is being executed. Because this bit can be referenced from all modules, it is useful in situations such as when you want to reference the bit before calling the macro to control macro execution.
To use this bit that is turned ON during macro execution, you need an ID number for each macro that you want to call.
In KV STUDIO, this device is represented by "_(macro name)_(ID number)."

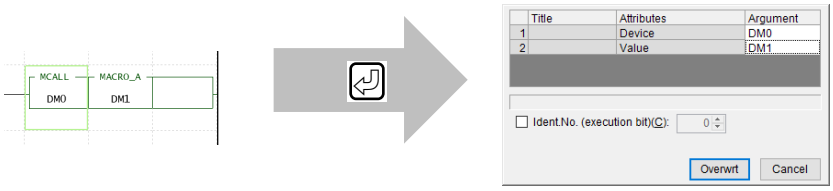
■ How to set the bit that Is turned ON during execution

The bit that is turned ON during execution can be used when " Ident.No. (execution bit)" is checked in the "Operand" dialog box.

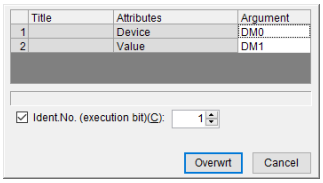
1 Move the cursor to the MCALL/MSTRT instruction and press the ⏎ key.

The "Operand" dialog box appears.

Alternative procedure Move the cursor to the MCALL/MSTRT instruction, and select "Edit (E)" -> "Change operand (O)" from the menu.



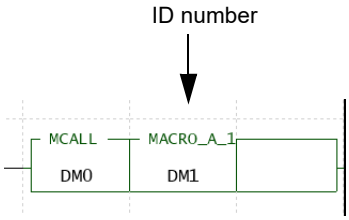
2 Check the "Ident.No. (execution bit)" checkbox, and enter an ID number.



Point **The bit that is turned ON during macro execution cannot be used if the "Ident.No. (execution bit)" checkbox is not checked.**

3 Click the "Overwrite" button.

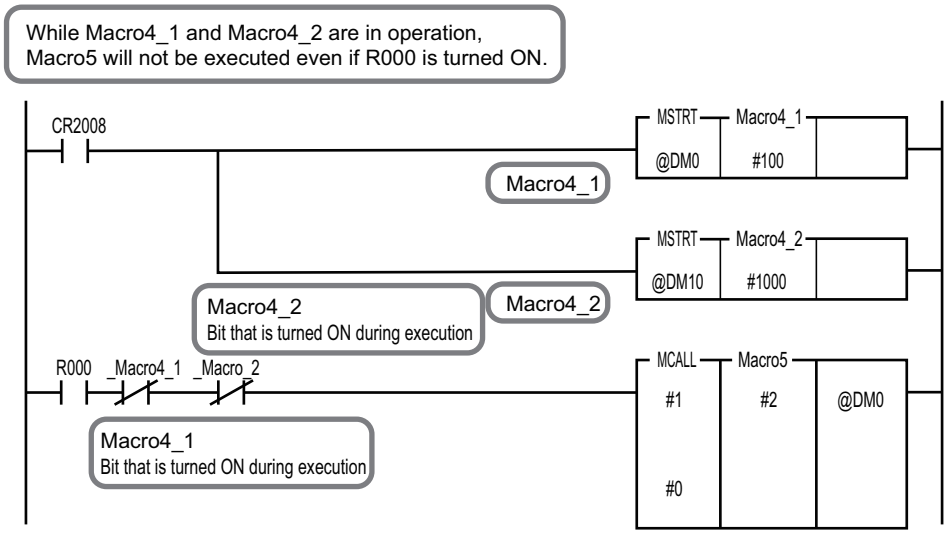
The ID No. (execution bit) is set.



To use the same macro repeatedly in the same project, change the number each time that you want to use the macro.

Point **If you specify the same number for different instances of the same macro, a conversion error will occur.**

● Usage example

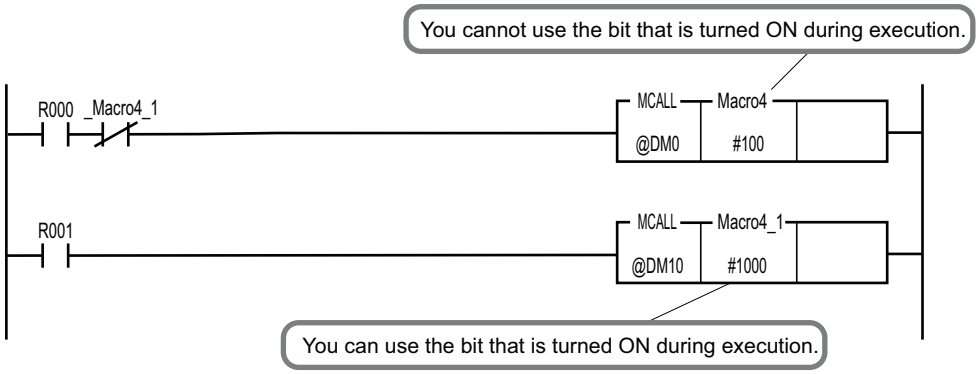


Point

If you write the bit that is turned ON during execution of an unused macro (a macro that is not specified by a macro call instruction in the project), a conversion error will occur.

Reference

You can use macro call instructions that call a macro with an ID number together with macro call instructions that call the same macro without an ID number.



Macro Passwords

■ Macro passwords

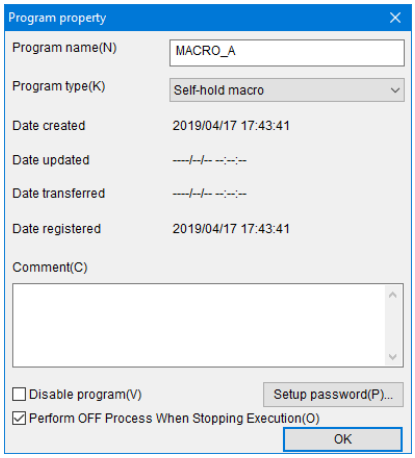
You can limit who can display macros in KV STUDIO by setting passwords on the macros.
Follow the procedure below to set a password.

1 Select the macro to set a password for in the workspace.

2 Right-click, and then select "Property" in the menu that is displayed.

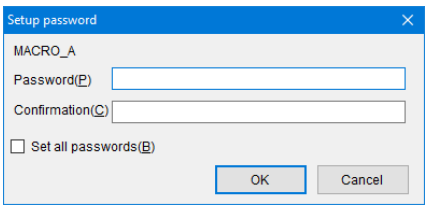
The "Program property" dialog box is displayed.

Alternative procedure
- Select the macro in the workspace, and select "Program" -> "Properties" from the menu.
- Select and right click the macro in the workspace and select "Setup password" from the menu.



3 Click "Setup password."

The "Setup password" dialog box is displayed.

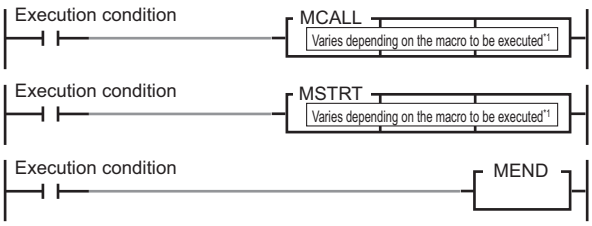


4 Enter the password in both text boxes (the second is for confirmation), and then click "OK."

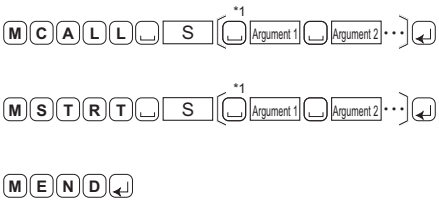
After you set the password, when an attempt is made to display the macro in KV STUDIO, a dialog box will be displayed asking the user to enter the password.
If the user does not enter the password, they will not be able to display the macro in KV STUDIO.

MCALL	MCALL	Macro call	Executes the specified subroutine macro (written within the module)
MSTRT	MSTRT	Macro start	Executes the specified self-hold macro (written within the module)
MEND	MEND	Macro end	Stops execution of the self-hold macro (written within the self-hold macro)

Ladder program



Entry mode



Operation	Explanation	Occupied size
S	Specifies the subroutine No.*3*4	—

*1 The number of operands other than the macro name and the data that can be specified differ as follows depending on the argument settings set in the macro to be executed.

- P : Devices, variables and labels can be specified. Constants cannot be specified.
- V : Devices, variables, labels and constants can be specified.
- Unit : Only constants can be specified.

*2 Handling of T, C, TIMER type variables and COUNTER type variables differs depending on the instruction used in the macro.
Example) For the MOV instruction, the current value is handled. For details, refer to the operation description of the respective instruction.

*3 "", # and $ cannot be used. Directly enter the name of the macro set in KV STUDIO.
Up to 12 half-width alphanumerics and 6 full-width alphanumerics can be specified.
However, up to 12 full-width alphanumerics can also be specified in the KV-8000/7000 series.

*4 When an ID No. is used, the format becomes "(macro name)_(ID No.)". Example) Batch transmit_2
For details on ID No., see the User's Manual of the CPU unit used.

Description of Operation

MCALL The sub-routine macro specified by [S] is executed at every scan when the execution condition is ON.
Execution of the specified macro is started from scans where the execution conditionchanges from OFF to ON.

MSTRT The self-hold type macro specified by [S] is executed at every scan when the execution condition is ON.
Execution of the specified macro is started from scans where the execution conditionchanges from OFF to ON.
Once execution is started, the macro continues to be executed even if the executioncondition is OFF.

MEND Self-hold type macros with a MEND instruction written to them stop when the executioncondition is ON.

The OFF process when stopped differs according to the CPU unit. Refer to the instruction manual foreach CPU unit for details.

Subroutine macros: "Operations when execution stops" (page 4-182)
Self-hold macros : "Operations when execution stops" (page 4-185)

Point

- **Writing in the RUN mode is not possible if the MCALL/MSTRT instruction is added or deleted.**
- **To stop execution of a self-hold macro, use the MEND instruction in the macro.**
  When the MSTRT instruction execution condition holds ON, the MEND instruction is ignored.
- **The MEND instruction cannot be programmed in modules and sub-routine macros.**
- **Other macros cannot be stopped by the MEND instruction.**
- **The previous state of differential execution type instructions in macros is initialized every time that execution of a macro is started. (rising edge differentiation: ON, falling edge differentiation: OFF)**
- **When T/C/CTH has been specified to an operand whose MCALL/MSTRT instruction attribute is a value, the following values are handed over to the macro:**
  T/C : Current value (32-bit data)
  CTH : Current value (32-bit data)
  CTC : Setting value (32-bit data)
- **In the following cases the macro being executed stops when each module stops or when the execution condition for each set of instructions turns OFF.**
  - **When an MCALL/MSTRT instruction is written to the module**
  - **When an MCALL/MSTRT instruction is written to master control (in the MC – MCR instructions)**
  - **When an MCALL instruction is written to stage (in the STG - JMP instructions)**
- **In the following cases the macro being executed is canceled when the execution condition for each set of instructions turns OFF.**
  The macro execution resumes when each set of instructions is executed.
  The previous state of differential execution type instructions in macros is not initialized at this time.
  (If it is written in the initialization module only, 1 scan is executed when operation starts.)
  - **When an MCALL/MSTRT instruction is written to sub-routine (in the SBN – RET instructions)**
  - **When an MCALL/MSTRT instruction is written to conditional branch (in the CJ/ NCJ - LABEL instructions)**
  - **When an MCALL/MSTRT instruction is written to step (in the STP – STE instructions)**
  - **When an MCALL/MSTRT instruction is written to interrupt (in the INT – RETI instructions)**
  - **When an MSTRT instruction is written to stage (in the STG - JMP instructions)**
- **MCALL and MSTRT instructions can not be used in macros.**
- **MCALL/MSTRT instructions cannot be used in function blocks/functions.**
- **Sub-routines can be used in macros, but interrupt programs can not be used.**
- **When performing a certain macro repeatedly in a project, the global devices / global variables used in OUT/OUB instruction in a macro will become double coil.**

Operation flag

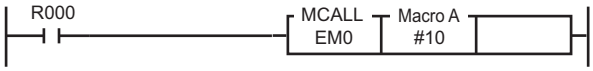
CR2009	The state depends on the processing result in the macro to execute.
CR2010	The state depends on the processing result in the macro to execute.
CR2011	The state depends on the processing result in the macro to execute.
CR2012	ON when the indirect specification or index modification range is inappropriate, the unit specified by UR/UV/UM while using the KV-8000/7000 series is not connected, or an access outside the operand range occurs. Otherwise itis OFF. It will not change when indirect specification or index modification is not specified in the operand, and the access outside the operand range is not checked.

* The macro is not executed when the argument is inappropriate.
When CR2012 is ON, error details are stored in CM5150 to CM5176.
"Error log" (page 4-86)

Point **The state of the operation flag does not change by the MEND instruction.**

Sample Program

The sub-routine macro (macro A) is executed with EM0 as argument 1 and #10 as argument 2 wheninput relay R000 is ON.



(Mnemonics list)

```
LD R000
MCALL Macro A EM0 #10
```

At the rising edge of input relay R001, execution of self-hold macro "Macro B" is started with #1 asargument 1 (unit No.) and #10 as argument 2.

(Mnemonics list)

```
LDP R001
MSTRT Macro B #1 #10
```

When input relay R002 is ON, execution of self-hold macro "Macro B" is stopped.

(Mnemonics list)

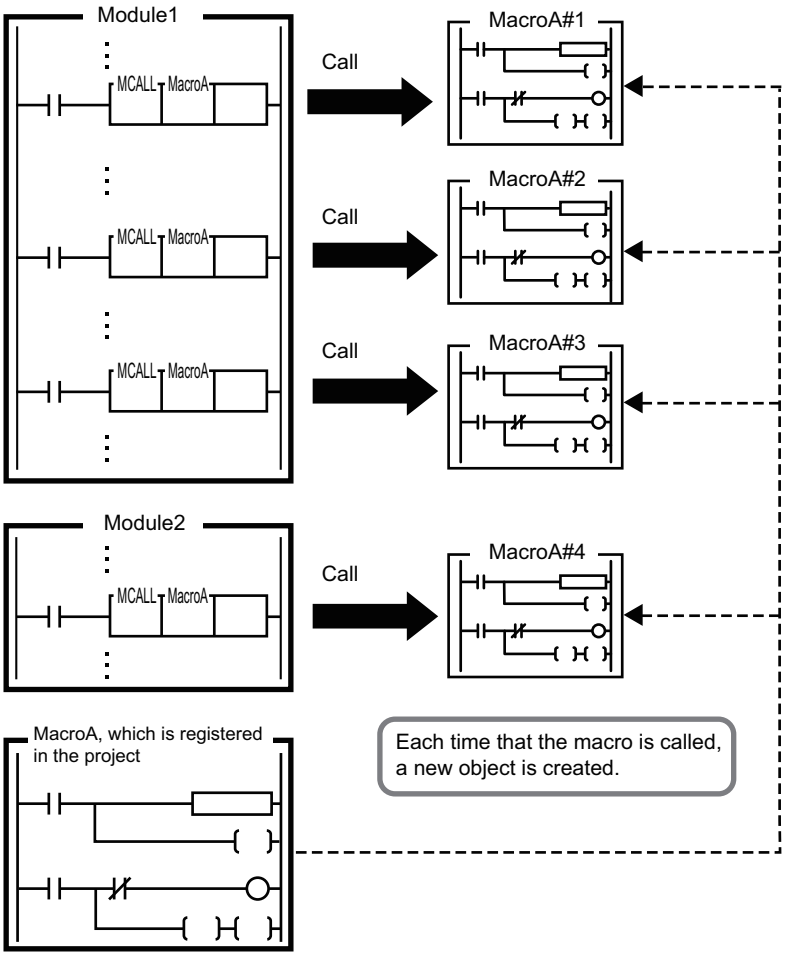
```
LD R002
MEND
```

Differences between Macros and Subroutines

The same as subroutine programs, call instructions are used to start macros. In this section we will show a comparison of the macro and subroutine operations.
For details on subroutine programs, see "Subroutine programs" (page 4-94).

● Object creation

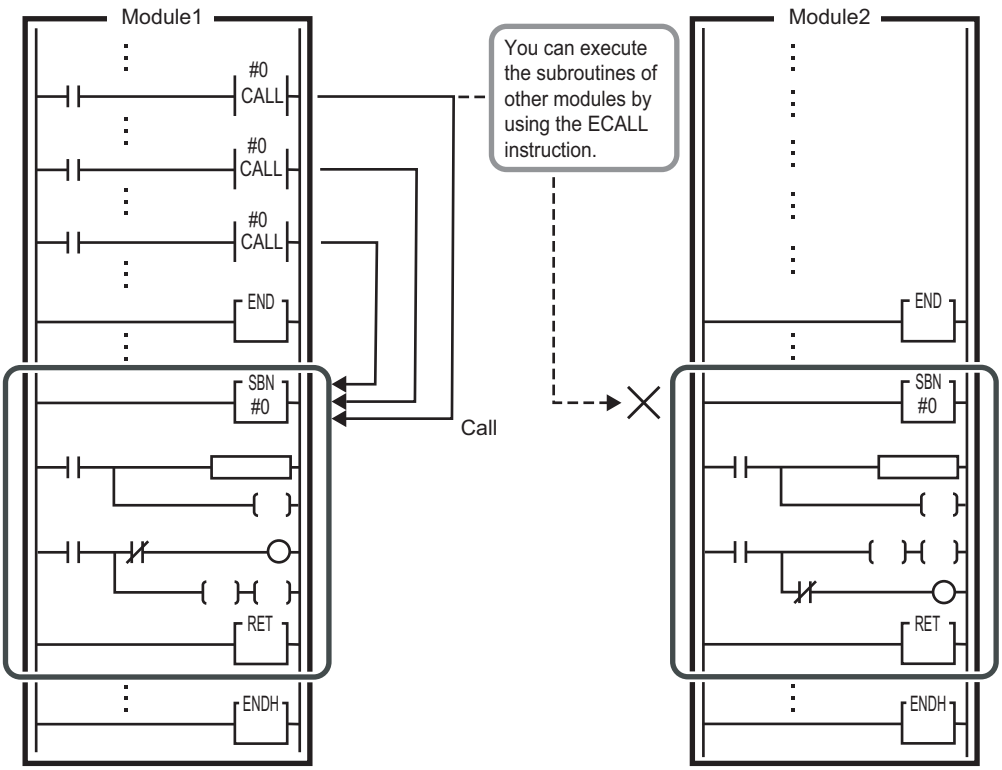
[Macro]
The number of objects that will be created is equal to the total number of macro call instructions (MCALL or MSTRT) that you write in a module.



Reference

- Macros that are registered in the project but that are not called (unused macros) are not transferred to the CPU unit. Unused macros are not included in the targets to be verified during project verification and during verification with the PLC.
- To differentiate between the objects of the created macro, "#(management number)" is appended after the macro name. In KV STUDIO, this is only displayed during monitoring and simulation.

[Subroutine]
Subroutines are written between the END and ENDH instructions of a program, and they are executed by the subroutine call instruction (CALL) that is written in the main routine program of the program. Objects are not created each time that a call instruction is executed.



● Callable position

[Macro]
In a module, you can write macro call instructions anywhere in the main routine, in subroutines, and in interrupt routines. You cannot write macro call instructions in macros.

[Subroutine]
Subroutine call instructions are only valid in the program in which the subroutine is created. To execute subroutines in other modules, use the inter-module subroutine call instruction.

● Name

[Macro]
You can freely assign names to macros.
"List of Characters That Cannot Be Used" (page A-39)

[Subroutine]
Numbers are used to manage subroutines.

● Arguments

[Macro]
Macros have arguments.

[Subroutine]
Subroutines do not have arguments.

● Operation when the execution condition is turned OFF

[Macro]
If "Perform OFF Process When Stopping Execution" is not checked, the program in the macro is not executed. Each device in which it is written maintains its status from the last time it was executed.
If "Perform OFF Process When Stopping Execution" is checked, when stopped all programs are executed with the execution condition OFF in the next scan only. For example, the devices written with the OUT instruction are automatically turned OFF.

"Macro Type" (page 4-181)

Point **In the case of KV-8000A, the setting is not displayed, and the operation is the same as when "OFF processing when execution stopped" item is always checked.**

[Subroutine]
The program in the subroutine is not executed. The written devices retain their status from the last time the subroutine was executed.

● Default status of differential execution instructions

[Macro]
The previous status that is used to detect differentials with instructions such as differential execution instructions is turned ON when a rising edge differential occurs at the start of execution and is turned OFF when a falling edge differential occurs at the start of execution. Differentials will not be detected immediately after execution starts.

[Subroutine]
The same as macros, except that the operation is only performed when execution starts for the first time. When execution starts for the second time and later, the status is set to that of the previous execution.

● Operation when the program is called between the MC (master control) and MCR (master control reset) instructions

• When the MC instruction's execution condition is turned ON
Macros and subroutines operate normally.

• When the MC instruction's execution condition is turned OFF
[Macro]
Stops in the same way as in "Operation when the execution condition is turned OFF" above. (Self-hold macros also stop.)

[Subroutine]
Subroutines are not executed.

● Applications

When you want to retain the status and values of the devices that you are using in the locations that programs are called

[Macro] .....Better

Because separate objects are created for each call instruction, if you use macros together with local devices, you can retain the status from the previous execution without having to consider problems such as device conflicts.

[Subroutine]....Good

Because the same program is executed, you have to use indirect specification or a similar method to create a program that stores and restores the status of the devices that you are using.

When you want to perform execution until a series of operations finishes

[Macro] .....Better

If you use self-hold macros, after the macro starts executing, it will continue to operate regardless of the status of the execution condition until the macro execution stop instruction (MEND) is executed.
Also, when you want to execute multiple instances of this type of processing at the same time, you can do so without having to consider device conflicts and the various operation statuses.

[Subroutine]....Good

You have to write the program so that the execution condition is not turned OFF until the operations are finished.
Also, when you want to execute multiple instances of this type of processing at the same time, you have to write the program while considering device conflicts and the various operation statuses.

When you want to reduce the number of steps

[Macro] .....Good

Because objects are created for each call instruction, you cannot reduce the number of steps.

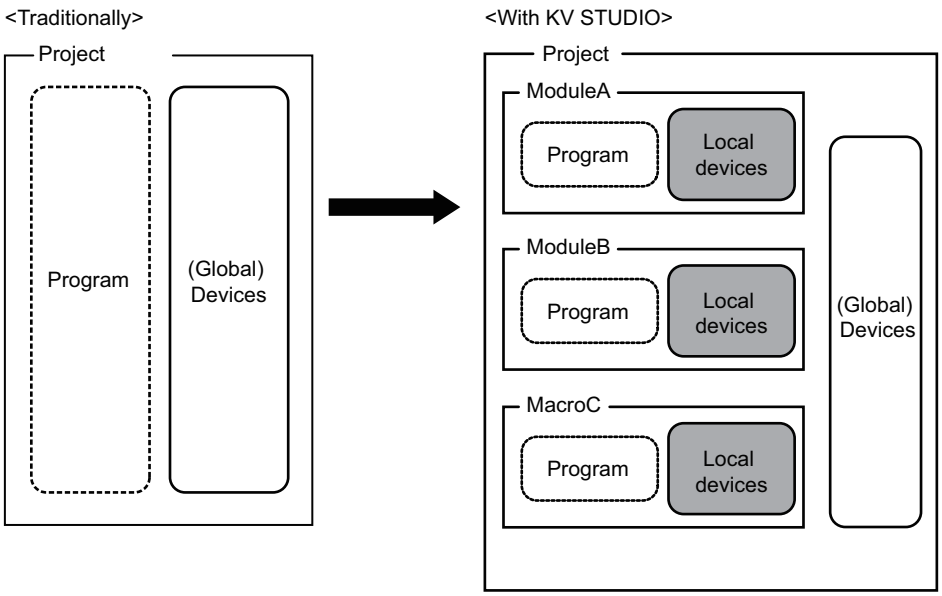
[Subroutine]....Better

Because the number of objects created remains the same no matter how many times the subroutine is called, the more times the subroutine is called, the more you can reduce the number of steps.

MEMO

4-12 Local Devices

Traditionally, the devices that were processed by the PLC were common to the entire project. Therefore, it was necessary for the user to manage all devices used in the program.
In KV STUDIO, for each program, local devices can be processed as separate devices. Even if you use a local device with the same number in a different program, because it will be processed as a separate device, you can easily manage the assignments of devices that are used only within the module or macro.
In contrast with the local devices that are valid only within a program, devices that are shared within the entire project are called global devices.



How to Write Local Devices

To specify a local device, prefix the device name with "@."

Example

Global devices	Local devices
EM0	**@EM0**
MR0	**@MR0**

Point

You have to write this as "@device number" only when you want to specify the device number as the operand of a timer or counter instruction.
Mnemonic input example: TMR @0 #100

List of Local Devices

The number of local and global devices that are reserved when you create a project and the default ranges of these devices are shown below. To change the reserved number of local devices, from the workspace, open the "CPU system setting" dialog box, and then click "Entire assignment setting of local devices."

Device Name*1	Local Device	Global Device	Number of points
	The number of reserved local devices as a ratio of all projects*2	Number, range	
Relay (R)*3	None	32000 points (2000 ch) R000 to R199915	32000 points (2000ch)
Internal auxiliary relay (MR)*3	32000 points（2000ch）	32000 points（2000ch） MR000 to MR199915	64000 points (4000 ch)
Latch relay (LR)*3	8000 points (500 ch)	8000 points (500 ch) LR000 to LR49915	16000 points (1000ch)
Timer (T)*4	1500 points	2500 points T0 to T2499	4000 points
Counter (C)*4	1500 points	2500 points C0 to C2499	4000 points
Data memory (DM)	25000 points	40535 points DM0 to DM40534	65535 points
Extended data memory (EM)	45000 points	20535 points EM0 to EM20534	65535 points
Temporary data memory (TM)	None	512 points TM0 to TM511	512 points

*1 The link relay (B), link register (W), file register (ZF), control relay (CR), control memory (CM) and index register (Z) cannot be used as local devices.
When @FMxxx is used, the work area is used as a local device.

*2 When new projects are created, collectively reserve a device area that can be used as a local device. (Calculated based on the assumption that 10 modules are used and that a macro is called 50 times.) The number of devices that can be reserved can be changed as required.

*3 Use channel (ch) units to set the use quantity.

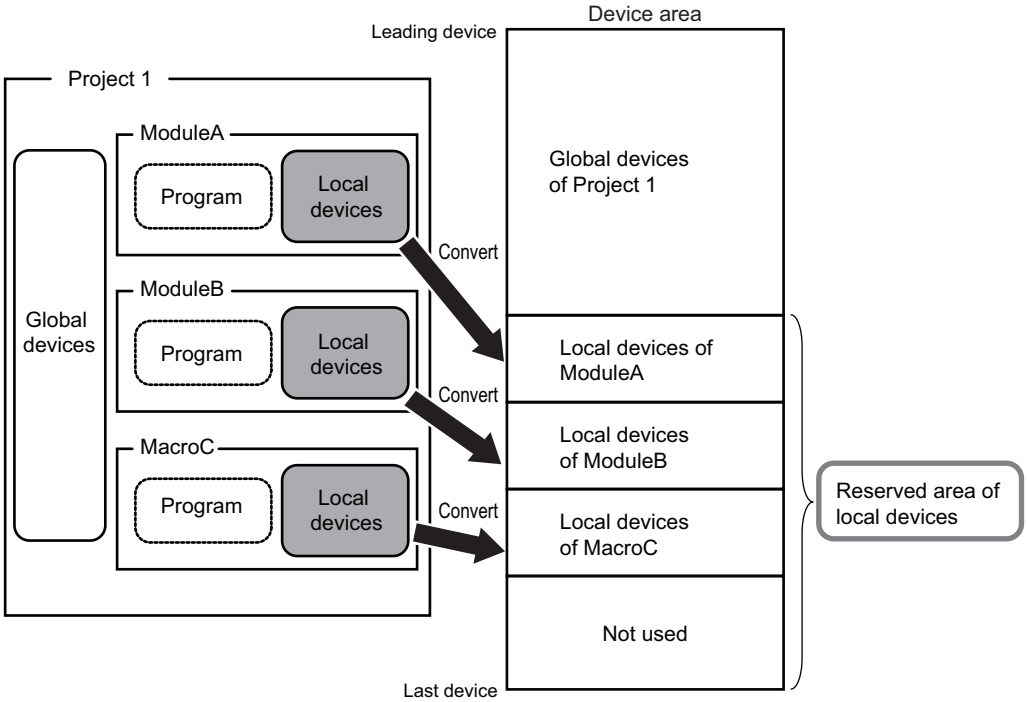
*4 Enter "@+(device no.)" only when a device number will be used as the operand of a timer counter instruction. (Example for entering mnemonic:TMR @0 #100).

Point

- **If you are using relays (R) and temporary memory entries (TM) as local devices, you have to set the number of devices that you will use.**
- **The following devices must be processed as global devices.**
  - **The device that is allocated to CPU unit and expansion unit**
  - **Devices that are referenced by connected peripherals such as touch panels**
  - **Devices that are common to the entire project**
  - **Devices that have to transfer data between modules**
- **The first local device is number 0. Use local devices with as small numbers as possible.**
- **Because R and DM devices are assigned to units, an area that is used as global devices is required. For local devices, use MR and EM devices with a higher priority.**
- **If you use a local device or global device outside of the usable range, a conversion error will occur.**
- **When you are using instructions in which you specify the leading device in the operands, index modification, and indirect specification, do not exceed the usable range of local and global devices.**

Local Device Assignment

During conversion, local devices are automatically assigned to the reserved areas that have been prepared for the different device types in advance.



Point

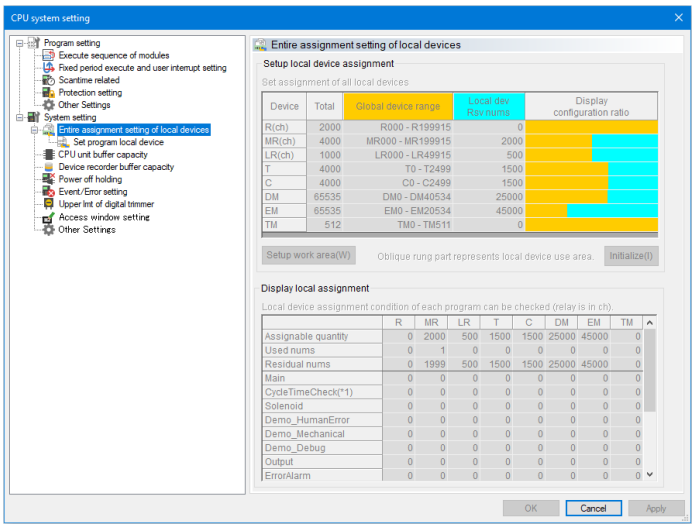
- **For function blocks and macros, a number of local devices equal to the number of times it is called in the project is assigned.**
- **For function type function blocks, only one local device is assigned regardless of the number of times it is called, to make sure there is always only one object.**

Reference

Just like global devices, you can assign device comments to local devices.

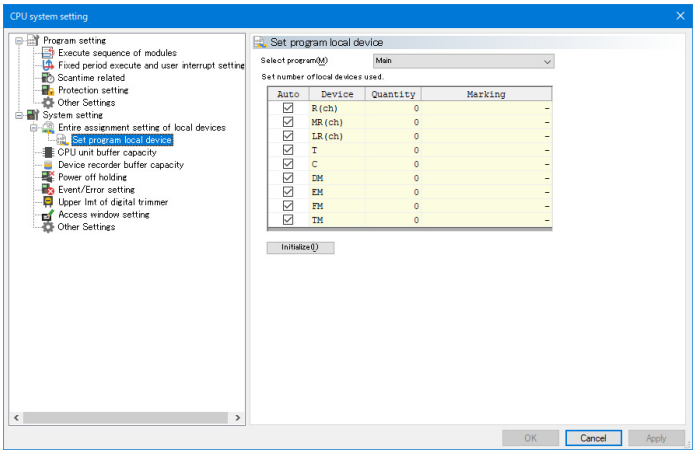
■ Setting assignments in batches

The number of local devices that are reserved for usage in a project is set in advance, but you can change the number of devices used per type of device. The changes that you make are saved in KV STUDIO, so there is no need to make the same changes each time that you create a new project.
From the workspace, open the "CPU system setting" dialog box, and then click "Entire assignment setting of local devices."



■ Configuring device settings for programs

You can set the number of reserved local devices for usage in each program, and you can make this setting for each type of device. The default setting is "Auto." With this setting, the number of local devices used in a program is set automatically, so there is normally no need to consider the number of local devices.
From the workspace, open the "CPU system setting" dialog box, and then click "Set program type device."



Local Device Input and Errors during Conversion

● During ladder entry

A typical example of an error occurring during ladder entry is shown below.

An attempt is made to use a device whose type cannot be used as a local device.

B, W, ZF, CR, CM, and Z cannot be used as local devices.

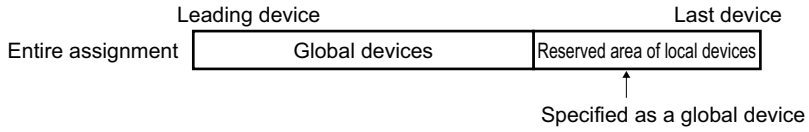
"List of Local Devices" (page 4-201)

● During ladder conversion

Typical examples of errors occuring during ladder conversion are shown below.

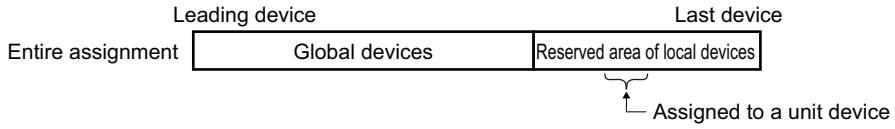
An attempt is made to use as a global device the device area that has been reserved for local devices.

•Check the "Global device range" settings on the "Entire assignment setting of local devices" screen.



"Setting assignments in batches" (page 4-203)

•There is a chance that the unit devices you have used the Unit Editor to set are using the local device area. Check whether this is the case.



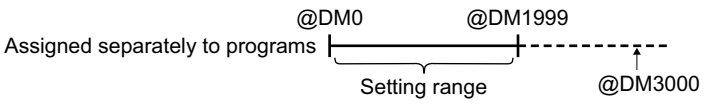
See "UNIT EDITOR" in the "KV STUDIO User's Manual."

An attempt is made to use a local device that has been assigned outside of the range for local devices.

This error can only occur when you configure settings manually.

Check the settings on the "Set program type device" screen.

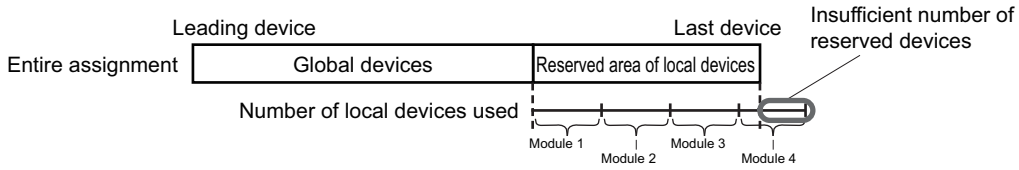
Example



"Configuring device settings for programs" (page 4-203)

The reserved number of local devices is insufficient.

Check the settings under "Display local assignment" on the "Entire assignment setting of local devices" screen. See if there are devices whose settings, such as the number of devices used, are displayed in red. Configure the settings—by increasing the reserved number of applicable devices or by decreasing the number of local devices set for each program—so that the number of devices that can be used (the number of reserved local devices) is greater than the number of devices used.

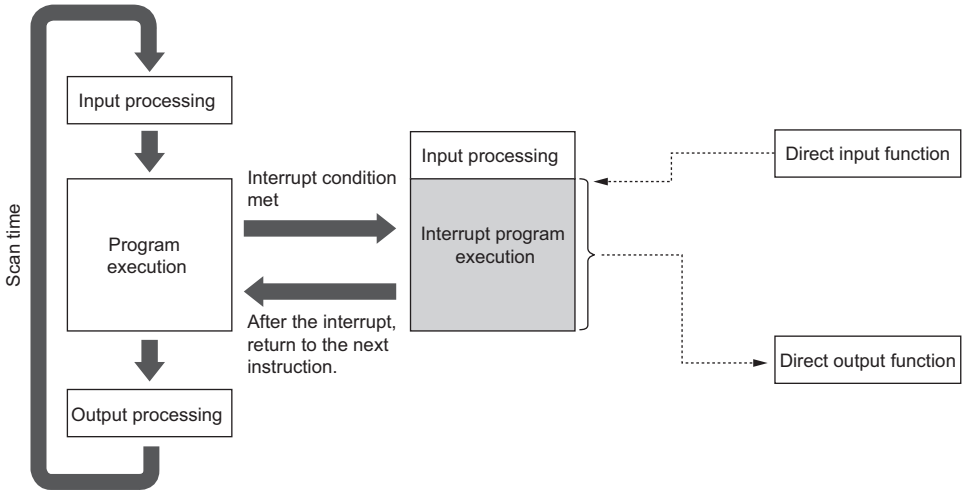


"Setting assignments in batches" (page 4-203)

4-13 Interrupts

Overview

Normal PLCs execute ladder sequences by repeating the following cycle: input processing, program execution, and then output processing. Therefore, signals shorter than the scan time cannot be applied. If you use the interrupt function, you can execute processing at the point in time that an interrupt occurs regardless of the scan time.
When an interrupt occurs, even if a program is being executed, the execution is temporarily interrupted, and the interrupt program that corresponds to the interrupt condition is executed. After the interrupt program is completed, the processing of the interrupted program begins again.



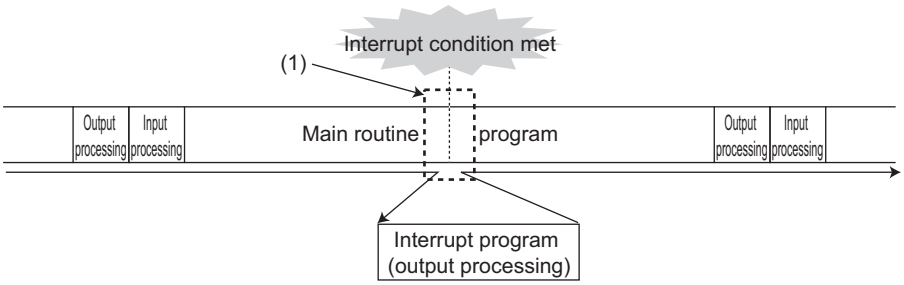
■ Types of interrupt conditions

The interruption sources can be set by the programs, are the followings: interruption from the expansion unit (max 64 items), fixed period modules (max 4 items), inter-unit synchronization module.

- "Fixed-Period Module" (page 4-103)

■ Device processing

Device processing when an interrupt occurs is shown below.
For details on I/O processing, see "I/O Processing during Interrupts" (page 4-212).



(1) Processing when the interrupt program starts and finishes.

When an interrupt program starts, the values of the internal registers, the operation flags, TM0 to TM3, and Z11 and Z23 are recorded in the system. When the interrupt program finishes, the recorded values are restored.
Even if these values are changed during interrupt processing, the main routine program will not be affected. Changed values are only valid within the interrupt program.
For other devices, if the values were changed during the interrupt program, these new values take effect in the main routine program after the interrupt program finishes.

Point

You can use the "User interrupt" screen on the "CPU system setting" dialog box to specify whether to save and restore index registers Z1 to Z10.

Reference

To update to the latest status device values that are not automatically refreshed:
- Use "DR" to write an I/O relay.
- Write the RFSX or RFSY instruction.
  "Direct processing" (page 4-83)

■ Processing time

When it suffices all of the following conditions, the time until the start of interruption will be 15μs (TYP.4μs) or shorter.
- When selecting the "Enable interrupt in instruction execution" checkbox in "Fixed period execute and user interrupt" setting of CPU System Settings"
- It does not use interruption program with higher priority than that interruption program
- It does not use the expansion unit for KV-5000/3000 series (maximum +14μs when it is used)
- When many devices are not refreshed by the inter-unit synchronization input or output refresh
- 1 ID is not registered with a device of 513 words or more during the recording/tracking setting.

In addition, the activation, finishing process time (refresh/ saving/recovery of each device) of interruption program is 2.5μs in total.

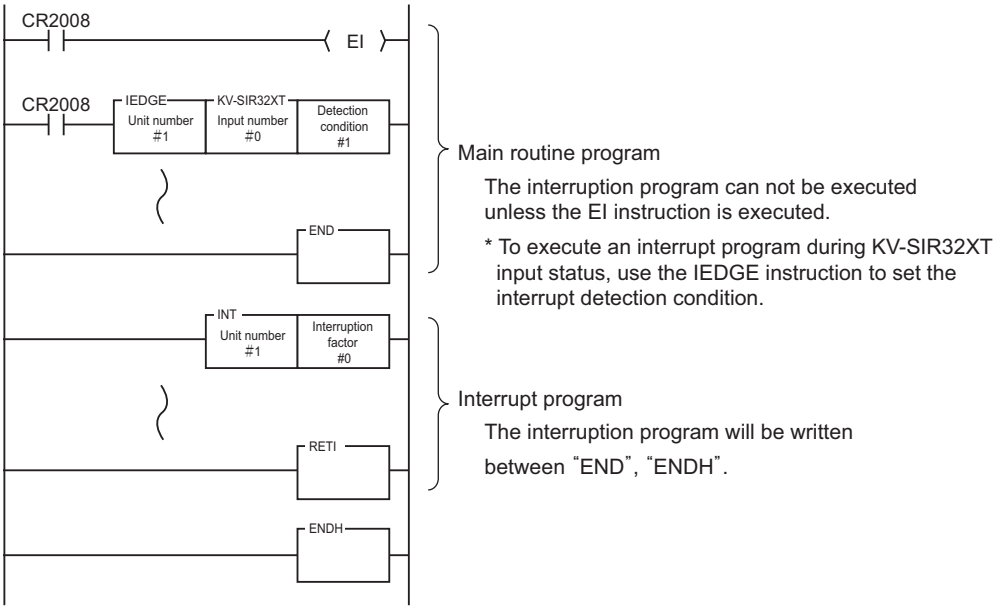
Reference

The time until processing of the interrupt starts is as stated above, but if a direct processing instruction is executed for an expansion unit for the KV-5000/3000 Series at the start of the interrupt program, it may take some time for the interrupt program to complete.
- When KL-N20V, KV-CL20, KV-DN20, KV-H20, KV-H20S and KV-H40S are used: 115 μs
- When an expansion unit for units other than the KV-5000/3000 Series above are used: 55 μs.

How to Execute Interrupt Programs

To make interrupt processing possible, write your programs in the following way.

Interrupt program example



■ Interrupt enable (EI) instruction

When operation starts, interrupts are disabled.
To execute interrupt programs, you have to execute the interrupt enable (EI) instruction.
Write the EI instruction in the main routine program.

Reference

- To disable interrupts, use the interrupt disable (DI or DIC) instructions.
- When interrupts are disabled, even if an interrupt occurs, the interrupt program will not be executed until interrupts are enabled.
- The interrupt program starts when interrupts are enabled.
- Even if you have not executed the interrupt enable (EI) instruction, system interrupts caused by peripheral processing are executed.

"EI Instructions" (page 4-218)
"DI Instructions" (page 4-218) and "DIC Instructions" (page 4-220)

■ INT and RETI instructions

Between the END and ENDH instructions, write the INT instruction with operands to set the interrupt condition.
After the INT instruction, program the interrupt processing details, and then finally write the RETI instruction.

Point

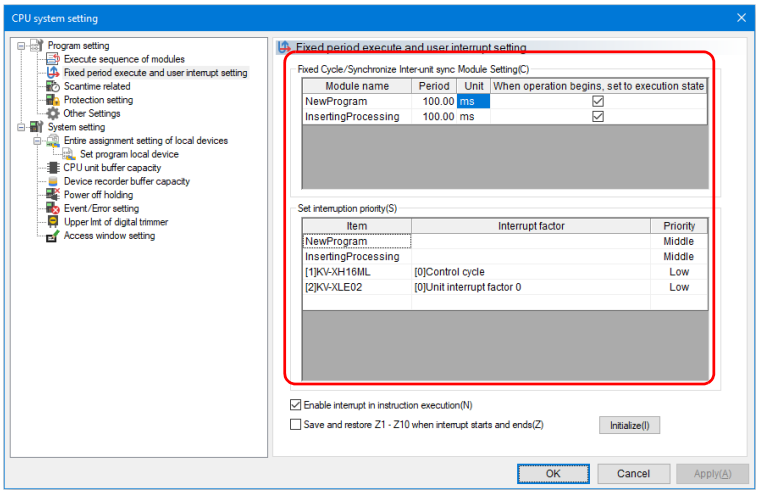
You can only write one INT instruction per interrupt condition.

- **Some instructions cannot be used in interrupt programs.**
  See "Instructions That Cannot Be Used in Interrupt Programs" (page 4-210).
- **To reduce the interrupt response time when executing interrupts using KV-SIR32XT, use the unit editor to reduce the input time constant (default value: 10 μs).**

"INT and RETI Instruction" (page 4-222)

Interrupt Priority

You can set the interrupt priority for the case in which multiple interrupt conditions are met at the same time.
To configure this setting from the KV STUDIO workspace, open the "CPU system setting" dialog box and select "Fixed period execute and user interrupt setting" in the "Program setting."



You can set the priority to "High", "Middle", or "Low". When an interrupt program is being executed, the programs of higher order interrupts will be executed, but the programs of same-level and lower order interrupts will have to wait until the interrupt program currently being executed is finished.
Also, if you select the "Enable interrupt in instruction execution" checkbox, you can suppress variations in interrupt response time because interrupts will be executed mid-process for instructions that have long processing times.
If a different interrupt occurs while an interrupt program is being executed, the operation varies according to the priority settings. See the following table.

Priority of Interrupt Program Being Executed (A)	Priority of Program Whose Interrupt Has Occurred (B)		
	High	Middle	Low
High	Wait	Wait	Wait
Middle	Execute immediately	Wait	Wait
Low	Execute immediately	Execute immediately	Wait

Wait: After the currently executing interrupt program (A) finishes, the program whose interrupt has occurred (B) will be executed.

Execute immediately: The currently executing interrupt program (A) is interrupted, and the program whose interrupt has occurred (B) is executed.
After interrupt program (B) finishes executing, interrupt program (A) continues from the point it was interrupted at.

Instructions That Cannot Be Used in Interrupt Programs

Instruction Type		Instruction	
Common		@XXXX instruction	Differential execution type
Basic instructions	Contact instructions	LDP and LDF	Load pulse and load pulse fall
		ANP and ANF	AND pulse and AND pulse fall
		ORP and ORF	OR pulse and OR pulse fall
		LDPB and LDFB	Load pulse bar and load pulse fall bar
		ANPB and ANFB	AND pulse bar and AND pulse fall bar
		ORPB and ORFB	OR pulse bar and OR pulse fall bar
	Output instructions	DIFU and DIFD	Differential up and differential down
		ONDL and OFDL	On delay and off delay
		SHOT	One shot
		FLIK	Flicker
		ALT	Alternate
	Timer and counter instructions	TMR, TMH, TMS, and TMU	Timer, high-speed timer, 1 ms high-speed timer, and 10 µs high-speed timer
		C	Counter
		OUTC	Out counter
		ITVL	Interval timer
		UDC	Up/down counter
		UDT	Up/down timer
	Connection/end instruction	MEP and MEF	Mep and mef
Application instructions	Shift instruction	SFT	Shift
	Stage processing instructions	STG, JMP, and ENDS	Stage, jump, and end stage
		W-ON and W-OFF	Wait on and wait off
		W-UE and W-DE	Wait rising edge and wait falling edge
	Flow instruction	SBN/RET/SCJ	Start subroutine/end subroutine/jump to LABEL instruction
	File register instruction	FRSTM/FRLDM	File register batch save/file register batch read
Operation instructions	Data control instructions	RAMP, TPOUT, and LLFLT	Ramp signal, time division proportion output, and advancement delay filter
	Character processing instruction	RCOM	Read character string

Instruction Type		Instruction	
Expansion instructions	Data processing instructions	HKEY and SORTN	Hexadecimal key input and binary data division sort
	Cam switch instructions	ABSENC	Absolute encoder
	PID instructions	PID and PIDAT	PID and PID with automatic tuning
	Storage instructions	MWRIT/MREAD/ MFREE/MMKDIR/ MPRINT/MREADL/ MCOPY/MMOV/ MREN/MFREEK/ MSTAT	Write storage/Read storage/Acquire storage capacity/ Create storage directory/Load storage character string/ Read one rung of storage/Copy storage file/Move storage file/Rename storage file/Acquire free storage capacity/Acquire storage file status
		MRMDIR/MDEL	Delete storage folder/Delete storage file
	Access window instruction	AWSHOW/ AWHIDE	Display user message/Cancel user message
	Sensor setting instructions	SPWR/SPRD/SSVC	Read sensor parameters, write sensor parameters and execute sensor parameters

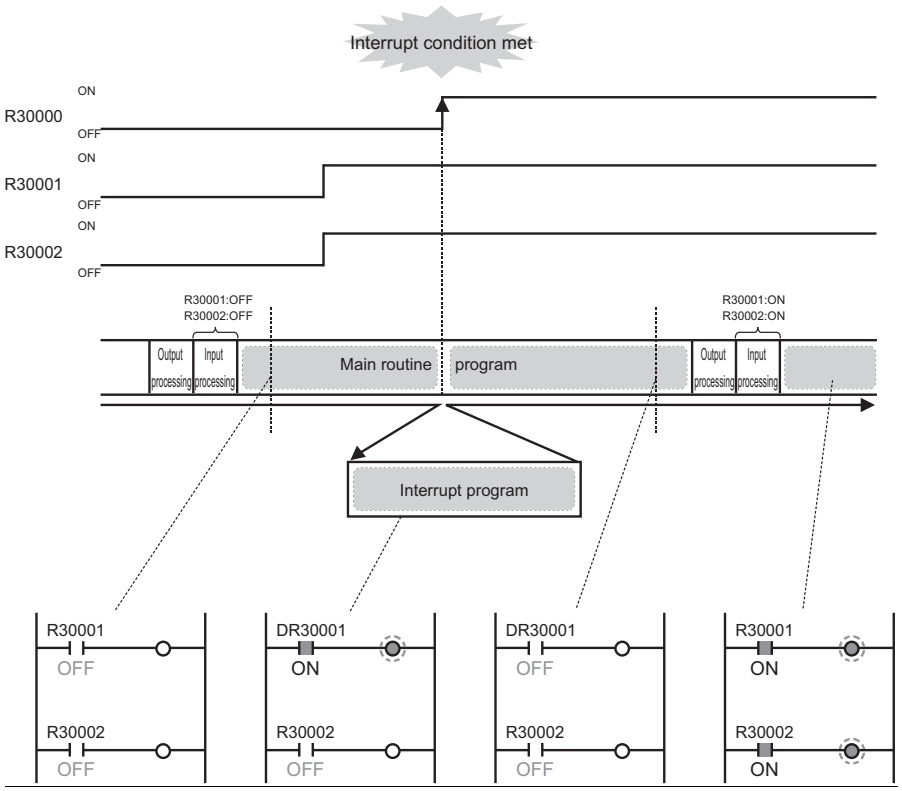
* Can be used in fixed period module or inter-unit synchronization module.

I/O Processing during Interrupts

■ Input processing

When the interrupt program starts, relay status is captured during the scan of input processing (I/O update).
Since the interrupt program captures the input relay status at that time, it is necessary to use DR to specify a relay number or use a direct refresh instruction (RFSX instruction) to update.

Example

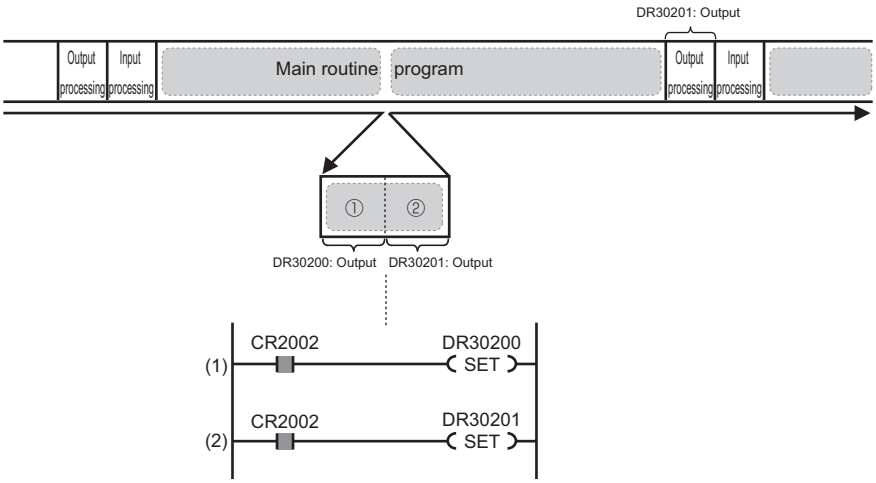


When the above interrupt program is executed:
(1) Input processing of the main routine program → R30001: OFF R30002: OFF.
(2) Input processing of the interrupt program → R30001: ON R30002: OFF.
The status shown above is captured.
Since input relay DR30001 is direct input, the status when the interrupt program is executed can be captured.
Input relay R30002 is handled with the status captured during the input processing of the main routine program even during an interrupt program.
Since the main routine program directly updates the interrupt program, it becomes the status of the interrupt program (ON).

■ Output processing

Interrupt programs have no output processing. After an interrupt program is finished, output is generated by the output processing (automatic refreshing) of the main routine program.
However, if you write a DR device (such as DR30200) on an output relay, output is generated immediately when the program is executed (direct output).

Example



When the above interrupt program is executed:

- Output relay DR30200 on rung (1) is a direct output relay, so output is generated immediately.
- Output relay R30201 on rung (2) is turned ON during internal processing, but because it is not a direct output relay , output is generated during output processing of the main routine program.

Example

If you want to perform direct output within an interrupt program, use a DR device on an output relay.

Reference

To update to the latest status device values that are not automatically refreshed:

- Use "DR" to write an I/O relay.
- Write the RFSX or RFSY instruction.
- It describes direct refresh instruction such as “UREAD/UWRIT”.
- Enter a unit dedicated instruction.

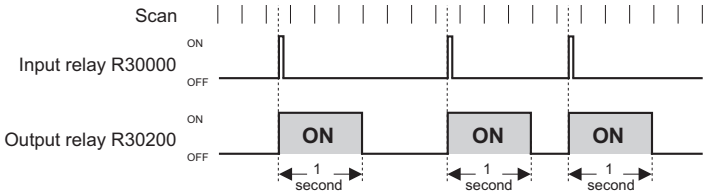
"Direct processing" (page 4-83)

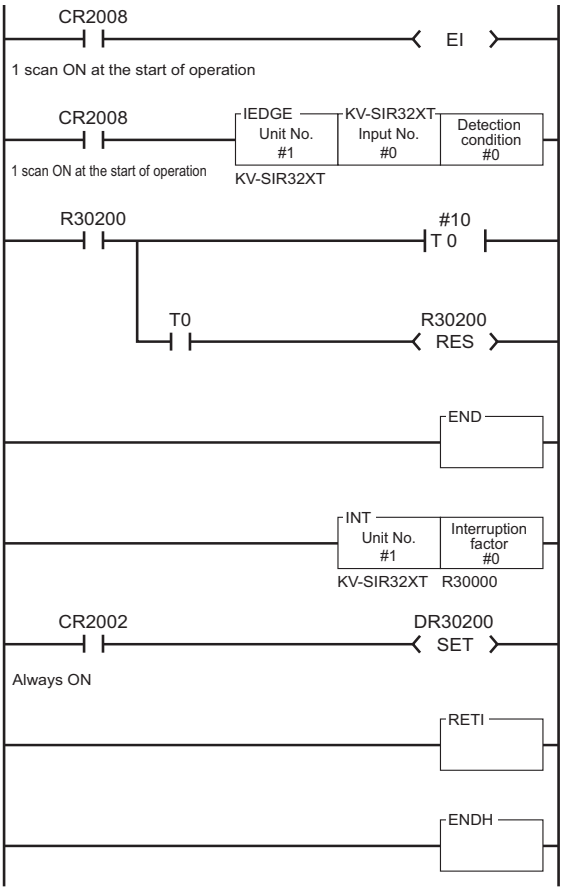
Sample Program

■ Application example of interrupts using a signal converter

In this example, interrupt processing is used to convert minute pulses into fixed width pulses, which are then output.

- The minute pulses applied to input relay R30000 are converted to pulses with an ON width of 1 second, which are output from output relay R30200.
- Output relay R30200 is turned ON for 1 second from the rising edge of input relay R30000.





When the power is turned on, interrupts are enabled.

The interrupt detection condition is set to the rising edge.
* Please configure the constant during input of input relay R300000 with the unit editor（Initial value:10 μs)

Output relay R30200 is turned ON for just 1 second.

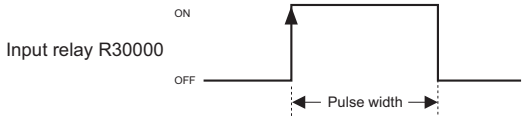
When input relay R30000 turns ON, output R30200 is turned ON.

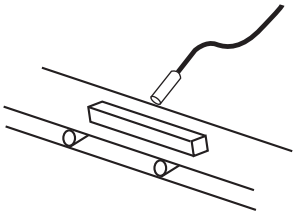
<Mnemonic List>

```
LD CR2008
EI
LD CR2008
IEDGE #1 #0 #0
LD R30200
MPS
TMR #0 #10
MPP
AND T0
RES R30200
END
INT #1 #0
LD CR2002
SET DR30200
RETI
ENDH
```

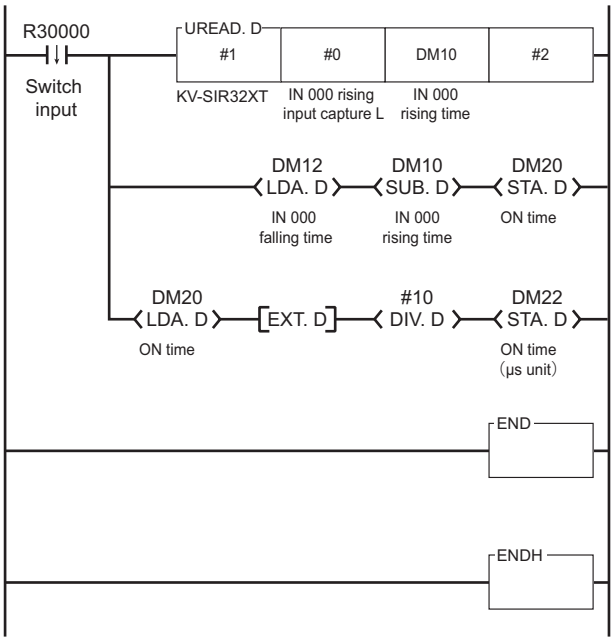
■ Application example of measuring the ON width of high-speed pulses

The pulse widths of pulses applied to the input relay are measured.





- The pulse width will be measured by 0.1 μs unit by using the input capture function.
- At the falling edge of the R30000 input relay, the input capture value of R30000 (IN000) rising edge (buffer memory #0, #1) is stored in data memory [DM10 and DM11] and the input capture value of the falling edge (buffer memory #2, #3) is stored in data memory [DM12 and DM13] by UREAD instruction.
- The value of [DM10/DM11] will be subtracted from [DM12/DM13].
- The measured values are stored in data memory entries [DM20 and DM21].
- The [DM20 and DM21] values are converted to 1 μs units and stored in [DM22 and DM23].

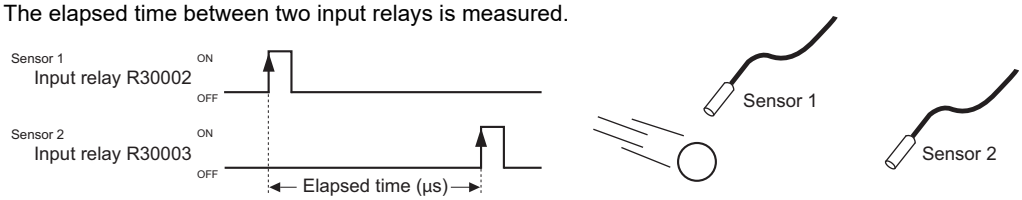


<Mnemonic List>

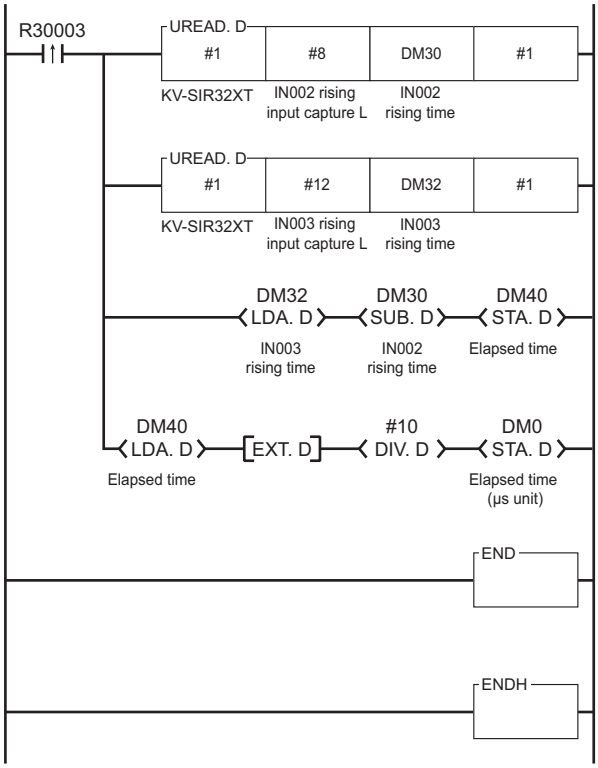
```
LDF R30000
UREAD.D #1 #0 DM10 #2
MPS
LDA.D DM12
CON
SUB.D DM10
CON
STA.D DM20
MPP
LDA.D DM20
CON
EXT.D
CON
DIV.D #10
CON
STA.D DM22
END
ENDH
```

■ Application example of measuring the elapsed time between two points

The elapsed time between two input relays is measured.



- It measures the passing time from input relay R30002 to input relay R30003.
- The measured value is stored in data memory entries DM0 and DM1 in units of 0.1 μs.



At the rising edge of R30003, the result of input capture is read.

The time from the IN002 rising edge to IN003 falling edge is obtained.

It converts the measured time to μs unit.

<Mnemonic List>

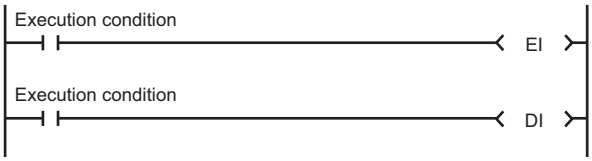
```
LDP R30003
UREAD.D #1 #8 DM30 #1
UREAD.D #1 #12 DM32 #1
MPS
LDA.D DM32
CON
SUB.D DM30
CON
STA.D DM40
MPP
LDA.D DM40
CON
EXT.D
CON
DIV.D #10
CON
STA.D DM0
END
ENDH
```

Interrupt Processing Instructions

Instruction	Mnemonic	Operation Overview	Page
Disable interrupts	DI	Disables the execution of interrupt programs	4-218
Enable interrupts	EI	Enables the execution of interrupt programs	4-218
Disable interrupts range	DIC	Sets the interrupt disabled range	4-220
Interrupt	INT	Executes the interrupt program from this instruction to the RETI instruction	4-222
Interrupt return	RETI	Indicates the end of the interrupt program	4-222

DI	DI	Disable interrupts	Disables the execution of interrupt programs
EI	EI	Enable interrupts	Enables the execution of interrupt programs

Ladder program



Entry mode

Operation Description

DI When the execution condition turns ON, the execution of interrupt programs is disabled. Use this instruction when you want to temporarily disable interrupts.

EI When the execution condition turns ON, the execution of interrupt programs is enabled. Use this instruction to enable the interrupt programs that have been disabled with the DI instruction.

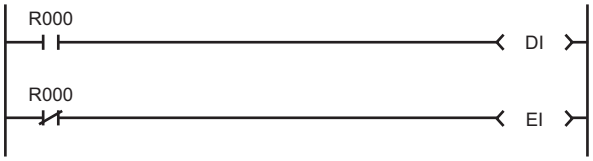
- When the EI instruction is executed, interrupts are enabled until the DI instruction is executed.
- When operations start, interrupts are disabled.
- If interrupts occur while the DI instruction is being executed (while interrupts are disabled), the execution of the interrupt programs will be put on hold.
  When the EI instruction is executed, the interrupts that were put on hold are executed.
- When an interrupt program starts, interrupts are enabled (EI). During the execution of an interrupt program, to execute further interrupt programs, the priority of these further interrupts must be set higher than that of the currently executing interrupt program. To set priorities, click "CPU system setting" and then "User interrupt." You can set up to three levels of priorities.

Operation flags

The operation flags are not changed.

Sample Program

When input relay R000 is turned ON, the execution of interrupt programs is disabled.
When R000 is turned OFF, the execution of interrupt programs is enabled.



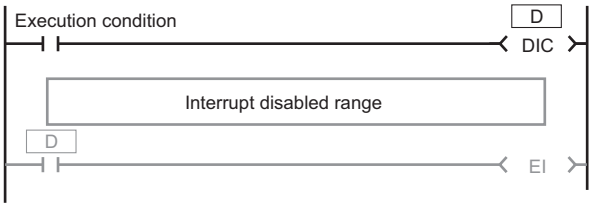
<Mnemonic List>

```
LD R000
DI
LDB R000
EI
```

DIC

≺ DIC ≻	Interrupt disabled range	Sets the interrupt disabled range

Ladder program



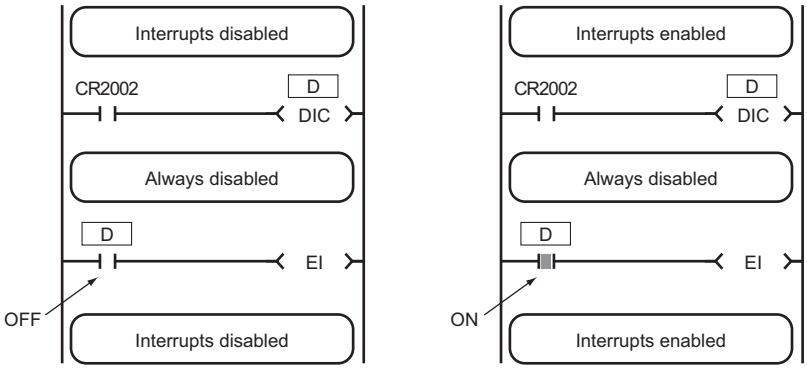
Entry mode

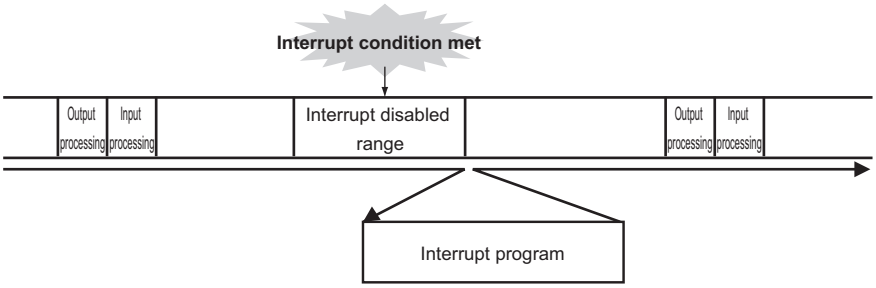
Operand	Description	Occupied size
D	Specifies the bit device that generates the interrupt enabled/disabled status	1 bit

Operation Description

DIC

Use this instruction together with the EI instruction.
When the execution condition turns ON, the execution of interrupt processing is disabled from the DIC instruction to the EI instruction.
The interrupt enabled/disabled status prior to the execution of the DIC instruction is generated by the device specified by D.
If interrupts were enabled prior to the execution of the DIC instruction, the device specified by D is set. If interrupts were disabled, the device is reset.





Operation flags

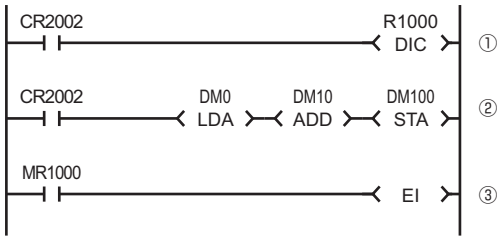
CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	This is turned ON if the range of indirect specification or index modification is incorrect. Otherwise, this is turned OFF. There is no change in status if the operands do not specify indirect specification or index modification.

* If CR2012 is turned ON, the instruction will not be executed.
When CR2012 is turned ON, the detailed error information is stored in CM5150 to CM5176.

"Error log" (page 4-86)

Sample Program

Interrupts are disabled during operation processing. After the DIC instruction is executed, the status of interrupt processing is returned to the status that was present prior to the DIC instruction being executed.



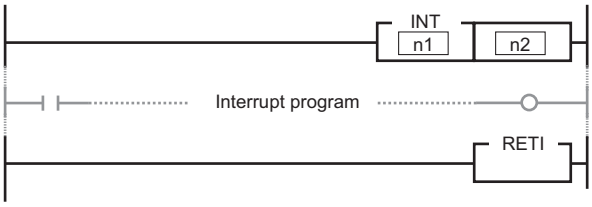
<Mnemonic List>

```
LD CR2002
DIC R1000
LD CR2002
LDA DM0
CON
ADD DM10
CON
STA DM100
LD R1000
EI
```

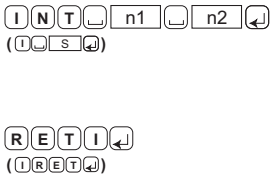
① Interrupt processing is disabled.
② DM0+DM10→DM100
③ The interrupt enabled or disabled status is recovered.

INT	INT	Interrupt execution	Executes the interrupt program defined by the instructions up to the RETI instruction upon the rising or falling edge of the operand
RETI	RETI	Interruption end	Indicates the end of the interrupt program

Ladder program



Entry mode



Operand	Description	Occupied size
n1 *1	Specifies a unit number (0 - 48).	16 bits
n2 *1	Specifies the unit interruption factor (0 - 31).	16 bits
S *2	Specifies the bit for interrupt.	1 bit

*1 Only constants and variables / labels set as constant can be specified.
*2 Constants can be specified.

Operation Description

INT This instruction specifies the start of an interrupt routine when an interrupt factor specified in n2 on the unit specified in n1 occurs.

RETI This instruction specifies the end of the interrupt routine.

Point

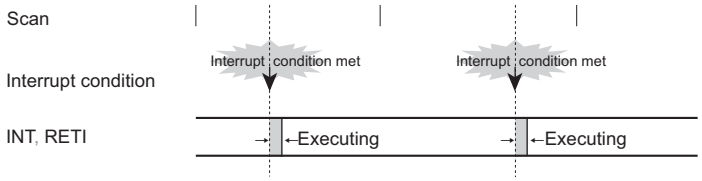
The internal register, index register, operation flag, TM0 to TM3 are automatically saved when an interrupt routine starts. Each saved value is returned when the interrupt routine ends. Under the default settings, only Z11 and Z12 index registers are saved/restored. You can use "CPU system settings" → "Fixed period execution and user interrupt" to configure settings to save and restore the Z1 to Z10 index registers.

Operation flags

CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	No change in status

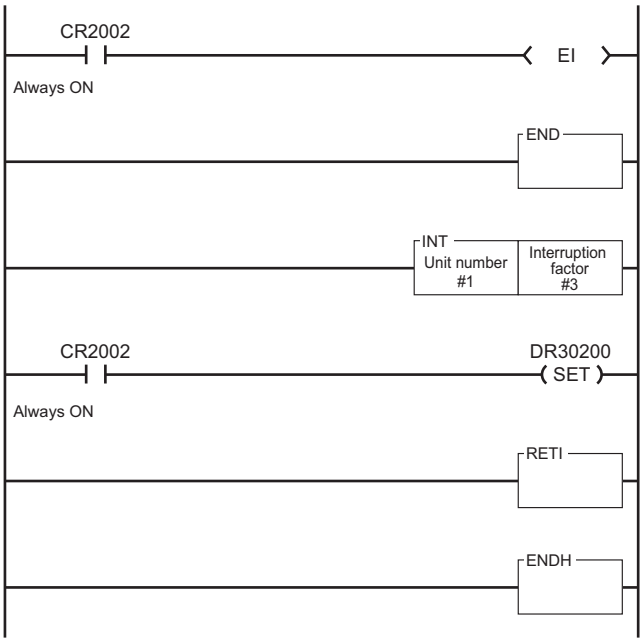
Instruction Execution Timing

INT and RETI The instruction is executed on the rising or falling edge of the operand.



Sample Program

Executes interrupt by occurrence of interrupt factor 3 of unit number 1, executes direct refresh and immediately outputs R30200.



<Mnemonic List>

```
LD CR2002
EI
END
INT #1 #3
LD CR2002
SET DR30200
RETI
ENDH
```

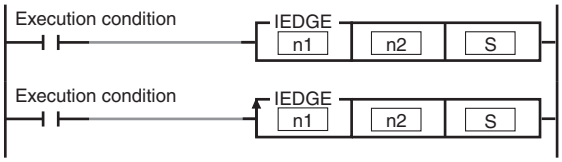
Reference For direct refresh, see "Direct processing" (page 4-83).

IEDGE
@IEDGE

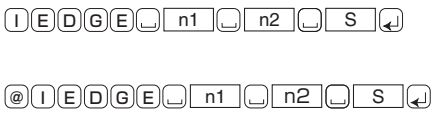
Input interrupt Detection conditions write

Sets detection conditions used for interrupt.

Ladder program



Entry mode



Operand	Description	Occupied size
n1	Specifies the unit No.(0 to 48). *1	16-bit
n2	Specifies the input number or the leading to store it. (0 to 31)*2	16-bit
S	Specifies the detection conditions or the leading to store it. (0 to 3)*2	16-bit

*1 Only constants can be specified.
*2 When specifying bit data, specify the leading of the channel.

Operation Description

IEDGE

When the execution condition is ON, sets 1 word of data specified in [S] as the interrupt detection condition of the [n2] factor of the [n1] unit.
The detection conditions are as follows according to the value of [S].

S	Detection condition
0 to 1	Rising edge
2	Falling edge
3	Both Rising and Falling Edges

Point

- **When 4 to 65535 is specified in [S], no calculation error occurs, but the detection conditions will not be changed.**
- **When this instruction is complete, the above detection condition setting is changed.**

@IEDGE

Only one scan is executed at the rising edge of the execution conditions.

Operation flag

CR2009	No change in status
CR2010	No change in state
CR2011	No change in state
CR2012	This flag is ON in the following instances. Otherwise, it is OFF. • When the indirect specification or index modify range is inappropriate • When the unit number specified with [n1] is 49 or more • When the input number specified with [n2] is 32 or more • When a 1-word device cannot be secured from the device number specified with [S] • When the unit specified by the unit number in [n1] is not KV-SIR32XT

* The instruction is not executed when CR2012 turns ON.
When CR2012 is ON, detailed error information is stored into CM5150 to CM5176.

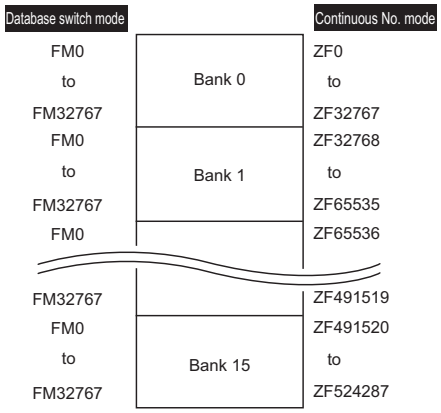
Sample Program

For Sample Program, refer to "Unit interruption factor" (page 6-5)

4-14 File Register

Overview

File registers are divided into "ZF" that handle all areas using a consecutive numbering scheme, and into 16 FM banks (0 - 15).



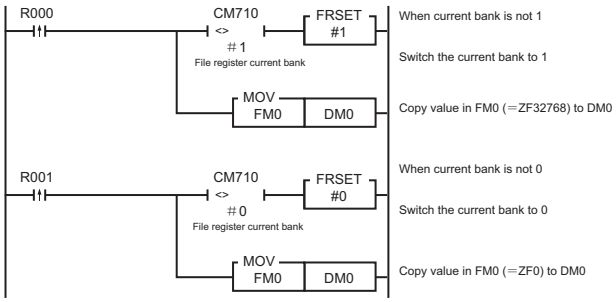
For example, FM0 and ZF32768 of memory bank 1 refer to the same data store part.

For detailed condition of device No. And data treatment method etc, please refer to "4-4 Devices", "File Register ZF/FM" (page 4-39).

Bank Switchover

To switch to another memory bank, use the "FRSET instruction".
Currently designated memory bank is called current memory bank.

Example



<Mnemonic List>

```
LDP R000
MPS
AND<> CM710 #1
FRSET #1
MPP
MOV FM0 DM0
LDP R001
MPS
AND<> CM710 #0
FRSET #0
MPP
MOV FM0 DM0
```

"FRSET" (page 4-230)

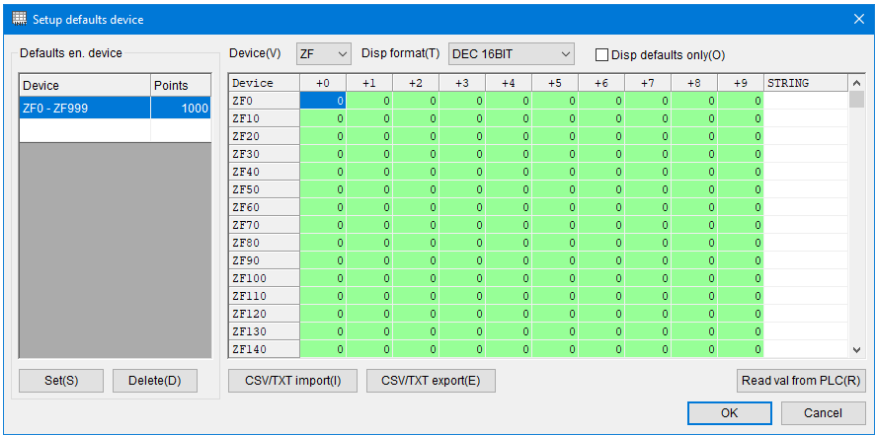
Reference

- No. Of current memory bank may be confirmed by CM710.
- Can use KV mode (host link) "BE" instruction, to switch memory bank.

KV-XL202/XL402 User's Manual

Defaults

When preset the value of file register (consecutive number mode), open "device default setting" dialog box on KV STUDIO work space to set.



After transmit the project set in this dialog box to CPU unit, when switch PRG → RUN mode, read set initial value automatically.

KV STUDIO User's Manual "Device Default Value Settings"

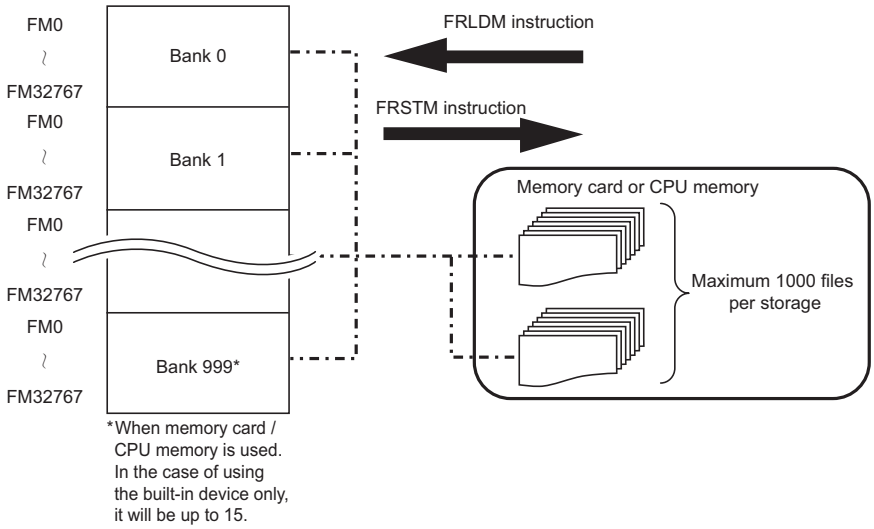
File Register Setting

Use special instructions to to save and read file register data (file register set values) on memory cards inserted in the CPU unit or in internal CPU memory .

FRSTM instruction: Saves the file registers in the specified bank to a memory card or CPU memory .

FRLDM instruction: Loads files saved to memory card or CPU memory to the file register in the specified bank.

"FRSTM" (page 4-232), "FRLDM" (page 4-234)

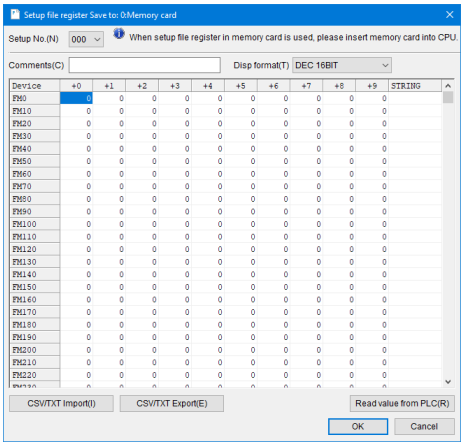


Create the file register setpoint for a total of 1,000 files in memory card or CPU memory in advance, and you will be able to load them whenever needed without having to consider how many devices are involved.

■ File register set value management

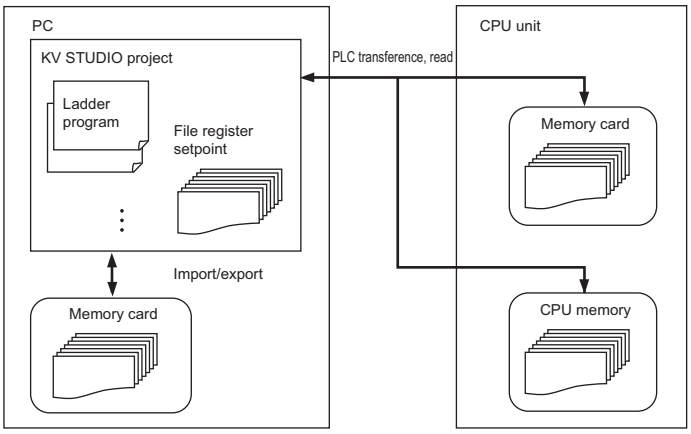
Files (file register setpoint) saved in memory card or CPU memory may be edited through KV STUDIO. Open "File Register Settings" dialog box on work space to carry out edit, new, CSV file import etc. File register setpoint should be managed according to different projects respectively.

KV STUDIO User's Manual "File Register Settings"



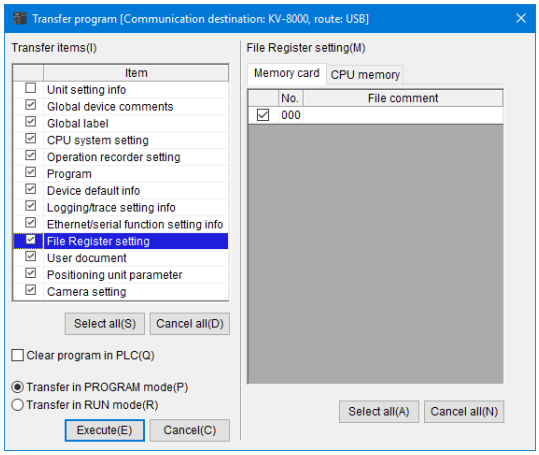
■ File register set value transfer

Use the following procedure to write the file register setpoint created or edited in KV STUDIO to memory card or CPU memory or read file register setpoint files from memory card or CPU memory .



● PLC transfer/read

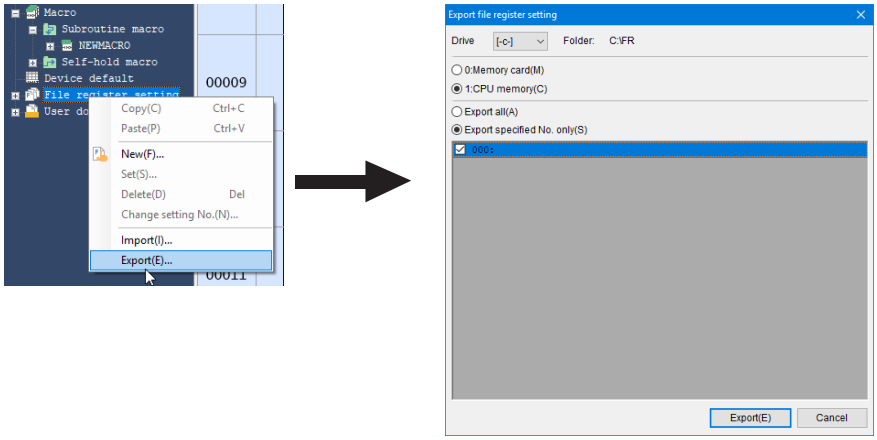
PLC is transferred and read just like other project information. Other project information is transferred to the CPU unit while the file register setpoint is saved to memory card or CPU memory.



KV STUDIO User's Manual “monitor"

● Import/export

Use the storage transfer tool to import and export information.

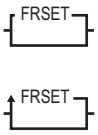


KV STUDIO User's Manual “file register setting"

File Register Instructions

Instruction	mnemonics	Instruction description	Page
File register memory bank switching	FRSET	Switch current memory bank no. of file register to determined no..	4-230
File register batch saving	FRSTM	Save all the file registers in the specified bank to memory card or CPU memory in the binary format.	4-232
File register batch read	FRLDM	Read all the data saved in memory card or CPU memory to the specified bank file register.	4-234

FRSET
@FRSET

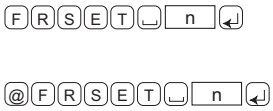


File register bank switching

Switch current memory bank no. of File Register to determined no..

Ladder program

Execute condition



Operation	Explanation	Occupied size
n	Specifies the current bank No. after switching or its storage location. (0 to 15 for the KV-8000/7000 Series, 0 to 3 for KV-5500/5000/3000)	16 bits

Description of Operation

FRSET When the execution condition is ON, switch the current bank No. of the file register to the specified No. of n.

@FRSET This instruction is executed for one scan at the rising edge of the execution condition.

Operation flag

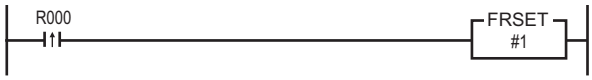
CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	• ON when n > 15. (For KV-5500/5000/3000, ON when n >3) • ON when the indirect specification and index modify range is inappropriate. • ON when an access outside the operand range occurs. Otherwise OFF

* The instruction is not executed when CR2012 turns ON.
When CR2012 is ON, error details are stored in CM5150 to CM5176.

"Latest error details (CM)" (page 4-87)

Example

When input relay R000 is ON at the rising edge, switch the current group of file register to Group 1.

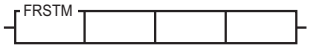


<Mnemonic List>

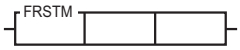
```
LDP R000
FRSET #1
```

FRSTM

For KV-8000 / 7000 series



For KV-5000 / 3000 series



File register allsaving

The file register within thedesignated bank will besaved at once to the memorycard (CPU memory) inbinary format.

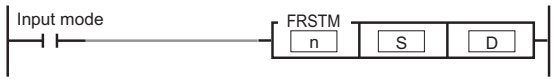
KV-8000/7000 Series

Ladder program

Execute condition

KV-5000/3000 Series

Ladder program



Execute condition

Operation	Explanation	Occupied size
n1	Specifies the drive No. or its storage location (0: Memory card, 1: CPU memory).*1	16 bits
n2	Specifies the bank No. or its storage location (0 to 15).*2	16 bits
S	Specifies the file No. to save the data or its storage location (0 to 999).	16 bits
D	Specifies the start address for notification bit data.	1 bitx2

*1 For KV-5000/3000, it is not necessary to specify the drive No.

*2 For KV-5000/3000, the bank No. range is 0~3.

Description of Operation

FRSTM At the rising edge of execution condition, it will save all file registers of the bank No. n2 in batch to the storage (memory card for KV-5000/3000 series) specified by n1 in binary format (the file will always be overwritten).

File No. S is specified with number.

The created file is named as \FR\FR (file No. is 3-digit).FRU*.

* The file extension for KV-5500/5000/3000 is FRB.
Create a new folder if there is no "FR" folder.

Example When 10 is stored in S, /FR/FR010.FRU (or FR010.FRB) will be created.

The end notification is written in D, and the error notification of abnormal end is written in [D+1].

- D turns OFF when execution of the instruction starts, and turns ON when write processing to the storage ends. It becomes ON either when the command is normally finished, or abnormally finished.
- D+1 turns OFF when the instruction ends successfully, and turns ON when execution ends in error.
  The error code is stored in CM2390.

Point **The command will not be executed when the storage device instructions in execution relay (CR3214) is ON. Please implement exclusive control using CR3214.**

Operation flag

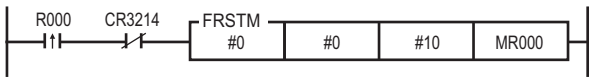
CR2009	No change in state
CR2010	No change in state
CR2011	No change in state
CR2012	• ON when the value saved in [n] is larger than "15" ("3" for KV-5500/5000/3000) • ON when the value saved in [S] is larger than "999" • ON when the indirect specification or index modify range is inappropriate. • ON when an access outside the operand range occurs Otherwise OFF

* The instruction is not executed when CR2012 turns ON.
When CR2012 is ON, error details are stored in CM5150 to CM5176.
"Latest error details (CM)" (page 4-87)

Sample Program

When input relay R000 is ON at the rising edge, store the file register of Group 0 into the memory cardwith the file No. 10.

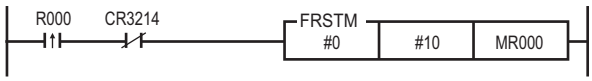
KV-8000/7000 Series



;(Mnemonics list)

```
LDP R000
ANB CR3214
FRSTM #0 #0 #10 MR000
```

KV-5000/3000 Series

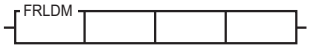


;(Mnemonics list)

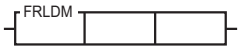
```
LDP R000
ANB CR3214
FRSTM #0 #10 MR000
```

FRLDM

For KV-8000 / 7000 series



For KV-5000 / 3000 series



Batch read fileregister

Read the batch data saved in memory card (CPU memory)into the file register of thespecified group.

KV-8000/7000 Series

Ladder program

Execute condition

KV-5000/3000 Series

Ladder program

Execute condition

Operation	Explanation	Occupied size
n1	Specifies the drive No. or its storage location (0: Memory card, 1: CPU memory).*1	16 bits
n2	Specifies the bank No. or its storage location (0 to 15).*2	16 bits
S	Specifies the file No. to read data or its storage location (0 to 999).	16 bits
D	Specifies the start address for notification bit data.	1 bitx2

*1 For KV-5000/3000, it is not necessary to specify the drive No.

*2 For KV-5000/3000, the bank No. range is 0~3.

Description of Operation

FRLDM

It will read the data saved in the storage (memory card for KV-5000/3000 series) specified by n1 to the file register of the bank No. n2 in batch (according to the FRSTM instruction or the "File Register Settings" of KV STUDIO) at the rising edge of the execution condition. (The file will always be overwritten).

File No. S is specified with number.

The read file is named as \FR\FR (file No. is 3-digit).FRU*.

* The file extension for KV-5500/5000/3000 is FRB.

Example When 10 is stored in S, /FR/FR010.FRU (or FR010.FR) will be read out.

The end notification is written in D, and the error notification of abnormal end is written in [D+1].

- D turns OFF when execution of the instruction starts, and turns ON when write processing to the storage ends. It becomes ON either when the command is normally finished, or abnormally finished.
- D+1 turns OFF when the instruction ends successfully, and turns ON when execution ends in error.
  The error code is stored in CM2390.

Point **The command will not be executed when the storage device instructions in execution relay (CR3214) is ON. Please implement exclusive control using CR3214.**

Operation flag

CR2009	No change in state
CR2010	No change in state
CR2011	No change in state
CR2012	• ON when the value saved in [n] is larger than "15" ("3" for KV-5500/5000/3000) • ON when the value saved in [S] is larger than "999" • ON when the indirect specification or index modify range is inappropriate. • ON when an access outside the operand range occurs Otherwise OFF

* The instruction is not executed when CR2012 turns ON.
When CR2012 is ON, error details are stored in CM5150 to CM5176.
"Latest error details (CM)" (page 4-87)

Sample Program

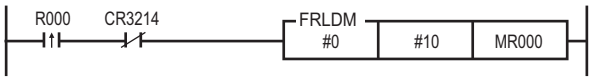
When input relay R000 is ON at the rising edge, read the data of file No. 10 in the memory card into thefile register of Group 0.

KV-8000/7000 Series

;(Mnemonics list)

```
LDP R000
ANB CR3214
FRLDM #0 #0 #10 MR000
```

KV-5000/3000 Series



;(Mnemonics list)

```
LDP R000
ANB CR3214
FRLDM #0 #10 MR000
```

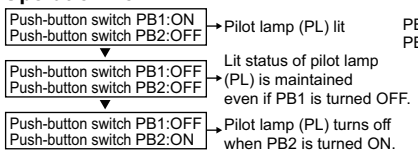
From Installation to Operation

The procedure from KV-8000 Series installation to operation is shown in the figure below.

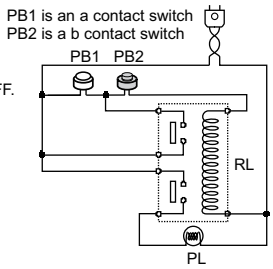
Operation details consideration

Use information such as the operation flow, specifications, timing charts, and circuit diagrams to design the operation details.

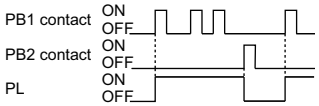
Operation flow



Circuit diagram



Timing chart



Program consideration

◇ **Unit Configuration**
When there are many pieces of I/O equipment used in control and when you are using expansion functions, consider using expansion units.

KV-8000 Series and Expansion Unit "User's Manuals."

◇ **Module Configuration**
Design the configuration of the modules that you will create for different aspects such as processes and functions.
"Module Configuration Examples" (page 4-110)

◇ **Sequence Program Configuration**
Design the outline while you consider the control contents of the program that you will create per module and while you consider the instructions that you will use.
For details, see the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

Program creation

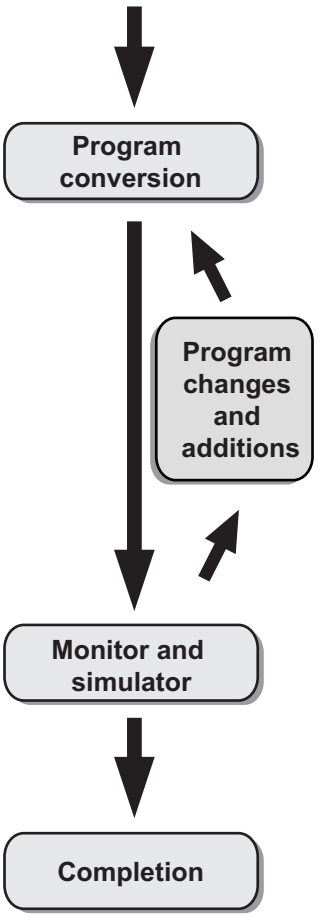
Use KV STUDIO to Create the Program

◇ **Create a New Project**
"New Project" in the "KV STUDIO User's Manual"

Key Points to Program Creation

◇ **Configure Unit Settings**
Use the Unit Editor to assign unit devices to the units.
"Chapter 3 UNIT EDITOR" in the "KV STUDIO User's Manual"

◇ **Create a Program**
"4-7 How Data Is Processed" (page 4-75)
"Program Creation Support" (page 4-238)
"Key Points to Program Creation" (page 4-258)
"4-10 Function Blocks" (page 4-117)
For details, see the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."



Programs are converted when they are transmitted to a PLC for monitoring and when you switch to the simulator. If a conversion error occurs, remove the cause of the error while referring to the output window.

"Conversion" (page 4-266)
"Local Device Input and Errors during Conversion" (page 4-204)
"Appendix 1 Error Message Lists" in the "KV STUDIO User's Manual"
"Key Points to Program Creation" (page 4-258)

Change and add programs.

"Creating Clear, Legible Programs" (page 4-252)
"Online edit" (page 4-269)
"Writing data during RUN mode" (page 4-267)
"Simulator edit" (page 4-271)

Use the useful monitor and simulator functions to debug the program.

Monitor "Monitor" (page 4-268)
Simulator "Simulator" (page 4-270)

Frequently used programs → Import programs

"Module management" (page 4-98)
"Function block management" (page 4-117)

Program Creation Support

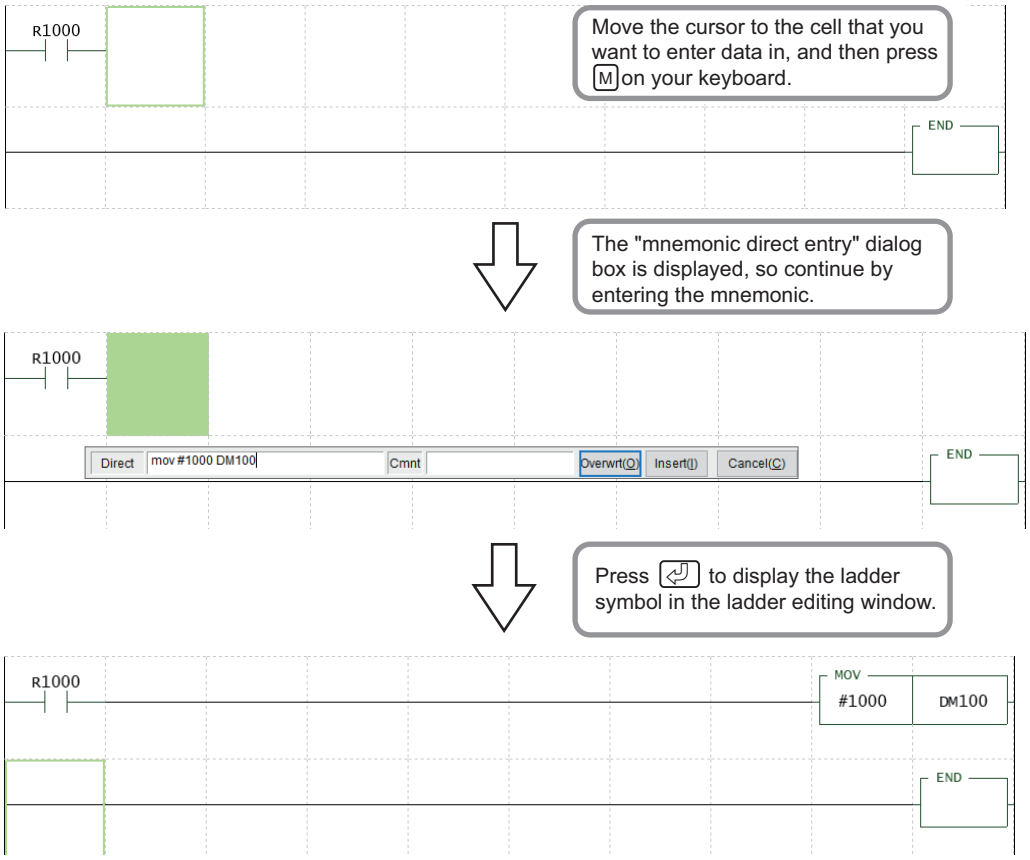
This section will explain some convenient functions that can reduce your entry and modification workload during ladder program creation.

Mnemonic entry

As mnemonic instructions can be entered without a mouse, programs can be created more quickly than when using the "Instruction palette".

■ Entry mode

Use the keyboard to enter the first character of the instruction.
(Example) Entering the MOV instruction



Reference On the "Tool" menu, click "Option," click the "Setup edit/key" tab, and then select the "Zoom out direct entry dialog box" check box to minimize the display of the "mnemonic direct entry" dialog box.

For details on the instructions, see the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

Reference

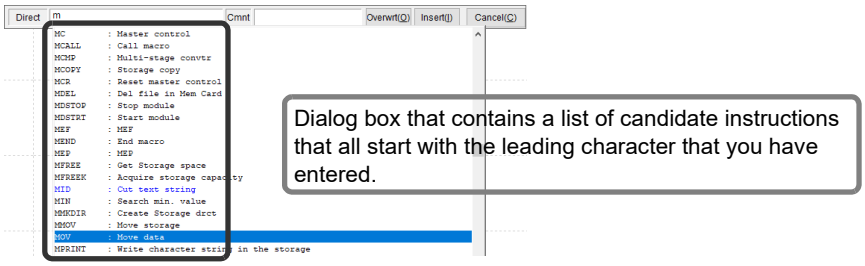
- You can omit the target devices and values that are specified by the instruction and operands when you enter the instruction.
  (Example) A 0 → LD R000
  "App-6 List of Abbreviated Mnemonics" (page A-23)
- To use the "mnemonic direct entry" dialog box to edit cells that have already been entered, press Esc or F2. (If you click "Option" on the "Tool" menu, click the "Setup edit/key" tab, and then select the "Start up direct entry by Enter key" check box, the ⏎ key will be assigned to the "mnemonic direct entry" dialog box.)

■ Entry assistance

There are several functions that can assist you in entering instructions when you are using mnemonic entry.

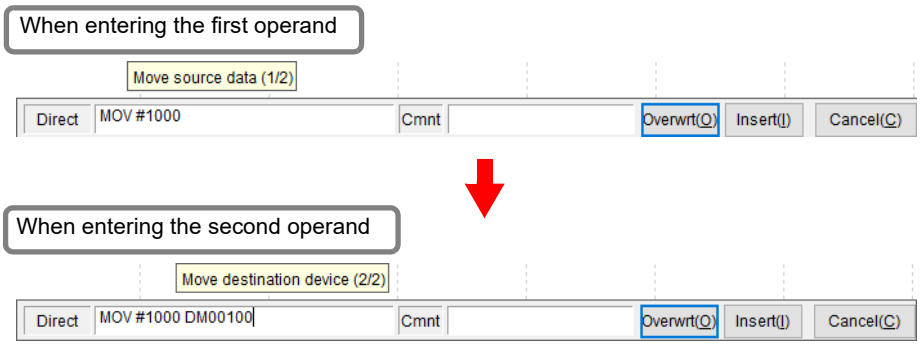
● Instruction list display

When you enter the first character of an instruction, a dialog box that contains a list of candidate instructions that all start with the leading character that you have entered is displayed, and you can select the instruction from the list.



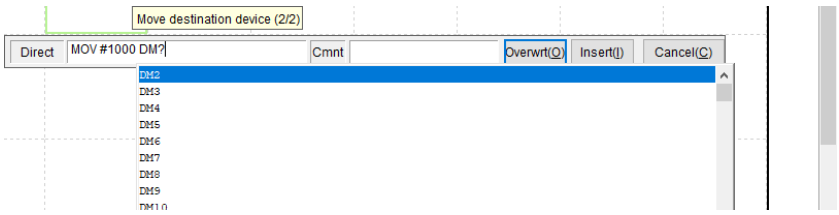
● Instruction operand simple explanation display

A simple explanation of the operand that you are currently entering and which number this operand is are displayed.



● Simple search for unused devices

If you enter part of a device type or number, and then enter "?," the unused devices will be listed.



Shortcut keys

If you use shortcut keys, you can perform operations with just the keyboard and without using a mouse. Also, because this reduces the number of keystrokes, it also reduces the amount of time spent entering data.

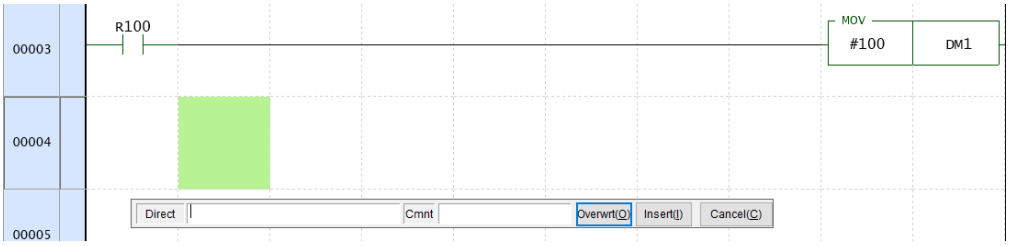
This section will explain typical shortcut keys.

"App-7 List of Shortcut Keys" (page A-25)

■ Entry and editing

● Esc F2

The "mnemonic direct entry" dialog box is displayed, and you can use it to directly enter and edit mnemonics.



"Mnemonic entry" (page 4-238)

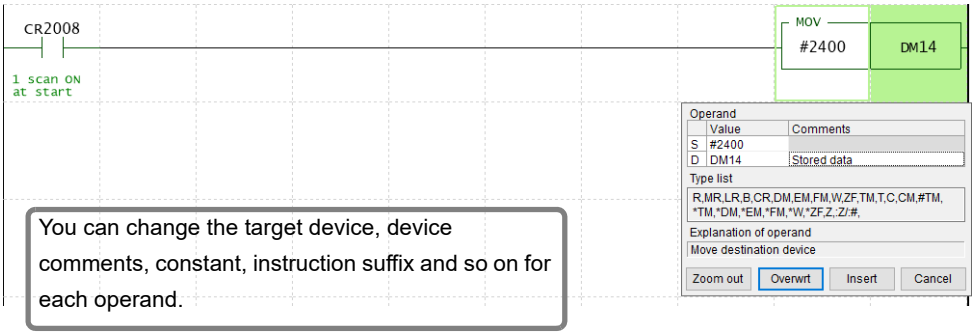
● ⏎

When a blank cell is selected in editor mode

The "mnemonic direct entry" dialog box is displayed, and you can use it to directly enter and edit mnemonics.

When a cell with an existing entry is selected in editor mode

The "Operand" dialog box is displayed.



In monitor mode or simulation mode

The "Search/replace" dialog box is displayed.

● Ctrl + M

If a cell with an existing entry has been selected, the "Operand" dialog box described above is displayed, and you can edit the comment.

- **Ctrl + X / C / V**

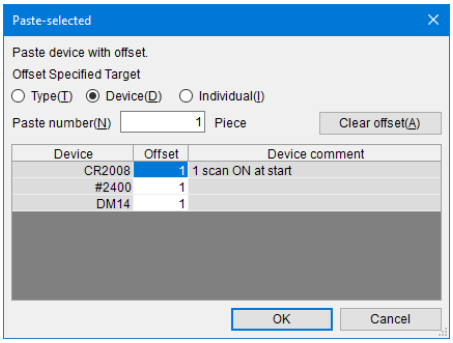
These are standard editing shortcut keys for Windows applications.

Ctrl + X Cut
Ctrl + C Copy
Ctrl + V Paste

- **Ctrl + Shift + V**

Set the number of items to paste to paste multiple times.
In addition, you can offset the included devices.

See in "5-1 About Edit Functions - Multiple Paste" in the "KV STUDIO User's Manual."



- **Ctrl + Z / Y**

These are standard editing shortcut keys for Windows applications.

Ctrl + Z Undo
Ctrl + Y Redo

- **Tab / Shift + Tab**

Move the cursor an amount specified by the ladder symbol unit.

- **Ctrl + ↑ / ↓**

Move the cursor in units of ladder circuit blocks.

- **Ctrl + →**

Select the entire line in which the selected cell resides.

- **Ctrl + Shift + ↑ / ↓**

Change the operand number of the selected cell.

■ Insert, Delete

- Insert

  Insert a horizontal connecting line in the selected cell.

- Delete

  Deletes the instruction/connection line in the selected cell.

- Shift + ↵ or Shift + Insert

  Insert a rung.

- Shift + Delete

  Delete a rung.

- Ctrl + ↵

  Insert rung comments.

  "Rung comments" (page 4-254)

■ Connecting line editing

- Alt + ↵ or Ctrl + Tab

  Draw a connecting line from the selected cell to the end of the line.

- Alt + ← / ↑ / → / ↓

  If there is no connecting line in the direction of the arrow key, draw a connecting line in this direction.
  If there is a connecting line in the direction of the arrow key, delete this connecting line.

■ Search and replace

- Ctrl + F

  The "Device search" tab of the "Search/replace" dialog box is displayed.

  "Search and replace" (page 4-244)

- Ctrl + H

  The "Device replace" tab of the "Search/replace" dialog box is displayed.

- Space

  Cross references are displayed.

  "Cross references" (page 4-244)

■ View

● Ctrl + F7

The "Device comment edit" window is displayed.

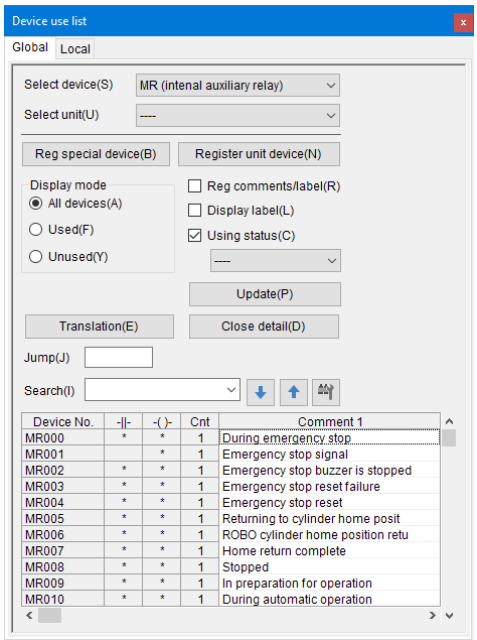
"Device comments" (page 4-252)

● Ctrl + E

The "Device use list" window is displayed.

You can use this to easily check the unused devices.

See "6-2 About View Functions - Device Use List" in the "KV STUDIO User's Manual."



● Ctrl + Page Up / Ctrl + Page Down

Zooms in to and out from the ladder editing window.

*When "Auto zoom" on the "View" menu is selected, pressing these shortcut keys has no effect.

● Ctrl + Space

Shows and hides comments in the ladder editor.

Search and replace

You can search for instructions and operands that match the specified conditions. You can also display the items that match the search conditions as a list in the output window.
When searching for devices, use the cross reference function. You can display this list just by pressing the space bar.

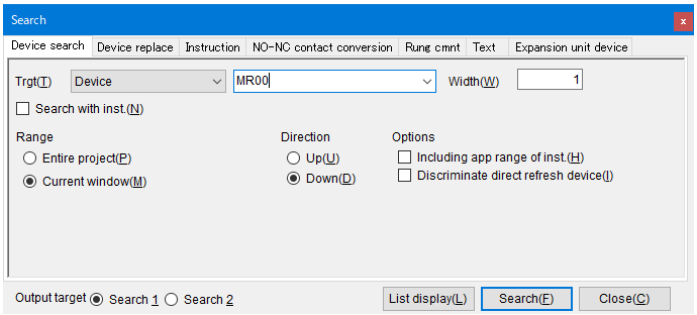
"Cross references" (page 4-244)

1 Select the ladder editing window.

2 Right-click, and then click "Search" in the menu that is displayed.

Alternative procedure
- On the "Edit" menu, click "Search."
- Ctrl + F

The "Search/replace" dialog box appears.



You can set the target to "Device", "Variable", "Constant", "Program", "Device comment", "Variable comment", "Hold / Invalid" or "Invalid line".

Cross references

Edit (E) → Cross reference (X) → Create operand (O) (space)

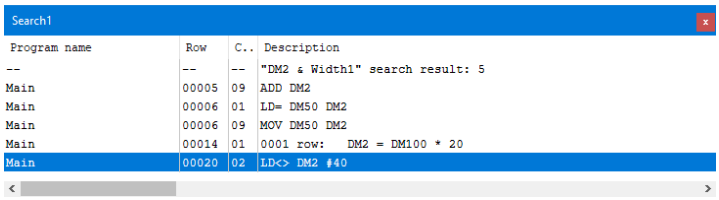
A list of search results for the specified device is displayed in the [Search 1]/[Search 2] output window.

1 Select the device (ladder symbol) whose cross references you want to create.

2 Select "Edit (E)" -> "Cross reference (X)" -> "Create operand list (O)" from the menu.

A list of search results is displayed in the [Search 1]/[Search 2] output window.

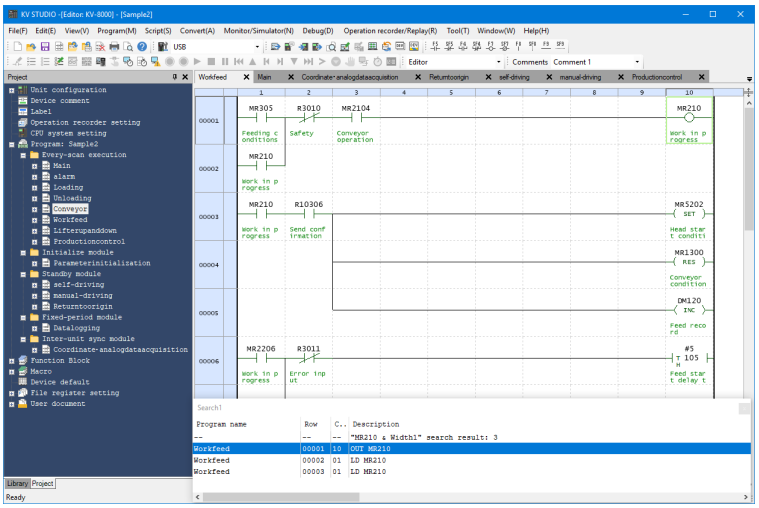
Alternative procedure
- (space) bar
- Right click and select "Cross reference (L)" -> "Create operand list (O)" from the menu.



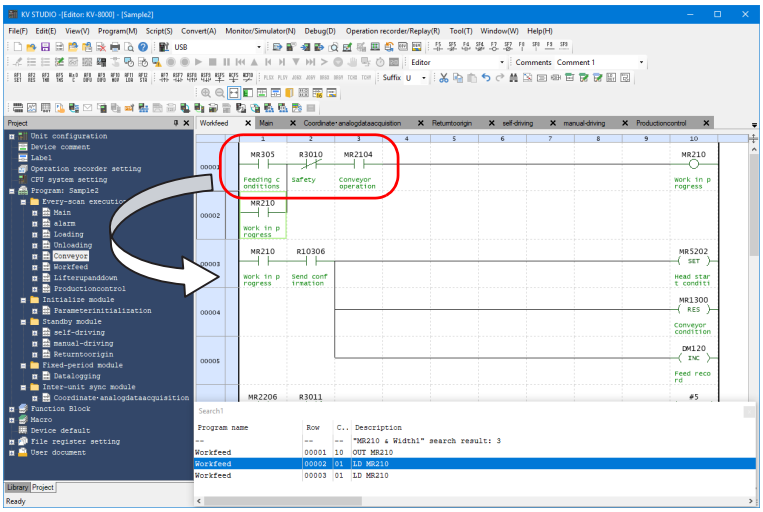
3 **Double-click the item that you want to check in the search result list.**

When you double-click a device that is displayed in the list, the cursor moves to the corresponding device (symbol) on the ladder.

Alternative procedure
- F3
- Shift + F3



4 **When you want to move the cursor focus to the next item, press F3.**



5 **To move the cursor focus to the previous item, press Shift + F3.**

Relation mapping

Edit(E) ▶ Relation Mapping(9) Ctrl + Shift + R

This is a function to search and display the circuit blocks that cause changes in the selected device, settings of KV STUDIO and device rewrite events from the human machine interface.

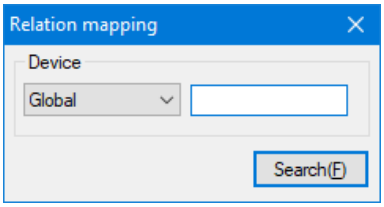
Point

It can be used only when the compatible model is KV-8000(A).

1 Select "Edit (E)"->"Relation Mapping (9)" in the menu.

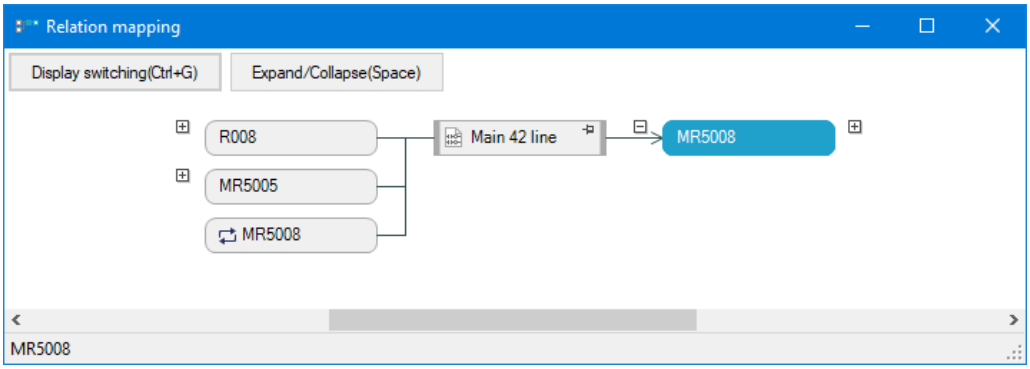
Alternative procedure
- With the target device selected in the ladder editing window, select "Relation Mapping 9)" in the right-click menu
- Ctrl + Shift + R

2 Enter the target device and click "Search (F)".

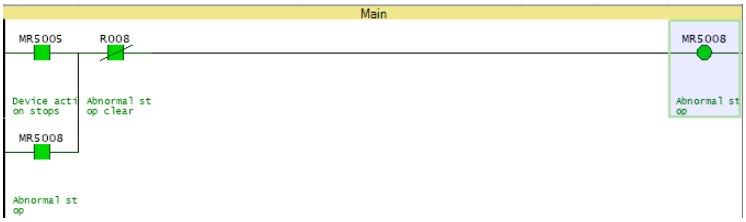


After entering the object device and clicking "Search (F)", the "Relation Mapping" window will be displayed with the input device as the object. The information on the ladder programs / functions with status change or the causes for status change can be displayed in the "Relation Mapping" window.

● Relation mapping window



● Ladder

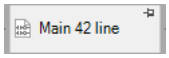


■ Description of blocks

● Device block

Display the devices used in the ladder block diagram. Click the "Display Switch" button on the upper left of the "Relation Mapping" window, and then press Ctrl + G to switch between the device comment display and device display.

● Ladder block

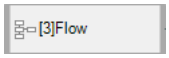


Display the module name and line number used by the searched device. When selected, the ladder program object is displayed in the selected ladder display window.

Point

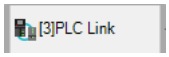
- **The ladder programs cannot be edited in the selected ladder edit window. To edit the ladder program object, right-click in the selected ladder edit window, select "Jump", and then edit the original ladder program.**
- **The scripts cannot be expanded.**

● Flow block



Display the unit number of the object unit when the searched device is used in the unit program (flow).

● Setting block



Display the unit setting name and unit number where the searched device is used. The unit settings displayed in the setting block are as follows.

- PLC link
- PROTOCOL STUDIO
- Industrial Ethernet (EtherNet/IP (Adapter), PROFINET, CC-Link IE Field, EtherCAT)
- Simple PLC link
- FL refresh setting
- Serial PLC link
- Device initial value setting
- Inter-unit synchronous refresh

● Device value change block

Displayed when the device value is changed for the searched device from the human machine interface or access window.

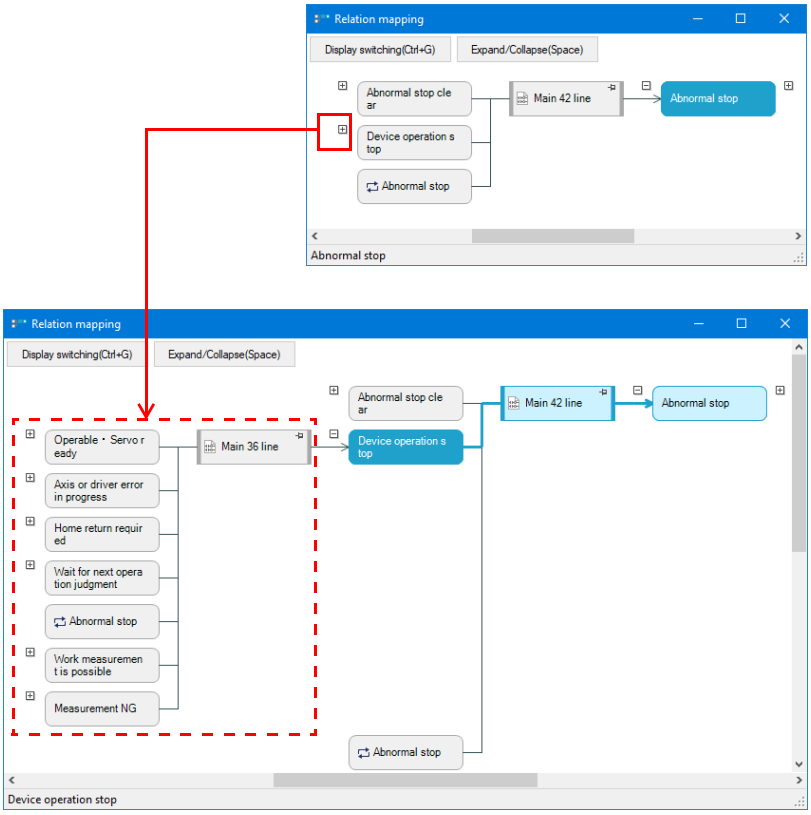
Point

The device value change block can be displayed only in the monitor / online edit / replay mode.

■ Relation mapping operation

● Additional search

Click the "+" button displayed in the device block to perform additional search and display for the cause that has changed the status of the device.



● Pinning

You can fix the display of the ladder block by clicking the icon displayed on the ladder block.

● Jump

It can only be used in ladder blocks, flow blocks and setting blocks. Right-click the block and click "Jump (J)" to display the following contents.

Item	Description
Ladder block	Move to the line of the object ladder.
Flow block	Display the object flow.
Setting block	Display the object setting screen.

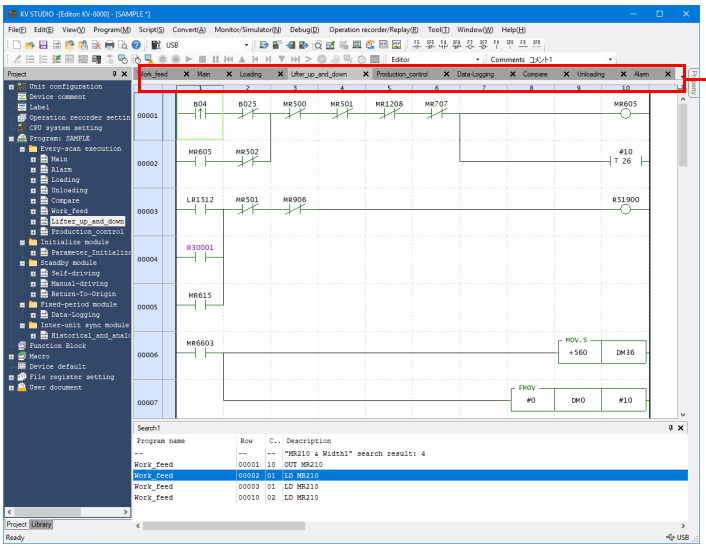
MEMO

Window menu

Using the Window menu can help you improve your program creation efficiency.
This section will explain how to use the "tabbed document" in the Window menu.

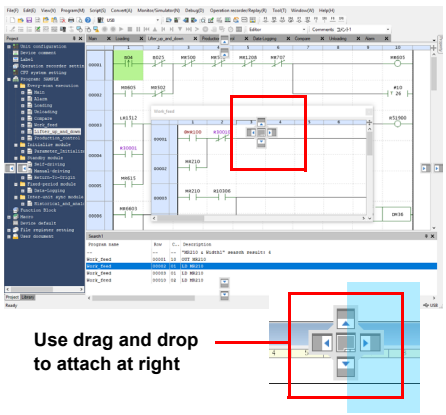
■ Tabbed documents

The module, macro can be displayed in tab format by selecting "Window(W)" → "Document with tab attached" from the menu of KV STUDIO Ver.8 or later.

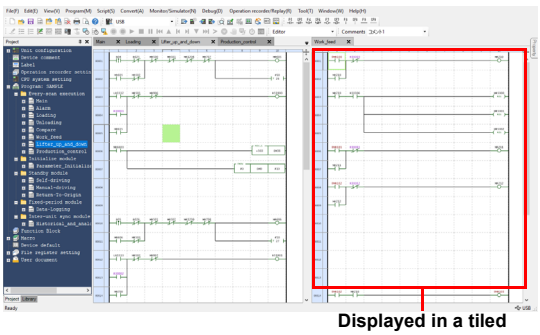


Tab
Use this key to select tabs and use Alt + F6 to switch between tabs.

Drag and drop a tab and use the docking function to easily display modules and macros in tiled format.



Use drag and drop to attach at right

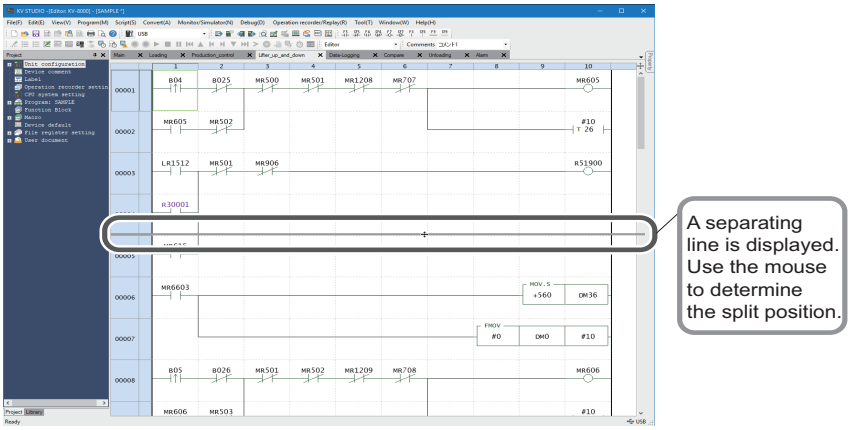


Displayed in a tiled format at right

■ Split

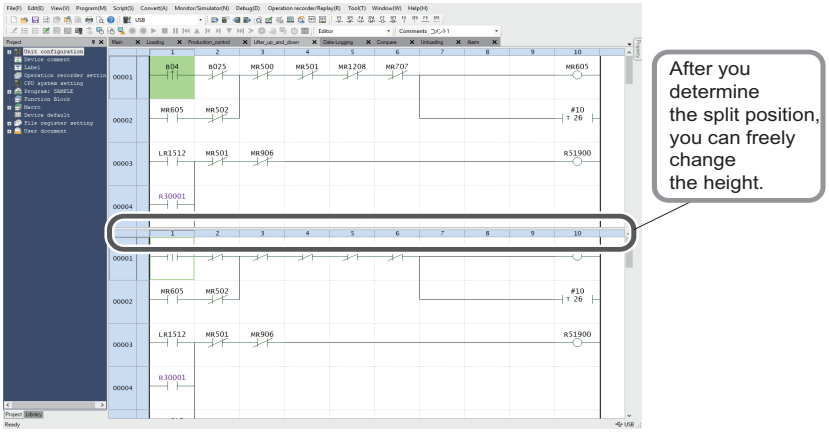
You cannot display a program in its entirety if the program is too long.
You can use the "Split" function to split the display of a program into two, so that you can view two parts of the program at the same time.

1 On the "Window" menu, click "Split."



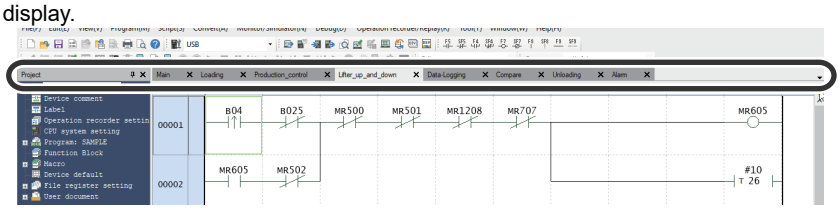
2 Click to fix the split position.

[Split screen]



When you determine the split position, the upper and lower screens both display the start of the program by default.
Use the scroll bars on the right side of the upper and lower screens to display the desired program positions.

Alternative procedure If you move the mouse cursor to the position shown in the following figure, the mouse cursor will change to "⇕," and you can drag the cursor to start splitting the program display.



Release the mouse cursor at the position that you want to fix the split.

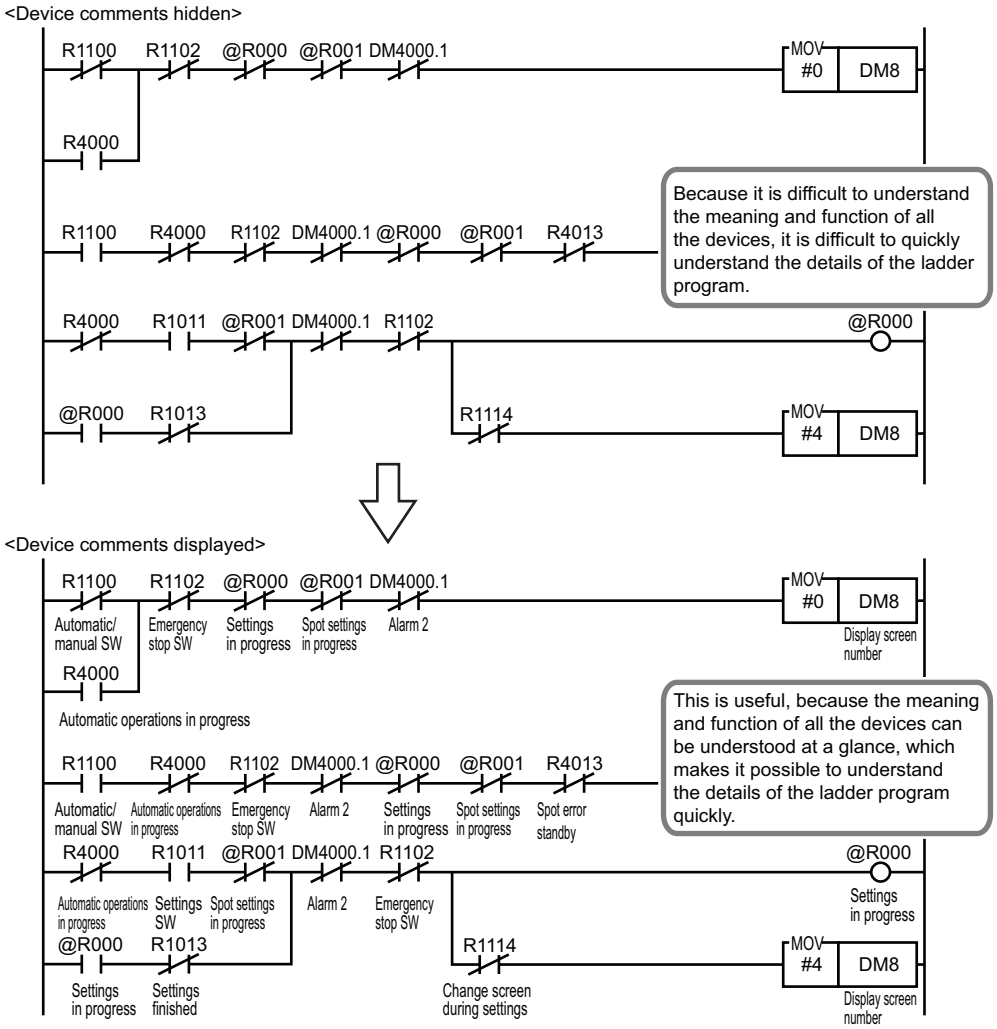
Reference

- You can copy and paste parts of the ladder program between the upper and lower screens.
- You can also set the display sizes of the screens independently.

Creating Clear, Legible Programs

Device comments

You can set comments for each device.
The set comments are displayed below each device when the ladder is displayed in KV STUDIO and can be displayed in the access window, making it easy to see the functions and contacts that are assigned to the devices.
Additionally, if device comments are present, it takes less time to understand the program when a person other than the program's creator views the program. Device comments can be used on both global devices and local devices. When bit processing is being performed on a word device, you can use device comments in units of bits.



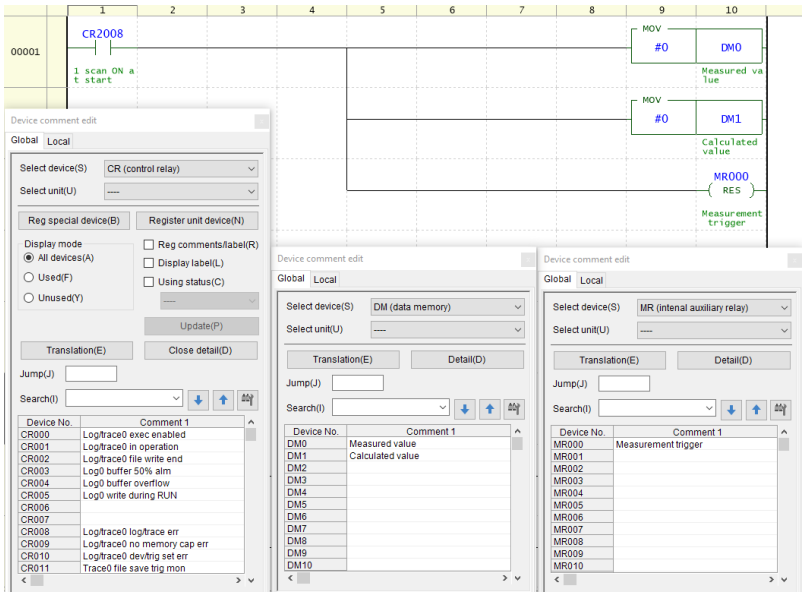
Point **You cannot set device comments on index registers (Z).**

■ Entry method

The following two methods are available for entering device comments.

- Enter the comments in the "Operand" dialog box (Ctrl + M).
- Enter the comments in the "Device comment edit" window.

Reference It is useful to write programs with the "Device comment edit" window displayed, because you can enter instructions while viewing the device comments. You can display up to three of these windows at the same time.



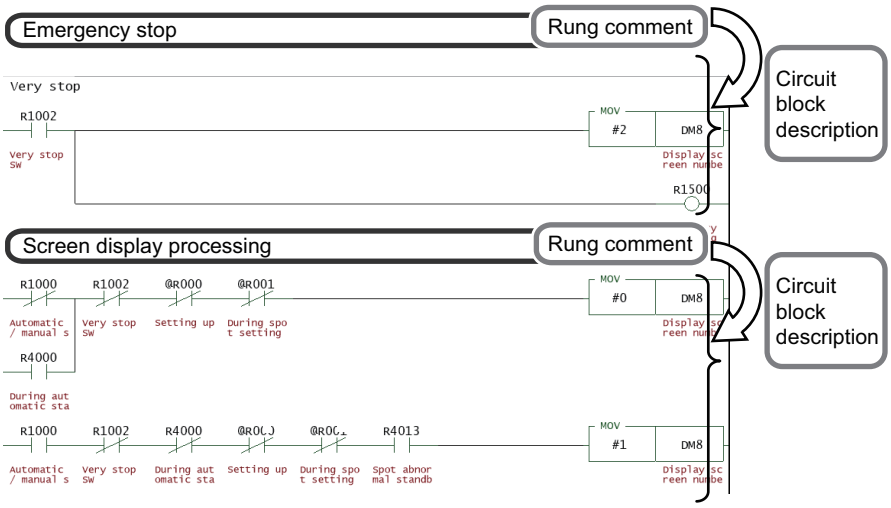
■ Display method

After you set device comments, on the "View" menu, click "Display cmnts" to show and hide the device comments on the ladder editor.

Alternative procedure Ctrl + Space

Rung comments

You can write comments on every rung in the ladder editing window.
Rung comments make it easy to understand the contents of each circuit block at a glance during program editing.
Also, you can display just rung comments in a list and move to the rung of a specific rung comment.



■ Entry method

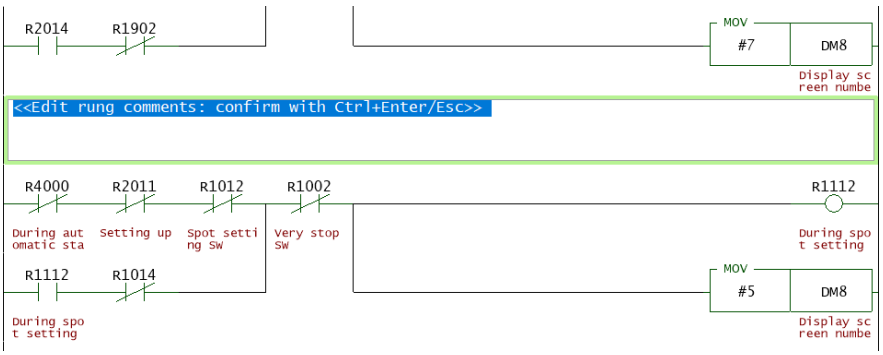
Follow the procedure below to enter rung comments.

1 **In the ladder editing window, position the cursor on the rung in which you want to enter a rung comment.**

2 **Right-click, and then click "Insert rung comment" in the menu that is displayed.**

Alternative procedure
- On the "Edit" menu, click "Insert," and then "Rung cmnt."
- Ctrl + ⏎

The rung comment editor is displayed as shown below. Enter the rung comment.



3 **Press** Esc **or** Ctrl **+** ⏎**.**

The rung comment is entered.

■ How to search for rung comments

Display the rung comments in the ladder editing window as a list. You can move to the rung whose comment you select from this list.

Follow the procedure below to search for rung comments.

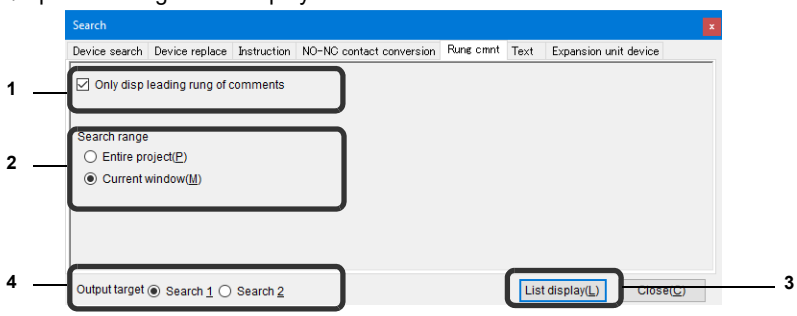
Alternative procedure Ctrl + L

1 Select the ladder editing window.

2 Right-click, and then click "Search" in the menu that is displayed.

Alternative procedure On the "Edit" menu, click "Search rung cmnts."

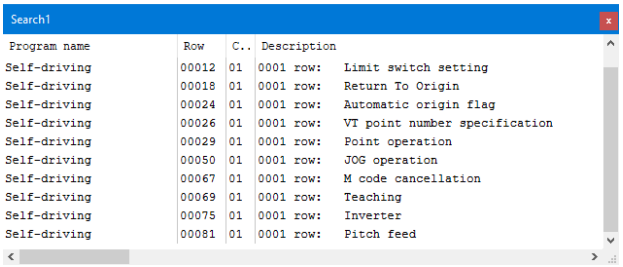
The "Search/replace" dialog box is displayed.



Number	Item	Description	Setting Range
1	Only disp leading rung of comments	If rung comments span multiple rungs, you can select whether all rungs will be displayed as search results or whether just the leading rung of multiple-rung comments will be displayed.	Checked/unchecked (Default: checked)
2	Search range	Select the range that the rung comment search will be performed over.	Default: whole program
3	List display	All rung comments in the target range are displayed in the output window.	-
4	Output taret	Select the output window to display when a list is displayed from [Search 1] and [Search 2].	Default: Search 1

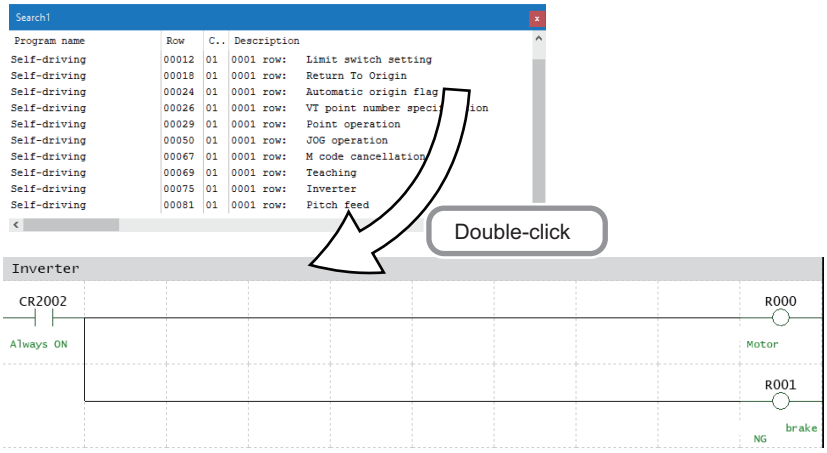
3 Click "List display."

The list of rung comments is displayed in the [Search 1]/[Search 2] output window.



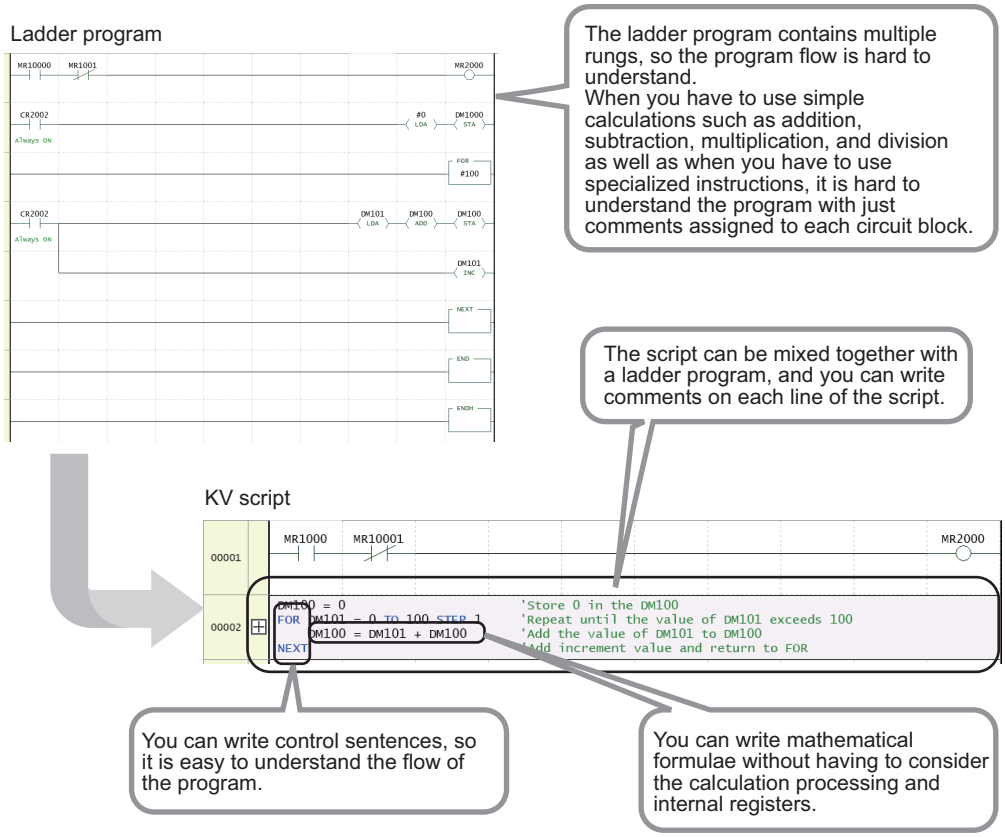
4 **Double-click the rung comment whose rung you want to jump to.**

The rung of the selected rung comment will be displayed in the ladder editing window.



KV script

Write calculations, character string processing, and other parts that are tedious to write in ladder programs as KV script.



For details, see the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Script Programming Manuals."

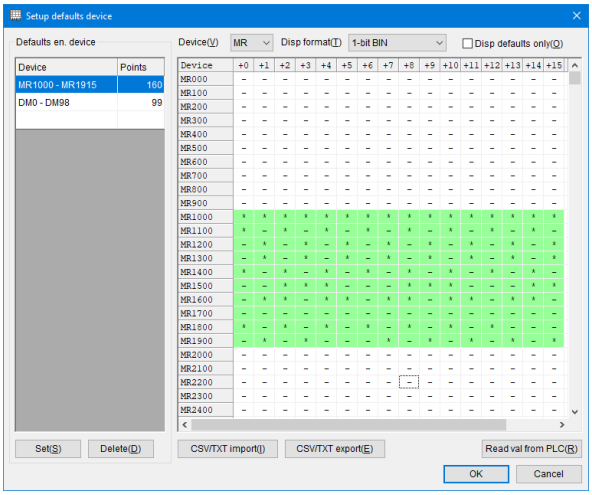
Program organization

To create programs that are easy to debug and reuse, write programs in modules (components) for each function and process, and assign easily understandable names to these modules. Also, turn frequently used ladder circuits into function blocks or macros (original instructions), and give them an easy-to-remember name so they can be used simply, like instructions.

"4-7 How Data Is Processed" (page 4-75)
"4-10 Function Blocks" (page 4-117)

Device defaults

Use the "Setup defaults device" dialog box to manage the programs in which initializing the devices is tedious when done by writing all the initializations in the ladder program. When operations start, the registered values are all written automatically.
Click "device default" in the workspace.



Key Points to Program Creation

This section explains the information and precautions that you should understand to create ladder programs.

■ Expanded ladder mode

In normal ladder notation, you can only position a single output instruction, such as OUT and TMR, on one rung of the ladder. If you use expanded ladder mode, you can position multiple output instructions on one rung, which reduces the number of rungs in the program, and thus clarifies the processing sequence.

In KV STUDIO, you can switch between expanded ladder input mode and normal ladder input mode. Follow the procedure below to switch between the modes.

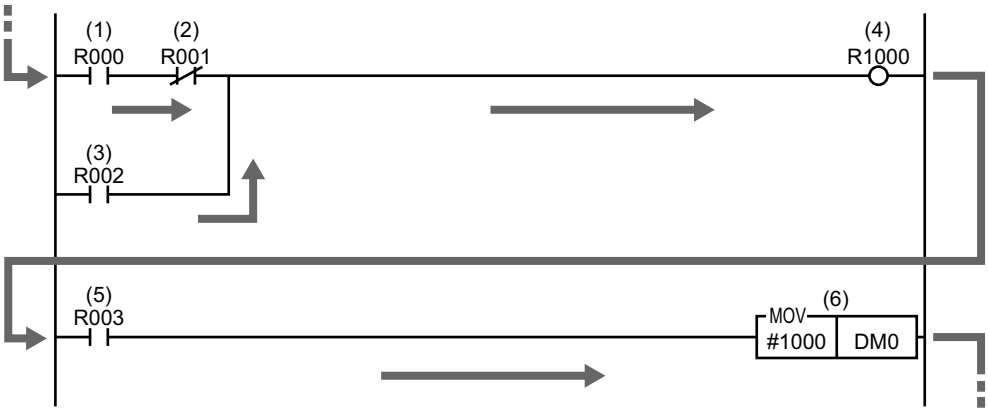
1 On the "Edit" menu, click "Expanded ladder mode."

Alternative procedure Click .

Reference
- When the expanded ladder mode is enabled, "Expanded ladder mode" on the menu is displayed with a check mark, and the icon is displayed with a frame surrounding it.
- When you select expanded ladder mode, the cursor does not automatically move to the right side of the rung when you enter an output instruction, and you can continue entering instructions on the same rung. When you are in normal ladder mode, the cursor automatically moves to the right side of the rung when you enter an output instruction.
- In the default settings, expanded ladder mode is disabled.

■ Ladder program execution sequence

As shown below, ladder programs are executed from left to right and from top to bottom.



Reference
- A program is executed in the order specified by its mnemonics list.

LD	R000	(1)
ANB	R001	(2)
OR	R002	(3)
OUT	R1000	(4)
LD	R003	(5)
MOV	#1000 DM0	(6)

- To set the module execution sequence, in KV STUDIO, open the "CPU system setting" dialog box, and then click "Execute sequence of modules."

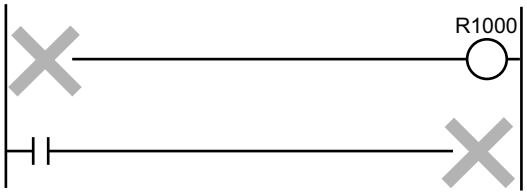
"Module Execution Sequence" (page 4-109)

■ Ladder programs that cannot be converted

Typical patterns that cause conversion errors are shown below.

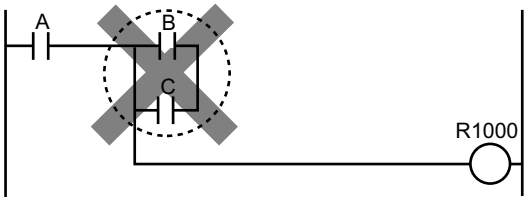
● Rail or output line not connected

Ensure that all contacts and coils are on lines that are connected from the rail to the output line, and that no items are unconnected.



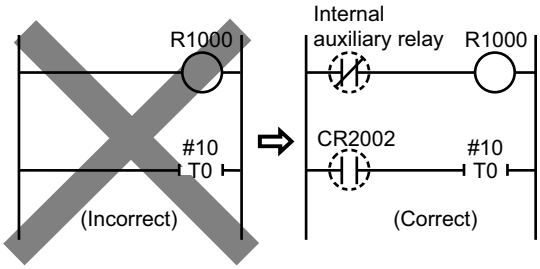
Reference

The ladder program to the right is unconnected, so a conversion error will occur.

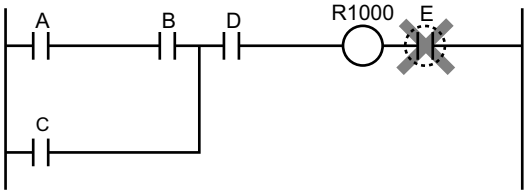


● An instruction that cannot be written directly from a rail is in use

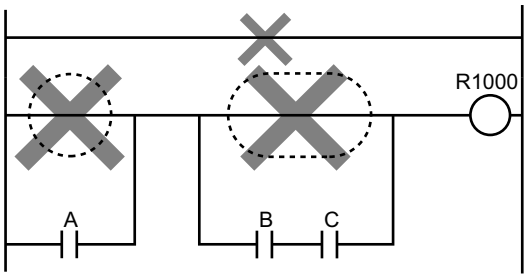
You cannot write instructions such as those for output coils, timers, and counters directly from the rail.
If execution conditions are not required, insert the b contact of an unused internal auxiliary relay or control relay CR2002 (the contact that is always turned ON) as a dummy.



● An instruction that cannot be written directly on an output line is in use

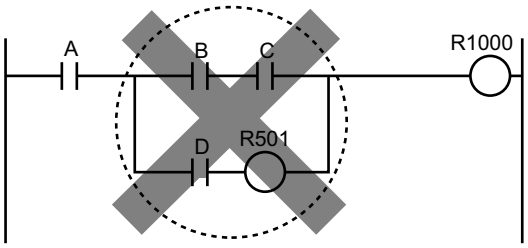


● Illegal short circuit present

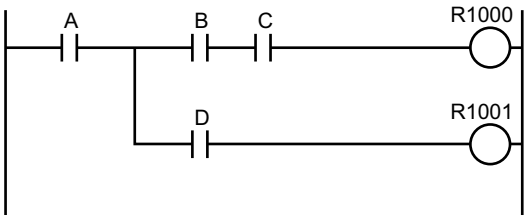


● Output coils are connected in parallel

● Circuit groups that contain output coils cannot be connected in parallel.



However, as shown in the figure to the right, you can connect circuit groups in parallel provided each circuit group is connected to the output line.



● Missing END or ENDH instruction

Each program needs an "END" or "ENDH" instruction.

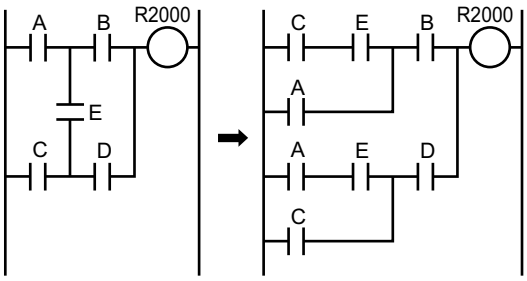
Reference

When you use KV STUDIO to create a new program, the "END" or "ENDH" instruction is automatically inserted. Write the "END" or "ENDH" instruction in a ladder program. If you write these in a script, the program may not operate correctly.

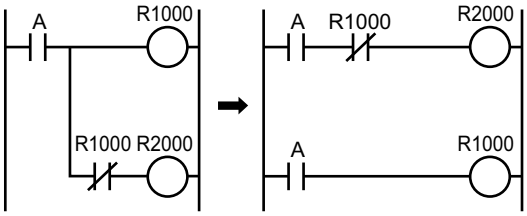
■ Ladders that must be rewritten

Non-convertible ladders | Convertible ladders

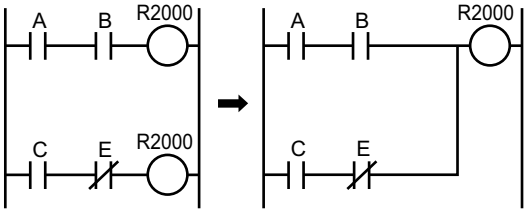
- You cannot program circuits such as those shown in the figure on the right. Rewrite these circuits.



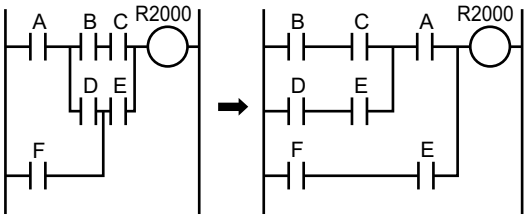
- In the circuit shown in the figure to the right, R2000 will not be output. By rewriting the circuit, the output can be generated for only one scan.



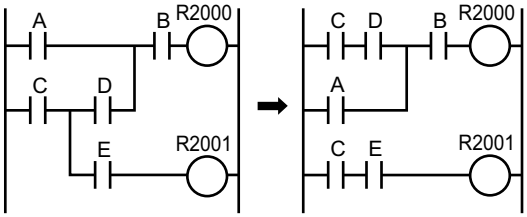
- If you use the same coil twice, as shown in the figure to the right, the later part of the program will be given priority, and the earlier part will be ignored.



- You cannot program circuits such as those shown in the figure on the right. Rewrite these circuits.



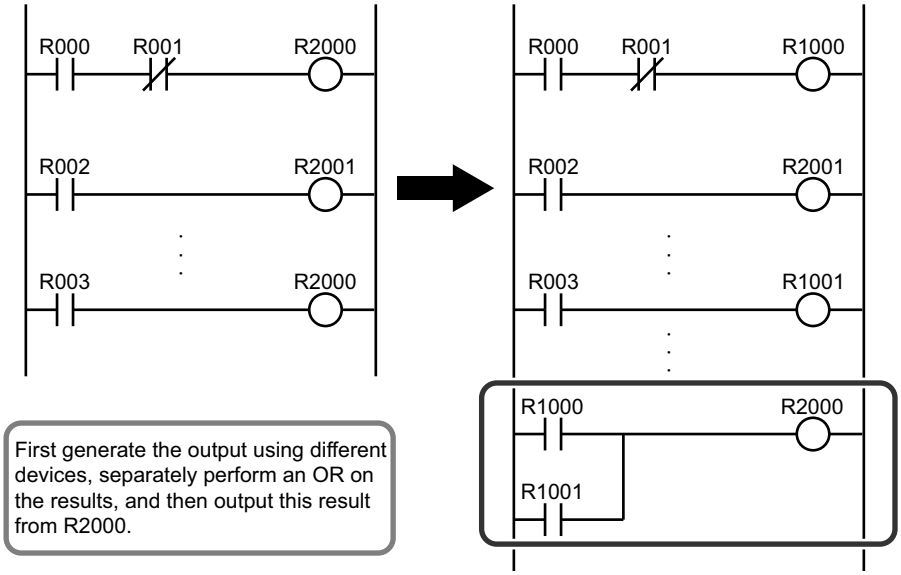
- You cannot program circuits such as those shown in the figure on the right. Rewrite these circuits.



■ Double coil

If you use two or more of the same OUT or OUB instruction for the same device within a project, the last instruction to be executed within a scan is given priority, and the other instructions are ignored.
If the instructions are on separate rungs or exist in different programs, they are processed as shown below.

(Example) As shown in the figure below, when the R2000 output instruction is used twice



A conversion error will not occur for double coil, but do not use them because they make operations complicated.
To check whether double coil exists, in KV STUDIO, on the "Convert" menu, click "Double coil check."

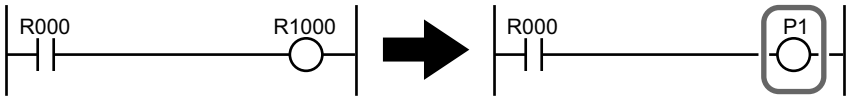
Point

- **If you call the same function block or macro two or more times, a double coil will occur if an OUT or OUB instruction that uses a global device is present.**
- **Even if a double coil, there is no effect if it is described in a module or function block that is not running.**

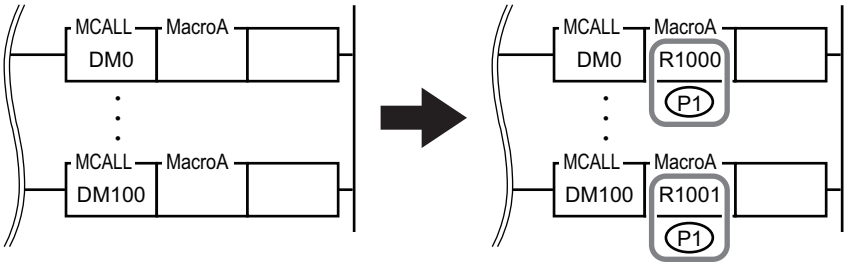
● How to avoid double coils between the same function block or macro

If you call a function block or macro with an OUT/OUB instruction that uses a global device 2 times or more, a double coil will occur. You can avoid this by processing as shown below.

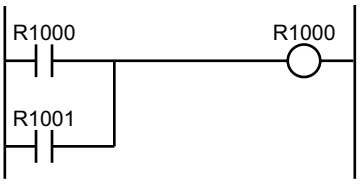
(1) Set the operand of the OUT/OUB instruction in the function block or macro to an IN-OUT type argument or argument P.



(2) Use the argument added in (1) that is specified in the function block or macro call instruction to specify a different internal auxiliary relay per call.



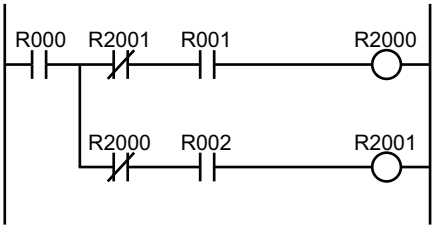
(3) Add the ladder program to output separately using OR outside the function block or macro.



■ Interlock

In programs, when one party acts, another party cannot act to prevent simultaneous actions by both parties. This exclusive processing composition mode is called interlock.
Although this is often necessary for actual equipment, its operations are very complicated to explain, so it is omitted from the 📖 sample programs introduced in the KV-8000/7000/5000/3000/1000 Series and KV Nano Series "Instruction Reference Manuals."

(Example) A circuit that prevents R2000 and R2001 from being turned ON at the same time



When R000 and R001 turn ON, R2000 turns ON.
In this state, even if R002 turns ON, R2001 will not turn ON because the b contact of R2000 is inserted.

If you use the bit that is turned ON during module or macro execution, you can easily create an interlock circuit.

📖 "Bit That Is Turned ON during Macro Execution" (page 4-189)
📖 "Macro System Devices" (page 4-188)

■ Reducing the scan time

When the scan time becomes long, the update period of I/O and values becomes long, so the response of the machines and equipment worsens. Also, if the scan time exceeds 300 ms, a "Scan time over error" (CPU error E30) will occur, and it will not be possible to continue operations.
You can reduce the scan time by using the following methods.

● Stopping unnecessary modules

If modules that do not need to operate are being kept in the execution state, use the module instructions (MDSTRT and MDSTOP) to only execute the modules when necessary.
"Module instructions" (page 4-115)

● Jumping over unnecessary ladder circuit blocks

Enclose ladder circuit blocks that do not need to operate in conditional branches (between CJ or NCJ and LABEL instructions) or steps (between STP and STE instructions) to only execute the circuit blocks when necessary.

● Reducing looped content (between FOR and NEXT instructions)

If there are unnecessary parts in the ladder circuit blocks that are repeatedly executed between FOR and NEXT instructions, the repeated processing time is added to the scan time. Write instructions for processing that does not need to be executed repeatedly outside of the FOR and NEXT instructions.

Reference
The same concept holds when you use the CJ or GOTO instructions to create programs with looped processing.

● Specifying devices with even numbers to 32-bit and 64-bit processing operands

Use even-numbered devices to specify 16-bit word devices such as DM, W, and TM for the operands of instructions that process 32-bit or 64-bit data such as those with the .D, .L, .F, or .DF suffix. If you use two words from an odd number, the processing speed will decrease.

Reference
In the same manner, for DM and TM, use even numbers for local devices. During conversion, even-numbered local devices are always assigned to even-numbered global devices, and odd-numbered local devices are assigned to odd-numbered global devices. Therefore, more device areas than those assigned to the program may be used.

■ Effective use of modules, function blocks and macros

To organize your programs by using modules, function blocks and macros effectively, use local devices and variable to the maximum extent. Precautions for using local devices are explained below.

- Except for the devices assigned to the CPU unit and the units that you are using, the devices referenced by the connected peripherals (such as touch panels), the devices that will be shared among the entire project, and the devices that need to transfer data between modules, use all devices as local devices.
- The first local device is number 0. Use local devices with as small numbers as possible.
- Because R and DM devices are assigned to units, an area that is used as global devices is required. Give priority to MR/LR and EM when using local devices.

"4-12 Local Devices" (page 4-200)
"4-5 Variable" (page 4-55).

■ Separate usages for index modification and indirect specification

There are two methods for using instruction operands to change the specified devices during operation: index modification and indirect specification.

<Index modification>
Better The specification method is simple.
Good You cannot change the type of devices that are specified.
Good You cannot use index register Z as a local device.
→You can use the "ZPOP" and "ZPUSH" instructions without considering how many index registers you can use.

<Indirect specification>
Better You can change the type of devices that are specified.
Good The specification method is complicated (to process addresses, you have to use dedicated instructions).

Unless you want to change the device type, we recommend that you use index modification, which is easier to handle.

"Modification according to index registers" (page 4-44)

How to Perform Debugging

Eliminating errors in a program is called debugging.
If you use the KV STUDIO monitor function, you can view the state of the CPU unit that is actually connected.
In addition, you can perform more efficient debugging by using the online editor function, which enables you to edit ladder programs during monitoring.
If you use the simulator function, which enables you to view the operation of a ladder program just from a PC even if you do not have a CPU unit, you can simulate the operation of the CPU unit on the PC to perform debugging.
By using the "Replay" function, the data when the equipment fails can be saved and played backed later with KV STUDIO, which can be used to analyze the fault cause.

Conversion

Before you use the monitor or simulator, you have to convert the ladder program that you have created to a format that can be processed by the CPU unit.

On the "Convert" menu, click "Convert."

Alternative procedure
- Click .
- Ctrl + F9

Reference
- On the "Monitor/Simulator" menu, click "PLC transfer → monitor mode" to automatically convert and transfer the data and to start the monitor.
- On the "Monitor/Simulator" menu, click "Simulator" to automatically convert the data and start the simulator.

<If conversion finishes normally>

The message "Conversion end OK" is displayed in "Conversion" in the output window.

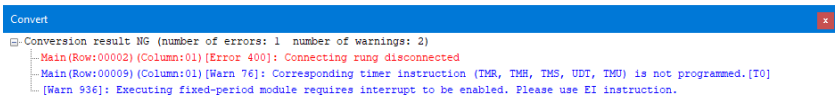
The usage percentage of the program capacity is displayed in the "Convert result" dialog box.

"Program Capacity" (page 4-14)

<If conversion fails>

The error details are displayed in "Conversion" in the output window.

Double click an error to jump directly to the corresponding location. Correct the errors one at a time in order.



"Key Points to Program Creation" (page 4-258)
"Appendix 1 Error Message Lists" in the "KV STUDIO User's Manual"

Transfer

On the "Monitor/Simulator" menu, click "Transfer to PLC" to transfer the converted ladder program to the CPU unit.

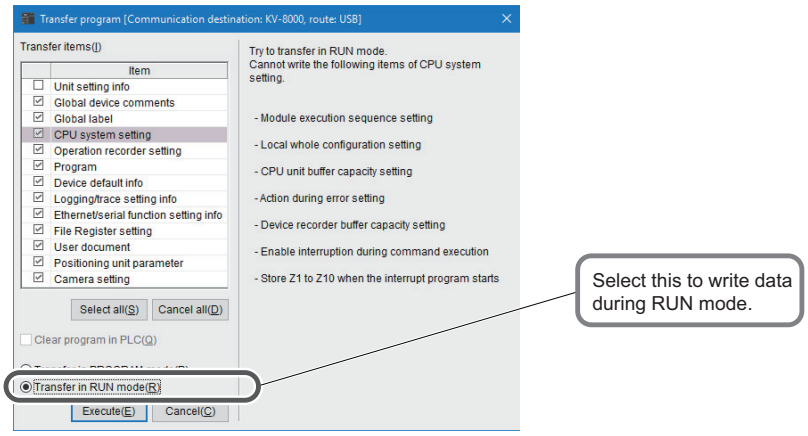
Alternative procedure Click .

Point
- **With the default settings, the programs stored on the PLC will not be deleted when you transfer a program. To delete unused programs when you transfer programs, select the "Clear program in PLC" check box in the "Transfer program" dialog box.**
- **If the function version of the project stored in the PLC unit and the function version of the project you are transferring differ, all clear must be executed before transferring.**

Reference On the "Monitor/Simulator" menu, click "PLC transfer → monitor mode" to automatically convert and transfer the data and to start the monitor.

■ Writing data during RUN mode

You can write data during RUN mode if the CPU unit is in the RUN state when you transfer a ladder program. This function enables you to overwrite the executing ladder program when the CPU unit is in the RUN state. You can make fine operation adjustments and parameter changes without stopping the device and the production line.
The scan time of writing during RUN can be extended by maximum 2ms.



CAUTION	**Depending on the status of the control equipment connected to the CPU unit, it may be dangerous to write data during RUN mode. Exercise caution when executing this function.**

● Data that you cannot write during RUN mode

- Projects that are different from the projects stored on the CPU unit
- Unit settings
- Execution sequence of modules
- Entire assignment setting of local devices
- Protection setting
- Whether interrupts are enabled during instruction execution
- Whether Z1 to Z10 are saved when interrupt programs start
- CPU unit buffer capacity
- Device recorder buffer capacity setting
- Action when error occurs

● Conditions under which you cannot write during RUN mode

- When the number of used local devices has been changed
- When a function block call has been added or deleted*1
- When the function block call sequence has been changed*1
- When a new function block call has been added*1
- When an instance is added, deleted or renamed
- When the order of instances is changed in the variable edit window*2
- When the function block/function argument settings have been changed
- When a macro instruction has been added or deleted
- When the sequence of macro instruction calls has been changed
- When macro argument settings have been changed
- When exceeding the local work area that automatically assigned by variable adding/changing/deletings or ST/Script editing
- When the project language configuration is modified

*1 For a function block executed using FBCALL/FBSTRT with CPU function version 2.0 or later, writing during RUN can be performed even if these operations are performed.
*2 When the CPU function version is lower than 2.0, these restrictions do not exist since it does not support instances.

Point

- **When data finishes being written during RUN mode, even if the execution condition ON/OFF state changes, differential instructions within the range of data written will not be executed.**
- **Even if you change the set value of a timer or counter and then write the data during RUN mode, the current values will not have changed when the data finishes being written.**

Monitor

The monitor is used to transfer the ladder program that you have created with the ladder editor to the CPU unit, execute the program, and view the operating status.

You can use the monitor to:

- View the program's operating status in real time with the ladder diagram and real-time chart monitor.
- Switch the operation mode of the CPU unit.
- Force the input contacts to turn ON or OFF from the monitor screen by carrying out forced set/reset.
- Easily change the device data and the set and current values of timers and counters.
- Specify the range of data, such as the current value of a device, to save or load the data.
- Return some devices to their initial values.

On the "Monitor/Simulator" menu, click "PLC transfer → monitor mode" to start the monitor. Before you carry out this process, connect the CPU unit and the PC, and configure both devices so that they can communicate with each other.

Alternative procedure
- Click .
- Ctrl + F8

Reference
The current status (such as monitor, online editing, or editor) is displayed in the KV STUDIO title bar.

Use the toolbar or the menus to operate the monitor.

For details on the different functions, see "Chapter 9 MONITOR & SIMULATOR" in the "KV STUDIO User's Manual."

Online edit

You can use the online editing function to change ladder programs directly from the KV STUDIO monitor screen. This function is useful when you want to make on-site adjustments during the final debugging of program creation and when you want to change simple programs.
While you are monitoring, on the "Monitor/Simulator" menu, click "Start Online Edit" to start online editing.
The location being edited is displayed in light blue, which enables you to see which cell is being edited at a glance.
When the program finishes being transferred, the light blue of the location being edited returns to the original background color.

Alternative procedure
- Select "Online edit" from the mode list.
- Click .
- F10

CAUTION	**If the online editing function is used while the PLC is in the RUN state, incorrect operations may cause serious damage to equipment and serious injury to workers. Thoroughly check the following precautions to ensure correct use of this function.** **• Thoroughly check the area around the equipment, and ensure that no damage to equipment and injury to workers will arise even if problems occur when you use the function.** **• Do not add circuits that operate immediately after the program is transferred (such as circuits whose execution condition is the B contact). Doing so may cause the equipment to run out of control.** **• If you delete output instructions that have been turned ON (such as OUT) and then transfer the program, the output will be retained as being turned ON.**

Reference

When you transfer data during online editing, only the changed circuit blocks are written to the PLC. However, changed programs are written in their entirety in the following situations.
- When several modules have been changed.

When editing the ST/Script which exceeds the local work area
- When variable have been added so that the local work area is exceeded.
- When the program inside of a function block has been edited (the function block is transferred in its entirety).
- When the program inside of a macro has been edited (the macro is transferred in its entirety).
- When the number of individual local device assignments has been changed.
- When a macro call has been added.
- When a function block call has been added.
- When the ENDH instruction has been edited.
- When a structure is edited (Modules or function blocks that use the edited structure are transferred.)
- When global variables and function block arguments used in multiple modules or function blocks are edited (Modules and function blocks that use the edited global variables and function blocks are transferred.)

For limitations and precautions of writing data during RUN mode, see "Writing data during RUN mode" (page 4-267).
For details on the online editing function, see "9-9 Online Edit" in the "KV STUDIO User's Manual."

Simulator

The simulator simulates on a PC without connecting the PLC the operation of the ladder programs that you have created with the ladder editor.
You can use the simulator to:

- View the operating status of the program by using the ladder monitor, registration monitor (time chart), and batch monitor.
- Perform simulations with various execution methods such as continuous scan/continuous step and single step/single scan.
- Perform efficient debugging by using convenient functions such as reverse step and reverse continuous step.
- Easily change the set values and current values of timers, counters, and devices.
- Register forced sets and resets of contacts.

On the "Monitor/Simulator" menu, click "Simulator" to start the simulator.

Alternative procedure
- Select "Simulator" from the mode list.
- Click .
- Ctrl + F2

Reference The current status (such as simulator, monitor, or editor) is displayed in the KV STUDIO title bar.

Use the toolbar or the menus to operate the simulator.

For details on the different functions, see "9-11 Simulator - Executing Simulator" in the "KV STUDIO User's Manual."

■ Differences with the monitor

You can use the simulator to control the operating status to a finer degree than the monitor by using functions such as single scan execution, single step execution, and single step reverse execution. Because you can only simulate the operation of the CPU, you cannot check the operation of expansion units. In addition, there are limits on the operation of some CPU built-in functions and devices.

For details, see "9-11 Simulator - Simulator Restrictions" in the "KV STUDIO User's Manual."

Reference There may be variations in the numeric values of timers because of the performance of your PC.

Simulator edit

The simulation editing function allows you to directly edit ladder programs during "KV STUDIO" simulator operation. This function is useful when you want to make on-site adjustments during the final debugging of a created program or editing a simple program.
When monitor/online edit is not running, select [Monitor/Simulator(N)] ⟶ [Simulation edit] from the menu to start simulation editing.
The location being edited is displayed in light blue, which enables you to see which cell is being edited at a glance.
When the light blue edited cells have been transferred, they return to their original background color.

Alternative procedure
- Select "Simulation edit" in the mode bar.

Point
- **Simulation edit is available only to the CPU unit of the KV-8000/7000 Series.**
- **You cannot switch from Simulation edit to monitor mode or online edit mode. Instead, first switch to edit mode, then to monitor mode and finally to online edit mode.**

For details on what functions are available during simulation edit, refer to "Online edit" (page 4-269).

Replay

Replay is a function that plays back the operation record before and after the set trigger condition is met. The operation record can be checked while replaying, stopping, rewinding and frame advance, so that the situation when the fault occurs can be mastered. Select if the replay mode is switched to after reading the operation record in the CPU unit or opening the already acquired operation record.

The operation record contents that can be played back during replay are as follows.

Type	Details
Device	Data for different scan times in the buffer memory of the PLC device and extension unit before and after the trigger condition is met
Camera	Video data shot with KV-CA02 by different frame rates before and after the trigger condition is met
Event/error	Event log of errors or device value changes, etc. before and after the trigger condition is met
Project data	Project data when the trigger condition is met

The following functions can be achieved during replay.

- The device data when a fault occurs can be checked with the ladder program for each scan, or checked against the frame-by-frame video shot by the camera.
- The events such as change in the device value or error can be selected to specify the position to play back the operation record, so that the status when an error occurs in the equipment can be easily mastered.
- The device data when a fault occurs can be registered to the real-time chart monitor and displayed as time series data.
- The device data when a fault occurs can be monitored on the HMI after used with the VT5 series.

To start replaying, select "Operation Record / Replay (R)" → "Replay Mode (R)" from the menu.

Alternative procedure

•Select "Replay" in the mode bar

•Select "Operation Record / Replay (R)" → "Read operation record in the CPU unit → Replay Mode (O)" from the menu

•Select "Debug (D)" → "Event / Error Monitor (E)" from the menu, and click "Replay Mode"

Point

- **To use Replay, KV STUDIO Ver.10.0 or later and KV REPLAY VIEWER are required.**
- **The replay function can be used only when KV-8000(A) is selected as the compatible model.**
- **It is impossible to change from the replay mode to other modes than the editor mode.**
- **For details on the replay mode, refer to "KV STUDIO User's Manual".**
- **If the project started when switching to the replay mode is different from the project of the operation record, a dialog will be displayed to close the active project and start the project of the operation record or play back the operation record using the active project.**

Monitor functions

This section explains the functions that are useful when using the monitor and simulator to debug efficiently.
For details on how to operate each function, see the "KV STUDIO User's Manual."

■ Changing the current value and status of devices

● Bit device ON/OFF

Double-click the cell whose bit device status you want to change. Each time that you double-click, the status will be inverted.

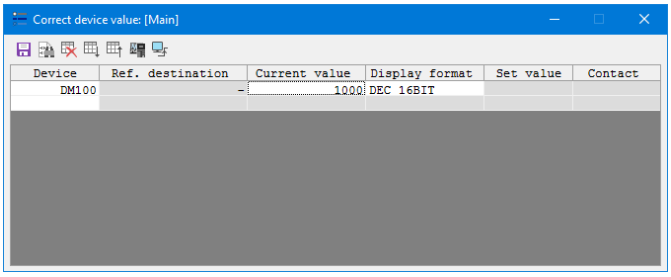
Alternative procedure Select the cell whose bit device status you want to change, and then press Space.

Point **During monitoring, the contacts and external I/O relays of timers and counters cannot be turned ON and OFF.**

Reference To forcibly turn the external I/O on and off when monitoring, select "Compulsory setting/ reseting registration (X)" from "Debug (D)" from the menu.

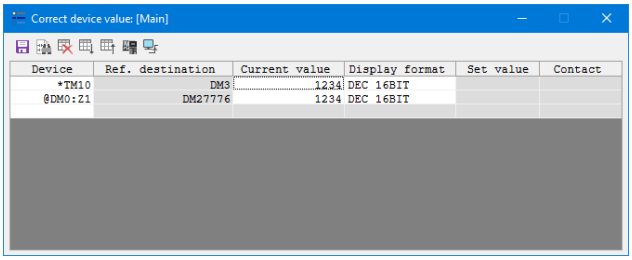
● Correct device value window

Select the cell that corresponds to the current value of the device that you want to change, and then press F2.



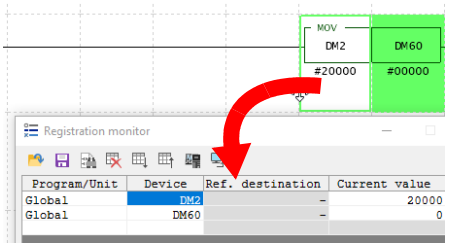
The "Correct device value" window is displayed. If you select multiple cells and then press F2, all the devices contained within the selected range are displayed.
You can change the values by entering values in the "Current value" column.

Reference If you select an instruction in which index modification or indirect specification is being used, the referenced device is displayed.



● Register and batch monitors

To register a device in the Register monitor or Batch monitor, select the instruction that includes the device you want to monitor. Move the cursor to the edge of the cell until it becomes a " ✥ ," and then drag the cell to the "Register monitor" dialog box.



Reference

- If you select multiple cells and drag them all to the dialog box, all the devices contained within the selected range will be registered.
- You can also enter devices directly in the "Register monitor" dialog box.

● Watch window

This function is used to automatically monitor all the devices that are contained within the circuit block of the selected cell.
To show and hide this window, on the "Monitor/Simulator" menu, click "Hide/show monitoring window."

● Real time chart monitor

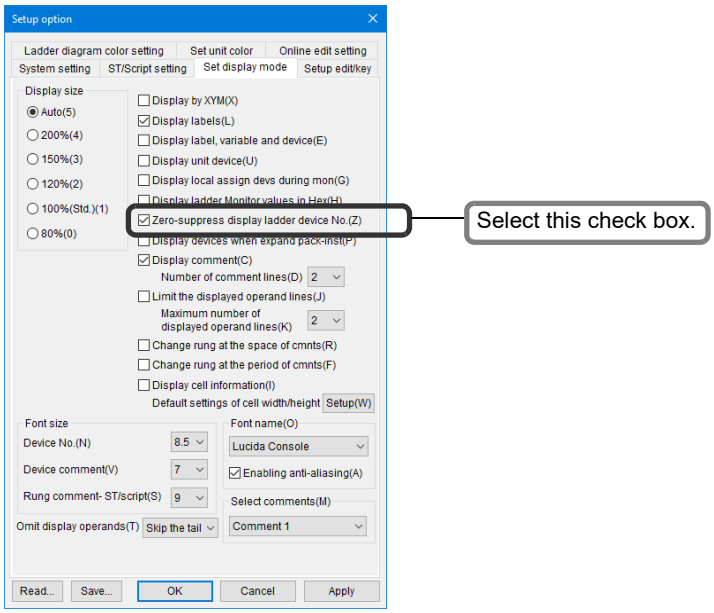
This displays the information of the devices registered in the Register monitor as a time chart.
Right-click the "Register monitor" dialog box, and then click "Real-time Chart Monitor" from the menu that appears.
You can display previously acquired data and display the values of the selected points.

Alternative procedure On the "Monitor/Simulator" menu, click "Real-time Chart Monitor."

See "9-6 Real-time Chart Monitor" in the "KV STUDIO User's Manual."

■ Displaying the assignment destinations of local devices and local labels

In the monitor or simulator, you can display the local devices and local labels in the ladder program that you have created as the assignment destination devices that are assigned at the time of conversion. On the "Tool" menu, click "Option" to display the "Setup option" dialog box. On the "Set display mode" tab, select the "Display local assign devs during mon" check box.



Alternative procedure From the menu, select [Display] → [Display local assign devs during mon(W)].

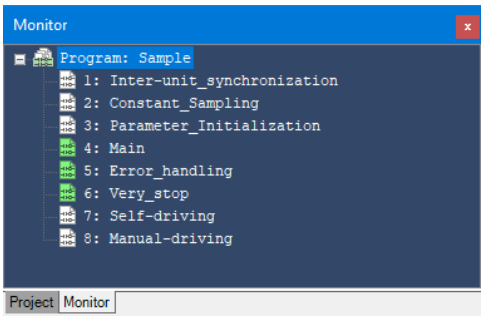
Reference Local labels are assigned to local work areas.

"Displaying the assignment destinations of local devices and local labels" (page 4-275)

Point **The destination of the variable is not displayed.**

■ Viewing the execution status of programs

In the monitor or simulator, click the "Monitor" tab at the bottom of the workspace to display the program monitor, which you can use to view this information.



■ Monitoring instructions

You can monitor the devices used in the PIDAT instructions that are written in the ladder program. Right-click a PIDAT instruction, and on the menu that appears, click "Command monitor" and then "PIDAT."
This is useful when monitoring measured values, set values, and other parameters that are used with PIDs.

Alternative procedure On the "Monitor/Simulator" menu, click "Command monitor," and then "PIDAT."
For details, see "Command monitor" in the "KV STUDIO User's Manual."

■ Monitoring rising and falling edges of devices

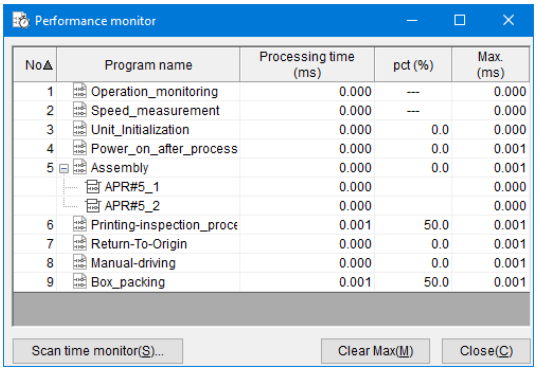
Right-click the instruction that includes the device you want to monitor, and then click "Derivation monitor" in the menu that appears.

Alternative procedure On the "Debug" menu, click "Derivation monitor."
For details, see "Derivation monitor" in the "KV STUDIO User's Manual."

■ Monitoring the execution time in program and interrupt program units (Performance monitor)

You can monitor the execution time of the programs or interrupt programs currently being executed.

1 **Switch to monitor mode. On the "Debug" menu, click "Performance monitor."**



No	Program name	Processing time (ms)	pct (%)	Max. (ms)
1	Operation_monitoring	0.000	---	0.000
2	Speed_measurement	0.000	---	0.000
3	Unit_Initialization	0.000	0.0	0.000
4	Power_on_after_process	0.000	0.0	0.001
5	Assembly	0.000	0.0	0.001
	APR#5_1	0.000		0.000
	APR#5_2	0.000		0.000
6	Printing-inspection_proce	0.001	50.0	0.001
7	Return-To-Origin	0.000	0.0	0.001
8	Manual-driving	0.000	0.0	0.001
9	Box_packing	0.001	50.0	0.001

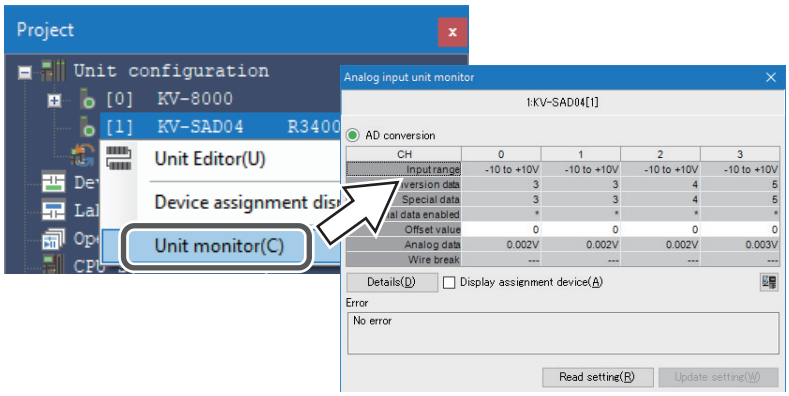
■ Monitoring units

In the workspace, right-click an expansion unit, and then click "Unit Monitor" in the menu that appears to display the unit monitor.

Alternative procedure

- On the "View" menu, click "Unit Monitor."
- When you are monitoring from the Unit Editor, you can display the unit monitor by right-clicking a unit and clicking the appropriate menu item in the menu that appears.

(Example) KV-SAD04



Point **This cannot be displayed when you are using the simulator.**

5

CPU BUILT-IN FUNCTIONS

This chapter explains the functions that the KV-8000 Series CPU unit makes available through the access window, communication ports and user memory.

5-1 Overview of Built-In Functions 5-2
5-2 Memory Card 5-3
5-3 CPU Memory (User Memory)..... 5-27
5-4 Recipe Load/Save Function 5-50
5-5 Logging/Tracing 5-57
5-6 Operation Recorder Function..... 5-86
5-7 USB Communication..... 5-109
5-8 Ethernet/EtherNet/IP Communication 5-110
5-9 Access Window.....5-111
5-10 Inter-unit Synchronization Function..... 5-165
5-11 Cam Switch 5-178
5-12 Clock Function 5-187

5-1 Overview of Built-In Functions

This section describes the built-in functions that you can use with the CPU unit.

What You Can Do with the Built-In Functions

Access window

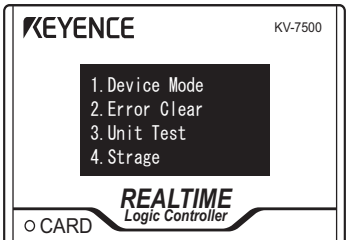
Device value monitoring

Alarm check/recovery

Memory card/CPU memory operations

Unit monitor

- "Access Window" (page 5-111)



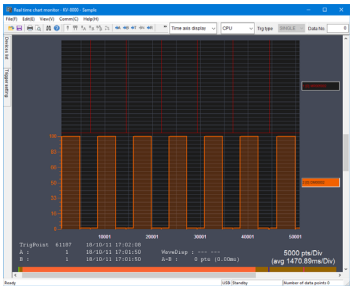
Logging/trace

Logging data

- "Logging/Tracing" (page 5-57)

Improved debugging efficiency

- "Logging/Tracing" (page 5-57)
- Real-time chart monitor
  "KV STUDIO User's Manual"



Memory card/CPU memory

Load/save

Performance management

Save logging data

Control of memory card and CPU memory through the ladder program

Save user documents

File register

operation recorder function

"Memory Card" (page 5-3)
"CPU Memory (User Memory)" (page 5-27)
"Operation Recorder Function" (page 5-86)

Built-in communication

Communication with PCs

- "USB Communication" (page 5-109)
- "USB Communication" (page 5-109)
- "Ethernet/EtherNet/IP Communication" (page 5-110)

Ethernet, EtherNet/IP device communications

- "Ethernet/EtherNet/IP Communication" (page 5-110)

5-2 Memory Card

What You Can Do with the Memory Card

You can exchange ladder programs, comments, and various unit setup information using the memory card. Also, even without "KV STUDIO", you can transfer ladder programs and save/read device values from the access windows or touch panel of CPU unit only by using the memory card.

- **Transferring data from memory card to the CPU**

You can select any (project) folder saved in the memory card from the access window for reading the data. Also, you can read the data stored in the (project) folder named "KVSAVE **" (** is the number: 01 ~ 99) saved on the memory card when CR5500 is turned from OFF to ON. Specify the drive number to CM 1880 and the folder number to CM 1881.

Data includes the following: program data such as ladder programs, device, comments, unit setup information and logging setup information, and device file data such as DM, EM, W, FM, ZF,CM, R, MR, LR, and CR.

"Writing/Reading Project Data" (page 5-13)

- **Auto-read function**

When powered on, read data to CPU unit automatically, these data are saved in "AutoLoad" or "AtLoad**" (** : 01 to 99) (item) folder. If they are saved in "RunLoad" or "RnLoad**" (** : 01 to 99) folder in advance, during PRG → RUN switching, it will read automatically.

"Auto Load Function" (page 5-24)

- **Saving data from the CPU unit to the memory card**

Program files (ladder programs, device, comments, unit setup information, logging setup information) and all device files (DM, EM, W, FM, ZF, CM, R, MR, LR, B, CR) can be saved from the CPU to the memory card. Saved files are stored to a newly created sub-directory (project folder). The name of folder is generated automatically, and displayed in the Access Window after normal completion of operation.

"Writing/Reading Project Data" (page 5-13)

- **Recipe load / save function**

The device / variable values can be saved in multiple data tables, and the device / variable values in the CPU unit are switched by switching the data tables. This can be used for process switching, etc.

"Recipe Load/Save Function" (page 5-50)

Point **When using the recipe load / save function, the CPU function version should be upgraded to 2.0 or later.**

- **Logging/tracing function**

If the logged device and loaded timing (triggering) are registered through KV STUDIO, the data will be saved automatically into the memory card. Since the data is saved in the format of CSV, it can be directly loaded into Microsoft Excel, etc.

"5-5 Logging/Tracing" (page 5-57)

- **operation recorder function**

After setting the type of data to be saved and the save time (trigger), the device value of the CPU or the video data of the camera input unit is automatically saved to the memory card when the trigger occurs. Saved data can be played back using the replay function of KV STUDIO.

"5-5 Logging/Tracing" (page 5-57)

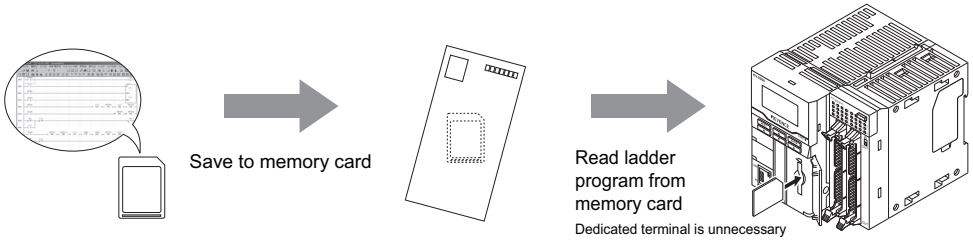
- **Ladder program-based memory card control**

The access to the memory card from the ladder program is possible when storage instruction is used. The ladder program makes it possible to save device data values on the CPU unit to memory card and load data onto another CPU unit.

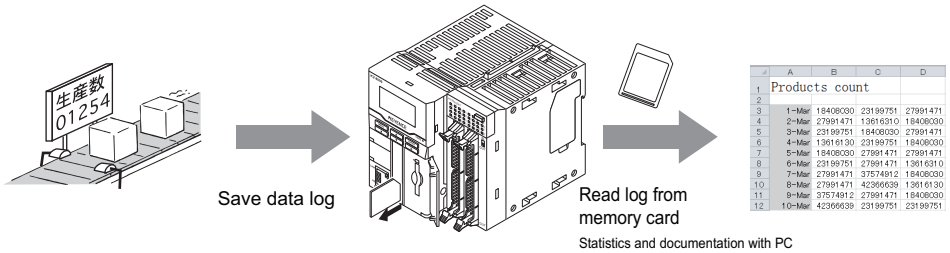
"Storage Instructions" (page 5-26)

● Example for use of a memory card

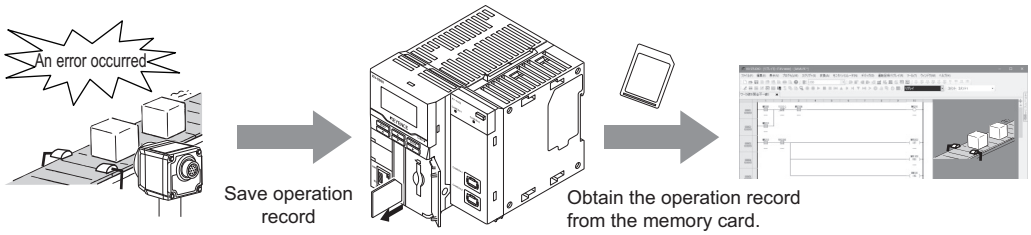
- Maintenance can be performed easily by a memory card. This allows a quick response to equipment maintenance and transmission of ladder data due to changes to specifications.



- Logging of device values is easy. Data is tabulated from logs to create data sheets with greater ease.



- It is very easy to identify the error cause. The ladder program when an error occurs can be checked for each scan, or checked against the frame-by-frame video shot by the camera, which can be used to identify the cause for the equipment error.



■ Differences in Memory Card and CPU Memory

Function	Memory card	CPU memory
Load/save programs	Yes	Yes
Recipe load / save	Yes	Yes
File register	Yes	Yes
Logging trace function	Yes	Yes
operation recorder function	Yes	Yes
File control provided by ladder program	Yes	Yes
Store user documents	Yes	Yes
Creating files in RAM	-	Yes

"5-3 CPU Memory (User Memory)"

About the Memory Card

■ Memory card that can be used

A memory card and adapter or card reader for reading the memory card are required. Support SDHC memory card and SD memory card.
Operation through memory cards not specified by KEYENCE is not assured.

SD memory card

Type	KV-M16G	KV-M4G	KV-M1G
Memory	Flash ROM		
Type	SDHC		SD
Memory capacity	16GB	4GB	1GB
Read/Write cycles (each block)	60,000 operations		100,000 operations
Operating temperature	-25 to +85°C		
Storage temperature	-40 to +85°C		

■ Reading and writing memory card data

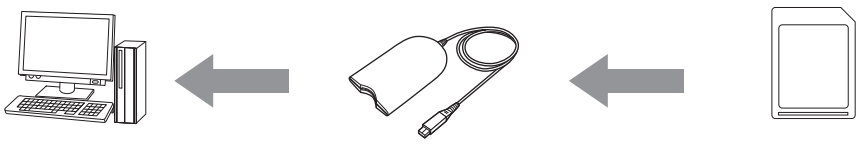
Use one of the following procedures for reading and writing memory card data.

(1) Connect directly to a PC to read and write data.
(2) Use the storage transfer tool to read and write via PLC.
(3) Use the FTP server function on a other device to read and write via PLC.

(1) Connect directly to a PC to read and write data.
The connection methods may differ depending upon PCs.

● When the PC has a USB port

Connect the USB card reader/writer to the USB port on the PC. Next, insert the memory card into the card reader/writer.



PC with USB　　Card reader/writer　　Memory card KV-M16G/M4G/M1G

Card reader/writer

This unit is for reading and writing data on memory cards on a PC that has a USB port. It cannot be used on PCs that do not have a USB port.
For details on the USB card reader/writer for the memory card, refer to the User's Manual provided with the card reader/writer.

● For devices provided with an SD memory card slot

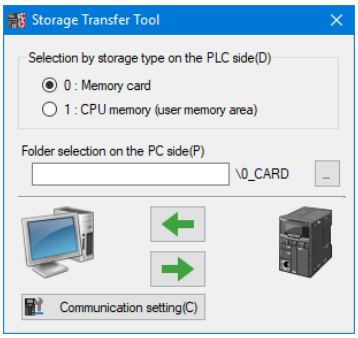
Insert a memory card into the card slot.

(2) Use the storage transfer tool to read and write via PLC.

Start "Tools (T)" → "Storage transfer tools" in the KV STUDIO menu.

Alternative procedure Menu → KEYENCE KV STUDIO ** → Memory Transfer Tool
(** is the version of KV STUDIO used)

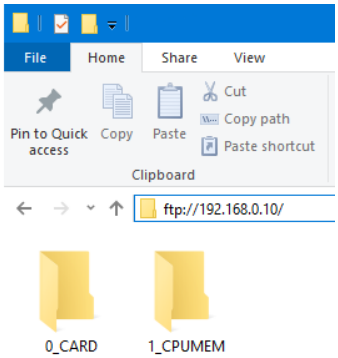
Folders on the memory card and the PC are backed up or restored.
This is only a batch transfer of all data and does not allow you to specify files or folders for reading or writing.



For details, see "KV STUDIO User's Manual."

(3) Use the FTP server function or other device to read and write via PLC.

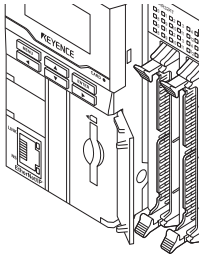
After using the CPU unit or Ethernet unit that supports the FTP server function, the data of the CPU memory card in the PLC can be read and written via an Ethernet cable.
Data on a memory card is stored in the "0 CARD" folder.



■ Used on the CPU unit

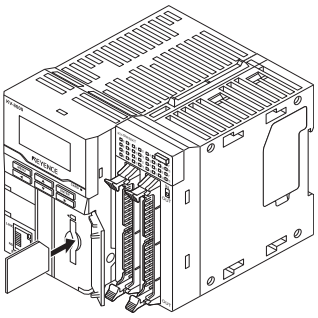
● Installing the memory card

1 Open slot cover of the memory card.



2 Insert the memory card into the memory card slot.

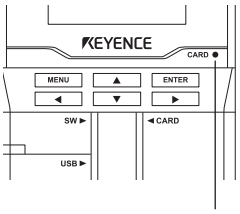
The labeling side at right hand is inserted properly with a "click" sound.



3 After inserting the memory card, close the memory card slot cover until it clicks into place.

● Removing the memory card

1 Please confirm that the LED lamp is off during memory card access and open the slot cover of the memory card.



Memory card access LED

2 Press the memory card in the insertion direction (inwards)

You hear a click sound when the card is inserted properly.
Draw out the memory card straight.

3 After removing the memory card, close the memory card slot cover until it clicks into place.

■ Precautions When Using a memory card

The following describes general precautions when using a memory card.

NOTICE	• **Data saved on a memory card may be lost in the following instances. Note that Keyence can assume no liability whatsoever if logged data is lost. You must be very careful.**
	  • **When the user or a third party has misused the memory card**
	  • **When the memory card has been subjected to static electricity and electrical noise**
	  • **When the memory card has been removed or other equipments have been turned OFF while the memory card is being accessed (e.g. during saving or deleting of data)**
	• **We recommend backing up important data on CD/DVD or other storage media.**
	• **Never remove the memory card or turn power OFF while the memory card access LED on the PLC is lit (indicating that memory card is being read/ written). Doing so might corrupt the data on the memory card.**

Point

- **In environments subject to severe noise, it may take longer time to access the memory card.**
  (In the worst case, this might damage the card or FAT (File Allocation Table). In this case, attach a ferrite core onto the power cable before use.
- **Also, install the unit away from equipment that emits strong electromagnetic fields.**
- **Accessing of the memory card can not be started with the memory card slot cover open. Be sure to use memory card with the memory card slot cover closed.**
- **When the memory card slot cover is opened while the memory card is being accessed, the process that was being performed is continued until it is completed. However, subsequent accessing of memory card will not be performed.**
- **The SD memory card is formatted for maximum performance before it is shipped from the factory. SD memory cards formatted in Windows can be used. However, the performance of the SD memory card may not be fully demonstrated. If the memory card needs to be restored to its initial shipped state, contact your nearest office.**

Devices using memory cards

Device No.	Attribute R: Load only Empty field: Read/write enabled	Name	Description
CR5400		Memory card mount request	Use for reidentification after unmounting a memory card. Since memory cards are normally identified automatically when mounted, there is no need to run the memory card mount request. * Use the SET instruction to run requests.
CR5401		Memory card unmount request	Unmounts inserted memory cards. When unmounting is completed, the memory card recognition end relay turns off. * Use the SET instruction to run requests.
CR5402	R	Memory card in use	Turns ON during memory card access by CPU unit or expansion unit functions.
CR5403	R	Memory card recognition end	Goes on when memory card is mounted. Goes off when unmounted.
CR5404	R	Memory card present	Goes on after verifying memory card insertion.
CR5405	R	Memory card is write protected	Goes on when memory card is write protected.
CR5500		Project load execution request	Project load execution request relay. Executing the project load by turning OFF to ON(Can not be executed during RUN).
CR5502		Project save execution request	Project save execution request relay. Executing the project save by turning OFF to ON.
CR5504	R	Project load execution complete	Project load execution end relay. Turns on loading of the project load ends. Turns OFF when the project load execution request relay turns OFF to ON.
CR5505	R	Project load execution failure	Project load execution failure relay. Turns ON when the project load fails. Turns OFF when the project load execution request relay turns OFF to ON.
CR5506	R	Project save execution complete	Project save execution end relay. Turns ON when the project save ends. Turns OFF when project save execution request relay turns OFF to ON.
CR5507	R	Project save execution failure	Project save execution failure relay. Turns ON when project save fails. Turns OFF when project save execution request relay turns OFF to ON.

Device No.	Attribute R: Load only Empty field: Read/write enabled	Name	Description
CR5508		Recipe load request	Recipe load request relay. When turned from OFF to ON, the recipe stored in the folder number specified by CM1871 of the drive specified by CM1870 is loaded to overwrite the device / variable values.
CR5510	R	Recipe load complete	Recipe load completion relay. ON when the recipe load is completed. OFF when the recipe load request relay turns from OFF to ON.
CR5511	R	Recipe load failed	Recipe load failure relay. ON when the recipe load is failed. OFF when the recipe load request relay turns from OFF to ON.
CR5512		Recipe save request	Recipe save request relay. When turned from OFF to ON, the recipe contents of the folder number specified by CM1876 of the drive specified by CM1875 are referenced, and the recipe of the folder number specified by CM1877 is overwritten using the device / variable values of the CPU unit.
CR5514	R	Recipe save complete	Recipe save completion relay. ON when the recipe save is completed. OFF when the recipe save request relay turns from OFF to ON.
CR5515	R	Recipe save failed	Recipe save failure relay. ON when the recipe save is failed. OFF when the recipe save request relay turns from OFF to ON.
CM1870		Drive number when loading recipe	Drive number when loading recipe. When CR5508 turns from OFF to ON, the recipe of the drive number stored is loaded.
CM1871		Folder number when loading recipe	Folder number when loading recipe. When CR5508 turns from OFF to ON, the recipe of the folder number stored is loaded.
CM1873	R	Completion code when loading recipe	When the recipe load completion relay turns ON, the completion code is stored.
CM1875		Drive number when saving recipe	Drive number when saving recipe. When CR5512 turns from OFF to ON, the recipe is saved to the drive number stored.
CM1876		Reference folder number when saving recipe	Reference folder number when saving recipe. When CR5512 turns from OFF to ON, the recipe contents of the folder number stored are referenced.
CM1877		Output folder number when saving recipe	Output folder number when saving recipe. When CR5512 turns from OFF to ON, the recipe is saved to the folder number stored.
CM1878	R	Completion code when saving recipe	When the recipe save completion relay turns ON, the completion code is stored.

Device No.	Attribute R: Load only Empty field: Read/write enabled	Name	Description
CM1880		Project load drive number (request)	Project load drive number (request). Load the project with the drive number stored when CR5500 turns OFF to ON.
CM1881		Project load folder number (request)	Project load folder number (request). Load the project with the folder number stored when CR5500 turns OFF to ON.
CM1882	R	Project load drive number (complete)	The loaded drive number is stored. 0 is stored when the project load fails.
CM1883	R	Project load folder number (complete)	The loaded folder number is stored. 0 is stored when the project load fails.
CM1884	R	Project load completion code	The completion code is stored when the project load execution end relay turns ON.
CM1888		Project save drive number (request)	Project save drive number (request). Saves the project folder to the drive specified when the project save execution request relay turns OFF to ON. 0: Memory card 1: CPU memory
CM1889		Project save folder number (request)	Project save folder number (request). Saves the project folder with the folder number specified when the project save execution request relay turns OFF to ON. Enter with 0, 1 to 99. If 0 is specified, searches for an empty number and sets it as the folder number.
CM1890	R	Project save drive number (complete)	The saved drive number is stored. 0 is stored if the project save fails.
CM1891	R	Project save folder number (complete)	The saved folder number is stored. 0 is stored if the project save fails.
CM1892	R	Project save completion code	The completion code is stored when the project save execution end relay turns ON.

Point

- **When a BOOL array type variable is assigned to CR, the element number of the array should be specified with a value. When specified with device or variable, the SET instruction cannot operate properly.**

Writing/Reading Project Data

■ Structure of folder

The data managed by (project) folder on the memory card is described below.

Software	File type		File name	Way of storing
KV-8000(A)	CPU System setting Ladder program Global variable Local variable Local Device Comment		□.YCD*1	CR5502 turns OFF to ON Access window KV STUDIO (File>Memory card)
	Variable information*7		□.YVR*1	
	Global device comment		□.YGC*1	
	Setting information of initial value of device.		□.YDR*1	
	Communication setting information		□.YAS*1	
	Unit setting information		□.YLB*1	
	Logging/tracing setting information		□.YLG*1	
	CPU memory capacity setting		□.YMC*1	
	operation recorder setting		□.YTR*1	
	Access window setup information		□.YAW*1	
	Device File	DM	DM.ZDV	CR5502 turns OFF to ON Access window KV STUDIO (Monitor/simulator >Batch change window)
		EM	EM.ZDV	
		FM	FM.ZDV*2 *5	
		CM	CM.ZDV	
		R	R.ZDV	
		MR	MR.ZDV	
		LR	LR.ZDV	
		CR	CR.ZDV	
		B	B.ZDV	
		W	W.ZDV	
		ZF	ZF.YDV*5	
	Email setting information.		MAIL*.YSH(*=0 to 48)*4	CR5502 turns OFF to ON Access window KV STUDIO (Email setup tool)
	FTP client-side setting information		FTPC*.YSH(*=0 to 48)	CR5502 turns OFF to ON Access window KV STUDIO (FTP client-side setup tool)
	OPC UA server setting information		UASVR*.YSH(*=0)	CR5502 turns OFF to ON Access window KV STUDIO
	Version information when memory card data are generated		Savelog.txt	CR5502 turns OFF to ON Access window KV STUDIO
KV-XD02	Data application unit parameter file		*_XD02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
			_XD02.YTV(=01 to 16)	
KV-CA02	Camera input unit parameter file		*_CA02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO

Software	File type	File name	Way of storing
KV-XLE02	Ethernet Unit parameter file	*_XLE02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XCM02	Communication unit parameter file	*_XCM02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XL402	Serial communication unit parameter file	*_XL402.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XL202		*_XL202.YTC(*=01 to 16)	
KV-L2*V	PROTOCOL STUDIO 2 File (PS1 compatible (standard/Large) mode)	PB*.ZSH(*=1A to 48B)*3	KV-L2*V settings (PROTOCOL STUDIO 2)
	PROTOCOL STUDIO 2 File (Protocol Studio2 mode)	PB*.YSH(*=1A to 48B)	CR5502 turns OFF to ON Access window KV-L2*V settings (PROTOCOL STUDIO 2)
KV-XH16ML	Positioning unit, motion control unit Parameter file	*_XH16.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XH04ML		*_XH04.YTC(*=01 to 16)	
KV-XH64EC		*_XH64EC.YTC(*=01 to 16)	
KV-XH32EC		*_XH32EC.YTC(*=01 to 16)	
KV-XH16EC		*_XH16EC.YTC(*=01 to 16)	
KV-ML16V/MC40V/ MC20V	Positioning unit, motion control unit Parameter file	KMP*.YSH(*=1 to 48)	KV-ML/MC settings (KV MOTION+)
KV-LH20V	Communication positioning unit parameter setting (Point parameter, control/system parameter)	SPSP**.YSH (*=1 to 48)	Communication positioning setting tool *6
KV-SH04PL	Positioning unit Parameter file	*_SH.YTC (*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-H20G/H20S/ H40S	Positioning unit Parameter file	MB*.ZSH(*=1 to 48)	KV-H20S/40S/20G settings (MOTION BUILDER)
	Positioning unit Comment file	MC*.ZSH(*=1 to 48)	
MV-L40	MV-L40 Parameter file	MLS*.ZSH(*=1 to 18)	MV-L40 settings (MV LINK STUDIO)
	MV-L40 Comment file	MLC*.ZSH(*=1 to 18)	
KV-DN20	KV-DN20 Parameter file	DN*.ZSH(*=1 to 48)	KV STUDIO (KV-DN20 Setup Tool)
KV-CL20	KV-CL20 Parameter file	CL*.ZSH(*=1 to 48)	KV STUDIO (KV-CL20 Setup Tool)

*1 "□" file name is the same with the name of project folder.
*2 FM saves the FM of current bank, and read to FM of current bank.
*3 It can not save to memory card and CPU memory from the access window. It will operate successfully even when reading out project data without these files since the configuration itself is saved in DM and CM.
*4 When performing email setup through CPU built-in Ethernet function, the setting will be saved by "MAIL0.YSH".
*5 If reading FM and ZF, it's required to firstly read the data of ZF and then data of FM.
*6 Not created when saved from the access window. However, the set point parameter etc. is stored in DM.ZDV.
*7 Data that exists only in KV-8000 with CPU function version 2.0 or later.

■ Access window

● How to save

If project data are saved from the Access Window of CPU unit, project folders "KVSAVE **" (**means serial No.: 01 to 99) will be generated automatically. The serial No. is the smallest two-digit number not yet used for folder name in the memory card.

"Memory Card" (page 5-3)

● How to read

Select and read the project folders from Access Window of CPU unit.

"Memory Card" (page 5-3)

Reference

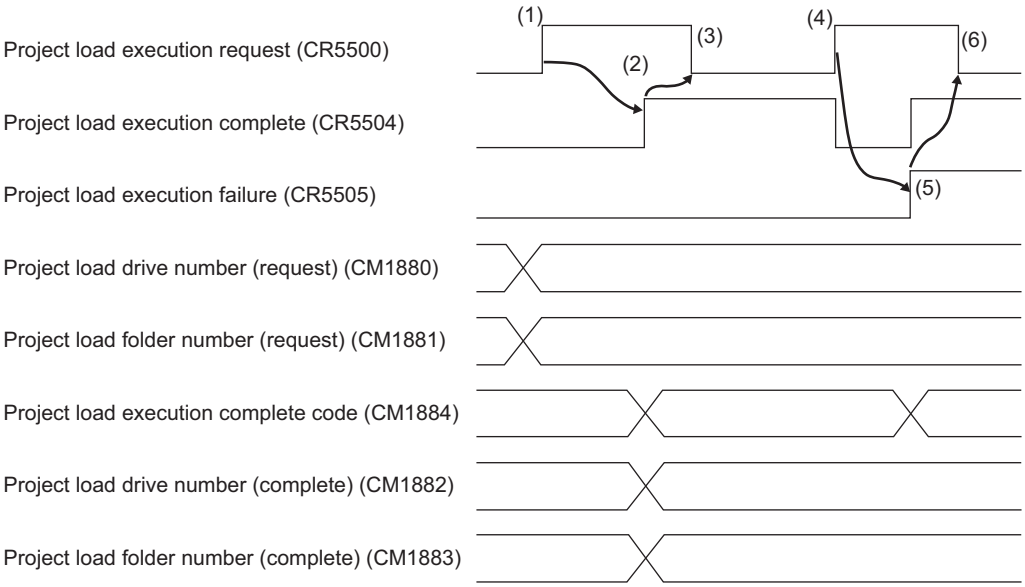
- Methods of only reading some data
  All data in (project) folder are generally read from memory card.
  For example, if you intend to save only the data of DM, or only save the program data, the specific data are only read from memory card to CPU unit after they are saved into folders in the memory card.
  Unnecessary files are deleted via PC from folders in the memory card.
- At power-on, project data saved in the "AutoLoad" (Folder selection: "Atload**"(**:01~99)) (project) folder are read automatically.
- At PRG -> RUN, project data saved in the "RunLoad" (Folder selection: "Rnload**"(**:01~99)) (project) folder are read automatically.

■ To load/ save with the control relay

● Load

The program load can not be executed in RUN mode. Switch to PRG mode to operate the control relay.

Time chart while loading



When the project load succeeds

(1) Set the drive number and folder number containing the memory card data to be loaded into the project load drive number (request) and the project load folder number (request) and turn ON the project load execution request relay. (for monitors, touch panels, or communications)

(2) When the project load ends, the project load execution end relay turns ON. The drive number and folder number from which the loaded memory card data was read will be stored to the project load drive number (complete) and the project load folder number (complete). (for CPU units)

(3) Turn OFF the project load execution request relay when the poject load execution end relay turns ON. (for monitors, touch panels, communications, etc.) When the next project execution request relay turns OFF to ON, the project load execution end relay turns OFF. (for CPU units)

When the project load fails

(4) Set the drive number and folder number containing the memory card data to be loaded into the project load drive number (request) and the project load folder number (request) and turn ON the project load execution request relay. (for monitors, touch panels, or communications)

(5) When the project load fails, the project load execution end relay and the project load execution failure relay turn ON. The project load drive number (complete) and the project load folder number (complete) will not be updated. (for CPU units) About the result at the time of completion, please check the project load completion code (CM1884). For the contents of the completion code, refer to "The error code during the execution of load, save" (page 5-19).

(6) Turn OFF the project load execution request relay When the project load execution failure relay turns ON. (for monitors, touch panels, communications, etc.) When the next project execution request relay turns from OFF to ON, the project load execution end relay and the project load execution failure relay turn OFF. (for CPU units)

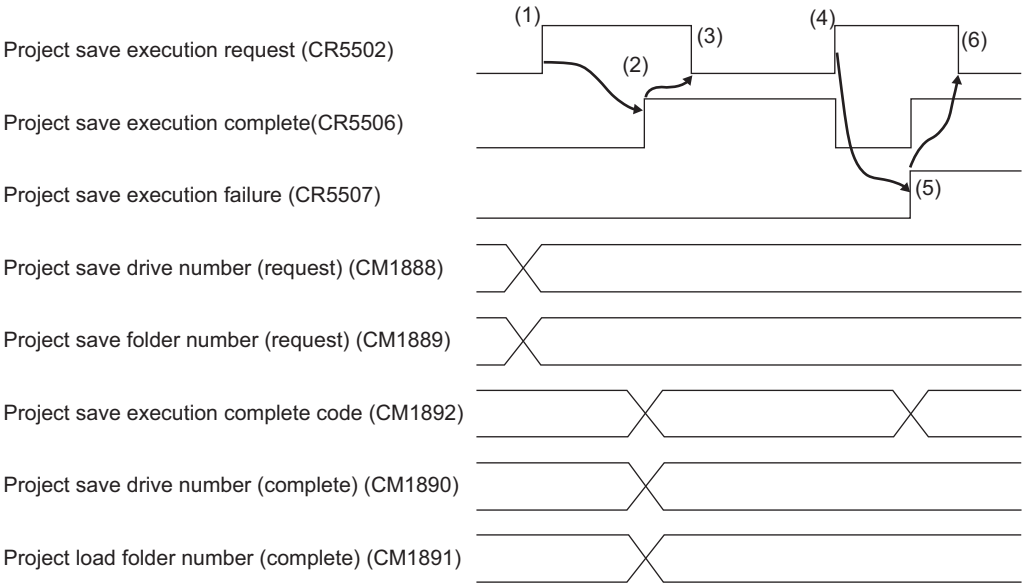
Sample program

Since the project load can not be executed in RUN mode, there is no sample program.

● Save

To perform the program save by controlling the control relay with the ladder program, execute with the SET or KEEP command.

Time chart while saving



* To specify the folder number when saving, use the project save folder number (request) (CM 1888).

When the project save succeeds

(1) Store the drive number of the save destination to the project save drive number (request) and turn ON the project save execution request relay. Store the folder number in the project save folder number (request) for storing in the specified folder. (for ladders, monitors, touch panels, or communications etc.)

(2) When the project save ends, the project save execution end relay turns ON. The project save drive number (complete) stores the save destination drive number, and the project save folder number (complete) stores the save destination folder number. (for CPU units)

(3) Turn OFF the project save execution request relay when the project save execution end relay turns ON. (for ladders, monitors, touch panels, or communication, etc.) When the next project execution request relay turns OFF to ON, the project save execution end relay turns OFF. (for CPU units)

When the project save fails

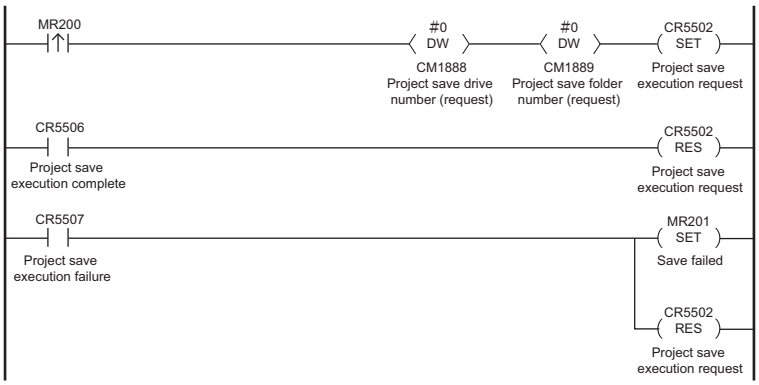
(1) Store the drive number of the save destination in the project save drive number (request) and turn on the project save execution request relay. (for ladders, monitors, touch panels, or communications, etc.)

(2) If the project save fails, the project save execution failure relay turns ON. (for CPU units)

(3) Turn OFF the project save execution request relay when the project save execution end relay turns ON. (for ladders, monitors, touch panels, or communication, etc.) When the next project execution request relay turns OFF to ON, the project save execution end relay and the project save execution failurerelay turn OFF. (for CPU units)

Sample program

To perform the program save, execute with the SET command.



<Mnemonic list>

```
LDP MR200
DW #0 CM1888
CON
DW #0 CM1889
CON
SET CR5502
LD CR5506
RES CR5502
LD CR5507
SET MR201
RES CR5502
```

The error code during the execution of load, save

The automatic loading complete code is stored in CM1791, the run load complete code is stored in CM1795, the project load completion code is stored in CM1884, and the project save completion code is stored in CM1892.

Number	Message	Cause	Countermeasure	At occurrence
2	空き番号がありません Unused number not available 没有空编号	The write in was not possible since there was a file already.	Please configure an empty number to the folder number to save.	Save
3	オートローディング失敗 Auto Loading Failed 自动加载失败	The folder number designated at CM was out of range during the execution of auto load or run load.	Please designate number from 0 to 99.	Load/Save
4	ロードできません Cannot load data 无法加载	The folder of CM designated to memory card/CPU memory does not exist.	Please confirm that project data is inside.	Load
5	メモリカードがありません ---------- CPUメモリアンマウント Memory card not found ---------- CPU Memory unmounted 没有存储卡 ---------- CPU存储器卸载	The load or save was executed during the condition without being able to recognize memory card/ CPU memory.	Please insert the memory card when loading from the memory card, or saving to memory card. Please mount the CPU memory.	Load/Save
6	アクセス競合 Access conflict 访问冲突	The save/load was not possible since transfer or read out of the program was done from another route.	Please execute save/load after the transfer or read out of the program is complete.	Load/Save
7	オートローディング失敗 Auto Loading Failed 自动加载失败	The autoload or run load was executed towards PLC with write protect configured.	Please execute after the write protect of PLC is released.	Load
9	CPUユニット機種不一致 PLC model mismatch CPU单元机型不一致	It tried to read in to the memory card file of the separate device model.	Please confirm the CPU unit model of project.	Load
10	非対応のCPU機能バージョン Unsupported Function Version 不支持的功能版本	It tried to read in function version that does not correspond or the project of separate model.	Please confirm the CPU unit model of the project.	Load
11	PRGモード時のみ可能です PRG mode only 仅PRG模式时可执行	It tried to load with RUN mode.	Please switch to PRG mode to load.	Load
12	空き容量がありません No free space 没有可用空间	There are no free space in the memory card/CPU memory.	Please increase the free space.	Load/Save
15	プロジェクト領域サイズオーバー Project area size over 项目区域容量超出	The write in of CPU memory capacity configuration has failed (project domain is insufficient).	Please review the configuration of CPU memory capacity configuration or reduce data of project domain inside the CPU.	Load
16	ユーザメモリ領域サイズオーバー UserMem area size over 用户存储器区域容量超出	The write in of CPU memory capacity configuration has failed (user memory domain is insufficient).	Please review the configuration of CPU memory capacity configuration or reduce data of user memory domain inside the CPU.	Load

Number	Message	Cause	Countermeasure	At occurrence
17	スイッチがPRG時のみ可能です Only when switch in PRG 仅开关处于PRG时可执行	Tried to upgrade the version of system program when the switch was in RUN mode.	Turn the switch to PRG mode then execute version upgrade of the system program.	Load
18	ユニット種別不一致 Unit Type　Mismatch 单元类型不一致	YTC file has loaded which is different kind of unit compared to the connected unit.	Please check unit type of YTC file.	Load
19	オールクリアが必要 All Clear Reqred Before loading 加载前　需全部清空	An attempt was made to load a project that requires all clear in advance. If the version of the project in the PLC and the project to load do not match, all clear must be executed before loading the project.	Execute all clear and then load the project. If you are loading from the CPU memory, back up the data beforehand as the data to load will also be cleared.	Load
20	ユニット非対応 機能バージョン Unsupported Unit Function Version 不支持的单元功能版本	An attempt was made to read save data with a function version not compatible with the expansion unit.	Check the function version of the expansion unit in the project. Check the function version of the expansion unit.	Load
100	メモリカードカバーオープン Memory card cover open 存储卡外盖打开	The memory card cover is opened.	Please close the memory card cover.	Load/Save
101	メモリカードがLOCKされています Memory card locked 存储卡被锁定	The memory card is in LOCK status.	Please release LOCK or use memory card that is writable.	Save
102	ロードできません ---------- セーブできません Cannot load data ---------- Cannot save data 无法加载 ---------- 无法保存	The memory card is damaged.	Please check the file system of the memory card.	Load ---------- Save
103	ファイル数上限を超過しました Max no. of files exceeded 超过了文件数上限	The sum of number of files and folders inside the user memory has gone over the upper limit.	Please reduce the file or folder inside the user memory.	Load/Save
104	フォルダ階層が深すぎます Folder layer too deep 文件夹层数过多	The folder hierarchy inside the user memory is too deep.	Please confirm the folder hierarchy inside the user memory.	Load/Save
1000	ロードできません Cannot load data 无法加载	• The file is damaged. • The memory card is damaged. • The card has been extracted during the loading process. • It tried to load from CPU memory with the configuration "Load after clearing the user memory". • It tried to execute load from CPU memory with methods other than the autoload, run load with the configuration "Load after all clear".	Please confirm if the project and memory card is appropriate.	Load

Number	Message	Cause	Countermeasure	At occurrence
1001	セーブできません Cannot save data 无法保存	• The memory card has been damaged. • The card has been extracted during save process.	Please confirm if the project and memory card is appropriate.	Save

Reference

The load/save completion code can be confirmed on the access window in addition to the CM.

■ Use KV STUDIO

You can read/write project data of the memory card in the menu of KV STUDIO.

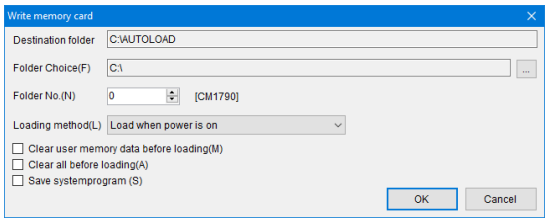
● Ladder program etc. (except device file)

1 Connect memory card to PC.

"Reading and writing memory card data" (page 5-6)

2 Select "File(F)"→"Memory card(M)"→"Write(W)" or "Read(R)" from the menu of KV STUDIO.

When save is chosen

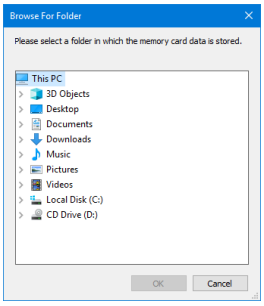


Item	Description
Output target folder	Displays the name of the output target folder for memory card data output. The folder name changes according to the folder selection, folder number, and load method.
Select folder	Select the folder to output memory card data to. Click the far right of the box ... to display the "Browse for folder" dialog box.
Folder No.	Specify a folder number. When 0 is input : No number is added to the end. When 1 to 99 is input : The 2 specified digits are added to the end of the folder name. The device number on the right indicates the device that specifies the folder number for each load method. Load when power is turned on : CM1790 Load when mode is changed from PROG to RUN : CM1794
Load method	Select the load method. The output target folder name changes according to the selected load method. Load when power is turned on : "AUTOLOAD", "AtLoad**" (**: 01 to 99) Load when mode is changed from PROG to RUN : "RUNLOAD", "RnLoad**" (**: 01 to 99)
Load after clearing user memory	It loads after clearing user memory when it is checked.
Load after all clear	It loads after all clear when it is checked.
Save system program	Saves the unit system program with the project. The system program is updated when it is loaded. "System Program Update" (page 3-21)

Point **If a folder which has the same name is chosen, the data is overwritten and previous data will be lost.**

When Read is chosen

The "Browse for folder" dialog box is displayed.

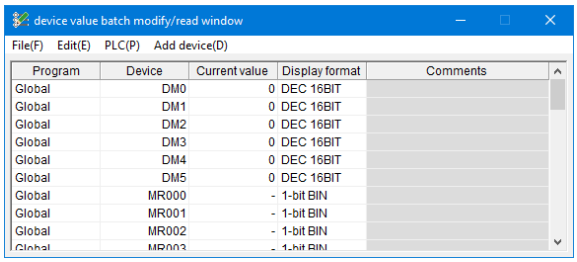


● Device file

1 Connect memory card to PC.

"Reading and writing memory card data" (page 5-6)

2 Select "Monitor/simulator (N)"→"Device value batch modify/read window (D)" from the menu of KV STUDIO.



Program	Device	Current value	Display format	Comments
Global	DM0	0	DEC 16BIT	
Global	DM1	0	DEC 16BIT	
Global	DM2	0	DEC 16BIT	
Global	DM3	0	DEC 16BIT	
Global	DM4	0	DEC 16BIT	
Global	DM5	0	DEC 16BIT	
Global	MR000	-	1-bit BIN	
Global	MR001	-	1-bit BIN	
Global	MR002	-	1-bit BIN	

3 Select "File (F)" → "Read memory card (R)" or "Save to memory card (Z)" from the menu of the batch change/read device values window.

Read: specify the read drive and folder name.

Save: save the registered device and current value.

Reference

If the device's default setting of KV STUDIO is employed, the device value can also be managed through the memory card.

Auto Load Function

Automatically load project data saved to the memory card when powered on or at PRG → RUN.
Control memory (CM) may be used to select the folder to be read from several folders.

■ Reading at power on.

1 KV STUDIO may be used to write item data under "AutoLoad" or "AtLoad" (** : 01-99) folder into the memory card.**

"Ladder program etc. (except device file)" (page 5-22)

2 Insert the memory card into CPU unit memory card slot, close the memory card slot cover, until “click”is heard.

3 Connect power supply of CPU unit.

Data are read automatically.

● Read folder selection method

For the selection of folder to be read, save value in CM1790 in advance.

- When CM1790 value is 0
  Read "AutoLoad" folder.
  If "AutoLoad" folder does not exist in the memory card, automatic loading is impossible. Please run the program before power on. Error will not occur.
- When CM1790 value is 01 - 99
  Read "AtLoad**" (** : CM1790 value (01-99)) folder.

Folder No. Selection steps

1 In case of power on, save the value from external equipments such as touch panel into CM1790.
2 Power off.
3 Power on.
4 Read the folder selected by CM1790.
5 After reading, 0 is saved in CM1790, the read folder No. Is saved in CM1792.

Reference

After automatic loading is completed, 0 is saved in CM1790, so when powered on at next time, the same folder No. will not be read. However, please note that, in case 0 is saved in CM1790, if the memory card contains “AutoLoad” folder, it will be read when powered on at next time.
The auto load execution status can be confirmed with the automatic loading complete code (CM1791). For the contents of the completion code, refer to "The error code during the execution of load, save" (page 5-19).

Point

- **When write protection is provided for CPU system setting of KV STUDIO, error will occur (auto loading failure), it will enter PRG mode. After the error is eliminated, in RUN mode, run the program before automatic loading.**
  In the following cases, error will occur (auto loading failure), it will enter PRG mode.
  - **When CM1790 value exceeds 100**
  - **The memory card does not contain any folder selected by CM1790**
- **If the function version of the project stored in the PLC unit and of the project to transfer differ, CPU all clear is required before transferring.**

■ Read for PRG → RUN

1 Use KV STUDIO to write in item data in memory card under "RunLoad" or "Rnload"(01~99) folder.**

"Ladder program etc. (except device file)" (page 5-22)

2 Insert the memory card in CPU unit memory card slot, close the memory card slot cover, until "click".

3 CPU unit executes PRG → RUN switching.

Read data automatically and begin to run.

● Read folder selection method

Select the folder to be read, save value in CM1794 in advance.

- When CM1794 value is 0
  Read "RunLoad" folder.
  If the memory card does not contain "RunLoad" folder, automatic loading could not be performed. Run the program before PRG → RUN switching. Error will not occur.

- CM1794 value is 0 1 - 99
  Read "RnLoad**"(CM1794 value (01-99)) folder.

Folder No. selection steps

1 Save the value in CM1794 from external equipment such as touch panel.
2 PRG → RUN switching.
3 Read the folder selected by CM1794.
4 After reading, 0 is saved in CM1794, the read folder No. Is saved in CM1796.

Reference

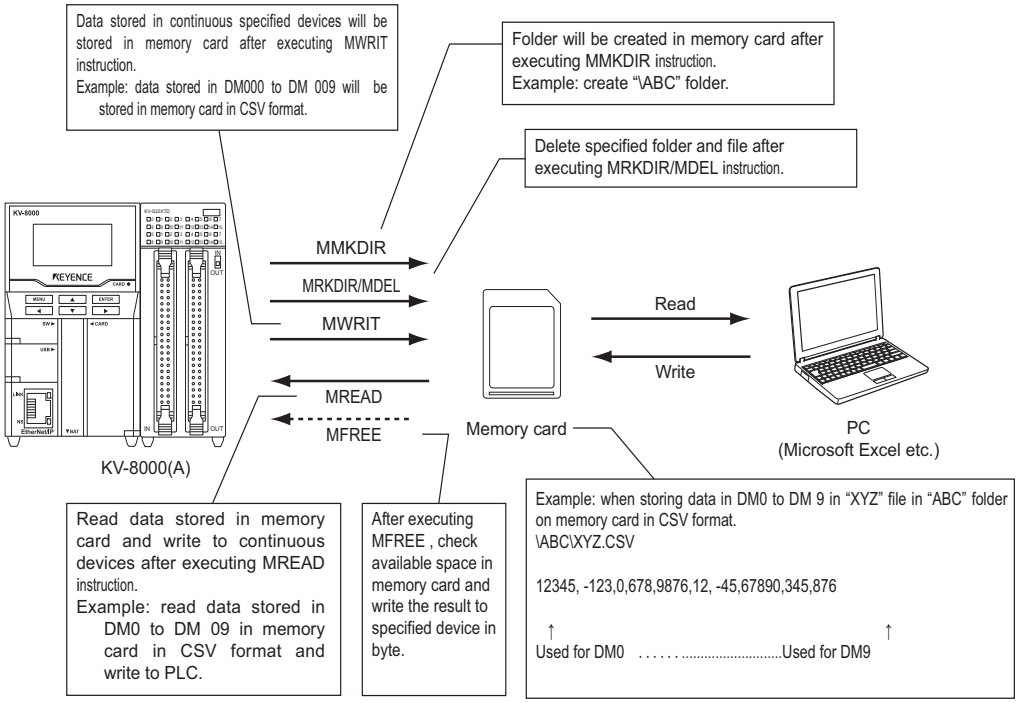
After automatic loading is completed, 0 is saved in CM1794, so when powered on at next time, the same folder No. will not be read. However, please note that, in case 0 is saved in CM1794, if the memory card contains "RunLoad" folder, it will be read upon next PRG → RUN.
The run load execution status can be confirmed with the run load complete code (CM1795). For the contents of the completion code, refer to "The error code during the execution of load, save" (page 5-19).

Point

- **When write protection is provided for CPU system setting of KV STUDIO, error will occur (auto loading failure), RUN mode cannot be set. After the error is eliminated, if memory card is taken out, in RUN mode, it will run the program before automatic loading.**
  Error will occur in the following conditions (auto loading failure), it cannot be set to RUN mode. After the error is eliminated, if take off memory card, run in RUN mode, it will run the program before automatic loading.
  - **When the value of CM1794 exceeds 100**
  - **When there is no folder selected by CM1794 in the memory card**
- **If the function version of the project stored in the PLC unit and of the project to transfer differ, CPU all clear is required before transferring.**

Storage Instructions

The memory card can be accessed from the ladder program when the storage instruction is used.



Reference

- The storage instruction can be used to access CPU memory as well as memory cards.
- The error code of Storage instruction is stored in CM2390. For details of CM2390, refer to 📖 "KV-8000/7000/5000/3000/1000/KV Nano Series Instruction Term Reference Manual".

5-3 CPU Memory (User Memory)

About CPU Memory (user memory)

Built-in CPU memory (user memory) areas can be used for operations such as loading/saving the ladder program, logging device values and creating files from the ladder program.

Point The equivalent can be realized even when the memory card is used.

- **Reading data from CPU memory to the CPU unit**

Data can be read by selecting the sub-directory (project folder) on the CPU memory in the access window.
Data includes the following: program data such as ladder programs, device, comments, unit setup information and logging setup information, and device file data such as DM, EM, W, FM, ZF,CM, R, MR, LR, and CR.

"Writing/Reading Project Data" (page 5-35)

- **Auto-read function**

When powered on, read data to CPU unit automatically, these data are saved in "AutoLoad" or "AtLoad**" (** : 01 to 99) (item) folder. If they are saved in "RunLoad" or "RnLoad**" (** : 01 to 99) folder in advance, during PRG→RUN switching, it will read automatically.

"Auto Load Function" (page 5-45)

- **Saving data from the CPU unit to the CPU memory**

Program files (ladder programs, devices, comments, unit setup information, logging setup information) and all device files (DM, EM, W, FM, ZF, CM, R, MR, LR, B, CR) can be saved from the CPU unit to the CPU memory. The files are saved into the newly-created (project) folder. The name of folder is generated automatically, and displayed in the Access Window after normal completion of operation.

"Writing/Reading Project Data" (page 5-35)

- **Recipe load / save function**

The device / variable values can be saved in multiple data tables, and the device / variable values in the CPU unit are switched by switching the data tables. This can be used for process switching, etc.

"Recipe Load/Save Function" (page 5-50)

Point **When using the recipe load / save function, the CPU function version should be upgraded to 2.0 or later.**

- **Log logging/tracking function**

If the logged device and loaded timing (triggering) are registered through KV STUDIO, the data will be saved automatically into the memory card. Since the data is saved in the format of CSV, it can be directly loaded into Microsoft Excel, etc.

"5-5 Logging/Tracing" (page 5-57)

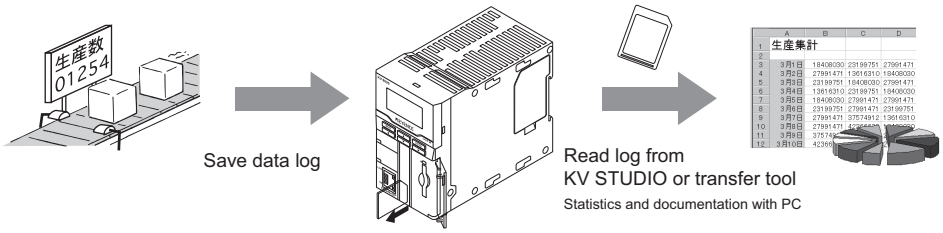
- **operation recorder function**

After setting the type of data to be saved and the save time (trigger), the device value of the CPU or the video data of the camera input unit is automatically saved to the memory card when the trigger occurs. Saved data can be played back using the replay function of KV STUDIO.

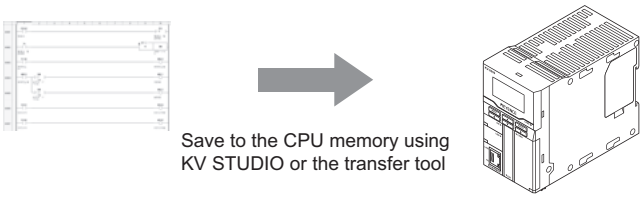
"5-5 Logging/Tracing" (page 5-57)

● Example of CPU memory usage

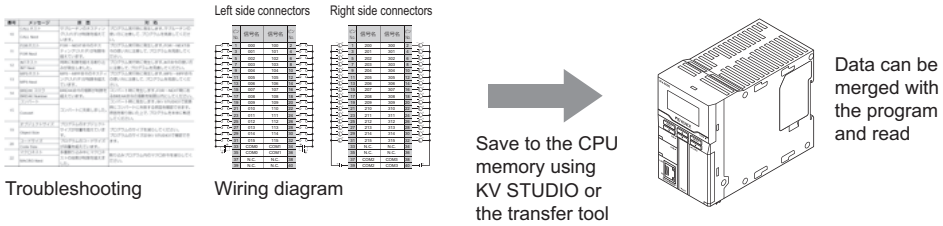
- Device values can be logged without a memory card. This makes it possible add up values from the log, which simplifies the preparation of data sheets.



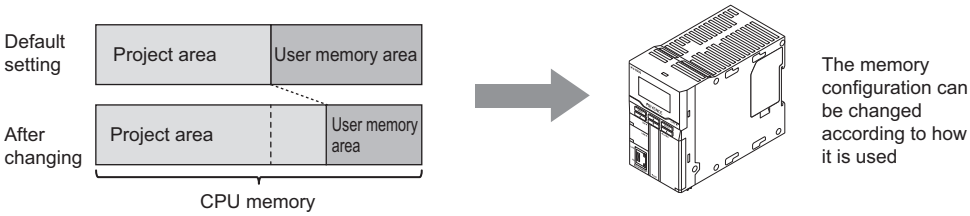
- Saving program data during normal operation will make it easier to restore the system to its original state if problems occur after making changes.



- Design data prepared in Excel, Word or PDF can be saved with the program allowing you to compare them with the design during maintenance work.



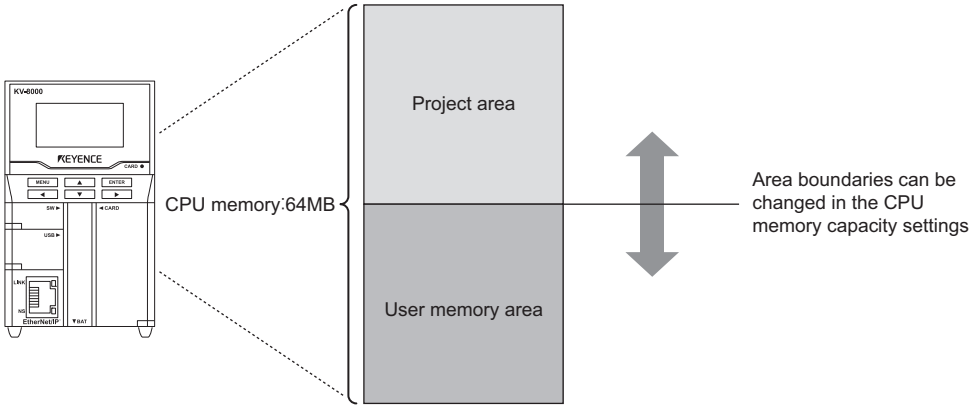
- This memory can be used as an area for storing device comments and variable.



CPU memory

■ CPU memory area

The sum of the program area and user memory area is the CPU memory area.



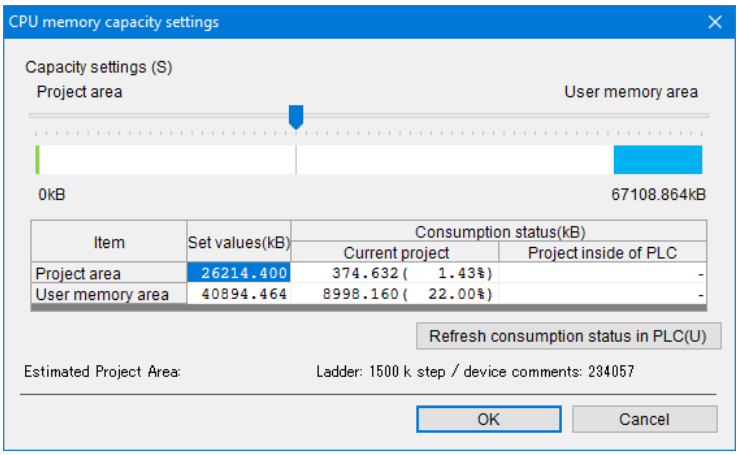
The following data is stored in these two areas.

Project area	User memory area
Unit setting information Device default values CPU system settings operation recorder setting Recording and tracking setting Program Device comment Variable Ladder monitor data	File register User documents Logging data Project data Other data

Unit settings, unit data, access window settings and other settings are stored in a separate area.

■ Setting method

Use KV STUDIO [Tools] → [CPU memory capacity setting] to change the size of these areas.



■ Procedures for reading and writing CPU memory (user memory) data

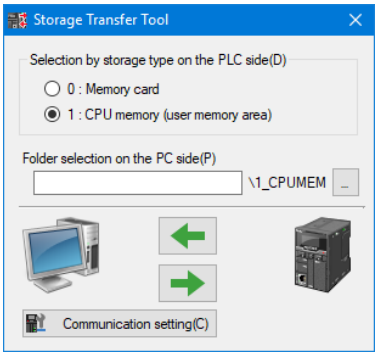
Use one of the following procedures for reading and writing memory card data.

(1) Use the storage transfer tool to read and write via PLC.
(2) Use the FTP server function or other device to read and write via PLC.
(3) Use the PLC transfer function from KV studio to read and write data*
(* Only user documents and file register settings)

(1) Use the storage transfer tool to read and write via PLC.
Start "Tools (T)" → "Storage transfer tools" in the KV STUDIO menu.

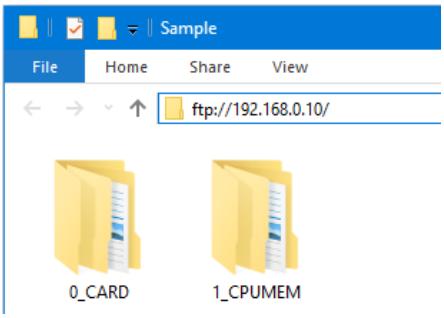
Alternative procedure Menu→All Programs→KEYENCE Applications→KV STUDIO Ver.10G→Tool→Storage Transfer Tool

Folders in CPU memory (user memory) and the PC are backed up or stored.
This is only a batch transfer of all data and does not allow you to specify files or folders for reading or writing.

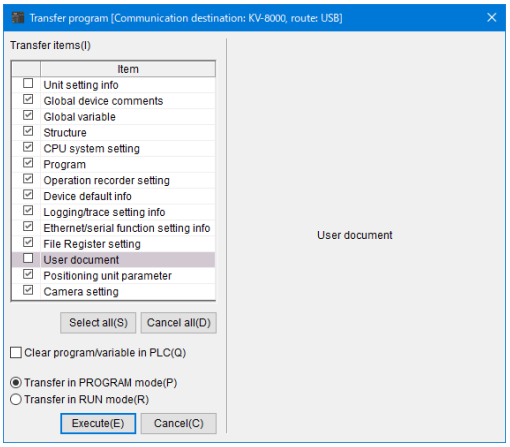


For details, see the (reference) "KV STUDIO User's Manual."

(2) Use the FTP server function or other device to read and write via PLC.
After using the CPU unit or Ethernet unit that supports the FTP server function, the data of the CPU memory card in the PLC can be read and written via an Ethernet cable.
The data inside the CPU memory is stored in the “1_CPUMEM” folder.



(3) Use the PLC transfer function from KV STUDIO to read and write data
In a PLC transfer, only CPU memory data such as user documents and file register settings can be read or written.



Point

The data such as logging data, load/save data can not be transferred during PLC transfer.

■ Operating Precautions

The following general precautions should be heeded in the use of CPU memory.

NOTICE	• In the following circumstances, data stored in CPU memory may be lost. Please note that Keyence accepts no liability whatsoever for loss of registered data. • Incorrect use of CPU memory by the user or a third person. • Exposure of CPU memory to static electricity and interference. • When the frequency of saving to CPU memory exceeds 10,000 times. (PLC main unit error 143: CPU Memory Write Cycle Over occurs)

Point

- You can access CPU memory during memory card access. (You cannot write to memory card/CPU memory when they are accessed.)
- If user memory is set to write data at all times, write operations will slow down. For higher write speeds, generate files in RAM and perform write operations before the system is shut down. "RAM mode" (page 5-47)
- These attributes will be released when file/folder with “Hidden file attribute” and “Read only attribute” is written to user memory.

Devices used by the CPU memory

Device No	Attribute R: Load only Empty field: Read/write enabled	Name	Description
CR3200		CPU memory save request	When this request goes on, the difference data is stored CPU memory. * Please use the SET instruction when executing demand.
CR3201	R	Saving to CPU memory	Goes ON when saving to CPU memory.
CR3202	R	Saving to CPU memory ended	Goes ON when saving to CPU memory ends. Turns off when a CPU memory save request is turned on.
CR3203	R	Save error in CPU memory	Goes ON when error occurs while saving to CPU memory. Turns off when a CPU memory save request is turned off.
CR3207		CPU memory instruction RAM mode	The file register instruction and storage instruction will operate in RAM disk when it is ON.
CR5408		CPU memory mount request	Use for reidentification after unmounting CPU memory. Since it will be mounted automatically during normal power insertion, there is no need to execute CPU memory mount demand. * Use the SET instruction to run requests.
CR5409		CPU memory unmount request	Unmounts CPU memory. When unmounted, the CPU memory identification complete relay turns off. * Use the SET instruction to run requests.
CR5410	R	CPU memory in use	Goes ON during CPU memory access.
CR5411	R	CPU memory recognition end	Goes on when CPU memory is mounted. Goes off when unmounted.
CR5500		Project load execution request	Project load execution request relay. Executing the project load by turning OFF to ON(Can not be executed during RUN).
CR5502		Project save execution request	Project save execution request relay. Executing the project save by turning OFF to ON.
CR5504	R	Project load execution complete	Project load execution end relay. Turns on loading of the project load ends. Turns OFF when the project load execution request relay turns OFF to ON.
CR5505	R	Project load execution failure	Project load execution failure relay. Turns ON when the project load fails. Turns OFF when the project load execution request relay turns OFF to ON.
CR5506	R	Project save execution complete	Project save execution end relay. Turns ON when the project save ends. Turns OFF when project save execution request relay turns OFF to ON.
CR5507	R	Project save execution failure	Project save execution failure relay. Turns ON when project save fails. Turns OFF when project save execution request relay turns OFF to ON.

Device No	Attribute R: Load only Empty field: Read/write enabled	Name	Description
CR5508		Recipe load request	Recipe load request relay. When turned from OFF to ON, the recipe stored in the folder number specified by CM1871 of the drive specified by CM1870 is loaded to overwrite the device / variable values.
CR5510	R	Recipe load complete	Recipe load completion relay. ON when the recipe load is completed. OFF when the recipe load request relay turns from OFF to ON.
CR5511	R	Recipe load failed	Recipe load failure relay. ON when the recipe load is failed. OFF when the recipe load request relay turns from OFF to ON.
CR5512		Recipe save request	Recipe save request relay. When turned from OFF to ON, the recipe contents of the folder number specified by CM1876 of the drive specified by CM1875 are referenced, and the recipe of the folder number specified by CM1877 is overwritten using the device / variable values of the CPU unit.
CR5514	R	Recipe save complete	Recipe save completion relay. ON when the recipe save is completed. OFF when the recipe save request relay turns from OFF to ON.
CR5515	R	Recipe save failed	Recipe save failure relay. ON when the recipe save is failed. OFF when the recipe save request relay turns from OFF to ON.
CM1870		Drive number when loading recipe	Drive number when loading recipe. When CR5508 turns from OFF to ON, the recipe of the drive number stored is loaded.
CM1871		Folder number when loading recipe	Folder number when loading recipe. When CR5508 turns from OFF to ON, the recipe of the folder number stored is loaded.
CM1873	R	Completion code when loading recipe	When the recipe load completion relay turns ON, the completion code is stored.
CM1875		Drive number when saving recipe	Drive number when saving recipe. When CR5512 turns from OFF to ON, the recipe is saved to the drive number stored.
CM1876		Reference folder number when saving recipe	Reference folder number when saving recipe. When CR5512 turns from OFF to ON, the recipe contents of the folder number stored are referenced.
CM1877		Output folder number when saving recipe	Output folder number when saving recipe. When CR5512 turns from OFF to ON, the recipe is saved to the folder number stored.
CM1878	R	Completion code when saving recipe	When the recipe save completion relay turns ON, the completion code is stored.
CM1882	R	Project load drive number (complete)	The loaded drive number is stored. 0 is stored when the project load fails.

Device No	Attribute R: Load only Empty field: Read/write enabled	Name	Description
CM1883	R	Project load folder number (complete)	The loaded folder number is stored. 0 is stored when the project load fails.
CM1884	R	Project load completion code	The completion code is stored when the project load execution end relay turns ON.
CM1888		Project save drive number (request)	Project save drive number (request). Saves the project folder to the drive specified when the project save execution request relay turns OFF to ON. 0: Memory card 1: CPU memory
CM1889		Project save folder number (request)	Project save folder number (request). Saves the project folder with the folder number specified when the project save execution request relay turns OFF to ON. Enter with 0, 1 to 99. If 0 is specified, searches for an empty number and sets it as the folder number.
CM1890	R	Project save drive number (complete)	The saved drive number is stored. 0 is stored if the project save fails.
CM1891	R	Project save folder number (complete)	The saved folder number is stored. 0 is stored if the project save fails.
CM1892	R	Project save completion code	The completion code is stored when the project save execution end relay turns ON.

Reference

In addition to CM1884, CM1892, you can check the load/save completion in the access window.

Point

- **When a BOOL array type variable is assigned to CR, the element number of the array should be specified with a value. When specified with device or variable, the SET instruction cannot operate properly.**

Writing/Reading Project Data

■ Structure of folder

The data that is managed with the (project) folder inside the CPU memory is as follows.

Software	File type		File name	Way of storing
KV-8000(A)	CPU System setting Ladder program Global variable Local variable Local Device Comment		□.YCD[*1]	CR5502 turns OFF to ON Access window KV STUDIO (File>Memory card)
	Variable information[*7]		□.YVR[*1]	
	Global device comment		□.YGC[*1]	
	Setting information of initial value of device.		□.YDR[*1]	
	Communication setting information		□.YAS[*1]	
	Unit setting information		□.YLB[*1]	
	Logging/tracing setting information		□.YLG[*1]	
	CPU memory capacity setting		□.YMC[*1]	
	operation recorder setting		□.YTR[*1]	
	Access window setup information		□.YAW[*1]	
	Device File	DM	DM.ZDV	CR5502 turns OFF to ON Access window KV STUDIO (Monitor/simulator >Batch change window)
		EM	EM.ZDV	
		FM	FM.ZDV[*2 *5]	
		CM	CM.ZDV	
		R	R.ZDV	
		MR	MR.ZDV	
		LR	LR.ZDV	
		CR	CR.ZDV	
		B	B.ZDV	
		W	W.ZDV	
		ZF	ZF.YDV[*5]	
	Email setting information.		MAIL*.YSH(*=0 to 48)[*4]	CR5502 turns OFF to ON Access window KV STUDIO (Email setup tool)
	FTP client-side setting information		FTPC*.YSH(*=0 to 48)	CR5502 turns OFF to ON Access window KV STUDIO (FTP client-side setup tool)
	OPC UA server setting information		UASVR*.YSH(*=0)	CR5502 turns OFF to ON Access window KV STUDIO
	Version information when memory card data are generated		Savelog.txt	CR5502 turns OFF to ON Access window KV STUDIO
KV-XD02	Data application unit parameter file		*_XD02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
			_XD02.YTV(=01 to 16)	
KV-CA02	Camera input unit parameter file		*_CA02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO

Software	File type	File name	Way of storing
KV-XLE02	Ethernet Unit parameter file	*_XLE02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XCM02	Communication unit parameter file	*_XCM02.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XL402	Serial communication unit parameter file	*_XL402.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XL202		*_XL202.YTC(*=01 to 16)	
KV-L2*V	PROTOCOL STUDIO 2 File (PS1 compatible (standard/Large) mode)	PB*.ZSH(*=1A to 48B)*3	KV-L2*V settings (PROTOCOL STUDIO 2)
	PROTOCOL STUDIO 2 File (Protocol Studio2 mode)	PB*.YSH(*=1A to 48B)	CR5502 turns OFF to ON Access window KV-L2*V settings (PROTOCOL STUDIO 2)
KV-XH16ML	Positioning unit, motion control unit Parameter file	*_XH16.YTC(*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-XH04ML		*_XH04.YTC(*=01 to 16)	
KV-XH64EC		*_XH64EC.YTC(*=01 to 16)	
KV-XH32EC		*_XH32EC.YTC(*=01 to 16)	
KV-XH16EC		*_XH16EC.YTC(*=01 to 16)	
KV-ML16V/MC40V/ MC20V	Positioning unit, motion control unit Parameter file	KMP*.YSH (*=1 to 48)	KV-ML/MC settings (KV MOTION+)
KV-LH20V	Communication positioning unit parameter setting (Point parameter, control/system parameter)	SPSP**.YSH (*=1 to 48)	Communication positioning setting tool *6
KV-SH04PL	Positioning unit Parameter file	*_SH.YTC (*=01 to 16)	CR5502 turns OFF to ON Access window KV STUDIO
KV-H20G/H20S/ H40S	Positioning unit Parameter file	MB*.ZSH(*=1 to 48)	KV-H20S/40S/20G settings (MOTION BUILDER)
	Positioning unit Comment file	MC*.ZSH(*=1 to 48)	
MV-L40	MV-L40 Parameter file	MLS*.ZSH(*=1 to 18)	MV-L40 settings (MV LINK STUDIO)
	MV-L40 Comment file	MLC*.ZSH(*=1 to 18)	
KV-DN20	KV-DN20 Parameter file	DN*.ZSH(*=1 to 48)	KV STUDIO (KV-DN20 Setup Tool)
KV-CL20	KV-CL20 Parameter file	CL*.ZSH(*=1 to 48)	KV STUDIO (KV-CL20 Setup Tool)

*1 "□" file name is the same with the name of project folder.
*2 FM saves the FM of current bank, and read to FM of current bank.
*3 It can not save from the access window to memory card and CPU memory.
*4 When performing email setup through CPU built-in Ethernet function, the setting will be saved by "MAIL0.YSH".
*5 If reading FM and ZF, it's required to firstly read the data of ZF and then data of FM.
*6 Not created when saved from the access window. However, the set point parameter etc. is stored in DM.ZDV.
*7 Data that exists only in KV-8000 with CPU function version 2.0 or later.

■ Access window

● How to save

If project data are saved from the Access Window of CPU unit, project folders "KVSAVE **" (**means serial No.: 01 to 99) will be generated automatically. The serial number is the smallest two-digit number not yet used for folder names in CPU memory.

"Storage" (page 5-139)

● How to read

Select and read the project folders from Access Window of CPU unit.

"Storage" (page 5-139)

Reference

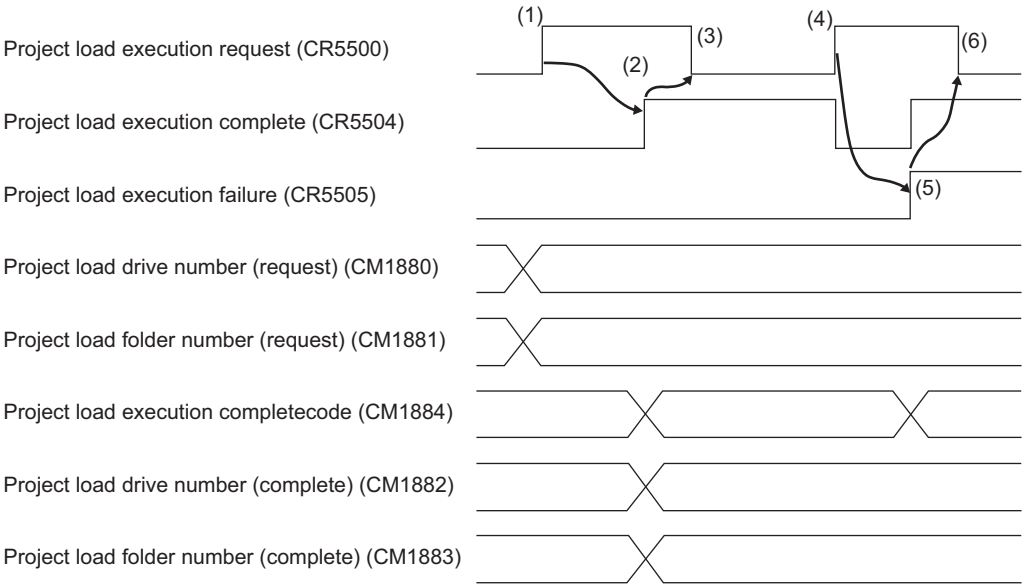
- Methods of only reading some data
  Normally, all data in the (project) folder is read from CPU memory at one time.
  You can read only the files you want to read by deleting all other files.
  To use the storage transfer tool, delete all files on the PC you no longer need before transferring data.
- At power-on, project data saved in the "AutoLoad" (Folder selection: "AtLoad**"(**:01~99)) (project) folder are read automatically.
- At PRG→RUN, project data saved in the "RunLoad" (Folder selection: "Rnload**"(**:01~99)) (project) folder are read automatically.

■ To load/ save with the control relay

● Load

The program load can not be executed in RUN mode. Switch to PRG mode to operate the control relay.

Time chart while loading



When the project load succeeds

(1) Set the drive number and folder number containing the project data to be loaded into the project load drive number (request) and the project load folder number (request) and turn ON the project load execution request relay. (for monitors, touch panels, or communications)

(2) When the project save ends, the project save execution end relay turns ON. The project load drive number (complete) stores the load destination drive number, and the project load folder number (complete) stores the load destination folder number. (for CPU units)

(3) Turn OFF the project save execution request relay when the project save execution end relay turns ON. (for monitors, touch panels, communications, etc.) When the next project execution request relay turns OFF to ON, the project save execution end relay turns OFF. (for CPU units)

When the project load fails

(4) Set the drive number and folder number containing the memory card data to be loaded into the project load drive number (request) and the project load folder number (request) and turn ON the project load execution request relay. (for monitors, touch panels, or communications)

(5) When the project load fails, the project load execution end relay and the project load execution failure relay turn ON. The project load drive number (complete) and the project load folder number (complete) will not be updated. (for CPU units)

(6) Turn OFF the project load execution request relay When the project load execution failure relay turns ON. (for monitors, touch panels, communications, etc.) When the next project execution request relay turns from OFF to ON, the project load execution end relay and the project load execution failure relay turn OFF. (for CPU units)

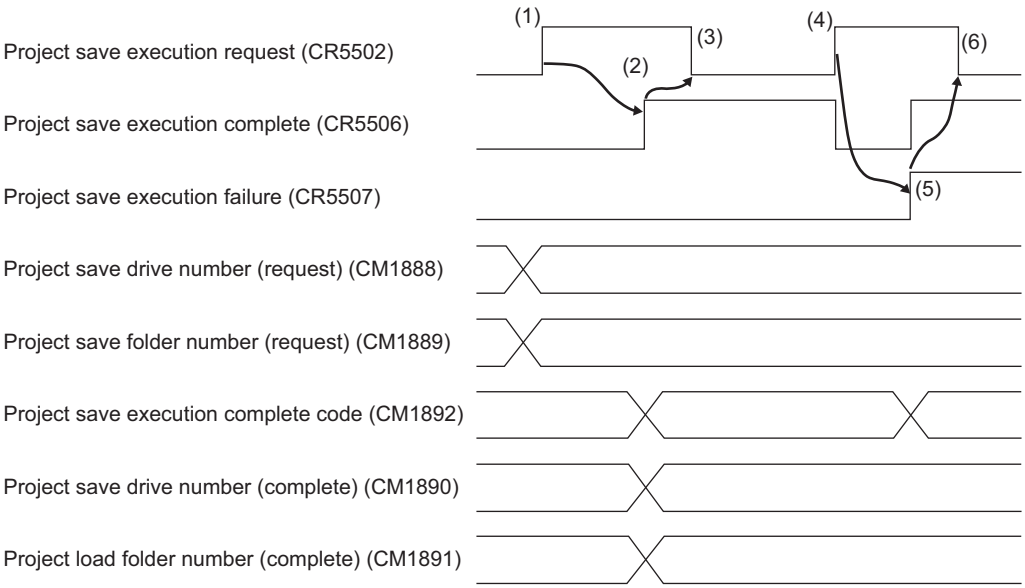
Sample program

Since the project load can not be executed in RUN mode, there is no sample program.

● Save

To perform the program save by controlling the control relay with the ladder program, execute with the SET or KEEP command.

Time chart while saving



* To specify the folder number when saving, use the project save folder number (request) (CM 1888).

When the project save succeeds

(1) Store the drive number of the save destination to the project save drive number (request) and turn ON the project save execution request relay. Store the folder number in the project save folder number (request) for storing in the specified folder. (for ladders, monitors, touch panels, or communications etc.)

(2) When the project save ends, the project save execution end relay turns ON. The project save drive number (complete) stores the save destination drive number, and the project save folder number (complete) stores the save destination folder number. (for CPU units)

(3) Turn OFF the project save execution request relay when the project save execution end relay turns ON. (for ladders, monitors, touch panels, or communication, etc.) When the next project execution request relay turns OFF to ON, the project save execution end relay turns OFF. (for CPU units)

When the project save fails

(4) Store the drive number of the save destination to the project save drive number (request) and turn ON the project save execution request relay. Store the folder number in the project save folder number (request) for storing in the specified folder. (for ladders, monitors, touch panels, or communications etc.)

(5) If the project save fails, the project save execution failure relay turns ON. (for CPU units)

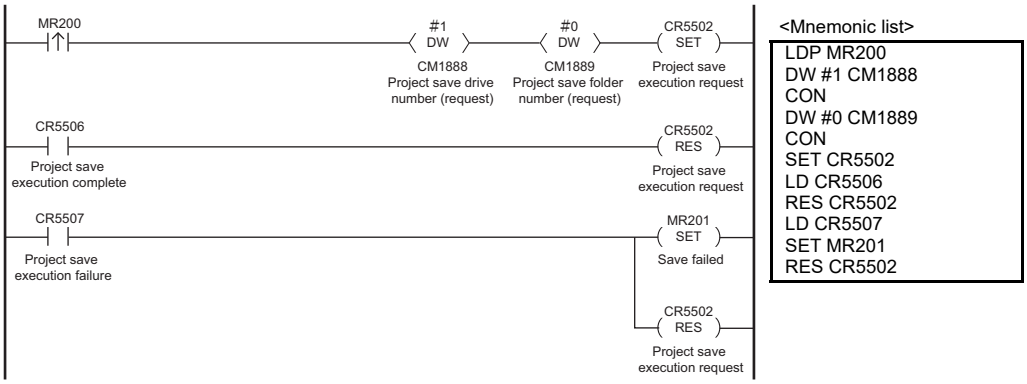
(6) Turn OFF the project save execution request relay when the project save execution end relay turns ON. (for ladders, monitors, touch panels, or communication, etc.) When the next project execution request relay turns OFF to ON, the project save execution end relay and the project save execution failurerelay turn OFF. (for CPU units)

Sample program

This is a program to save the project in the CPU memory (drive number: 1) when MR200 rising edge occurs.

When 0 is stored in the project save folder number (request), it searches for an empty number and sets it as the folder number.

To perform the program save, execute with the SET command.



The error code during the execution of load, save

The automatic loading complete code is stored in CM1791, the run load complete code is stored in CM1795, and the project save completion code is stored in CM1892.

Number	Message	Cause	Countermeasure	At occurrence
2	空き番号がありません Unused number not available 没有空编号	The write in was not possible since there was a file already.	Please configure an empty number to the folder number to save.	Save
3	オートローディング失敗 Auto Loading Failed 自动加载失败	The folder number designated at CM was out of range during the execution of auto load or run load.	Please designate number from 0 to 99.	Load/Save
4	ロードできません Cannot load data 无法加载	The folder of CM designated to memory card/CPU memory does not exist.	Please confirm that project data is inside.	Load
5	メモリカードがありません ---------- CPUメモリアンマウント Memory card not found ---------- CPU Memory unmounted 没有存储卡 ---------- CPU存储器卸载	The load or save was executed during the condition without being able to recognize memory card/ CPU memory.	Please insert the memory card when loading from the memory card, or saving to memory card. Please mount the CPU memory.	Load/Save
6	アクセス競合 Access conflict 访问冲突	The save/load was not possible since transfer or read out of the program was done from another route.	Please execute save/load after the transfer or read out of the program is complete.	Load/Save
7	オートローディング失敗 Auto Loading Failed 自动加载失败	The autoload or run load was executed towards PLC with write protect configured.	Please execute after the write protect of PLC is released.	Load
9	CPUユニット機種不一致 PLC model mismatch CPU单元机型不一致	It tried to read in to the memory card file of the separate device model.	Please confirm the CPU unit model of project.	Load
10	非対応のCPU機能バージョン Unsupported Function Version 不支持的功能版本	It tried to read in function version that does not correspond or the project of separate model.	Please confirm the CPU unit model of the project.	Load
11	PRGモード時のみ可能です PRG mode only 仅PRG模式时可执行	It tried to load with RUN mode.	Please switch to PRG mode to load.	Load
12	空き容量がありません No free space 没有可用空间	There are no free space in the memory card/CPU memory.	Please increase the free space.	Load/Save
15	プロジェクト領域サイズオーバー Project area size over 项目区域容量超出	The write in of CPU memory capacity configuration has failed (project domain is insufficient).	Please review the configuration of CPU memory capacity configuration or reduce data of project domain inside the CPU.	Load
16	ユーザメモリ領域サイズオーバー UserMem area size over 用户存储器区域容量超出	The write in of CPU memory capacity configuration has failed (user memory domain is insufficient).	Please review the configuration of CPU memory capacity configuration or reduce data of user memory domain inside the CPU.	Load

Number	Message	Cause	Countermeasure	At occurrence
17	スイッチがPRG時のみ可能です Only when switch in PRG 仅开关处于PRG时可执行	Tried to upgrade the version of system program when the switch was in RUN mode.	Turn the switch to PRG mode then execute version upgrade of the system program.	Load
18	ユニット種別不一致 Unit Type　Mismatch 单元类型不一致	YTC file has loaded which is different kind of unit compared to the connected unit.	Please check unit type of YTC file.	Load
19	オールクリアが必要 All Clear Reqred Before loading 加载前　需全部清空	An attempt was made to load a project that requires all clear in advance. If the version of the project in the PLC and the project to load do not match, all clear must be executed before loading the project.	Execute all clear and then load the project. If you are loading from the CPU memory, back up the data beforehand as the data to load will also be cleared.	Load
20	ユニット非対応 機能バージョン Unsupported Unit Function Version 不支持的单元功能版本	An attempt was made to read save data with a function version not compatible with the expansion unit.	Check the function version of the expansion unit in the project. Check the function version of the expansion unit.	Load
21	*1 *1 *1	When executing CR, the drive number specified by CM is out of the specified range.	Specify the drive number as 0 or 1.	Load/Save
100	メモリカードカバーオープン Memory card cover open 存储卡外盖打开	The memory card cover is opened.	Please close the memory card cover.	Load/Save
101	メモリカードがLOCKされています Memory card locked 存储卡被锁定	The memory card is in LOCK status.	Please release LOCK or use memory card that is writable.	Save
102	ロードできません ---------- セーブできません Cannot load data ---------- Cannot save data 无法加载 ---------- 无法保存	The memory card is damaged.	Please check the file system of the memory card.	Load ---------- Save
103	ファイル数上限を超過しました Max no. of files exceeded 超过了文件数上限	The sum of number of files and folders inside the user memory has gone over the upper limit.	Please reduce the file or folder inside the user memory.	Load/Save
104	フォルダ階層が深すぎます Folder layer too deep 文件夹层数过多	The folder hierarchy inside the user memory is too deep.	Please confirm the folder hierarchy inside the user memory.	Load/Save

Number	Message	Cause	Countermeasure	At occurrence
1000	ロードできません	• The file is damaged. • The memory card is damaged. • The card has been extracted during the loading process. • It tried to load from CPU memory with the configuration "Load after clearing the user memory". • It tried to execute load from CPU memory with methods other than the autoload, run load with the configuration "Load after all clear".	Please confirm if the project and memory card is appropriate.	Load
	Cannot load data			
	无法加载			
1001	セーブできません	• The memory card has been damaged. • The card has been extracted during save process.	Please confirm if the project and memory card is appropriate.	Save
	Cannot save data			
	无法保存			

*1 No error message is displayed in the access window as this error occurs only during execution by operating CR.

■ Use KV STUDIO

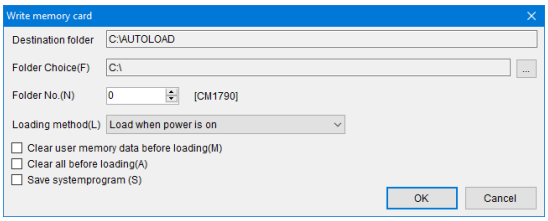
From the KV STUDIO menu, you can read and save project data for loading and saving.

● Ladder program etc. (except device file)

Select "File(F)"->"Memory card(M)"->"Write(W)" or "Read(R)" from the menu of KV STUDIO.

Point **The procedure of creating project data for load save is the same as procedure of saving to memory card. Please save created folder to the CPU memory.**

When save is chosen

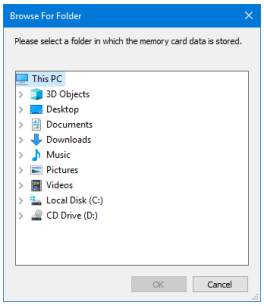


Item	Description
Output target folder	Displays the name of the output target folder for memory card data output. The folder name changes according to the folder selection, folder number, and load method.
Select folder	Select the folder to output memory card data to. Click the far right of the box ... to display the "Browse for folder" dialog box.
Folder No.	Specify a folder number. When 0 is input : No number is added to the end. When 1 to 99 is input : The 2 specified digits are added to the end of the folder name. The device number on the right indicates the device that specifies the folder number for each load method. Load when power is turned on : CM1790 Load when mode is changed from PROG to RUN : CM1794
Load method	Select the load method. The output target folder name changes according to the selected load method. Load when power is turned on : "AUTOLOAD", "AtLoad**" (**: 01 to 99) Load when mode is changed from PROG to RUN : "RUNLOAD", "RnLoad**" (**: 01 to 99)
Load after clearing user memory	It loads after clearing user memory when it is checked.
Load after all clear	It loads after all clear when it is checked.
Save system program	Saves the unit system program with the project. The system program is updated when it is loaded. "System Program Update" (page 3-21)

Point **If a folder which has the same name is chosen, the data is overwritten and previous data will be lost.**

When Read is chosen

The "Browse for folder" dialog box is displayed.



Point **For procedures for reading saved data from CPU memory or transferring saved data to PLC, refer to "Procedures for reading and writing CPU memory (user memory) data" (page 5-30).**

Auto Load Function

Automatically load project data saved to the CPU memory when powered on or at PRG->RUN.
Control memory (CM) may be used to select the folder to be read from several folders.

■ Reading at power on.

1 Using KV STUDIO, you can create project data in the "AutoLoad" or "AtLoad＊＊" (＊＊: 01 - 99) folder names. Using storage transfer tools and FTP server function and so on, you can write project data to CPU memory (user memory).

"Ladder program etc. (except device file)" (page 5-44)

2 Turn off the CPU unit.

3 Connect power supply of CPU unit.

Data are read automatically.

● Read folder selection method

For the selection of folder to be read, save value in CM1790 in advance.

- When CM1790 value is 0
  Read "AutoLoad" folder.
  If the "AutoLoad" folder does not exist in CPU memory (user memory), data will not be automatically loaded. Run the program before power on.
- When CM1790 value is 01 - 99
  Read "AtLoad**" (** : CM1790 value (01-99)) folder.

Folder No. selection steps

1 In case of power on, save the value from external equipments such as touch panel into CM1790.
2 Power off.
3 Power on.
4 Read the folder selected by CM1790.
5 After reading, 0 is saved in CM1790, the read folder No. is saved in CM1792.

Reference

After automatic loading is completed, 0 is saved in CM1790, so when powered on at next time, the same folder No. will not be read. However, please note that, in case 0 is saved in CM1790, if the memory card contains “AutoLoad” folder, it will be read when powered on at next time.

Point

- **When it is write protected with CPU system configuration of "KV STUDIO", an error (auto load failure) will occur and switches to PRG mode. After the error is eliminated, in RUN mode, run the program before automatic loading.**
  In the following cases, error will occur (automatic loading folder), it will enter PRG mode.
  - **When CM1790 value exceeds 100**
  - **When the folder specified by CM1790 cannot be found in CPU memory (user memory)**
- **If the function version of the project stored in the PLC unit and of the project to transfer differ, CPU all clear is required before transferring.**

■ Read for PRG → RUN

1 Use KV STUDIO, the "RunLoad" or "RnLoad＊＊" (＊＊: 01 - 99) folder names to write project data to CPU memory (user memory).

"Ladder program etc. (except device file)" (page 5-44)

2 CPU unit executes PRG → RUN switching.

Read data automatically and begin to run.

● Read folder selection method

Select the folder to be read, save value in CM1794 in advance.

- When CM1794 value is 0
  Read "RunLoad" folder.
  If the "RunLoad" folder does not exist in CPU memory , the data will not be automatically loaded. Run the program before PRG → RUN switching. Error will not occur.

- CM1794 value is 0 1 - 99
  Read "RnLoad**"(CM1794 value (01-99)) folder.

Folder No. selection steps

1 Save the value in CM1794 from external equipment such as touch panel.
2 PRG → RUN switching.
3 Read the folder selected by CM1794.
4 After reading, 0 is saved in CM1794, the read folder No. is saved in CM1796.

Reference

After automatic loading is completed, 0 is saved in CM1794, so when powered on at next time, the same folder No. will not be read. Note that if 0 is stored in CM1790 and CPU memory (user memory) contains a "RunLoad" folder, it will be read in the next PRG → RUN..

Point

- **When it is write protected with CPU system configuration of "KV STUDIO", an error (auto load failure) will occur and can not switch to RUN mode. Starting RUN mode after clearing the error will run the program before auto loading was executed.**
  Error will occur in the following conditions (automatic loading folder), it cannot be set to RUN mode. Starting RUN mode after clearing the error will run the program before auto loading was executed.
  - **When the value of CM1794 exceeds 100**
  - **When the folder specified by CM1794 cannot be found in CPU memory (user memory)**
- **If the function version of the project stored in the PLC unit and of the project to transfer differ, CPU all clear is required before transferring.**

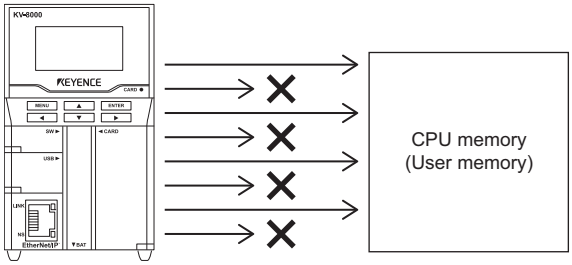
RAM mode

RAM mode is a function for saving files in the CPU memory by creating files in RAM before saving the completed files to CPU memory, which is faster than using CPU memory alone.

Point

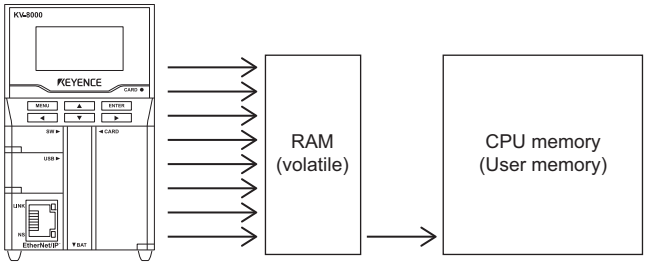
- **RAM which used in RAM mode will be cleared when the power is turned off. Be sure to save to CPU memory before turning off the power.**

When RAM mode is not used



During high-speed saving, write speed is restricted since the CPU memory write time limit is quickly reached.

When RAM mode is used



Since files are created in RAM, it is possible to restrict the number of load operations to CPU memory.

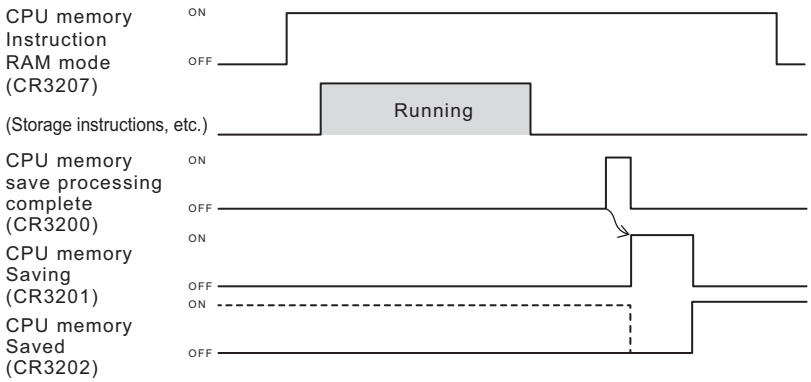
Point

- **When using the file register instructions or storage instruction to access CPU memory, the CPU instruction RAM mode relay (CR3207) goes on to engage RAM mode.**
- **Deselect the "Save files to non-volatile memory" option in the logging/trace setting to run in RAM mode during logging when accessing CPU memory.**
- **The write speed to CPU memory is restricted to prevent memory corruption from write overload.**

■ Setting method

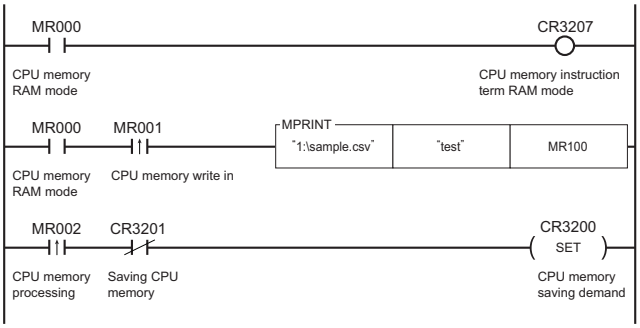
During ladder program operation, the CPU memory instruction RAM mode relay (CR3207) is ON.
When loading to CPU memory, the CPU memory save processing request relay (CR3200) is ON.

Time chart



(1) The CPU memory instruction RAM mode relay (CR3207) turns ON, and storage instructions etc. are executed. (Ladder)
(2) Processing of the storage instruction ends and the CPU memory save processing request relay (CR3200) goes ON when saving becomes necessary. (Ladder)
(3) When saving data to CPU starts, the CPU save processing relay (CR3201) goes ON. If the CPU memory saved relay (CR3202) is ON, it turns OFF. (CPU)
(4) When data has been saved to CPU memory, the CPU save processing relay goes OFF and the CPU memory save processing completed relay goes ON. (CPU)

Ladder program example



<Mnemonic List>

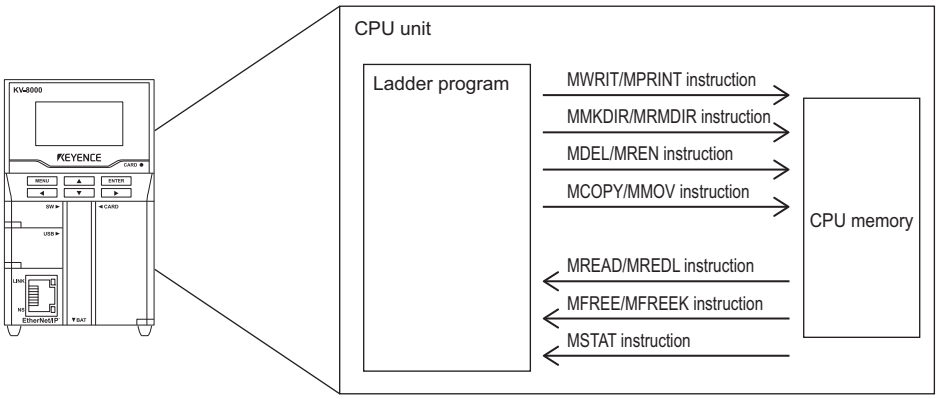
```
LD MR000
OUT CR3207
LD MR000
ANP MR001
MPRINT "1:\sample.csv" "test" MR100
LDP MR002
ANB CR3201
SET CR3200
```

Reference

- The CPU memory save processing demand can be executed even when the CPU memory instruction term RAM mode relay is ON.
- Use the SET instruction to process a CPU memory save request.

Storage Instructions

The storage instruction enables the ladder program to write device values to CPU memory and output values stored on CPU memory.

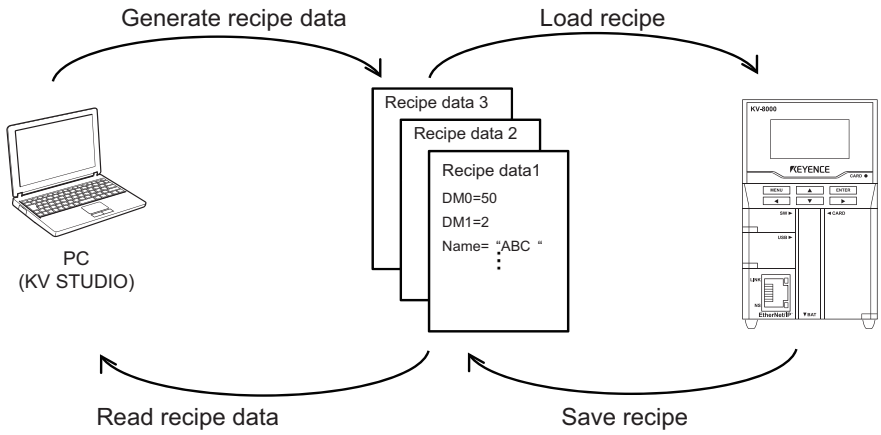


KV-8000/7000/5000/3000/1000 and KV Nano Series Instruction Reference Manual.

5-4 Recipe Load/Save Function

Overview of Recipe Load Function/Recipe Save Function

KV-8000(A) can save device / variable values in multiple data tables, and switch the device / variable values in the CPU unit and the unit variable values in the KV-XD02 by switching data tables. This data table is called recipe data.
The recipe data saved in advance can be read and reflected to the device / variable values after using the recipe load function.
New recipe data can be created with the current device / variable values after using the recipe save function.
Recipe data is generated in KV STUDIO and saved in the storage (memory card / CPU memory) of the CPU unit. It is also possible to read the recipe data in the storage of the CPU unit with KV STUDIO.



The specifications of the recipe load / save function are as follows.

Item	Content
Object of recipe data	Device / variable
Maximum number of recipe data	1000 (0～999)
Storage location of recipe data	Memory card / CPU memory
Load / save in RUN mode	Load / save can be performed even in RUN mode
How to execute recipe load / save	Operate CR/CM from ladder program for execution

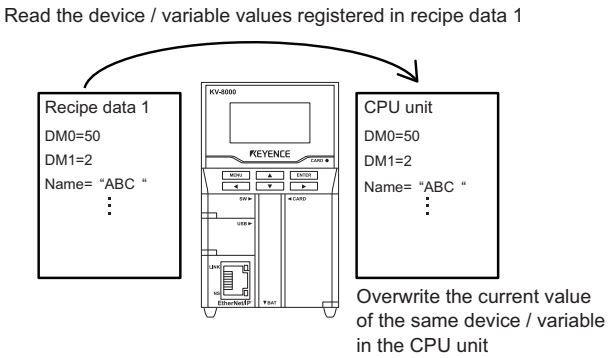
Point

- **The recipe load / save function can be used only with KV-8000 with CPU function version 2.0 or later.**
- **TIMER type and COUNTER type variables cannot be read / saved with the recipe loading / saving function.**

● Recipe load function

The recipe load function is a function that reads the contents of recipe data created in advance and saved in the memory card / CPU memory, and overwrites the device and variable values of CPU unit with the values in the recipe. The recipe to be read is selected by specifying the folder number of the recipe.

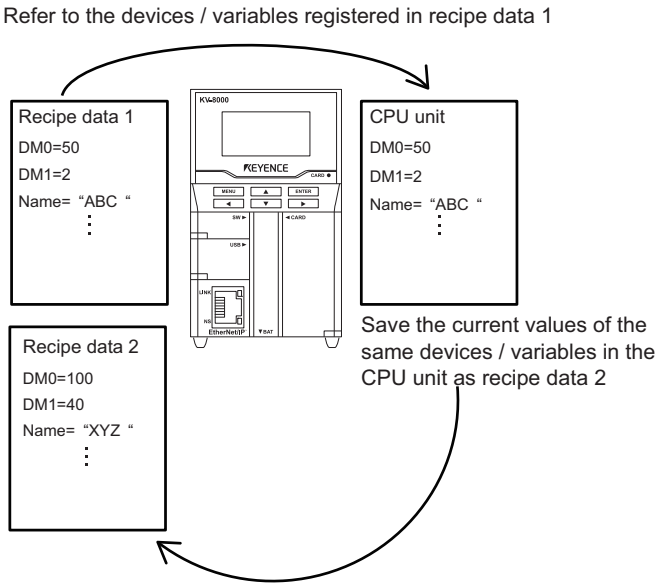
The contents of the recipe data are set in KV STUDIO and transferred to the CPU unit.



● Recipe save function

The recipe save function is a function that refers to the devices / variables registered in the recipe data saved in the reference folder number, and outputs the values of the same devices / variables as the recipe data specified by the output folder number.

The contents of the output recipe data can be read with KV STUDIO.



Reference

The project save function is used to save a backup of all set variables.

"Procedures for reading and writing CPU memory (user memory) data", (page 5-30)

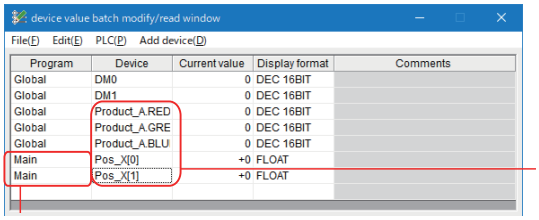
Procedure for generating / reading recipe data

This section describes the procedure for creating recipe data in KV STUDIO and transferring it to the CPU unit storage, and the procedure for reading the recipe data from the CPU unit storage to KV STUDIO.

■ Generation of recipe data (KV STUDIO→CPU unit)

1 **Select "Monitor / Simulator (N)" → "Device Value Batch Change / Read (D)" from the KV STUDIO menu.**

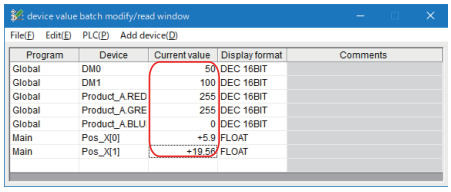
2 **Register the devices / variables included in the recipe data in the device value batch change / read window.**



Specify structure / array type variables up to members or elements.

Specify the program when entering a local variable.

3 **Enter the values to be registered as recipe data.**

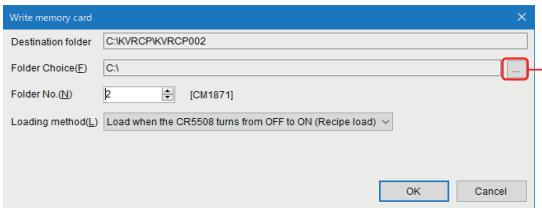


Reference

After selecting "PLC (P)" → "Read all items from PLC (R)" from the device value batch change / read window menu, the device / variable values in the connected CPU unit can be read in batch and reflected to the current values.

4 **Select "File (F)" → "Save to memory card (Z)" from the device value batch change / read window menu.**

For the load method, select "Load when CR5508 turns from OFF to ON (Recipe load)". Specify the folder for creating the recipe data and the recipe data number (Folder number: 0～999), and click "OK".



The dialog for selecting a folder appears

Point

- **When saving directly to the memory card, the location directly under the memory card should be specified. When saving to a location other than the memory card such as a PC folder, the "KVRCP" folder should be copied directly under the memory card.**
- **The load method should be selected before selecting a folder.**
- **When a PC is not granted the permission to write data out, saving to external media such as a memory card will fail. For details, contact your PC administrator.**

5 **Store the recipe data in the storage.**

When recipe data is saved in a memory card, insert the memory card into the CPU unit. When saving recipe data to the CPU memory, use the storage transfer tool, FTP server function or FTP client function, etc. to transfer the "KVRCP" folder directly under the CPU memory.

■ Read recipe data (CPU unit → KV STUDIO)

1 **Transfer the recipe data in the storage of the CPU unit to a PC.**

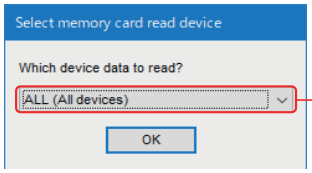
When recipe data is saved in the memory card, remove the memory card and insert it into the PC. When recipe data is saved in the CPU memory, use the storage transfer tool, FTP server function or FTP client function, etc. to transfer the "KVRCP***" folder (*** is the number of recipe data to be read) in the "KVRCP" folder to the PC.

2 **Select "Monitor / Simulator (N)" → "Device Value Batch Change / Read (D)" from the KV STUDIO menu.**

3 **Select "File (F)" → "Read (R)" from the device value batch change / read window menu.**

The dialog for browsing folders is displayed. Select the "KVRCP***" folder transferred to the PC in Step 1.

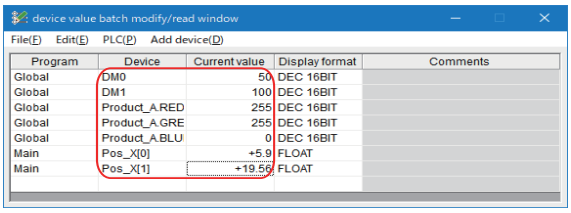
4 **Select the devices / variables to be read from the recipe data to KV STUDIO.**



When "ALL (All devices)" is selected, the values of all devices / variables in the recipe data are read.

5 **The contents of the read recipe data are reflected in the device value batch change / read window.**

After selecting "File (F)" → "Save CSV/TXT file after naming (A)" from the device value batch change / read window menu, the contents of the recipe data can be saved as a CSV/TXT format file.

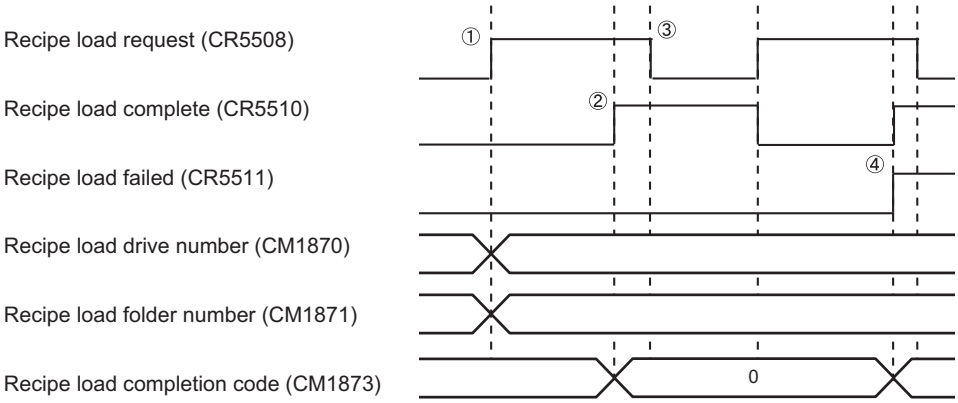


Program	Device	Current value	Display format	Comments
Global	DM0	50	DEC 16BIT	
Global	DM1	100	DEC 16BIT	
Global	Product_A.RED	255	DEC 16BIT	
Global	Product_A.GRE	255	DEC 16BIT	
Global	Product_A.BLU	0	DEC 16BIT	
Main	Pos_X[0]	+5.9	FLOAT	
Main	Pos_X[1]	+19.56	FLOAT	

Recipe load / save procedure

This section describes the procedure for reflecting the contents of the recipe data saved in the storage of the CPU unit to the devices / variables of the CPU unit, and the procedure for overwriting / creating recipe data using the device / variable values of the CPU unit.

■ Recipe load procedure



① Specify the recipe load drive number (0: Memory card / 1: CPU memory) and the drive number and folder number that save the recipe data to be loaded in the recipe load folder number, and turn the recipe load request ON.

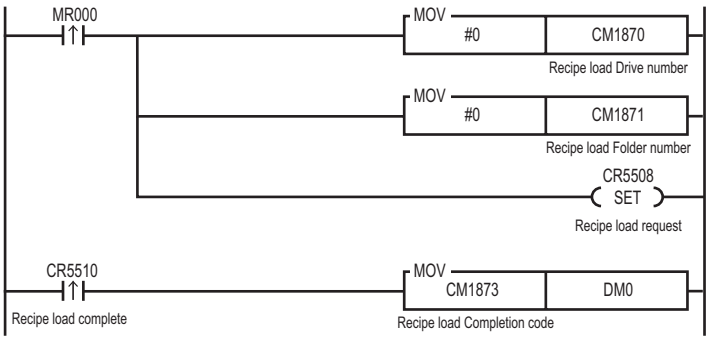
② When the recipe load is completed normally, the recipe load complete turns ON. "0" is also stored in the recipe load completion code.

③ Check that the recipe load complete is ON, and turn the recipe load request OFF from the ladder program. The recipe load complete automatically turns OFF when the next recipe load request is turned from OFF to ON.

④ When the recipe load fails, both the recipe load complete and recipe data execution failed turn ON. The completion code corresponding to the cause of failure is stored in the recipe load completion code.

Reference program

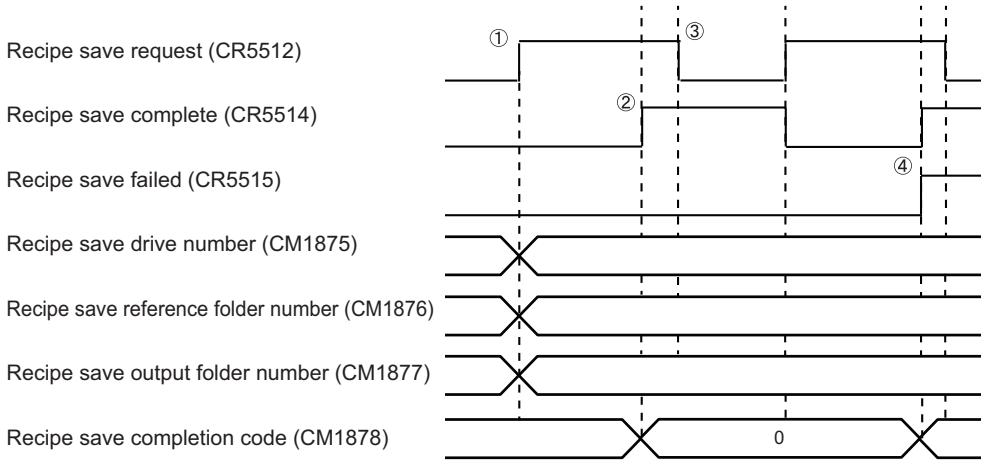
The following is a reference program to load the recipe data of folder number "0" in the memory card.



<Mnemonic list>

```
LDP MR000
MOV #0 CM1870
MOV #0 CM1871
SET CR5508
LDP CR5510
MOV CM1873 DM0
```

■ Recipe save procedure



① Specify the output location of the recipe data with the recipe data drive number (0: Memory card / 1: CPU memory) and the output folder number when saving the recipe, and turn the recipe save request ON. At this time, the devices / variables to be output as recipe data are the devices / variables written in the recipe data with the number specified by the reference folder number when saving the recipe.

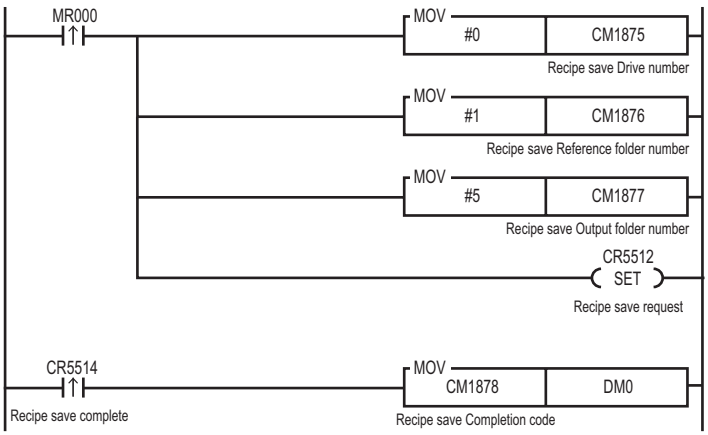
② When the recipe save is completed normally, the recipe save complete turns ON. "0" is also stored in the recipe save completion code.

③ Check that the recipe save complete turns ON, and turn the recipe save request OFF from the ladder program. The recipe save complete automatically turns OFF when the next recipe save request is turned from OFF to ON.

④ When the recipe save fails, both the recipe save complete and recipe data execution failed turn ON. The completion code corresponding to the cause of failure is stored in the recipe save completion code.

Reference program

The following is a reference program that refers to devices / variables registered in the recipe data of folder number "1" in the memory card and saves the values of devices / variables in the PLC as recipe data of folder number "5".



<Mnemonic list>

```
LDP MR000
MOV #0 CM1875
MOV #1 CM1876
MOV #5 CM1877
SET CR5512
LDP CR5514
MOV CM1878 DM0
```

Recipe load completion code / recipe save completion code

List of completion codes stored in the recipe load completion code (CM1873) and recipe save completion code (CM1878).

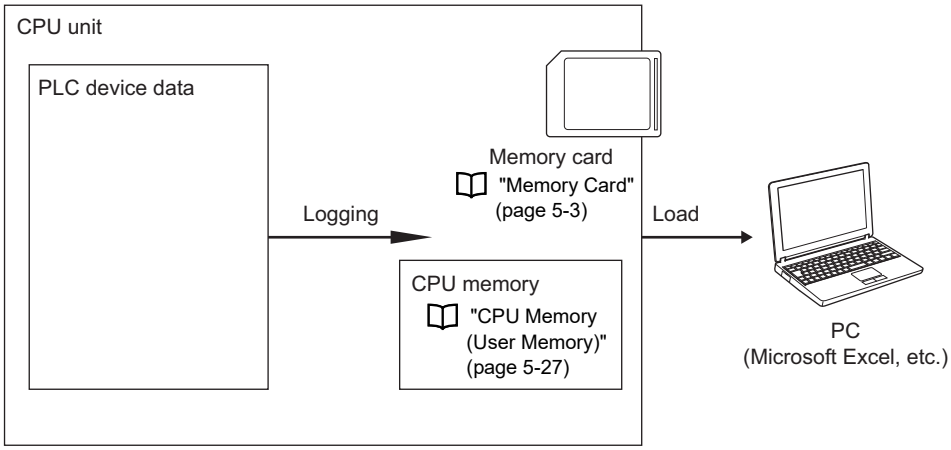
Number	Content	Cause	Countermeasure	At occurrence
2	Unused number not available	The write in was not possible since there was a file already.	Please configure an empty number to the folder number to save.	Save
3	Out of the specified folder number range	The folder number designated at CM was out of range during.	Please designate number from 0 to 999.	Load/Save
4	No data	The folder of CM designated to memory card/CPU memory does not exist.	Please confirm that recipe data is inside.	Load
5	Memory card not found ---------- CPU Memory unmounted	The load or save was executed during the condition without being able to recognize memory card/ CPU memory.	Please insert the memory card when loading from the memory card, or saving to memory card. Please mount the CPU memory.	Load/Save
6	Access conflict	The save/load was not possible since transfer or read out of the program was done from another route.	Please execute save/load after the transfer or read out of the program is complete.	Load/Save
12	No free space	There are no free space in the memory card/CPU memory.	Please increase the free space.	Load/Save
21	Out of the specified drive number range	When executing CR, the drive number specified by CM is out of the specified range.	Specify the drive number as 0 or 1.	Load/Save
22	Unregistered variables included	The variables included in the recipe data have not been defined in the variable edit window.	Please define the variables (including structure type and function block type, etc.) included in the recipe data in the variable edit window.	Load / Save
23	Variable data type mismatch	The data type of variables included in the recipe data does not match the data type of variables registered in the PLC.	Make sure that the data type of variables included in the recipe data matches the data type defined in the variable edit window.	Load / Save
24	Unsupported variable data type	The recipe data contains variables of unsupported data type.	Check the data type of variables included in the recipe data.	Load / Save
100	Memory card cover open	The memory card cover is opened.	Please close the memory card cover.	Load/Save
101	Memory card locked	The memory card is in LOCK status.	Please release LOCK or use memory card that is writable.	Save
102	Memory card error	The memory card is damaged.	Please check the file system of the memory card.	Load ---------- Save
103	Max no. of files exceeded	The sum of number of files and folders inside the user memory has gone over the upper limit.	Please reduce the file or folder inside the user memory.	Load/Save
1000	Cannot load data	• The file is damaged. • The memory card is damaged. • The card has been extracted during the loading process. • It tried to load from CPU memory with the configuration "Load after clearing the user memory".	Please confirm if the project and memory card is appropriate.	Load
1001	Cannot save data	• The memory card has been damaged. • The card has been extracted during save process.	Please confirm if the project and memory card is appropriate.	Save

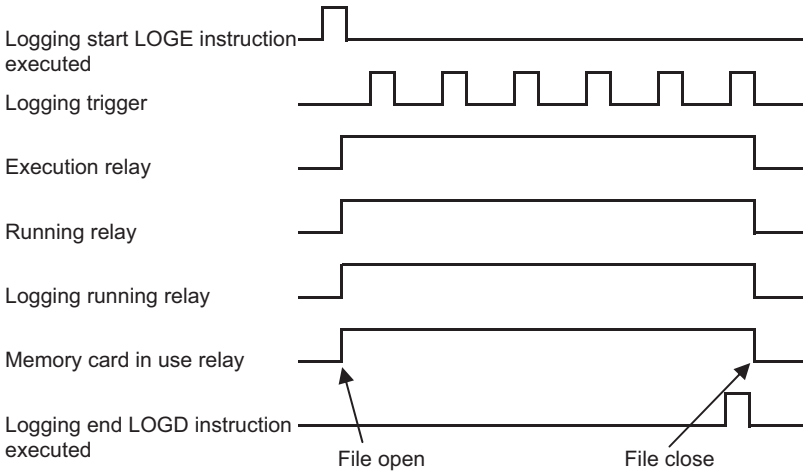
Overview

● Logging function

The recording function is a function to load the status or value of the specified device at any time (trigger). The captured data is in the CSV file format and is automatically saved to memory card or CPU memory (user memory). Device loading or triggering can be easily set through KV STUDIO.
Since the data is saved in CSV format, it can be directly loaded into Microsoft Excel, etc.
Data saved in CSV format can be read in the following way.

- Extract the memory card and read it with a PC
- Use the storage transfer tool or FTP function to read via PLC
- Use the real time chart monitor to read





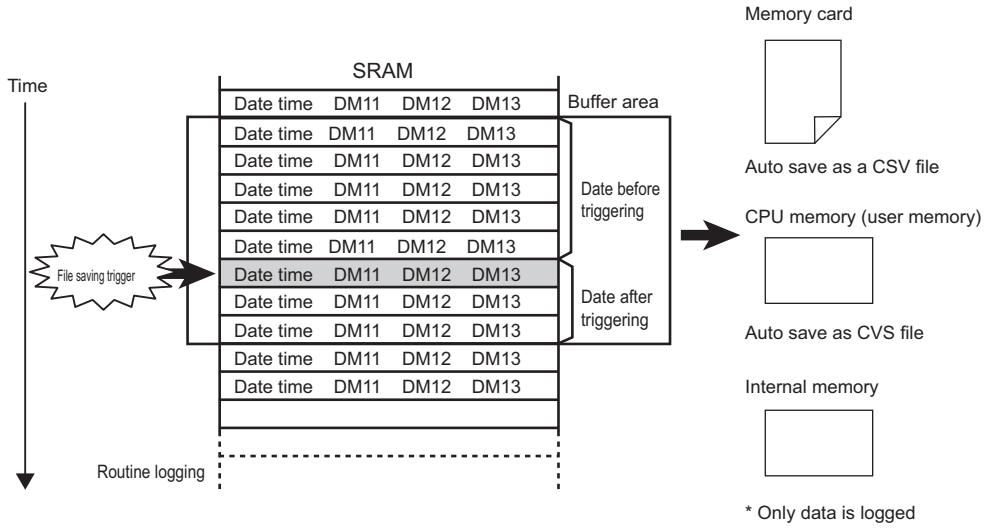
*Relay action may vary depending on the option settings.

Reference

- If logging/tracing is started without the file being present, a new file will be automatically created.
- Logging/Tracing function and Storage instruction can be used at the same time. The order of priority is Storage instruction> Logging/Tracing function.

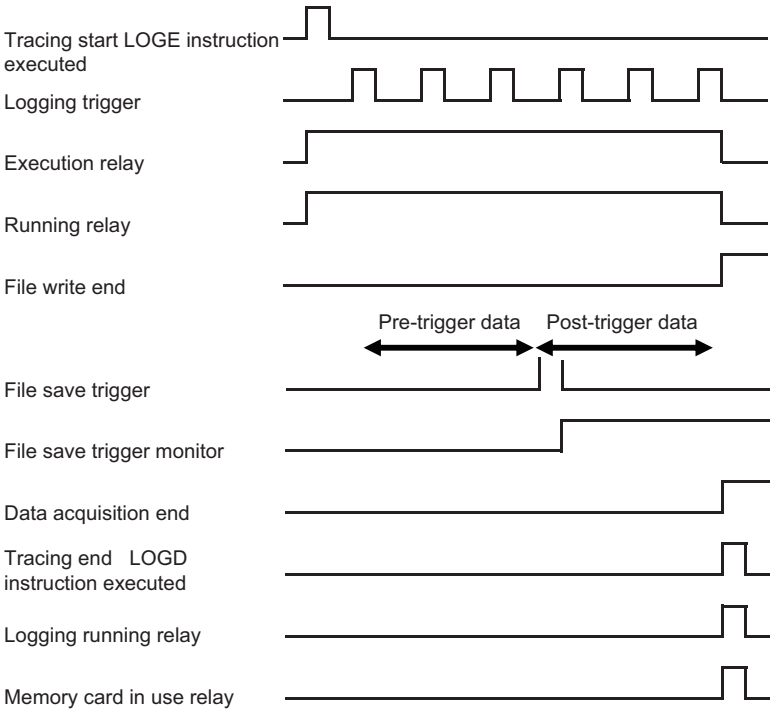
● Tracing Function

As in the case of logging function, the state or value of device will be logged in buffer memory. When files are saved, only the specified amount of data preceding and following is extracted and logged in memory card, CPU memory (user memory) or internal memory.
For example, after trigger setting is made for saving files in the event of abnormality of equipment, the data prior to or after occurrence of abnormality will be saved automatically, which facilitates abnormality analysis.



Reference

After the file save destination is set as "internal memory", the data saved can be read from the real-time diagram monitor of KV STUDIO.



Point

When saving data acquired by the tracing function on a memory card, execute the LOGD instruction by using the data acquisition end as a trigger.
Files will not be generated even if the LOGD instruction is executed before data acquisition is completed.

How to Use

The steps of using logging/tracing functions are described below.

(1) Setup device and trigger

"Tool (T)" - "Logging/tracing setting (L)"

"Setup" (page 5-60)

(2) Create the ladder program

"LOGE" "LOGD" "TRGD" instruction

"Logging Instructions" (page 5-73)

(3) Transfer to KV-8000

"Monitor/simulator (N)" - "PLC transfer" → "Monitor (C)"

(4) Execute Logging/tracing

Execute the LOGE instruction to start logging.
If logging triggers occur (file save trigger is also required while track is being used) while the log is being executed, data will be written into the file.
Execute the LOGD instruction and close the file.

Setup

Set up through KV STUDIO to use logging/tracing function.
The following settings are required when using logging/tracing function.

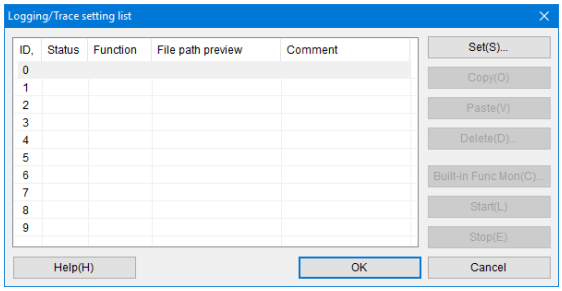
- •Data save destination
- •Save the settings of file name
- •Function (logging/tracing) selection
- •Setting of logging/tracing device
- •Setting of trigger
- •Options setting
- •Detailed setting of tracing (only when tracing function is selected)

logging/tracing function permits to create up to 10 files, and divides them through logging/tracing ID (0 to 9).
Set separately the file names, device and trigger depending upon every logging/tracing ID.

1 Select "Tool (T)" → "Setup logging/trace" from the menu.

"List of logging settings list" dialog box is displayed.

Other ·Click the " " button.



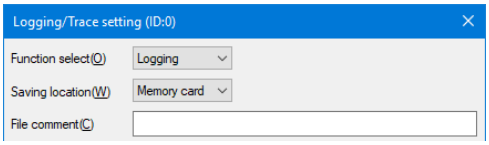
Item	Description
Status	Displays the current logging and tracing function execution status (stopped / running / error) only during monitoring.
Function	The functions (logging/tracing) already set are displayed
File path preview	An example of file saving position (file path) is displayed.
Comment	File comment already set is displayed.
Setup	"Logging/tracing setting" dialog box is displayed. The selected logging/tracing ID is set.
Copy	Copy the setting of the selected Logging/tracing ID.
Paste	Paste the copied setting to the selected Logging/tracing ID.
Delete	Delete the setting of selected logging/tracing ID.
OK	Confirm the setting contents and close the dialog box.
Cancel	Cancel the setting contents and close the dialog box.
Built-in function monitor	The built-in function monitor of the selected logging/tracing ID is displayed during monitoring process.
Start	The selected logging/tracing ID is implemented only during monitoring process (Ladder Program first)
Stop	The selected logging/tracing ID is implemented only during monitoring process (Ladder Program first)

2 Select logging/tracing ID to be set, and press "Set up" button.

Logging/tracing setting" dialog box is displayed.

Item	Description
Function selection	Selects function (Logging/trace).
Data location	Select "memory card/CPU memory/internal memory." ("Internal memory" is available only when (Trace) is selected in "Function selection." *1)
File comment	In the Logging/trace setting, choose [Option] >[Insert file comment in CSV file] to select the checkbox and enter a file comment. A comment of (up to 64 single-byte characters) can be entered on the first line of a CSV file.

*1 Use [real-time chart monitor] to load data when internal memory is selected.
See "Real-time chart monitor" in the "KV STUDIO User's Manual."
Data saved in internal memory will be lost when the power is turned off.



"Logging/Tracing" (page 5-57)

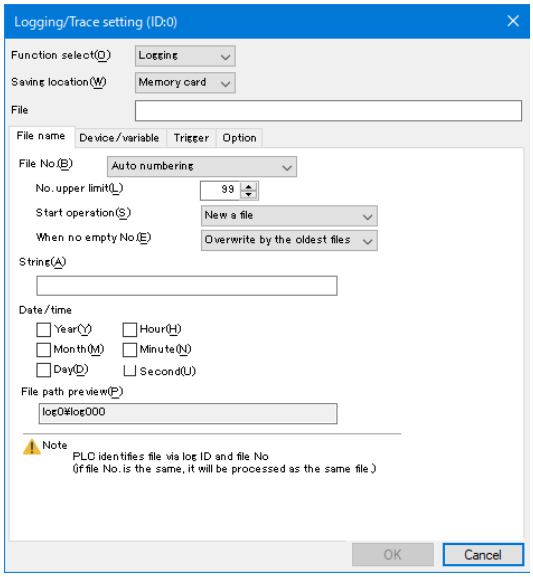
File name

To set the files to be saved.

Name of folder for saving files will be determined depending on every logging/tracing ID. (LOG0 to LOG9)

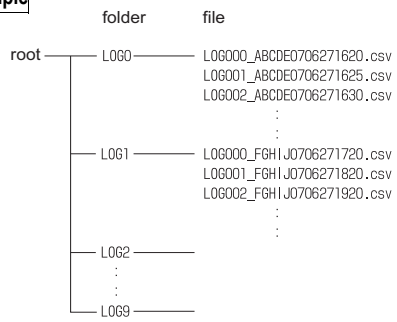
Reference

When the data save destination is "internal memory", each item of "file name" can not be set.



Data of log files are saved on the memory card or CPU memory (user memory) by the following forms.

Example



Item			Description
File No.	File No.		Select from **"Auto numbering/Fixed/Dev val (lower 3 dgts)"**.
	Auto numbering*1		Between 0 to 999, use the minimum unused file No.
		No. upper limit	Set the upper limit (0 to 999) of file NO.
		Start operation	Select operation at logging start from **"New file/Add to latest file"**.
		When no empty No.	If there isn't remaining No., please select **"Overwrite from the oldest file/ Stop logging"**.
	Fix		Specify the fixed value between 0 to 999.
		No.	Set up file No.(0 to 999).
		When the same files	When there are same files, select **"Add to the old file/Add"**.
	Device value (use last 3 digits)		Specify using word device value (0 to 999) when enabling is performed.
		Device	Set the device to store the file number or UINT/UDINT/TIMER/COUNTER type variables.
		When the same file	When there are same files, select **"Add to the old file/Add"**.
String*2			Set the character strings attached to the file names.
Date/time*2			If this option box is selected, the file name will contain creation date and time information when the LOGE instruction is executed. The date and time information is fixed as the time when reading the new. So, the file name isn't changed when adding. Original files are deleted when overwriting the existing ones, so, the date and time information of new files is the information when file is created. Year (00 to 99), month(01 to 12), day(01 to 31), hour(00 to 23), minute (00 to 59), second (00 to 59)
File path preview			The file path (the location where the file will be saved) is displayed.

*1 After retrieving of remaining No., CPU unit may prepare to save data to the files. In this case, the relay for logging the working condition is ON. Buffer overflow is liable to occur if trigger is previously activated, so don't activate the trigger after the relay for logging the working condition is ON.
The higher the upper limit of No., the longer the time for retrieving remaining NO. The approximate time necessary when the upper limit of number is 999 is 1 second.

*2 Select Logging ID4, file number 2, character string "KEYENCE," date information "YYMMDDHHMMSS"
If an update is made at 15:16:17 on March 14, 2019, a "log002_KEYENCE190417.csv" will be created in the "log4"route folder on memory card or CPU memory.

File name: $\underbrace{\text{log (File No.) _ (Character string) (Date)}}_{\text{Up to 32 half-width characters}}$

For information about what to do in case there is no room in the file No., please see "Processing at start of logging/tracing" (page 5-84).

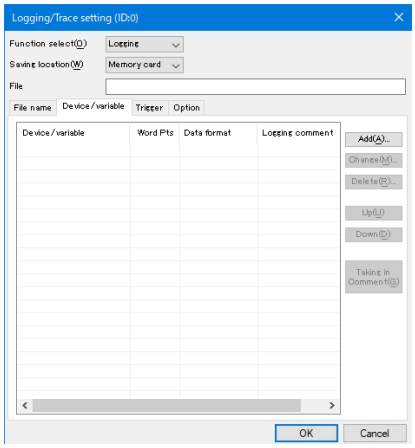
Point

The files created with the recording and tracking function will be identified using the folder (recording and tracking ID) and file number.
In a storage device, do not create multiple files with the same file number but containing different character strings or date and time data in the same folder (logging ID).

Device / variable

To set the saved device / variable by mixing the bit data and word data.

At most 512 devices can be set for every logging ID.



Example

(1) When all 512 devices are set to the bit data:

ON/OFF state (0 or 1) of 512 bit data will be logged.

(2) When all 512 devices are set to the word data with 1 words:

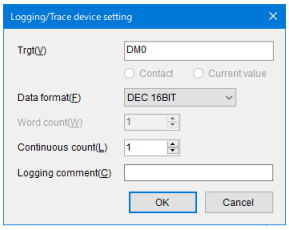
Values of 512 word data with 1 words will be logged.

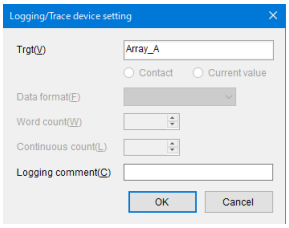
(3) When all 512 devices are set to the word data with 2 words*1:

Value of 512 word data with 2 words will be logged.

*1 It also includes devices that occupy 32 bits in 1 word, such as TC/CC/TRM/Z.

Item	Description
Add	"Logging/tracing device setting" dialog box is displayed when "Add" button is pressed. The logging/tracing device is added.
Change	Select the changed device. Press "Change" button to display "Logging/tracing device setting"dialog box. The logging/tracing device setting is changed.
Delete	Press “Delete" button to delete the selected device. (Multiple device can be selected)
Up	Shift the selected device up in the registering sequence.
Down	Shift the selected device down in the registering sequence.
Taking in device comment	When you press the "Taking in device comment" button, the device comment of the set device will be imported into the logging comment area.

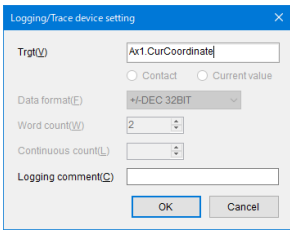




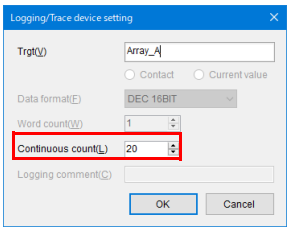
Project	Description
Target*1	Set the devices or variables to be recorded / traced.
Data format	For devices Select from "Decimal 16BIT / Decimal 32BIT / ± Decimal 16BIT / ± Decimal 32BIT / Hexadecimal 16BIT / Hexadecimal 32BIT / ASCII16BIT / FLOAT". Please set to ASCII16BIT to obtain the SJIS code. When "FLOAT" is selected, the data will be saved in exponential format. (Example: 0.123E-03) To obtain data with a decimal point, convert it to a string using the ladder program, and select "ASCII16BIT" to save it. For variables Only global variables can be specified, and the data type set in the variable edit window is displayed and cannot be set. Variables with data types other than LREAL can be specified.
Number of words	This can be set only when "ASCII16BIT" is selected as the data type or a variable set to STRING type is input. Setting range: 1 to 512
Continuous number	Place the device set through destination device on the front part, and set the number of devices for continuous logging/tracing. Only array type variables can be set with variables registered.

Reference

When setting a structure type variable, specify up to the member name.

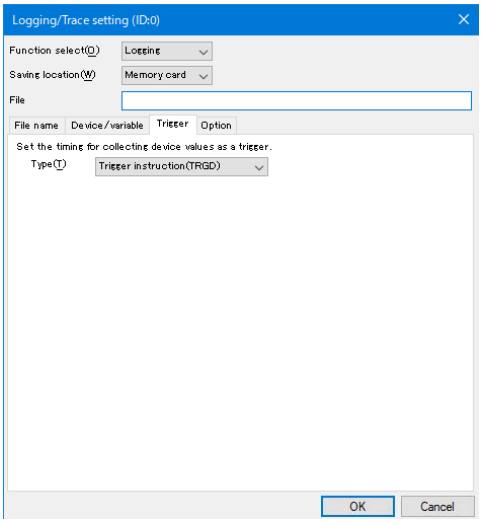


When setting an array type variable, enter the array name only
The array elements can be set in batch by setting the continuous number.



Trigger

To set timing of logging/tracing of registered device / variable.



Item		Description
Type *1		Select the trigger type from the following. "Bit device" "Period" "Trigger command (TRGD)" "Data acquisition trigger (flow)"
Bit device		Enter the bit device or BOOL/TIMER/COUNTER type variable name to be used as a trigger.
cycle	cycle	Set the cycle (1 to 86400000) (unit: ms).
	Start trigger when starting logging	If the checkbox is selected, trigger is activated when logging/tracing is started.
Trigger instruction		Triggering occurs when TRGD instruction is executed. "TRGD Instruction", Page 5-77

***1 Bit device**

Logging/tracing is run when the specified bit data is in ON state.
Referenced to the processing state of END, the state of bit data isn't logged when ON/OFF during 1 scan.
Device value at END processing of detecting trigger is saved.
If bit data is ON, the device / variable value for each scan and END processing will be saved.

Cycle

Through the built-in timer of CPU unit, logging/tracing according to the specified interval time is run.
Device / variable value at END processing of detecting trigger.
Every scanning is performed when specified time is less than scanning time.
Time interval is calculated starting from the operation of logging/tracing.
In PRG mode, the timing of time interval is reset.

Point

Through END processing after trigger is activated, the specified device / variable value is read. Thus, at most 1 scan can be delayed from trigger activation to device / variable value saving.

TRGD instruction

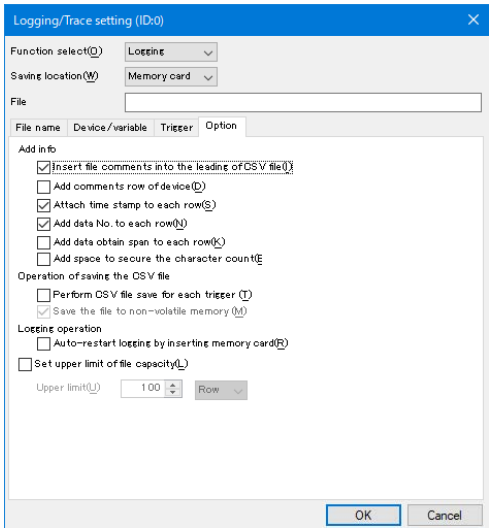
The device / variable value when performing TRGD instruction is loaded, without the need of waiting for END processing.
At most 8 value can be logged in the program for 1 ID.
Can be also programmed to interruption program or fixed period module, inter-unit synchronization module.

For the Data acquisition trigger (flow)

With the unit program (flow) of the X-Unit function, the device / variable value at the END processing time after the "Operation: Data acquisition trigger " execution starting in logging / trace block is loaded.
The block ends after loading the value.
Place the unit supported by unit editor to create a unit program.

Option

To set the detailed condition of additional information or operation. When the tracing functions is selected, no setting can be done if the file saving destination is "internal memory".



Item		Description
Add information*3	Insert file comments into the leading of CSV file	If checked, a file comment is entered at the start of a CSV file.
	Add comments row of device *1	If checked, a device comment is entered. If a comment has not been set, the device number will be added.
	Attach time stamp to each row *2	If checked, a time stamp is added to every line.
	Add data No. to each row	If checked, a time stamp is added to every line. Logging/tracing will start from 0. When "Add" is selected in a file number setting, numbers will start from 0 at the middle of the file (the position where numbers are first added). When a buffer overflow occurs, the data number will reveal how much data was lost. The buffer capacity that is targeted in an overflow can be set by opening the [CPU system setting] in the work space and setting [System setting] and [CPU unit buffer capacity setting].
	Add data obtain span to each row	The time elapsed from the most recent data acquisition will be logged in 1μs unit.
	Add space to secure the character count	If checked, blank spaces are added for every numeric value that can be zero suppressed. The file size increases, but the location of the first character can be obtained through calculation.
Save CSV file	Perform CSV file save for each trigger	If checked, data is written to the memory card at each trigger occurrence. High-speed logging is possible because the result of each trigger is stored in the internal buffer area, and the memory card writing is executed only when the LOGD instruction is executed.
	Save the file to non-volatile memory	If checked, data is written to nonvolatile memory at each trigger occurrence.
Logging operation	Auto-restart logging by inserting memory card	If checked, logging data will be stored in CPU memory should the memory card be removed during logging until the memory card is reinserted. When reinserted, the data is automatically saved to the memory card. The operation related to saving to memory card at automatic restart depends on "operation at start". "File name" (page 5-62)
Set upper limit of file capacity (Only logging)		If checked, you can set the upper limit of file capacity. See "Troubleshooting for overflow of file capacity settings" (page 5-85).
The upper limit *6(only logging)		The upper limit is set in line or KB units.

*1 The device for which the device comment is not transferred to CPU unit will be stored to a device No.(e.g. DM0).

*2 The year, month, day, hour, minute and second in the time stamp are saved in the format of CSV.

Example

2019/04/17,15:16:17

*3 "CSV File sample" (page 5-72)

*4 **When the saving destination is memory card**

The file will stay open during login.

When power is turned OFF while the file is open, logging result will not remain in the memory card.

Always remember to execute LOGD instruction before turning power OFF.

When "Save CSV file for every trigger " is checked, the logging result will remain even if power is turned OFF as long as the write in after trigger has been completed while the logging is executed.

But when power is turned OFF while writing in to memory card, there is a risk of losing content of the memory card.

In addition, since write in to memory card occurs for every trigger, the accessing time to memory card will become longer.

Therefore, the buffer overflow will become more frequent.

When the interval of trigger is short, please uncheck the "Saving CSV file for each trigger".

When the saving destination is CPU memory

It always executes save to CSV file for each trigger, but it differs by the check status of "Executes file save to non memory" if it will remain in the CPU memory even if power is turned OFF.

When it is checked

Since it will be saved to CPU memory for every trigger, the file will remain in CPU memory even when power is turned OFF. But the accessing time will become longer.

When check is off

The high speed login is possible since data will be created in RAM.

However, since the data created in the RAM mode will be cleared when the power is turned off, you need to turn on CR3200 and execute the CPU memory save save request before turning off the power supply.

"RAM mode" (page 5-47)

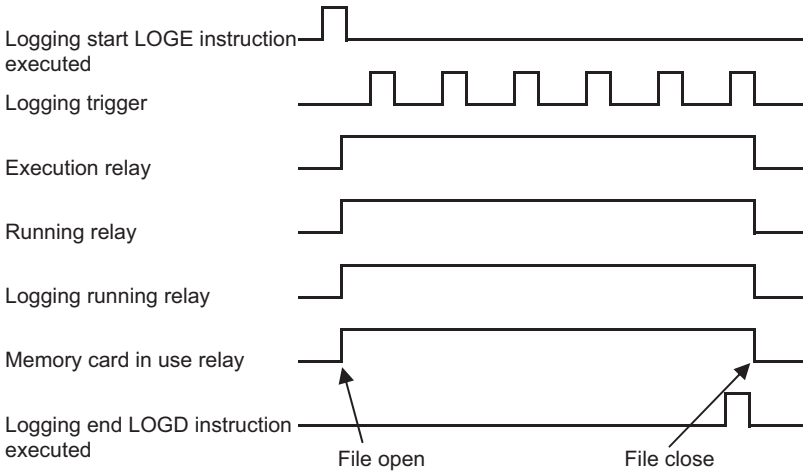
*5 This function is "Auto restart", not "Auto start". When LOGE instruction is executed and memory card isn't inserted into the unit, the log error relay will be opened without starting logging.

"Auto restart" refers to the function that a restart will be performed when the memory card is inserted.

*6 The number of lines, file comment and device comment occupy 1 line respectively.

File size indicates the size including all file comments and device comments.

"Program Capacity" (page 4-14)

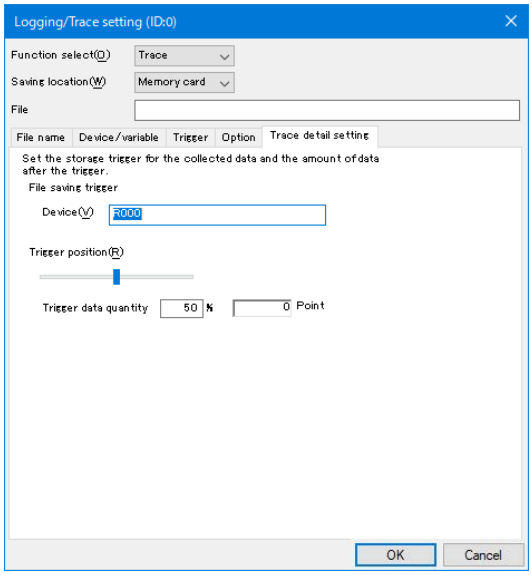


*Relay action may vary depending on the option settings.

Trace detail setting

Setting is allowed when tracing function is selected.

Triggering or saving method when the files are saved will be set.



Item		Description
File Saving trigger	Device	Specify a bit device or BOOL type variable.*2
Trigger position	Trigger data size (%)	Press % to set the data size after trigger. Slider can also set up.
	Trigger data size (point)*1	The data size set by % is displayed by number of points.

*1 Open "CPU system setting" from work space, Set it with [CPU unit buffer capacity setting]. For details, see the "CPU unit buffer capacity setting" (page 5-70)

*2 When trigger type is "TRGD instruction", the file saving/triggering state will be confirmed only when TRGD instruction/function executed.

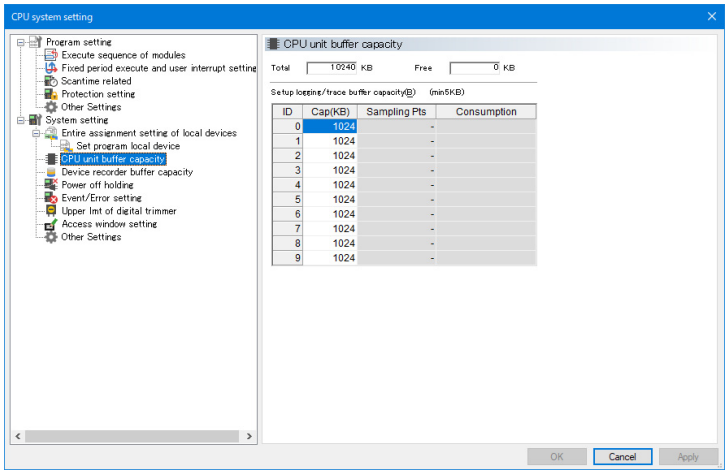
Reference

The traced data saved using the tracing function can be read by the real time chart monitor. For details, see the "KV STUDIO User's Manual."

CPU unit buffer capacity setting

Set the capacity of the buffer to save data with the logging/tracing function.

1 **Open [CPU system setting] from the workspace and click in order of [System setting] → [CPU unit buffer capacity setting].**



2 **Set the capacity of the buffer to be used for ID0 to ID9. If you increase the capacity of the ID to be used, the amount (the number of samples) of data that can be saved also increases. Please set within the range not exceeding the total capacity 10240KB of buffer.**

About CSV File

■ CSV files

CSV files generated automatically through logging/tracing function comprise device data, additional data information and additional file information. A half-width character is counted as 1 character, and a full-width character is counted as 2 characters. In addition, these letters align left if the data format is ASCII character string, otherwise align right.

● File additional information

The following items will be saved from the first line of the file according to the setting contents "Options" tab of "logging/tracing setting" dialog box.

Item	1 column character length	Example	Note
File Comment	64	Header	The length of 1st line includes 64 characters." " is not included in the character length.
File saving trigger state (only tracing)	32	"TRIGGER DATA NO" "COMM TRIGGER DATA NO"	" " is not included in the character length.
File saving trigger occurring position (only tracing)	10	1234567890	The data No. is logged when file saving trigger is activated.
Logging Comments	32	…, "Starting relay" ,…	" " is not included in the character length.

*Doesn't include comma.

● Additional data information

According to the setting contents in the "Option", the following items are saved from the left side of device data in sequence.

Item	1 Column character length	Example
Data No.	10	1234567890
Timing flag(date)	10	2019/04/17
Timing flag(time)	8	11:12:13
Data capture interval	10	1234567890

*Doesn't include comma.

● Device data

According to the setting of "Device", it's saved by the following character length.

Item	1 Column character length*	Example	Note
Bit device	1	…,1,…	ON:1 OFF: 0
±DEX16 bits	6	…,␣ -1234,…	No "+" is added in the case of positive value.
DEX 16 bits	5	…,␣ 1234,…	-
±DEX32 bits	11	…,␣1234567890,…	No "+" is added in the case of positive value.
DEX 32 bits	10	…,␣123456789,…	-
HEX 16 bits	4	…,BA98,…	
HEX 32 bits	8	…,BA987654,…	
FLOAT	13	…,␣␣␣2.4123E+12,…	Exponent format is fixed. To log with decimal point, convert the floating-point type data to a character string using the FASC instruction in the ladder program and log it as ASCII 16 BIT.
ASCII16BIT	(words)×2+2	…,"a123"␣…␣,…	If NULL exists in the character string, the subsequent character strings are not logged.

■ CSV File sample

Data No. File Comment Time stamp (Logged time) Data acquisition interval (μs) Logging comment

Manufacturing line A dimensional error							
DATA No	DATE	TIME	INTERVAL	Production c	Dimensiona	Cumulative	Abnormal re
0	2019/4/19	11:20:09	0	851	-2	207326	0
1	2019/4/19	11:20:10	999987	852	-6	207327	0
2	2019/4/19	11:20:11	1000002	853	8	207328	0
3	2019/4/19	11:20:12	1000025	854	16	207329	0
4	2019/4/19	11:20:13	999980	855	31	207330	0
5	2019/4/19	11:20:14	1000003	856	22	207331	0
6	2019/4/19	11:20:15	999998	857	-38	207332	0
7	2019/4/19	11:20:16	999991	858	22	207333	0
8	2019/4/19	11:20:17	1000023	859	35	207334	0
9	2019/4/19	11:20:18	999998	860	-34	207335	0
10	2019/4/19	11:20:19	1000009	861	12	207336	0
11	2019/4/19	11:20:20	999984	862	21	207337	0
12	2019/4/19	11:20:21	1000001	863	-1	207338	0
13	2019/4/19	11:20:22	1000013	864	-37	207339	0
14	2019/4/19	11:20:23	999978	865	44	207340	0
15	2019/4/19	11:20:24	1000002	866	40	207341	0
16	2019/4/19	11:20:25	1000009	867	14	207342	0

← File comment line
← Comment line of device
Data line

Reference

The logging comment set in the [Logging / Tracing Setting] dialog is displayed in the comment line of the device. When the logging comment is blank, the device name or variable name is displayed.

Logging Instructions

To implement the logging operation, a program for using logging instruction can be written.

Instruction	mnemonics	Description about instruction	Page
Enable logging/tracing	LOGE	Set logging of ID logging/tracing to enabled/inhibited state	5-75
Inhibited logging/tracing	LOGD		5-75
Capture log data	TRGD	Capture log data of ID specified.	5-77

■ Calculation of file size

File size can be calculated as follows. The unit to be used is byte.

(size of a file comments line)
+ (size of a device comments line)
+ (size of a data line) x (number of data lines)
= (file size)

(size of file comments line) = (the number of characters set up with KV STUDIO) + 4
A half-width character is counted as a letter and a full-width character is counted as 2 letters.

(size of a device comments line) ={35x(number of device comments)}+1
={35x((number of devices)+A+B)}+1

"attach a data No. to each line": Use (A=1) Not Use (A=0)
"Attach a time stamp to each line" : Use (B=2) Not Use (B=0)

(size of data line) = (total size of devices)+1

Size of device

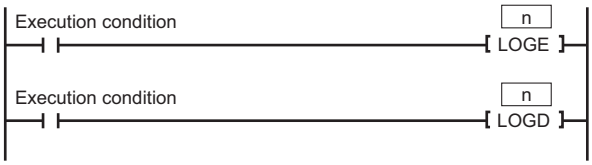
Item			Size
Bit device			2
Word device	1 word	Unsigned	6
		Signed	7
	2 words	Unsigned	11
		Signed	12
	Floating point real number		14
	ASCII character string (n words)		(n x 2) + 3
Data No.			11
Time stamp (date + time)			20
Data capture interval			2~11

NOTICE	**An approximate number of files that can be stored can be calculated using the file size, memory card capacity and CPU memory capacity calculated in the above steps. However, the actual number of files that can be stored may vary depending on disk use. Please ensure sufficient free space is available on the disk.**

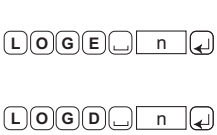
MEMO

LOGE	-[LOGE]-	Enable logging/tracing	Set logging of logging/tracing to enabled/inhibited state
LOGD	-[LOGD]-	Inhibited logging/tracing	

Ladder program



Entry mode



Operand	Description	Occupied size
n	Logging/tracing specifying ID (0 to 9).*1 *2	16 bits

*1 Cannot be specified in "$".

*2 When bit device is specified in n, continuous 16 bit will be occupied.
If word device is specified, 1 word will be occupied.

Description of Operation

LOGE

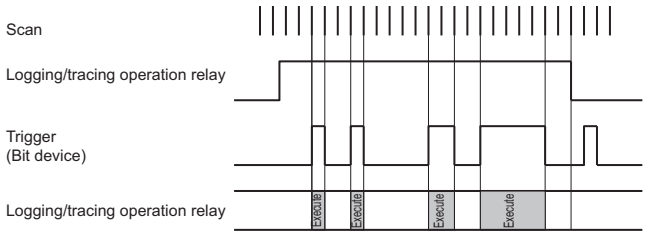
When execute condition is ON, n will set ID of specified logging/tracing to enabled state.
When the instruction is performed, CPU unit buffer of specified ID is cleared, and the logging/tracing operation relay is set. When a trigger is activated, logging/tracing will be performed.
Operation enable state will continue before LOGD instruction is executed, independent of the execution condition of the LOGE instruction.
The logging/tracing of all log IDs can enter into operation enable state under the same execution condition.

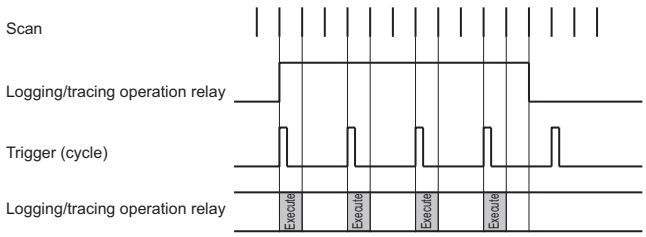
LOGD

When execute condition is ON, n will set ID of specified logging/tracing as inhibited state.
After the instruction is executed, logging/tracing operation relay of every ID is reset, then logging/tracing is stopped. All log contents till that time are output to the memory card or CPU memory.

Log ID	Logging operation relay
0	CR000
1	CR100
2	CR200
3	CR300
4	CR400
5	CR500
6	CR600
7	CR700
8	CR800
9	CR900

Timing chart of logging/tracing





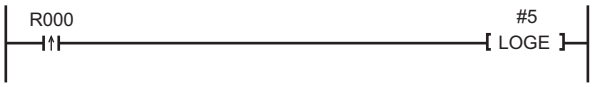
Operation flag

CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	It's ON when a value larger than 10 is specified in [n], or in direct specifying and index change range are not suitable; it is OFF in other cases. There is no change when indirect specifying and index change are not specified in operand.

* If CR2012 is ON, the instruction will be not executed.
When CR2012 is ON, the detailed error message is saved into CM5100 to CM5176.
"Latest error details (CM)" (page 4-87)

Sample Program

When input R000 is ON, logging/tracing of ID5 enters into operation enable state.



<Mnemonic List>

```
LDP R000
LOGE #5
```

When input R001 is ON, logging/tracing of ID5 enters into operation inhibited state.

<Mnemonic List>

```
LDP R001
LOGE #5
```

TRGD

-[TRGD]- **Capture log data** **Capture log data of specified ID.**

Ladder program

Input mode

Operand	Description	Occupied size
n	Specify the logging/tracing ID for capturing, or save the first device of logging/tracing ID. *1	

*1 Specify the logging ID within the range of 0 to 10. Specify "10" when executing instruction trigger using real time chart monitor or real time logging of KV COM + for Excel.

Description of Operation

TRGD When execution condition is ON, data of logging/tracing ID in n will be captured.

- Capture the data until TRGD instruction is performed.
  Data can be Captured during END processing when bit device or period for triggering logging/tracing; if TRGD instruction is used, data can be Captured anytime from the ladder.

Point

- **when using the TRGD instruction to capture data, please select Trigger Instruction (TRGD) as the Trigger Type in the Logging/Tracing Settings of KV STUDIO.**
- **To use Instruction Trigger in the Real-time chart monitor, select Instruction Trigger from the Trigger Type in the Trigger setting of the Real-time chart monitor screen. Otherwise no data will be captured even if the TRGD instruction is executed.**
- **At most 8 TRGD instructions can be used by 1 project for 1 ID.**

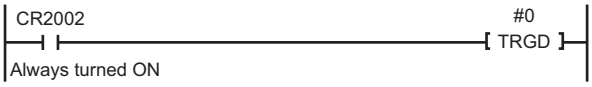
Operation flag

CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	· It is ON when the operand value is beyond the specified range. · It is ON when indirect specifying, and pointer change range are not suitable. Otherwise, the value is OFF. · No change will occur when n is specified as a constant.

* If CR2012 is ON, the instruction will not be executed.
When CR2012 is ON, detailed error info are stored in CM5100 to CM5176.
"Latest error details (CM)" (page 4-87)

Sample Program

If the following ladder program is included in the fixed period module, logging will be performed every period.



<Mnemonic List>

```
LD CR2002
TRGD #0
```

Point

When fixed period module is implemented, it's necessary to set it to interrupt enable state (perform EI instruction).

"Fixed-Period Module" (page 4-103)
"4-13 Interrupts" (page 4-206)

Device List for Logging/Tracing

Control information and error information for logging/tracing are assigned to the control relay and control memory. Control relays include control relays for each logging/tracing ID and a relay only reserved for the system.

Point

The control relay/control memory is reflected through END processing. Under locking state, no results can be reflected prior to END processing, even if operation is performed.

■ Devices of IDs

Device No. of logging/tracing IDs (0 to 9)					Attribute R: Read only Blank: R/W	Content
ID0	ID1	ID2	…	ID9		
CR000	CR100	CR200	…	CR900		Execute logging/tracing
CR001	CR101	CR201	…	CR901	R	Logging/tracing operation in progress
CR002	CR102	CR202	…	CR902	R	Logging/tracing file write end Turns off automatically when the LOGE instruction is executed.
CR003	CR103	CR203	…	CR903	R	Alarm for 50% logging buffer or less
CR004	CR104	CR204	…	CR904	R	Logging buffer overflow
CR005	CR105	CR205	…	CR905	R	RUN writing during logging
CR008	CR108	CR208	…	CR908	R	Logging/tracing error
CR009	CR109	CR209	…	CR909	R	Insufficient memory capacity error
CR010	CR110	CR210	…	CR910	R	Device/trigger setting abnormal
CR011	CR111	CR211	…	CR911	R	Monitoring of tracing file writing trigger
CR012	CR112	CR212	…	CR912	R	Tracing data capture complete
CM1630	CM1631	CM1632	…	CM1639	R	Minimum buffer free space (in KB units)*1
CM1680	CM1681	CM1682	…	CM1689	R/W	File save counter (memory card only)
CM1500	CM1501	CM1502	…	CM1509	R	File save counter (shared)
CM1660	CM1661	CM1662	…	CM1669	R	Buffer free space (in KB) *2

*1 Displays minimum remaining space of the buffer after executing the LOGE instruction. It will be reset when the next LOGE instruction is executed.

*2 Displays the remaining space of current buffer.

■ Common devices

Device No.	Attribute R: Read only Blank: R/W	Content
CR3210	R	Optional logging trace
CR3513	R	Internal buffer overflow relay
CR5400		Memory card mount request
CR5401		Memory card unmount request
CR5402	R	Memory card is being accessed
CR5403	R	Memory card identification complete
CR5404	R	Memory card is available
CR5405	R	Memory card is write protected
CR5408		CPU memory mount request
CR5409		CPU memory unmount request
CR5410	R	CPU memory is being accessed
CR5411	R	CPU memory identification complete
CM2391	R	Power OFF during storage access

■ Details of devices

● Logging/tracing operation relay

To enable and inhibited logging/tracing. It is set when logging/tracing instruction (LOGE) is performed, or logging/tracing is started through KV STUDIO, generally no direct operation is required.
When this relay is turned ON, all status relays with the same logging/trace ID will temporarily go OFF.
It will be reset when an error occurs.

● Logging/tracing in progress relay

In fact, it's ON when logging/tracing is run.

● Logging/tracing file write complete relay

When logging is used: the relay is ON when data larger than upper limit of the specified file capacity are saved in every file.
When tracing is used: the relay is ON when data have been written into file or internal memory
When the relay is ON, logging/tracing will be stopped; the logging/tracing operation relay will be OFF.
When LOGE is executed in the next time, the logging/tracing file write end relay is OFF.

"Troubleshooting for overflow of file capacity settings" (page 5-85)

● Alarm relay when the logging buffer is below 50%

Since the triggering frequency is higher than writing speed, the relay is ON when the residual capacity of buffer is below 50%. When data larger than upper limit of the specified file capacity are stored, the relay is ON.

● Buffer overflow relay

This relay shows that the logging/tracing buffer has overflowed as the data could not be duly written into the memory card or CPU memory.
When the relay is in ON state, data abnormity occurs. Observe the data No. to know when data abnormity occurs.
Even if overflow occurs, data will continue to be written into the memory card.
When overflow occurs, the following measures can be taken to alleviate overflow.

- Reduce the frequency of access to the memory card or CPU memory
- Reduce data size and frequency trigger
- Prolong the scanning time.

● Store in RUN during logging/tracing relay

It will be ON when logging/tracing is in progress, and writing is performed in RUN.
When writing is performed in RUN, logging/tracing continues but buffer overflow occurs more easily as writing to the memory card or CPU memory is slower.

● Logging/tracing error relay

When the logging/tracing stopped because of the memory card error, it turns ON and logging/tracing ends. Please check if various error relays were not turned on. In this case, the residual data in the buffer will disappear. When LOGE instruction is executed the next time error will be cleared.

● No memory space error relay

Turns ON when there is not sufficient space remaining in the memory card or CPU memory.
When the relay is ON, logging/tracing will be completed. When logging/tracing failure relay is ON, the logging/tracing relay will be OFF. At this time, data remaining in the buffer disappears and no logging file is created. Error will be cleared by the next "LOGE instruction".

● Device/trigger setting abnormal relay

In the case of abnormal device setting and trigger setting for logging/tracing, it's ON.
When the relay is ON, logging/tracing will be completed. When logging/tracing error relay is ON, the logging/tracing relay will be OFF.

● Tracing file storing trigger monitor relay

When tracing function is used, it's in ON state after file storing is triggered.
It turns OFF when the LOGE instruction is executed in the next time.

● Tracing data capture complete relay

When tracing function is used, and after file writing is triggered, it's in ON state when the data size set correspondingly has captured.
It turns OFF when the LOGE instruction is executed in the next time.

● Buffer free space (in KB)

The remaining space of the current buffer is stored in KB. The buffer capacity of KV-8000 series is 10240KB.

When logging is used: you can control the buffer's free space. In case the value becomes smaller, the possibility of buffer overflow increases.

When tracing is used: the buffer's overall capacity is shown before file storing is triggered. After file writing is triggered, the captured residual data size will be displayed after tracing is finished.

● File storing counter

Increase 1 each time file storing is performed.
Counting is performed within the range of 0 to 65535, and changes to 0 after 65535. It will be saved even when the power is turned off. It does not automatically change to 0 when the LOGE instruction is executed.
Please save 0 when the file save counter is reset.

● Optional Logging Trace Operating Relay

It turns ON when either 1 or more of logging trace of logging trace ID from 0 to 9 is operating towards the memory card.
When using other functions, it is used to confirm if logging trace to the memory card is operating or not.

● Memory card is in use relay

When memory card is used, it's ON. This relay is ON when memory card is used independent of logging/tracing functions.

● Memory card identification complete relay

When memory card is inserted, slot cover of memory card closed, initialization finished, and ready for the relay will be ON when it's ready.

● With memory card relay

When memory card is inserted and slot cover of memory card is closed, it will be ON.

● Memory Card write protect relay

It will be ON when memory card is inserted and write protected.

● Power OFF during storage access

It becomes an error when the power is turned OFF during storage access, and value other than 0 will be stored.
The value will not be cleared automatically.

● Internal buffer overflow relay

It turns ON when the buffer range used inside overflows. It turns ON when large quantity of device is collected at high frequency using logging trace real time chart monitor function.
It can be reduced by decreasing the number of collected device and sampling frequency.

Logging/Tracing Function Precautions

The following describes the precautions on using the logging function.

"Used on the CPU unit" (page 5-8) and "Precautions When Using a memory card" (page 5-9).

● Storing speed

- The maximum scan time during logging/trace operation is extended up to (20 μs × number of used IDs) ms.
- When multiple triggers are simultaneously generated and the number of access instances to memory card or CPU memory increases, scan times increase and buffer overflows occur more often.

Reference

The data failure can be improved by setting of certain scanning time or END processing time.

"Fixed scan time operation" (page 4-90)
"Setting the END processing time" (page 4-92)

● Priority order of accessing storage devices

- During logging/tracing the priority order of accessing storage devices is shown below.
  Storage instruction > logging/trace instruction > FTP server function, FTP client function, file operations in the access window, access from the KV-XD02
  For this reason, if a great number of logging/trace triggers occur, other functions will not be able to use storage devices.

● Power OFF, memory card insertion/removal

Before removing the memory card, make sure that the power of the main unit turned ON, the memory card unmount request relay turns ON, and the memory card recognition completion relay turns off. Data is not saved during logging/tracing if the Memory Card is not removed by this procedure.

Handling of the Memory Card slot cover

- Logging/tracing will not be started if the Memory Card slot cover is not closed.
- When open the slot cover of memory card, the current processing will be implemented, but the following processing will not be implemented.

Precautions during power off and card insertion/removal

- Never remove the Memory Card or turn the CPU unit OFF while the Memory Card access LED on the PLC is lit (indicating that data is being read from or written to the Memory Card). Doing so might corrupt the data on the Memory Card.
- When power is turned OFF while logging/tracing is being executed, data will not be saved to the Memory Card.
- Wait for the CPU memory unmount request relay to go ON and the CPU memory verification completion relay to go off, before turning off the power. Unmounting is to make the memory card safely removable.
- If RAM mode is used, turn on the CPU memory save request relay to save the necessary data. Then confirm that the CPU memory save completed relay goes on before turning off the power.

Replacing Memory Card

- Replace Memory Card by the following procedure.
  "Used on the CPU unit" (page 5-8)
- If auto-restart is logging/tracing by Memory Card insertion is set in the logging settings of KVSTUDIO, when Memory Card is removed during execution of logging, the logging data is stored in buffer until the Memory Card is next inserted. When the Memory Card is inserted, data stored in memory will be written to the Memory Card. At this time, the logging error relay will not be ON. If there is a long period between the Memory Card is inserted next time or many triggers have been generated, a buffer overflow will occurs and some data may be lost.
  When "auto-restart logging/tracing by Memory Card insertion is not set, never remove the Memory Card while logging is being executed. Doing so might corrupt the data on the Memory Card.

● Operation during rewriting in RUN mode

- Logging execution slows down very much when rewriting is performed in RUN mode. This makes it more likely for buffer overflow to occur.
- Logging/Tracing settings cannot be written in the runtime (RUM). The settings will be applied when Logging/Tracing is executed in the next time.

● Operation when switching between RUN and PRG mode

- The logging function is active only in the RUN mode. When RUN → PRG switching is performed, all logging/tracing operations will stop.
  Like when the LOGD instruction is executed, the logging/trace data up to this moment is output to memory card (or CPU memory).
  Also, when PRG → RUN switching is performed, the LOGE instruction must be executed to start logging/tracing, as all logging/tracing operations are in a stopped state.

● Processing at start of logging/tracing

If a specified file No. already exists when logging/tracing is enabled after power-on, the following action will be executed.

When specifying method of file No. is "Auto numbering."

- For operations when no No. is available in "Creating a New File", please refer to Troubleshoot: Overflow of File Capacity Settings
- Execute the Add To Latest Files and the latest file exceeds file capacity limit:
  Use the file with the next available No. In case no No. is available, please follow the instruction in the Troubleshoot: Overflow of File Capacity Settings.
  If no file capacity limit is defined, the new file can be added.
  For Troubleshoot: Overflow of File Capacity Settings, please see it on the next page.

When the specifying method of file No. is "Fixed/Dev val"

- When same file exists and "Overwrite old file" is selected: old file will be deleted and file will be saved by the same file name.
- When same file exists, and "Add" is selected:
  Add data in the specified file. When the file exceeds file capacity limit, if logging/tracing is started, the logging/tracing error relay turns on, the logging/tracing is then stopped.

"Device List for Logging/Tracing" (page 5-79)

● Troubleshooting for overflow of file capacity settings

The operation when the file capacity is exceeded may vary with specifying methods of file No.

About file capacity upper limit setting, see "Option" (page 5-67)

When specifying method of file No. is "Auto numbering."

If there is remaining space in the file number, creates a new file with the next file number.

When all file No.s are used and the last file capacity is exceeded.

- When there are no remaining No.s, and "Overwrite old file" is selected:
  Oldest data are deleted, and new file will be created according to same file NO.
- When there are no remaining No.s, and "stop logging/tracing" is selected:
  When logging/tracing starts, the logging/tracing error relay turns ON and logging/tracing stops.

When specifying method of file No. is "Fixed/Dev val"

When the same file exists whether "Overwrite old file"/"Add" is selected, logging/tracing will be stopped.

Reference

When "logging / tracing transfer setting" is set with the FTP client function, even if the LOGD instruction has not been executed, files generated by "file capacity upper limit setting over" will be uploaded to FTP every time.

● Character restriction

(1) Character string in CSV file: file comment, device comment

(2) File name character string : file name character string

(3) ASCII character string : there are restrictions on characters that can be used when "ASCII character string" is selected as the device's data format.

Item	Explanation
Character string in CSV files	• "," cannot be used.
Character string in file names	The following characters cannot be used. • 12 half-width characters * / < > ? \ | % , : ; " • 7 full-width characters ／ ： ？ ￥ ＊ ［ ］ • NEC special characters (8740H to 879CH) • IBM extended characters for NEC (ED40H to EEFCH) • IBM extended characters for IBM (FA40H to FC4BH)
ASCII string	All ASCII codes are saved directly. Control characters and " etc. are also saved directly.

● Upper limit of file capacity

The upper limit per file is 256 MB.

● Upper limit of storage capacity

When remaining capacity drops below 10 KB, the insufficient memory capacity error may go on.

Overview of operation recorder

The operation recorder function can be used to save the PLC operation records before and after any timing (trigger). The data that can be saved in the operation recorder is as follows.

Type	Details
Device	Data for each scan time in the PLC devices, variables and buffer memory of the expansion unit before and after the trigger condition is met
Camera	Video data shot with KV-CA02 by different frame rates before and after the trigger condition is met
Event / error	Event log of errors or device value changes, etc. before and after the trigger condition is met
Project data	Project data when the trigger condition is met

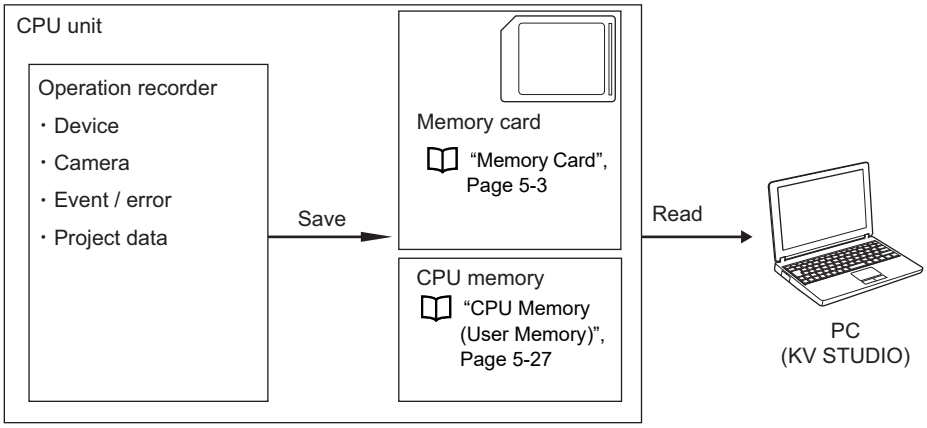
Point **The operation recorder function is a function that can be used only by KV-8000.**

The saved operation records can be played back with KV STUDIO or KV REPLAY VIEWER. The ON/OFF status of the bit device before and after a fault occurs, the value of the word device and the video shot with the camera*1 can be checked with the ladder program, real-time chart monitor and Replay VT Viewer*2. Since the time series data can be checked while playing, stopping, rewinding and frame advancing, etc., it can be used to analyze the fault causes difficult to check after a sensor chatters or a chucking error occurs due to incomplete stop before chucking, etc.

*1 The small standard camera KV-CA1H or the wide-angle, high-resolution camera KV-CA1W and the camera input unit KV-CA02 are required.

*2 This is a function to simulate the device data of operation recorder in the PC using the screen data of the human machine interface. VT STUDIO Ver.7.10 or later is required.

Operation records can be saved in the memory card inserted to the CPU unit or CPU memory, and the contents can be checked by reading them via the CPU unit or opening files copied to the PC, etc.



Point
- **The saved operation records can be checked via the replay mode of KV STUDIO or KV REPLAY VIEWER. Refer to "KV STUDIO User's Manual" for details on the replay mode.**
- **The video data of the camera is recorded in the camera input unit KV-CA02, and saved in the CPU unit when the trigger condition is met. Refer to "KV-CA02 User's Manual" for details.**
- **The data application unit KV-XD02 can be used to analyze operation records and display the devices and variables that were the likely causes of problems. Refer to "KV-XD02 User's Manual" for details.**

Operation steps

The steps for using the operation recorder function are as follows.

① Setting saved data and trigger

Operation Record / Replay (R) - Operation Record Setting (P)

Set the type of data to be saved as operation record, save condition and period. The data to be saved is automatically set from the unit settings or the contents of the ladder program, and can be added and deleted manually.

“Settings”, page 5-88

② Set the camera input unit

Camera setting

Setting is required only when recording the camera video.

Refer to "KV-CA02 User's Manual" for details.

③ Operation record collection and save

Transfer the setting to the PLC. When the set file save trigger conditions are met, the operation recorder is saved in the CPU unit.

④ Replay of operation record

The saved operation recorder is played back via the replay mode of KV STUDIO or KV REPLAY VIEWER. Refer to "KV STUDIO User's Manual" for how to use the replay mode.

Settings

To use the operation recorder function, set the following in KV STUDIO.

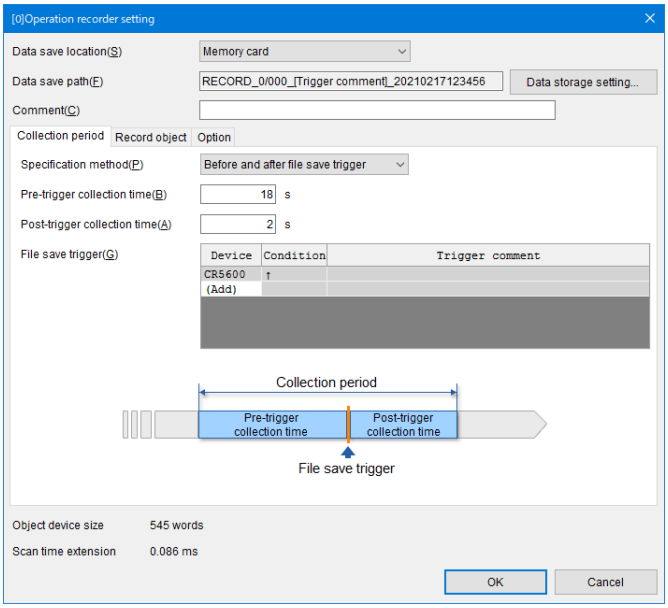
- Data save location
- Data acquisition method
- File save trigger
- Data collection time
- Data collection start timing
- Recording object

Up to 4 operation recorder settings can be created (ID0~ID3). Set the operation recorder for each ID.

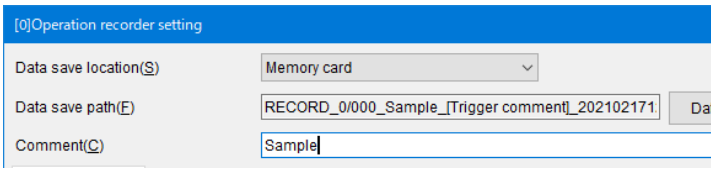
1 **Click "Operation Record / Replay (R)" ▶ "Operation Record Setting (P)" ▶ "New (N)" from the menu.**

The "Operation Record Setting" dialog is displayed.

Other •Double-click "Operation Record Setting" in the workspace



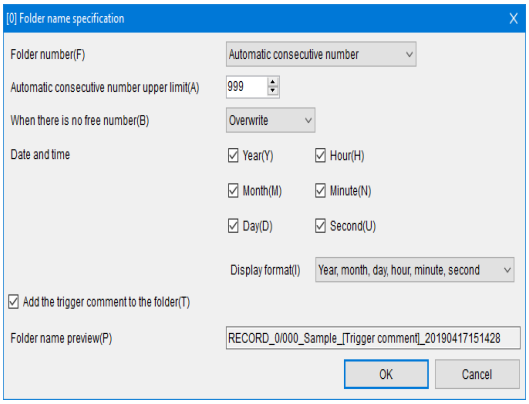
2 Perform settings related to the data save location or folder name.



Item	Description
Data save location	Select from "Memory Card / CPU Memory".
Data save path	Set the data save path. Click Data storage setting... to display the "Specify Folder Name" dialog.
Comment	Set the comment to be inserted into the saved file name.

Point

Some characters cannot be used in comments. Refer to "Characters that cannot be used in the data save path", Page 5-108 for characters that cannot be used.



Item			Description
Folder number	Automatic consecutive number	Automatic consecutive number upper limit	Use the minimum folder number from the unused number between 0 and the value set for the automatic consecutive number upper limit.
		When there is no free number	Select the processing when there is no free number from "Overwrite / Stop".
	Fixed	No.	Specify a fixed value between 0 and 999.
	Device value	Device	Use the value of the device or UINT/UDINT/TIMER/COUNTER type variable (0~999).
Date and time			If checked, the file name contains the date and time information when saving the operation record. Since the date and time information is fixed to the date and time information at the time of file creation, the file name is not changed when adding a record. When an existing file is overwritten, the old file is deleted and a file will be created with a new file name. Year (2000~2099) Month (01~12) Day (01~31) Hour (00~23) Minute (00~59) Second (00~59)
Add the trigger comment to the folder			If checked, the trigger comment is added to the folder name.
Folder name preview			An example of the folder name is displayed.

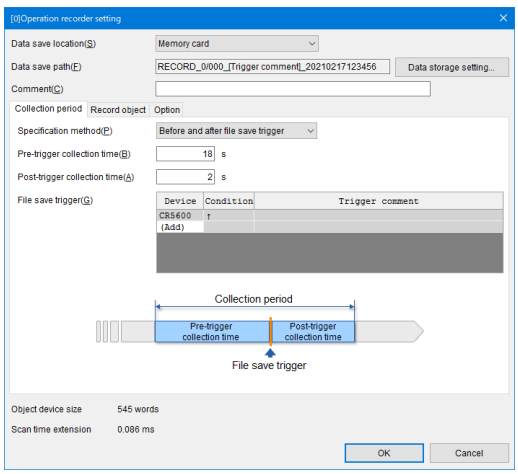
Point

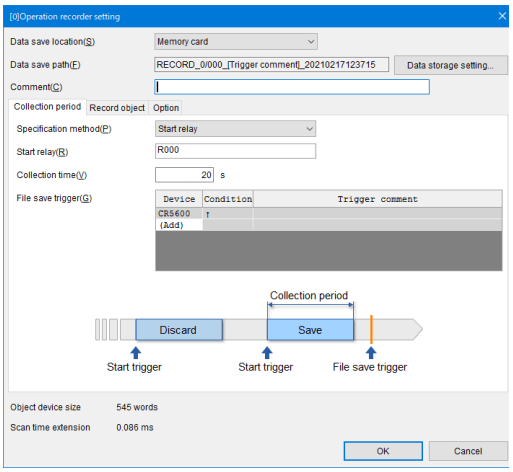
If the free space on the memory card becomes low before the automatic serial No. upper limit reached, additional data cannot be saved. In that case, you can continue to save by reducing the automatic serial No. upper limit and setting to "Overwrite" when there is no more free number (please note that the oldest file will be deleted first).

Collection period

Set the timing or time to collect data.

The setting items are partially different depending on whether "Before and after file save trigger" or "Start relay" is selected for "Specification method".





Item		Description
Specification method		Select from "Before and after file save trigger / Start relay".
When "Specification method" is "Before and after file save trigger"	Collection time before trigger	Set the time to save the data before trigger.
	Collection time after trigger	Set the time to save the data after trigger.
When "Specification method" is "Start relay"	Start relay	Set a relay to start collecting data.
	Collection time	Set the time to collect data.
File save trigger	Device	Specify the bit device or BOOL/TIMER/COUNTER type variable as a trigger to save the collected data.
	Conditions	Select from " ↑ " (rising edge) or " ↓ " (falling edge).
	Trigger comment	Set the file save trigger comment. The set comment can be added to the saved folder name.

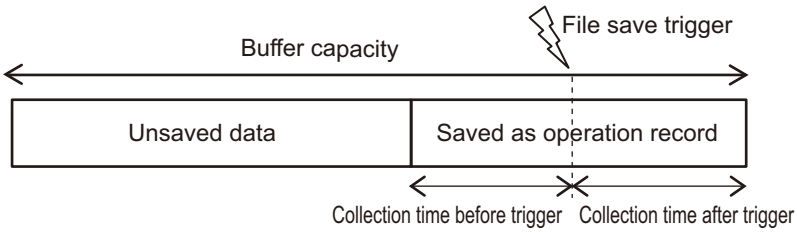
Point

Some characters cannot be used in trigger comments. Refer to “Characters that cannot be used in the data save path”, Page 5-108 for characters that cannot be used.

■ Operation of "Before and after file save trigger" and "Start relay"

● When "Before and after file save trigger" is selected

Data for the buffer capacity assigned to each ID in the operation recorder setting is always collected. When the file save trigger condition is met and the collection period after trigger elapses, the data collection is stopped, and the data for the collection time before trigger and the collection time after trigger is saved as operation records. The data prior to the collection time before trigger is not saved. This is selected when checking the operation recorder before and after the rising edge of the error relay, etc.

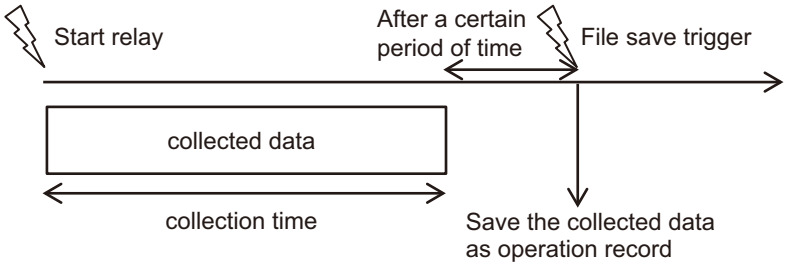


Point

- **When the condition for file save trigger is met without collecting data for the time specified in the collection time before trigger, only the data collected successfully is saved as the operation record.**
- **When the data for the buffer capacity is collected, the data saved at the start is overwritten in order.**

● When "Start relay" is selected

When the device specified by the start relay becomes ON, the data collection starts, and stops after the set collection time. When the completion signal or the fact that the time until the next start relay is ON exceeds a certain time is set
as the file save trigger condition, the data in the collection time is saved as the operation recorder after the file save trigger condition is met.
This is selected when it cannot be detected as an error relay, such as when the equipment stops without response.



Reference

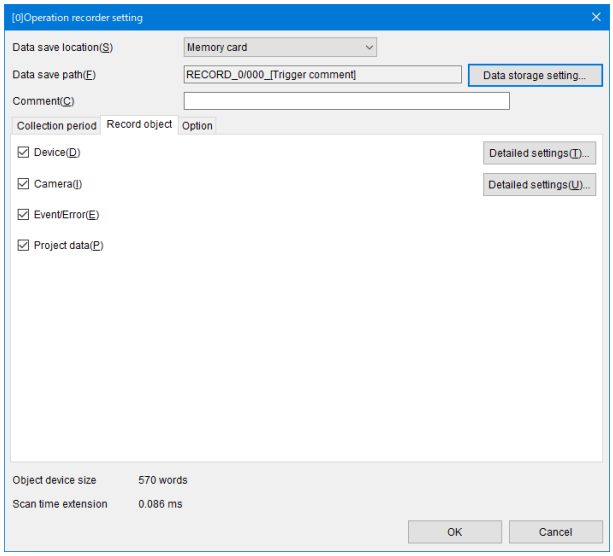
All device data for the buffer capacity can be saved when the file save trigger condition is met while the operation recorder buffer area full save mode (ID0: CR5614) is ON, so the device data outside the collection period can also be checked.

Point

- **If the collected data exists when the start relay is ON, the collected data is discarded and then the data collection starts.**
- **If the file save trigger condition is met without the collected data present, no operation recorder is saved.**
- **If the file save trigger condition is met before the collection time has elapsed after the start relay is ON, the data for the collection time will be collected and then the operation recorder will be saved.**

Recording object

Set the object of data to be saved as operation record.



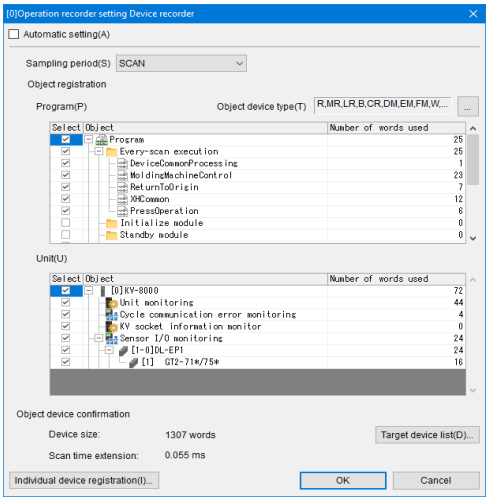
Item	Description
Device	Saves PLC devices and variables. The devices used in the unit configurations or projects are registered in batch automatically in the initial value state. To register the devices manually, click "Detailed Settings (T)".
Camera	Save the video data shot with the camera input unit KV-CA02. The video data of all connected cameras is saved for the collection time in the initial value state. To change the saved camera or collection time, click "Detailed Settings (U)".
Event / error	Save the log of events, errors, etc. occurred in the PLC. The log that can be saved is as follows. • Error log (Serious error / minor error) • Power ON/OFF log • PLC event log • Device value change log
Project data	Save the project data when the file save trigger occurs. The saved project data is used as a project when the ladder is displayed in the replay mode of KV STUDIO or via KV REPLAY VIEWER.

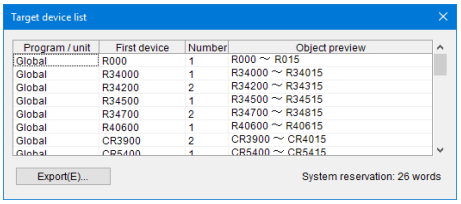
Point

- The scan time will be extended when the recording object contains a device. The time reference value for scan time extension is displayed on the lower left of the "Operation Record Setting" dialog. To suppress the scan time extension, please set manually and reduce the number of registrations.
- Scan time is extended only during data collection. The object when "Specification method" is "Before and after file save trigger" is other than the period during file saving; the object when "Specification method" is "Start relay" is from the start relay ON until the collection time elapses. The scan time extension does not exist at other timings.
- To keep the scan time constant without changing the number of devices registered, use the constant scan time operation function.
  “Fixed scan time operation”, Page 4-90

■ Device details

Click "Detailed Settings (T)" to display the following dialog. Click "Details (D)" to display a list of selected device and variable.





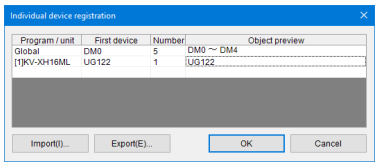
Item	Description	
Automatic setting	When unchecked, the device and variable to be collected can be set manually for each function or module of the unit. The device and variable used in the unit settings or ladder programs are registered in batch in the initial value state*1.	
Sampling period	Select from "SCAN / Fixed-period / Trigger Instruction (TRGR)".	
	SCAN	Data is collected for each scan at the timing of END processing.
	Fixed-period	Data will be collected in the initial END processing after the period set in "Period" displayed on the right when selected.
	Trigger instruction	Data is collected at the execution timing of the TRGR instruction.

*1 Device and variable used by instructions with an unfixed scope or device and variable specified by index modification or indirect specification cannot be registered automatically. The data of these device and variable is registered manually when recording.
Example: When using "FMOV #0 DM0 EM0", the range of the target device and variable changes depending on the value of EM0, so it cannot be registered automatically.

Reference

The maximum collection period changes depending on the number of data saved. The maximum collection period can be checked from the operation recorder monitor when KV STUDIO is in the monitor mode. Check the maximum collection period, and adjust it as necessary by increasing the device recorder buffer capacity or reducing the number of device and variable saved, etc.
“Device recorder buffer capacity setting”, Page 5-97

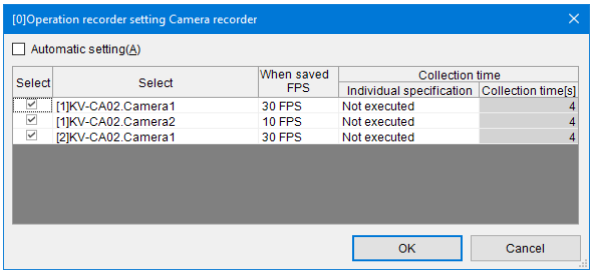
To manually set the device and variable saved, uncheck "Automatic Setting(A)" and then click "Individual Device Registration(I)".



Item	Description
Program / unit	Select the type of device and variable to be saved from the drop-down list.
Leading device	Specify the leading device and variable of the device to be saved.
Count	Specify the continuous count when saving multiple device and variable in succession.
Target preview	Display the result set in "Leading device" and "Count".
Import	Import the setting information of the saved device and variable, and overwrite the current settings.
Export	Export the setting information of the saved device and variable.

■ Camera details

Click "Detailed Settings(U)" to display the following dialog. Uncheck "Automatic Setting(A)" during manual setting.



Item		Description
Select		Check to save the shooting data of the target camera.
FPS when saving		Set the FPS when saving the camera shooting data.
Collection time	Individual specification	Select from "Not Executed / Executed". When "Executed" is selected, "Collection Time" can be set manually. When "Not Executed" is selected, the video within the collection time of operation recorder will be collected.
	Collection time	This can be set only when "Individual Specification" is set to "Executed". In addition to the collection time of the operation recorder setting, the collection time of the camera shooting data can be set.

■ Event / error details

The event or error log of the PLC that can be saved using the operation recorder function is described.

● Error log

Save the log of errors occurring in the PLC. Up to 32 severe errors and 200 minor errors can be saved.

● Power ON/OFF log

Save the power ON/OFF log of the PLC. Up to 20 power ON records and power OFF records can be saved.

● PLC event log

Save the log of the following events that occurred in the PLC.

Event	Target unit	Content
Operation mode switching	CPU unit	The operation mode (PROG⇔RUN) has been switched.
Project change	CPU unit	The project has been changed by transferring the project from KV STUDIO or loading the project from the memory card or CPU memory.
Operation record save trigger occurred	CPU unit	Operation record save trigger has occurred.
Operation record save completed	CPU unit	Saving of operation recorder succeeded or failed.
Memory card	CPU unit	Memory card slot cover has been opened or closed. Memory card has been mounted or unmounted.
Calendar timer setting change	CPU unit	Calendar timer settings have been changed.
Alarm	CPU unit	Alarm relay status has changed.
PLC clear	CPU unit	Various logs of the PLC have been cleared.
Event buffer overflow	CPU unit	The log recording cannot catch up as the events have occurred continuously.
Ethernet communication	CPU unit KV-XD02 KV-XLE02 KV-XCM02	The link status of Ethernet (Normal ↔ Abnormal) has changed.
Serial communication error	KV-XL202 KV-XL402	A parity error, structure error or overflow error has occurred during serial communication.
Serial communication error	KV-XL202 KV-XL402	Checksum error has occurred during serial communication.
Application event	KV-XD02	A warning or alarm has occurred in the symptom observation application.
OPC UA server restart	CPU unit	The OPC UA server has restarted.

● Device value change

Save the log of device values changed from external equipment, such as human machine interface, PLC and PC.

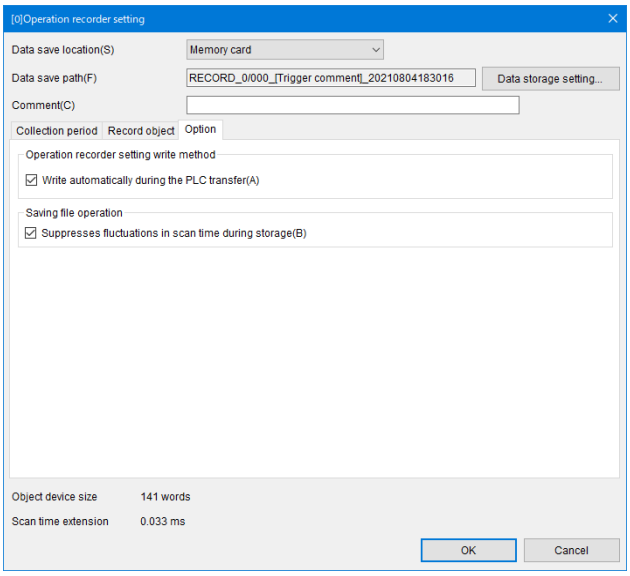
Target unit	Content
VT5/VT3	Change of device value from VT5/VT3 connected to the CPU unit or expansion unit
Access window	Change the device value by operating the access window of CPU unit
Modbus server	Change the device value from an external equipment connected using the Modbus server function of KV-XLE02/XCM02/XL202/XL402
KV host link	Change the device value from an external equipment connected using the KV host link function of KV-8000(A) or KV-XLE02/XCM02/XL202/XL402
OPC UA server	Change the device value from an external equipment connected using the OPC UA server function of the KV-8000(A)

■ Project data

Set whether to save the project data transferred to the PLC. When checked, the project data will be saved as operation record. To reduce the capacity of the operation record, uncheck it. When unchecked, it is necessary to prepare the project data when playing in the replay mode.

Options

Set the options.



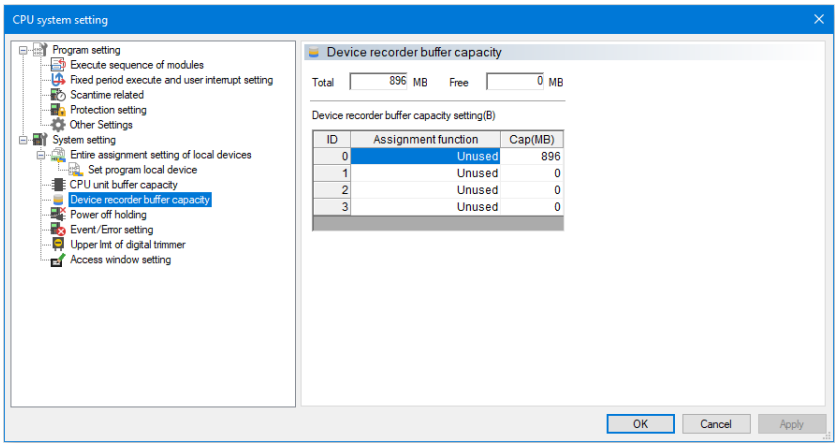
Item	Description
Write automatically at the time of PLC transfer	When checked, the operation recorder settings are written automatically at the time of project transfer. If you uncheck it, the information of Device/Variable to be assigned will not be updated, and you can prevent overwriting of the operation record setting when designing a project with multiple people.
Suppresses fluctuations in scan time during storage	When checked, the scan time is kept at as close to a constant speed as possible during operation record saving by suppressing increases in the speed.

Device recorder buffer capacity setting

Set the buffer capacity used to record the device value when using the operation recorder function.

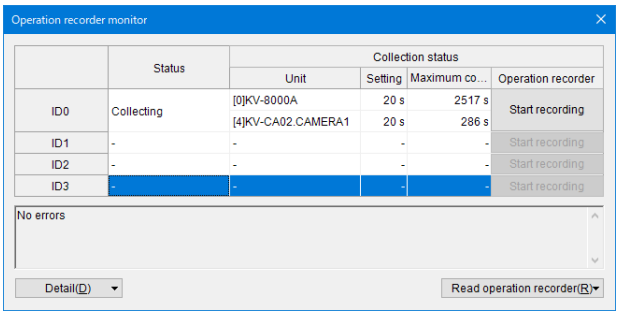
1 **Click "Display (V)" ▶ "CPU System Settings (P)" from the menu, and select "Device recorder buffer capacity setting".**

Other •Double-click "CPU System Settings" in the workspace, and select "Device recorder buffer capacity setting"



2 **Set the buffer capacity to be used in the operation recorder settings of ID0~ID3. After the buffer capacity of the ID used is increased, the amount of data that can be saved (data collection time) will also increase, and the total buffer capacity should not exceed 896MB during setting.**

Reference When KV STUDIO is in the monitor mode, the maximum collection period (in seconds) of the device can be checked with the operation recorder monitor that can be displayed from "Operation Record / Replay (R)" ▶ "Operation Record Monitor (M)" in the menu.
When the maximum collection period is shorter than the set value, make fine adjustments as necessary, such as increasing the buffer capacity and reducing the number of collected devices.



Device list for operation recorder function

The execution information and error information of the operation recorder function are assigned to the control relay (CR) and control memory (CM). The control relay can be divided into the control relay provided for each ID, and 1 control relay for all IDs.

■ Device of each ID

Device numbers of different IDs				Property	Device name
ID0	ID1	ID2	ID3	R: Read only Blank: Read / write	
CR5600	CR5700	CR5800	CR5900		Operation record save request
CR5601	CR5701	CR5801	CR5901	R	Operation recording
CR5602	CR5702	CR5802	CR5902	R	Operation record save trigger monitor
CR5603	CR5703	CR5803	CR5903	R	Operation record saving
CR5604	CR5704	CR5804	CR5904	R	Operation record collecting
CR5605	CR5705	CR5805	CR5905	R	Operation record setting error
CR5607	CR5707	CR5807	CR5907	R	Operation record save completed
CR5608	CR5708	CR5808	CR5908	R	Operation record save error
CR5609	CR5709	CR5809	CR5909	R	Device recorder buffer shortage
CR5614	CR5714	CR5814	CR5914		operation recorder buffer area full save mode
CM2000~ CM2001	CM2010~ CM2011	CM2020~ CM2021	CM2030~ CM2031	R	Device record average sampling period
CM2002~ CM2003	CM2012~ CM2013	CM2022~ CM2023	CM2032~ CM2033	R	Device record average change device number
CM2004~ CM2005	CM2014~ CM2015	CM2024~ CM2025	CM2034~ CM2035	R	Device record maximum collection period
CM6000	CM6200	CM6400	CM6600	R	Operation record save completion code
CM6002	CM6202	CM6402	CM6602		Operation record save counter
CM6004	CM6204	CM6404	CM6604	R	Operation record save progress detail 1
CM6005	CM6205	CM6405	CM6605	R	Operation record save progress detail 2
CM6006	CM6206	CM6406	CM6606	R	Operation record save progress
CM6010	CM6210	CM6410	CM6610	R	Operation record save path
:	:	:	:		
CM6171	CM6371	CM6571	CM6771		

■ Device common to IDs

Device number	Property R: Read only Blank: Read / write	Device name
CR5402	R	Memory card in use
CR5403	R	Memory card recognition completed
CR5404	R	Memory card available
CR5405	R	Memory card write protection
CR5410	R	CPU memory in use
CR5411	R	CPU memory recognition completed

■ Details of each device

● Operation record save request relay

When the operation recorder save request relay is turned from OFF→ON, the data collected by the CPU unit is saved as an operation record. Even if the file save trigger is not set by the operation recorder function, the operation recorder can be saved by turning the operation recorder save request relay from OFF→ON. Use the SET instruction when executing from a ladder program.

Point

- **When a BOOL array type variable is assigned to CR, the element number of the array should be specified with a value.**
  When specified with device or variable, the SET instruction cannot operate properly.

● Operation recording relay

ON while the operation recorder function operates.

● Operation record save trigger monitor relay

ON when the file save trigger occurs during the operation recording. OFF when saving of the operation recorder is completed or fails.

● Operation record saving relay

ON when the operation recorder collection is completed and saving is started. OFF when saving of the operation recorder is completed or fails.

● Operation record collecting relay

ON while collecting the operation record.

● Operation record setting error relay

ON when the set value of the operation recorder function is out of the specified range.

● Operation record save completion relay

ON when saving of the operation recorder is completed. OFF when the next file save trigger is ON. During abnormal analysis with the KV-XD02, this relay turns ON when this analysis is completed after the saving of the operation record is completed.

● Operation record save error relay

ON when saving of the operation recorder fails. OFF when the next file save trigger is ON.

● Device recorder buffer shortage relay

ON when the buffer capacity is insufficient and the data cannot be collected for the set collection time.

● operation recorder buffer area full save mode relay

When the operation recorder is saved with this relay turned ON, not only the set collection time, but also all data present in the buffer will be saved.

● Device record average sampling period

The average of sampling periods (in μs) of the device record is stored.

● Device record average change device number

The average of the number of devices (words) with the state changed is stored.

● Device record maximum collection period

The period (in seconds) of collection under the current setting conditions is stored. When the status of the device to be collected changes,
the period of collection becomes short, so the value change may occur during monitoring even under the same setting conditions.

● Operation record save completion code

A value is stored after the operation recorder is saved. 0 is stored when saving is completed, and the completion code related to the cause for failure is stored when saving fails. Refer to "Operation record save completion code", Page 5-103 for the content of the completion code.

● Operation record save counter

1 is added after an operation recorder is saved. The addition is performed in the range of 0~65535, and the next one of 65535 is 0. The value is maintained even when the power is turned off. 0 is stored when resetting.

● Operation record save progress detail 1

The file type during saving is stored. The relationship between the number and the file type is as follows.

No.	File type	Content
0	No	Not saving.
1	CPU log	Event / error log of the CPU.
2	Ladder project	Project executed by the CPU unit when saving.
3	Device record	Device data set during the operation recorder setting.
4	Camera record	Video data shot by KV-CA02.
5	Savelog.txt	Text file that describes the trigger type, time, folder creation date and time, folder name and data size. Savelog.txt is not saved when saving of the operation recorder fails.

● Operation record save progress detail 2

The progress of the operation recorder save is stored in the range of 0 to 100 for different file types.

● Operation record save progress

The progress of the operation recorder save is stored in the range of 0 to 100.

● Operation record save path

The path to save the operation recorder is stored.

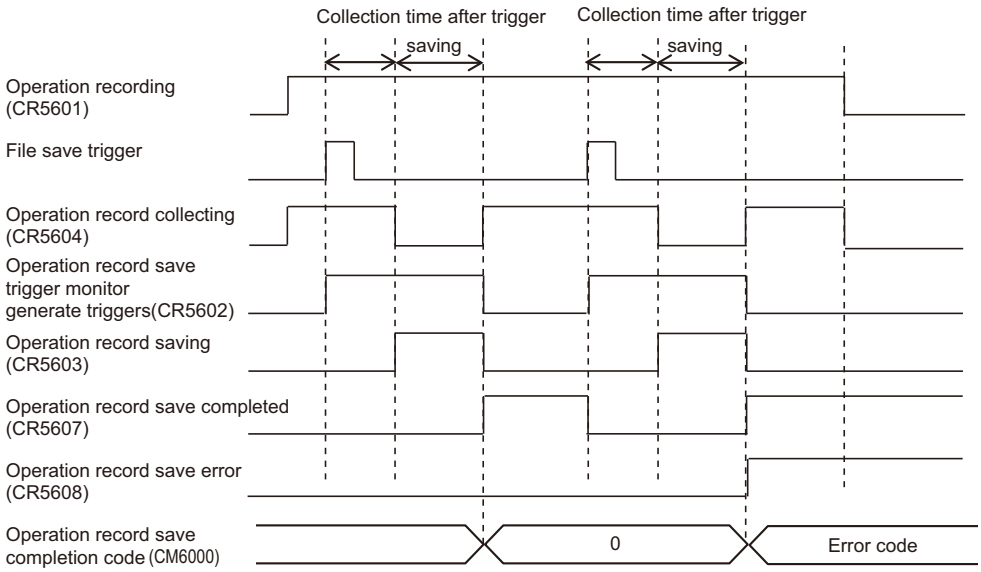
Point

Refer to "Device List for Logging/Tracing", page 5-79 for details on the devices common for IDs such as memory card in use.

Device operation when saving the operation record

The device to be used or the ladder program to be created differs depending on whether "Before and after file save trigger" or "Start relay" is selected as the specification method of the collection period.

● When "Before and after file save trigger" is selected



① When the operation recorder function is in operation, the operation recording relay (CR5601) and operation recorder collecting (CR5604) become ON, and the CPU unit starts data collection.

② When the condition for file save trigger is met, the operation recorder save trigger monitor (CR5602) becomes ON, and the data is collected until the collection time after trigger elapses.

③ When the collection time after trigger has elapsed, the CPU unit starts saving the operation record, and the operation recorder saving relay (CR5603) becomes ON. While the operation recorder saving relay is ON, the operation recorder collecting relay is OFF, and the CPU unit stops the data collection.

④ When the saving of the operation recorder is completed, the operation recorder save trigger monitor relay and the operation recorder saving relay become OFF, and the operation recorder save completion relay (CR5607) becomes ON. 0 is stored in the operation recorder save completion code (CM6000).

⑤ When the file saving fails, the operation recorder save trigger monitor occurrence relay and the operation recorder saving relay become OFF, and the operation recorder save completion relay and the operation recorder save error relay (CR5608) become ON. The error code is stored in the operation recorder save completion code.

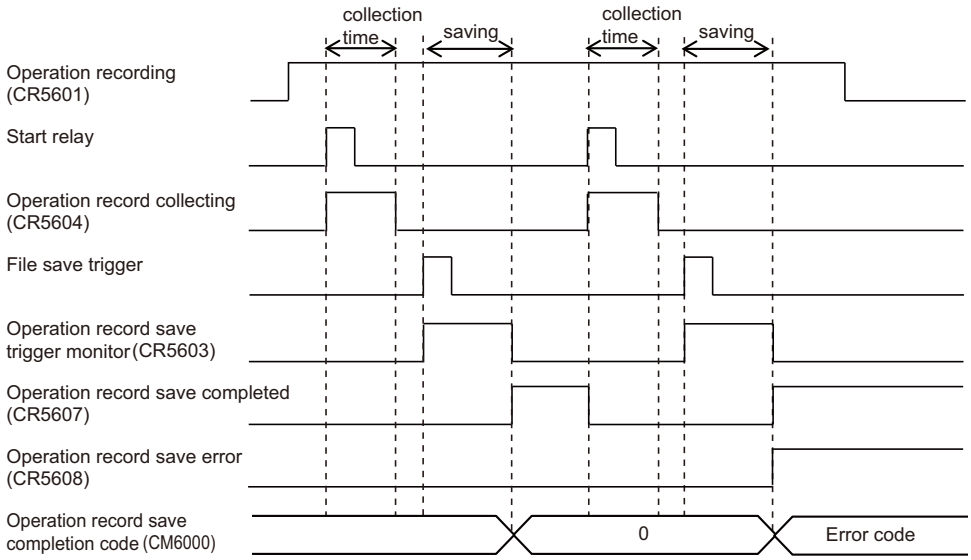
Reference

- The operation recorder function is in operation when the CPU unit switches from PROG mode to RUN mode.
- The CR and CM related to the operation recorder function are controlled automatically by the CPU unit, so there is no need to operate from the ladder program.

Point

During abnormal analysis with the KV-XD02, the operation record save completion relay turns ON when this analysis is completed after the saving of the operation record is completed.

● When "Start relay" is selected



① When the operation recorder function is in operation, the operation recording relay (CR5601) becomes ON.

② When the device specified as the start relay in the operation recorder setting is turned ON, the operation recorder collecting (CR5607) becomes ON, and the CPU unit starts data collection. When the collection time has elapsed, the data collection is ended, and the operation recorder collecting becomes OFF.

③ When the condition for file save trigger is met after the collection time has elapsed, the CPU unit starts saving the operation record, and the operation recorder saving relay (CR5603) becomes ON (If the condition for file save trigger is met before the collection time elapses, the operation recorder will be saved after the collection time has elapsed).

④ When the saving of the operation recorder is completed, the operation recorder save trigger monitor relay and the operation recorder saving relay become OFF, and the operation recorder save completion relay (CR5607) becomes ON. 0 is stored in the operation recorder save completion code (CM6000).

⑤ When the file saving fails, the operation recorder save trigger monitor occurrence relay and the operation recorder saving relay become OFF, and the operation recorder save completion relay and the operation recorder save error relay (CR5608) become ON.
The error code is stored in the operation recorder save completion code.

Reference

- The operation recorder function is in operation when the CPU unit switches from PROG mode to RUN mode.
- The CR and CM related to the operation recorder function are controlled automatically by the CPU unit, so there is no need to operate from the ladder program.

Point

During abnormal analysis with the KV-XD02, the operation record save completion relay turns ON when this analysis is completed after the saving of the operation record is completed.

Operation record save completion code

This is a list of completion codes stored in the operation recorder save completion code (ID0: CM6000).

Completion code	Title	Error cause	Handling method
0	Saved successfully	Saved successfully	Saved successfully
10	No free number	Not written as a file already exists	Modify the folder number to be saved or create a free number.
11	There is no memory card or CPU memory is unmounted.	An operation recorder is saved when the memory card / CPU memory cannot be recognized	Please mount the CPU memory or insert a memory card
12	No free space	There is no free space in the memory card / CPU memory	Please increase the free space of the memory card / CPU memory
100	Cover OPEN	Memory card cover is open	Please close the memory card cover
101	Card LOCK	Memory card is locked	Please release LOCK or use a writable memory card
102	Memory card access error	Access to the memory card has failed due to the influence of external causes, such as memory card deterioration, abnormal contact point, abnormal insertion state or interference.	Check the insertion status of the memory card. Check the file system of the memory card. Execute the chkdsk instruction from the Windows instruction prompt to check the file system of the memory card. Example) When the memory card is E drive C:\WINDOWS>chkdsk E: When an interference source (for example, a cable with large current flowing, such as a motor) is in the vicinity, keep it away from the equipment.
103	Maximum number of files exceeded	The total number of files and folders in the user memory exceeds the upper limit	Reduce the files or folders in the user memory.
104	Project data save failed	• An attempt was made to save the project data, but it cannot be executed (other errors when saving the project data) • Memory card damaged • The memory card is removed while saving the project data	Please check if the project or memory card is proper
200	Project access conflict	Project save cannot be performed as the program is transferred or read from another path	Please save the operation recorder after transferring or reading the program.
201	Failed to access the camera input unit	The camera input unit does not operate normally, or the unit configuration information and the actual unit configuration are different.	Check if an expansion bus communication error has occurred. Check if an error has occurred in the camera input unit. Check if the unit configuration of the Unit Editor matches the configuration of the actual equipment.
202	Variable mismatch	The variables included in the target device for operation recorder setting are not defined in the variable edit window.	Please define the variables (including structure type and function block type, etc.) included in the target device for operation recorder setting in the variable edit window.
1000	Failed to create the folder	An attempt was made to create a folder, but it cannot be created successfully (The memory content has been changed during processing)	Conflict with other memory access Please save again

Completion code	Title	Error cause	Handling method
1001	Failed to create the file	An attempt was made to create a file, but it cannot be created successfully (The memory content has been changed during processing)	Conflict with other memory access Please save again
1002	Memory access conflict	Failed to achieve the memory LOCK (Conflict with other memory access processing)	Conflict with other memory access Please save again

TRGR

⊣[TRGR]⊢	Device record data acquisition	Device data of the specified ID

Ladder program

Entry mode

Operand	Description	Occupied area
n	Specify the ID (0~3) of the operation recorder setting to be acquired or the device storing it.	16 bits

Operation description

TRGR When the execution condition is ON, the device data of the ID specified by n will be acquired.
When "SCAN" or "Fixed-period" is set as the sampling period, the data at the time of END processing will be acquired, but the data at any time point of the ladder can be acquired when "Trigger Instruction" is set as the sampling period and the TRGR instruction is used.

Point **When the data is acquired with the TRGR instruction, it is necessary to set the sampling period to "Trigger Instruction (TRGR)" in "Operation Record Setting" — "Device Record Setting" of KV STUDIO in advance.**

Calculation flag

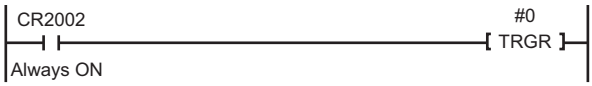
CR2009	No change
CR2010	No change
CR2011	No change
CR2012	• ON when the value specified for the operand is out of the specified range. • ON when the range of indirect specification or index modification is improper. OFF except the above • n No change when specified as a constant.

* When CR2012 is ON, no instruction will be executed.
When CR2012 is ON, the error details will be stored in CM5100~CM5176.

"Latest error details (CM)", page 4-87

Sample program

If the following ladder program is described in the Fixed-period module, the device data can be acquired each time the Fixed-period module is executed.




```
Mnemonic list
 LD CR2002
 TRGR #0
```

Point **When the Fixed-period module is executed, it is necessary to set to the interrupt enable state (execute the EI instruction).**

" Fixed-Period Module", page 4-103
"4-13 Interrupts", page 4-206

Precautions for operation recorder function

The precautions for using the operation recorder function are as follows.

● Save speed

- Scan time will be extended when collecting operation records. The actual scan time can be checked in the scan time monitor.
- The scan time will not be extended as the data collection stops while the operation recorder is saved.

Reference

If the scan time fluctuation is an issue, it can be improved using the fixed scan time operation or via the setting of END processing time.

"Using the fixed scan time operation function", page 4-91
"Setting the END processing time", page 4-92

● Memory access priority

The memory access priority is as follows.
Memory instruction > Recording and tracking instruction > operation recorder function, FTP server function, FTP client function,
Access window file operations
Therefore, when there are a large number of file save triggers for the operation recorder function, other functions may not be able to use the memory.

● Power off, card insertion and removal

The memory card unmount request relay should be turned ON with the main unit power ON, and the memory card should be removed after checking that the memory card recognition completion relay becomes OFF. Do not remove the memory card or open the memory card slot cover, etc. while saving the operation record.

Handling of memory card slot cover

- When the memory card slot cover is not closed, the operation recorder cannot be saved.
- After the memory card slot cover is opened, the processing in execution will continue, but the subsequent processing will not be accepted.

Precautions for power off, card insertion and removal

Do not perform any operation that interrupts saving until the saving is completed while the operation recorder is being saved. After the saving is interrupted before completed, the operation recorder cannot be saved correctly. The operations of saving interrupt are as follows.

- Turn off the CPU unit
- Turn the memory card unmount request relay and the CPU memory unmount request relay ON to unmount the memory
- Open the memory card slot cover (when the file save location is a memory card)

● Operation at RUN/PRG mode switching

The operation recorder function operates only when the CPU unit is in the RUN mode. When switching from RUN→PRG, all operation recorder functions stop. When switching from PRG→RUN, the operation recorder functions restart automatically.

● About processing at the time of operation recorder saving

When saving the operation record, the operations are as follows if the specified folder number already exists.

When the specification method of the folder number is "Automatic consecutive number"

- When "Overwrite" is selected in "When there is no free number", the oldest folder will be deleted and overwritten with the same folder name.
- When "Stop" is selected in "When there is no free number", the saving will fail, and the collection of operation recorder will restart.

When the specification method of folder number is "Fixed / device value"
Delete the existing folder number and overwrite it with the same folder name.

● Characters that cannot be used in the data save path

The following characters cannot be used for data save path comments and trigger comments.

- Half-width 12 characters* / < > ? ￥ ｜ % ， : ; “
- Full-width 7 characters ／ ： ？ ￥ ＊ ［ ］
- NEC special characters (8740H~879CH) ①、㌢, ㌔, ㍻, etc.
- NEC selected IBM extended characters (ED40H~EEFCH)
- IBM selected IBM extended characters (FA40H~FC4BH)

5-7 USB Communication

You can use the USB port of the CPU unit to monitor and debug ladder programs from KV STUDIO. You cannot use the USB port to perform communications such as transmitting commands.

Reference

If you use "KV COM+ for Excel/KV COM+ Library," you can read and write PLC device values.

How to Connect to a PC

Use a USB cable (OP-35331) to connect the PC and the CPU unit.

Point

- **Before you connect the CPU unit to the PC's USB connector, be sure to install KV STUDIO.**
- **The USB port's SG is shared with 0 V of the power supply. A potential difference between the product being connected to and the SG will lead to damage to the CPU unit and to the product being connected to.**

Installing the USB Driver

When a Windows 11/10/8/7 PC is used, USB is automatically recognized and no driver need to be installed.

The Role of CPU Built-In EtherNet/IP Communication Function

Item	Reference manual	Reference page
Remote transmission/remote maintenance of the ladder diagram program	"KV STUDIO User Manual"	"9-1 Monitor/Simulator Functions"
Testing via EtherNet/IP		
Connection with VT5/VT3 Series	"VT/DT PLC Connection Manual"	"Chapter 22 Ethernet connection"
Host link communication	"EtherNet/IP function user's manual"	"Chapter 8 Host link communication function"
MC protocol communication		"Chapter 9 MC protocol communication function"
Mail sending/receiving		"Chapter 10 Mail sending/ receiving function"
FTP server		"Chapter 11 FTP server function"
FTP client		"Chapter 12 FTP client-side function"
KV socket communication		"Chapter 14 KV socket communication function"
BOOTP client		"Appendix-2 Communication without Unit Setting/BOOTP"
Implicit (I/O) message		"Chapter 4 EtherNet/IP communication function"
Explicit message (client)		
Explicit message (server)		
Sensor application function		"Chapter 7 Sensor application function"

For specific content of the available functions, please refer to "EtherNet/IP function user's manual" .

The Role of the OPC UA Server Function

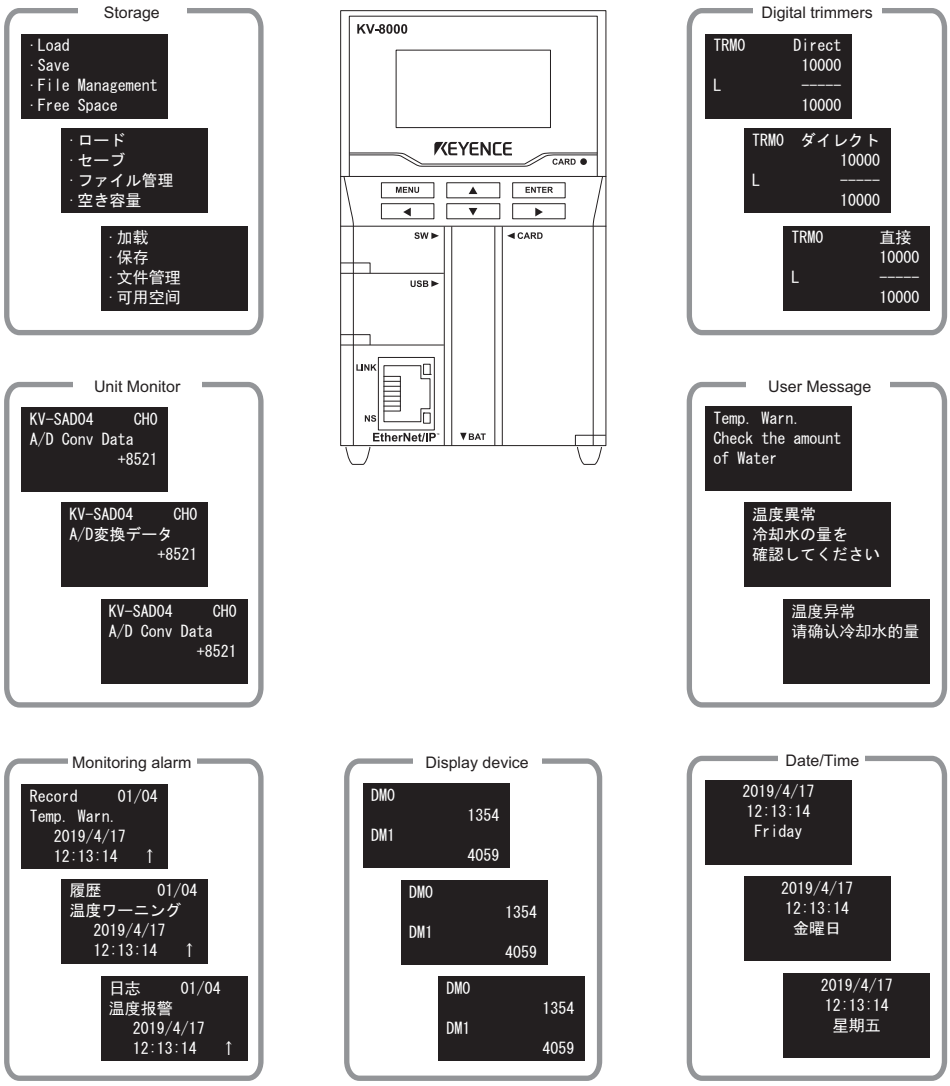
Item	Reference manual	Reference page
OPC UA server device	"OPC UA Server Function User's Manual"	"Chapter 1 OPC UA Function Overview"

5-9 Access Window

This section describes the functions of the Access Window that can be used to monitor or change CPU unit or expansion unit data easily.

Operations that Can be Implemented Through Access Window

The CPU unit is provided with a compact LCD display with 2-color backlight.
The LCD lights white in RUN mode and lights red in PRG mode (halted).



■ Function list

The device data can be changed and monitored through the access window equipped in the CPU unit.
The Access Window also allows the user to check error codes that may occur.
The Access Window has the following functions.

Unit Monitor

In this mode, state of the unit can be monitored on the Access Window by pressing the direct access switch connected to the CPU.

"Unit Monitor" (page 5-115)

Device mode

In this mode, internal devices can be displayed and their settings changed.

"Device Mode" (page 5-118)

Storage

The operation such as project save/load, file management, confirmation of free space inside the memory card and CPU memory is possible.

"Storage" (page 5-139)

Error clear

Clear serious errors of the CPU unit.

"Error Clear" (page 5-127)

Digital trimmer

In this mode, trimmers can be set digitally. The maximum resolution is 4294967296.

"Digital Trimmer" (page 5-128)

Unit Test

In this mode information about the expansion unit, such as the input/output status, can be checked.

"Unit Test" (page 5-130)

Display and setting of time

In this mode, internal clock can be displayed and time can be set.

"Date/Time" (page 5-132)

CPU monitor

In this mode, the internal functions of the CPU can be monitored.It also allows you to verify names of projects in the unit, unit serial number and version.

"CPU Monitor" (page 5-134)

Monitoring alarm

This mode allows you to check alarm monitor settings and log.

"Alarm" (page 5-136)

Access Window setup

In this mode, you can set the contrast of the Access Window, and other related settings.

"Access Window Setup" (page 5-151)

Language
The language displayed can be selected from Japanese, English, Chinese (simplified).

RAM clear mode
In this mode, device values can be cleared.

"RAM clear" (page 5-153)

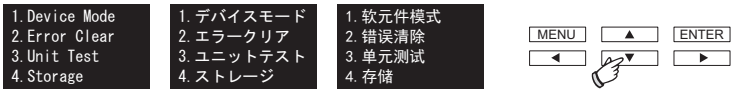
All Clear mode
In this mode, all settings can be cleared.

"All Clear" (page 5-154)

Clear CPU Memory
The data in the CPU memory can be cleared.

■ Menu Structure

Switching among modes is implemented by menu.
Pressing the "MENU" key displays a menu. (When other units are selected, press the direct access key for the unit, or press "MENU" key twice.)
If not all of the menu items fit in the screen, press the "▼" key to display the rest of the menu. Press the "▲" and "▼" keys to scroll the menu up and down.



In the examples shown, Japanese is displayed on the left, English is displayed in the center and Chinese is displayed on the right. Refer to "11.LANGUAGE" for information on how to switch between Japanese, English and Chinese.

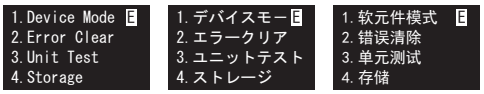
"Access Window Setup" (page 5-151)

Point **Part of the menu does not correspond to written Chinese. When a non-corresponding phrase is displayed in Chinese, it becomes written English.**

The following table shows the CPU menu items.

English	Japanese	Chinese
1.Device Mode	1.デバイスモード	1.软元件模式
2.Error Clear	2.エラークリア	2.错误清除
3.Unit Test	3.ユニットテスト	3.单元测试
4.Storage	4.ストレージ	4.存储
5.Monitor	5.CPUモニタ	5.CPU监控
6.Data/Time	6.時計	6.时钟
7.Trimmer	7.トリマ	7.微调器
8.Alarm	8.アラーム	8.报警
9.AW Setting	9.AW設定	9.访问窗设定
10.Clear	10.クリア	10.清除
11.Language	11.LANGUAGE	11.LANGUAGE

Reference When serious error occurs and its log hasn't been cleared, "E" will be displayed on the upper right of Access Window.

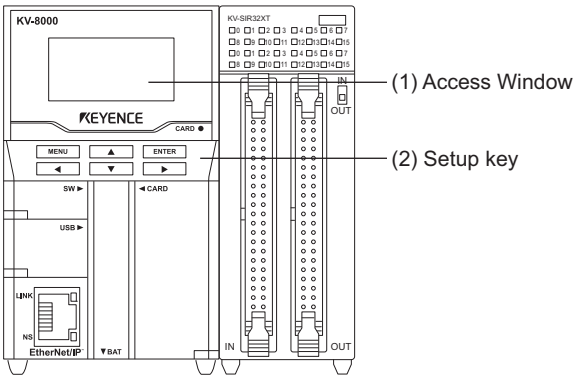


For the Method of use, see "Error Clear" (page 5-127).

The Access Window menu is configured as follows:

- 1.Device Mode（1.デバイスモード/1.软元件模式）
- 2.Error Clear（2.エラークリア/2.错误清除）
- 3.Unit Test(3.ユニットテスト/3.单元测试）
- 4.Storage（4.ストレージ/4.存储）
  - Load（ロード/加载）
  - Save（セーブ/保存）
  - File Management（ファイル管理/文件管理）
  - Free Space（空き容量/可用空间）
- 5.CPU Monitor（5.CPUモニタ/5.CPU监控）
  - Project Name（プロジェクト名/项目名）
  - Version（バージョン/版本）
  - Ethernet（イーサネット/以太网）
- 6.Data/Time（6.時計/6.时钟）
- 7.Trimmer（7.トリマ/7.微调器）
- 8.Alarm（8.アラーム/8.报警）
  - Monior（モニタ/报警监控器）
  - Monitor Record（履歴モニタ/日志监控器）
  - Clear Record（履歴クリア/日志清除）
- 9.AW Setting（9.AW設定/9.访问窗设定）
  - Contrast（コントラスト/对比）
  - Comment Number（コメント番号/注释编号）
- 10.Clear（10.クリア/10.清除）
  - RAM Clear（RAMクリア/RAM 清除）
  - User Mem Clear(ユーザメモリクリア/用户存储器清除）
  - All Clear(オールクリア/全部清除）
- 11.Language(11.LANGUAGE/11.LANGUAGE）

■ Names and Functions of Parts



(1) Access Window　　Displays the current values and setting values of each device, and error messages.
Information of each unit is displayed by pressing the direct access switch for the respective unit.

(2) Setup key　　Used for switching between modes and changing settings in each mode.
Key functions differ according to each mode. For details, see the operation instructions for each mode.

Unit Monitor

Units can be monitored and set in the Access Window by pressing the direct access switch for special units or expansion units such as the I/O unit connected to the CPU.
To return to the original screen from the expansion unit or special unit screen, press the direct access switch of the unit displayed in the access window again, or press the "MENU" key twice.

Reference When unit error occurs, the direct access switch will turn red, and the error contents will be displayed in the Access Window.

For details on monitor items and settings for each unit, refer to the User's Manual for the respective unit.

■ Functions of setup key

The following table summarizes key functions when monitoring or setting up units.

Setup operation keys	General Function	When Changing Numerical Values
MENU	Displays the menu.	It returns to normal display.
ENTER	Hold down 1 second or longer to change settings.	Hold down 1 second or longer to change settings.
▲	Turn the page and select an option to change settings.	Calculates the amount of change from the setting value.
▼	Turn the page. Select an option to change settings.	Subtracts the amount of change from the setting value.
◀	Displays the menu.	Changes the number of digits in the amount of change.
▶	Not used.	

Reference

Device number of configuration operation key

When the configuration operation key of the unit is pressed, the following control relay（CR）will turn ON.

Configuration operation key	Device number	Name of comment
MENU	CR4404	AW MENU key is being pressed
ENTER	CR4405	AW ENTER key is being pressed
▲	CR4400	AW up key is being pressed
▼	CR4401	AW down key is being pressed
◀	CR4402	AW left key is being pressed
▶	CR4403	AW right key is being pressed

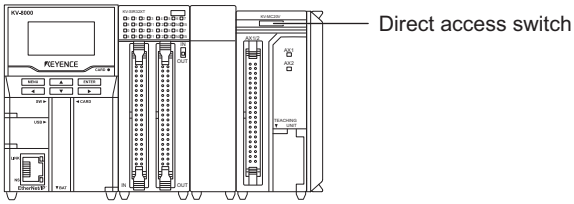
■ Method of Operation (example: on I/O Unit)

We will take KV-SIR32XT for example to describe the operation methods.

● Switch to Unit Monitor

1 Press the direct access switch on the unit.

The direct access switch for the selected unit flashes.



2 The ON/OFF status of relay will be displayed.

The access window changes as follows according to the selected unit.

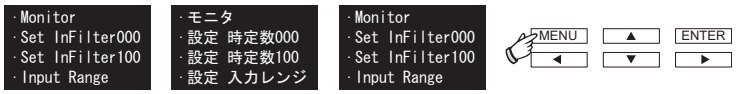
Press "▲" and "▼" to switch the pages.

Point **The unit monitor does not correspond to Chinese signage.**

● Menu display

1 Press the "MENU" key with the direct access switch for the unit flashing (selected).

The menu for the unit is displayed.



● Changing Settings

Let's take altering the input time constant as an example.

1 Select constant 000 when configured and then press the "ENTER".

The currently set time constant can be checked.

2 Hold down the "ENTER" key for at least one second in this state

The setting value flashes, and the unit enters the change mode. (Not possible during RUN, can only be modified during PRG mode.)

3 Change the setting value using the up/down keys, and hold down the " ENTER" key for at least one second.

The setting value stops flashing and is set. (It cancels when the MENU key is pressed.)

Device Mode

In this mode, PLC device values can be monitored and changed.

■ Displaying devices and changing their settings

The following table summarizes devices that can be displayed and set.

Display	Name	Display	Name
R	Relay	ZF	File Register (consecutive number mode)
MR	Internal auxiliary relay	W	Link register
LR	Latch relay	CM	Control memory
B	Link relay	TM	Temporary memory
CR	Control relay	T	Timer
DM	Data memory	C	Counter
EM	Extended data memory	TRM	Digital trimmer
FM	File register (memory bank switching mode)	Z	Index register

Point **The contact states of timers, and counters cannot be monitored.**

● About XYM marking

XYM marking can be performed on relay (R), internal auxiliary relay (MR), latch relay (LR), data memory (DM), expansion data memory (EM) and file register (FM).
Default marking/XYM marking can be automatically switched by the setting of items (KV STUDIO's Tool > Options > Display mode setting > XYM marking).

Device name	Default marking	XYM marking		Device No. XYM marking
Relay	R	Input	X	0000 to 1999F
		Output	Y	0000 to 1999F
		Internal auxiliary	R	0000 to 199915
Internal auxiliary relay	MR	M		00000 to 63999
Latch relay	LR	L		00000 to 15999
Data memory	DM	D		00000 to 65534
Extended data memory	EM	E		00000 to 65534
File register	FM	F		00000 to 32767

■ Functions of Setup key

The following table summarizes key operations in the device mode:

Setup operation keys	General Function	When Changing Numerical Values
MENU	Displays the menu.	Returns to general function display.
ENTER	Switches to change in numerical status. Hold down 1 sec or more.	Set/Cancel changed value Hold down 1 sec or more to set. Ignored if held down less than 1 second.
▲	Selects the device number.	Increases the value being changed.
▼	Selects the device number.	Reduces the value being changed.
◀	Displays the device type setting.	Moves the digit in the value being changed to the left.
▶	Displays the Device display setting screen.	Moves the digit in the value being changed to the right.

■ Operation method

● Switch to the Device mode

1 Press the "MENU" key.

A menu is displayed.(If other units are selected, press the "MENU" key twice.)

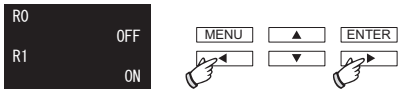
2 Select "1. device mode ", and press the "ENTER" key.

The monitor screen is displayed in the device mode.

The device No. to be displayed can be changed by the "▲" "▼" keys.

● Changing type of device to be monitored

1 Press the "◀ / ▶" key in device mode



2 A list of device types is displayed.

Select the device to display using the "▲" and "▼" keys.

In this example, select device "TM".

The device types are displayed in the following order: R→MR→LR→B→CR→DM→EM→FM→ZF→W→CM→TM→T→C→TRM→Z. When the "▼" key is pressed with Z selected, "R" will be selected again.

Likewise, when the "▲" key is pressed with "R" selected, "Z" will be selected .

3 "·TM" is displayed flashing.

Press the "ENTER" key.

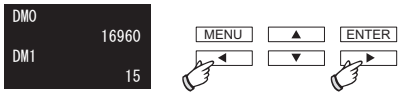
The screen for determining the desired device No. is displayed.
The current device No. is displayed on the 3rd line.

The numerical value displayed on the 4th line is the device No. change increment.
Change the device No. using the "▲" and "▼" keys.
As shown in this diagram, device No. will increase by an increment of 10 after pressing "▲" key.
The device No. change increment will be multiplied by 10 or 1/10 when the "◀ / ▶" keys are pressed.
Determine the device No. to display using the "▲", "▼" and "◀ / ▶" keys.

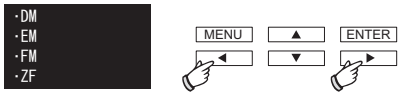
Press the "ENTER" key. This displays the TM of the specified No.

■ Changing the device value display format

1 Press the "◀ / ▶" key in device mode



2 Press the "◀ / ▶" key with the list of device types displayed.



3 This displays the display format selection screen.

4 Select the display format.

Select the display format using the "▲" and "▼" keys.

5 **Press the "ENTER " key. This displays the monitor value in the selected display format.**
The display format selected here is applied to all devices.

The following table summarizes display formats that can be selected.

English	Japanese	Chinese	Display Contents
Default	デフォルト	默认格式	Vary depending on device type (see appendix)
1-bit BIN	2進数1BIT	1位二进制	ON/OFF display
16-bitDEC	10進数16BIT	无符号16位十进制	Data width 16-bit, unsigned, decimal display
±16-bit DEC	±10進数16BIT	有符号16位十进制	Data width 16-bit, signed, decimal display
16-bit HEX	16進数16BIT	16位十六进制	Data width 16-bit, hexadecimal display
16-bit BIN	2進数16BIT	16位二进制	Data width 16-bit, binary display
32-bit DEC	10進数32BIT	无符号32位十进制	Data width 32-bit, unsigned, decimal display
±32-bit DEC	±10進数32BIT	有符号32位十进制	Data width 32-bit, signed, decimal display
32-bit HEX	16進数32BIT	32位十六进制	Data width 32-bit, hexadecimal display
ASCII 16BIT	ASCII 16BIT	16位ASCII码	ASCII character representation
FLOAT	FLOAT	单精度浮点型	Data width 32-bit, single-precision floating point representation
DOUBLE FLOAT	DOUBLE FLOAT	双精度浮点型	Data width 64-bit, double-precision floating point representation

Formats of devices that can be displayed:

◎: Can be displayed (Default); Yes: Can be displayed; No: Cannot be displayed

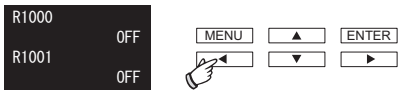
Device type	ON/ OFF	16Bit-BASE 10 unsigned	16Bit-BASE 10 signed	16Bit-BASE16	16Bit-BASE2	32Bit-BASE 10 unsigned	16Bit-BASE 10 signed	32Bit-BASE16	ASCII	FLOAT	DOUBLE FLOAT
R/MR/LR/B/CR	◎	Yes	Yes	Yes	Yes*1	No	No	No	No	No	No
DM/EM/FM/ZF/W/CM	No	◎	Yes	Yes	Yes*1	Yes	Yes	Yes	Yes*1	Yes*1	Yes*1
T/C/TRM	No	No	No	No	No	◎	Yes	Yes	No	No	No
Z	No	No	No	No	No	No	◎	Yes	No	No	No

The display format that can be selected varies according to the device type. When a display format that cannot be applied is selected, the default display format is selected.

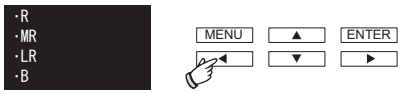
*1 Display is avalible, but device value rewriting from the access window in this format is not supported.

● Device name/comment display switching

1 Press the "◀ / ▶" key in device mode.



2 Press the "◀ / ▶" key with the list of device types displayed.



3 Press the "◀ / ▶" key in the display format selection screen.

4 The screen for selecting device name/comment display is displayed.

Select whether to display the device name or comment using the "▲" and "▼" keys.

Pressing the "▼" key displays the comment.

5 Press the "ENTER" key. This selects comment display, and the display returns to the device mode.

The comment "sensor A" is registered in R1000 and when the comment is not registered to R1001, the display will be as follows.

About comments that can be displayed in the Access Window

- When more than 16 single-byte characters are entered, characters are automatically scrolled.
- The screen displays Unicode characters.
- Comments can not be set up for Z.

● Changing device values

- When display format is ON/OFF

1 Hold down the "ENTER" key for at least one second with the device to change displayed.

If two devices are currently displayed in one screen, the device whose value can be changed is the device displayed on the previous row of the screen. (In this case, the device on the next row is no longer displayed).

2 The value of the device is displayed flashing.

In this state, press the "▼" and "▲" keys. This toggles the display in the order ON->OFF->ON->···. (In this case, device value will not be stored)

3 To set the desired device value and hold down the "ENTER" key for at least one second.

Flashing stops and device value will be stored. Press the "MENU" key to stop loading and return to monitor display.

- When displaying format is other than ON/OFF (except T, C, TRM)

The following example describes how to change the value of DM1000 No. from "100" to "201".

1 Hold down the "ENTER" key for at least one second with the device to change displayed.

2 The value of the desired device flashes.

The "increment" is displayed on the bottom row of the screen.

3 Press the "▲" key.

The device value is incremented by the amount set , and the device value changes to "101".

(In this case, the device value will not be stored into device.)

4 Press the "◀ / ▶" key once.

This changes the digit of the increment. Each press of the key changes a digit one time.

To change the hundreds digit, press the "◀" key twice.

5 The increment value changes to "100".

6 Press the "▲" key once.

The device value is incremented by the "100" and the display changes to "201".

(In this case, the device value is not stored into device.)

7 Adjust the device value to a suitable value, and hold down the "ENTER" key for at least one second.

This stores the value, and the screen returns to the monitor display.

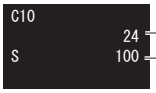
It returns to monitor display without executing write in when the "MENU" key is pressed.

- When display format are not ON/OFF(T, C, TRM)

The setting value and current value can be changed in the case of T, C, and TRM devices.

Current value of the timer

Set value of the timer



Current value of the counter

Set value of the counter

Value of trimmer

Upper limit of trimmer

If a device is not being used by a ladder program (the device is not registered), it is displayed as shown in the figure to the left.

(Ladder Not displayed when upper limit is not set.
"Digital Trimmer", Page 5-128)

<How to change setting values>

To change device values, hold down the "ENTER" key for at least one second. The following items can be changed:

T : Timer setting value

C : Counter setting value

TRM : Trimmer

<How to change current values>

The following items can be changed by holding down the "ENTER" key for at least one second with the "◀ / ▶" key pressed.

T : Timer current value

C : Counter current value

TRM : Trimmer upper limit value (The trimmer upper limit value is assigned to CM.)

"Digital Trimmer", Page 5-128

Confirm error details

There are two kinds of CPU unit errors: minor error and serious error.

	Minor error	Serious error
Colors of Access window	White	Red
Displays of Access Window	Blinking while displaying error contents CPU error W128 Ladder calculation error	Blinking while displaying error contents CPU error E52 Unit Unconnected
Status of CPU	Leave in RUN mode (Ladder program works)	Switch to PROGRAM mode (Ladder program will stop)
Error clearing methods	• Press a certain key in Access Window • Set CR 3909 on Ladder Program	Since CPU entersPROGRAM mode, error can not be cleared from Ladder Program. For error clearing methods, refer to "Error Clear", Page 5-127.

Reference

In the case of a serious error occurs, pressing a certain key in Access Window will erase the error display, but the error cannot be cleared. If you don't operate the Access Window for 30sor longer, or switch the RUN⬌PRGselector to RUN, the error contents will be displayed again.

■ What to do when an error occurs

1 Check the error code.

Check the error code displayed in Access Window.

The display of the error code differs between slight and serious error as follows.

Minor error…"W○○○"

Serious error…"E○○○"

CPU error E52
Unit Unconnected

In the example on the right, you can see that a serious error with error code 52 has occurred.

2 Check the error contents and solving methods.

Check the error contents and solving methods in "List of PLC Unit Errors" (page A-2) and remove the cause of the error.

52	Configuration error	ユニット未接続 Unit Unconnected 単元未連接	In the unit setting information,there is an extension unit, but itis not actually connected.	After turning off the power,connect the extension unit set inUnit Editor.

In case of error occurred in the extension unit, check the user's manual of each unit.

Reference

When a serious error occurred, CR 3905 turns ON, and error code is stored in CM 5105.
If a minor error occurred,CR 3906 turns ON, and error code is stored in CM 5155.
With CRsand CMs, you can check the error contents from the touch panel. See "Latest error details (CM)" (page 4-87)

3 Execute Error Clear.

In the case of a serious error has occurred, pressing a certain key in Access Window, or Set CR 3909 on Ladder Program.

If a serious error has occurred, you need to clear the error by operating Access Window.

For error clearing methods, refer to "Error Clear", Page 5-127.

Error Clear

Clear serious error of the CPU unit.

■ Functions of Setup key

The following summarizes key functions when clearing the error.

Setup operation keys	Function
MENU	Menu display
ENTER	Hold down for 1 second or longer to clear displayed errors. Hold down less than 1second to disable. Disabled also when no error has occurred.
▲	Disabled
▼	Disabled
◀	The same as MENU key
▶	The same as ENTER key

■ How to operate

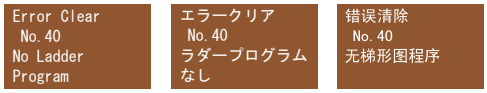
1 Press the "MENU" key.

A menu is displayed. (If other units are selected, press the "MENU" key twice.)

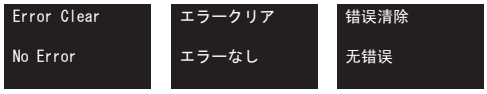
2 Select "2. Clear error" and press "ENTER" key.

The following screen is displayed.

- Screen example when an error occurs



- Screen example when no error occurs



3 After changing the value, hold down the "ENTER" key again for at least one second.

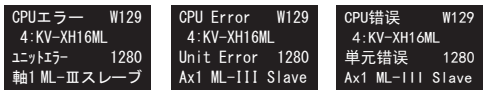
Serious error displayed can be cleared.

If several severe errors occur, the other ones will be displayed. Press "ENTER" key for more 1s to clear them. "No error" will be displayed when all of the severe error is cleared.

Reference

If an error occurs on the extension unit, the following message will be displayed in the access window.

In this example below, you can see that error number 1280 (axis 1 ML-III slave station communication error) has occurred in KV-XH16ML of unit number 4.



If an error occurs on the extension unit, it is necessary to clear the error on the extension unit side after removing the error cause. For how to clear the extension unit error, refer to the user's manual of each unit.

Digital Trimmer

In numeric tuner mode, the trimmer's value can be changed simply through device mode.
Use TMIN instruction to read the value of trimmer in ladder program.

"KV-8000/7000/5000/3000/1000 and Nano Series Instruction Reference Manual, "TMINInstruction""

■ Functions of setup key

The following table summarizes key operations when changing the trimmer value:

Setup operation keys	When Changing Numerical Values
MENU	Displays the menu.
ENTER	Changes the displayed trimmer number. Each press increments the device number by 1. (TRM0 to 9)
▲	Calculates the amount of change from the trimmer value.
▼	Subtracts the amount of change from the trimmer value.
◀	Changes the number of digits in the amount of change. Each press increments the digits by 1. (When the upper limit of digits is exceeded, the least significant digit is displayed.)
▶	Changes the number of digits in the amount of change. Each press decrements the digits by 1. (When the lower limit of digits is exceeded, the most significant digit is displayed.)

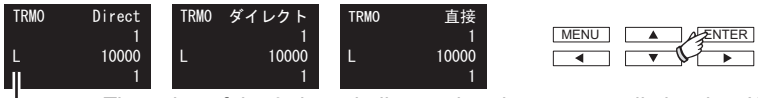
Changing values in this mode differs from changing values in the device mode. In the digital trimmer mode, changes to values are reflected directly without pressing the "ENTER" key.

■ How to operate

1 Press the "MENU" key.

A menu is displayed. (If other units are selected, press the "MENU" key twice.)

2 Select '2.trimmer" and press "ENTER" key.



The value of the 3rd row indicates the trimmer upper limit value. When the upper limit value setting is not used, it will not be displayed.

3 The trimmer No. displayed can be changed by the "ENTER" key.

Select the trimmer No. to change.

4 Change the increment value by the "◀ / ▶" key.

5 Pressing the "▲" key increases the trimmer value by the increment value, and pressing the "▼" key decreases the trimmer value by the decrement value.

The digital trimmer mode differs from the device mode in that changes to values are reflected directly without pressing the "ENTER" key.

Reference

- The current trimmer value can be read by the TMIN instruction.
- To enable setting of upper limit, it's necessary to set on KV STUDIO. ("CPU system setting" -> "Setting of digital trimmer's upper limit".)
- Upper limit can be set in KV STUDIO or Access Window's device mode (CM1700 to CM1715).
- When the setting of upper limit is enabled, the upper limit will change to 4294967295 automatically if the setting value of upper limit is set to "0".
- The upper limit is stored in the following CM

Trimmer No.	Device
TRM0 upper limit	CM1700, CM1701
TRM1 upper limit	CM1702, CM1703
TRM2 upper limit	CM1704, CM1705
TRM3 upper limit	CM1706, CM1707
TRM4 upper limit	CM1708, CM1709
TRM5 upper limit	CM1710, CM1711
TRM6 upper limit	CM1712, CM1713
TRM7 upper limit	CM1714, CM1715

Unit Test

These allow you to verify the input/output status and other information of an expansion unit.

Point **It can only be executed during PRG mode.**

Reference This can also be verified when the settings of unit configuration set through KV STUDIO haven't been transmitted to CPU.

■ Functions of setup key

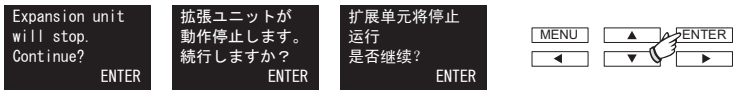
Setup operation keys	When Changing Numerical Values
MENU	Returns to the unit selection screen.
ENTER	Input unit: Disabled Output unit: Moves the cursor position forward by 1 bit.
▲	Invalid
▼	Invalid
◀	The same as MENU key
▶	The same as ENTER key

■ How to operate

● Migration to unit test

1 Press the "MENU" key.
A menu is displayed. (If other units are selected, press the "MENU" key twice.)

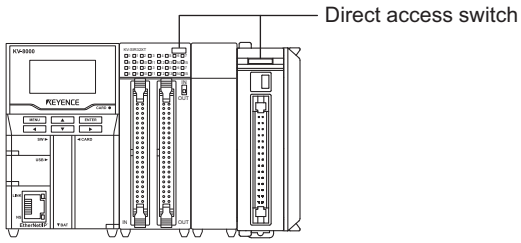
2 Select "3.Unit Test" and press the "ENTER" key.
The following screen appears.



3 Press the "ENTER" key for 1 second or longer.
The following screen will appear.

4 Select the direct access switch of verified expansion I/O unit.

The direct access switch for the selected unit flashes.



The unit that corresponds to unit test is the expansion unit for KV-8000/7000 series and expansion I/O unit for KV-5000/3000 series.

Please refer to the user manual of each unit regarding the unit test content when the expansion unit for KV-8000/7000 series is used. When expansion I/O unit for KV-5000/3000 series is used is as follows.

● In the case of input unit

Example) KV-C32XC

Shift the verified relay position according to every 16 digits by pressing "▲" "▼".

● In the case of output unit

Example) KV-C32TD

Shift the verified relay position according to every 8 digits by pressing "▲" "▼".

Press "◀" "▶" key to change the cursor position.

Press "ENTER" key to reverse the output state of cursor position (ON/OFF).

Point **Please operate after verifying the state of connected control equipment.**

● I/O unit

Example) KV-C16XTD

The relay position to confirm can be moved in groups of 16 bits (8 bits for output relay) with the "▲" "▼" key.

"◀" "▶" key may be used to change cursor position.

"ENTER" key may be used to inverse output state (ON/OFF) of the cursor position.

Point **Please operate after confirm the state of connected control equipment carefully.**

Date/Time

The date/time can be displayed and set.

The time will be displayed in 24-hour format.

The day of the week is displayed as follows:

Sunday	：日曜日	：星期日
Monday	：月曜日	：星期一
Tuesday	：火曜日	：星期二
Wednesday	：水曜日	：星期三
Thursday	：木曜日	：星期四
Friday	：金曜日	：星期五
Saturday	：土曜日	：星期六

■ Functions of setup key

The following table summarizes key operations when setting the date/time.

Setup operation keys	Clock monitor	Date and time setting	Adjust screen
MENU	Displays the menu.	Clock monitor display	Clock monitor display
ENTER	Pressing in the time display for 1 second or longer will activate the time setting.	Press for one second or longer to apply the settings and display the monitor screen	Make adjustments.
▲	Disabled	Change settings	Disabled
▼	Disabled	Change settings	Disabled
◀	The same as MENU key	Change setting items.	The same as MENU key
▶	Switch to the adjust screen.	Change setting items.	Invalid

■ How to operate

● Switching clock

1 Press the "MENU" key.

A menu is displayed. (If other units are selected, press the "MENU" key twice.)

2 Select "5.Clock" and press the "ENTER" key.

The following screen appears.

● Date/time setting

1 Press "ENTER" key for more than 1s in date/time.

The "Year" segment flashes.

2 Select a settings by pressing the "◀" "▶" key.

Each time the "◀""▶" key is pressed, the items are switched between as follows: "Year", "Month" , "Day", "Hour", "Minute", "Second", and "Day of week".

3 Select the item using the "▲" and "▼" keys.

4 Repeat steps 2 and 3. After setting, hold down the "ENTER" key for at least one second.

"Exec" is displayed at the bottom row of the screen indicating that the time setting request was accepted.
(In this case, the date/time has not been set.)
Release the "ENTER" key. This applies setting of date/time, and returns the screen to the date/time display.

It returns to time display without executing time configuration when the "MENU" key is pressed.

Point

Set year, month, day, hour, minute, second, and day of the week accurately.
The day of the week is not checked corresponding to the year, month, day, hour, minute, and second.

■ 0 second adjustment

0 second adjustment is a function for setting the seconds value to "0".

Current Seconds Value	Operation
0 to 29	Only sets seconds value to "0".
30 to 59	Add 1 minute, and sets seconds value to "0".

1 Press the "▶" key in the date/time display state.

The "Adjust will be executed when ENTER is pressed" will be displayed in 3rd, 4th line and time adjust becomes possible.

(To cancel time adjustment, press the "◀" key. The "Adjust" display disappears, and the screen returns to the date/time display.)

2 Press the "ENTER" key.

CPU Monitor

CPU's internal functions can be monitored. Meanwhile, the names of projects stored in CPU unit can be displayed.

■ The items that can be monitored

The items that can be monitored via Access Window are shown as follows.

English	Japanese	Chinese	Function
Project Name	プロジェクト名	项目名	Indicates the names of projects stored in the CPU unit. (The name is set to KV STUDIO)
Version	バージョン	版本	Indicates the version of the unit system program and the serial number. * The expansion unit for KV-8000/7000 series is the only one that corresponds.
Ethernet	イーサネット	以太网	Can monitor CPU built-in EtherNet/IP/Ethernet function and OPC UA server function. For the contents that can be monitored, refer to the EtherNet/IP Function User's Manual.

■ How to operate

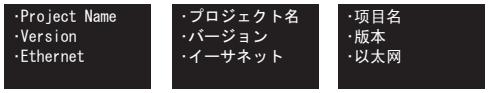
● Switch to CPU monitor

1 Press the "MENU" key.

A menu is displayed. (If other units are selected, press the "MENU" key twice.)

2 Select "6. CPU Monitor" and press "ENTER" key.

CPU monitor menu is displayed.



● Project Name

1 Select "Project name" in the CPU monitor menu and press the "ENTER" key.

Project name will be displayed.

Project names that can be displayed in the Access Window

- Up to 8 full-width (16 half-width) characters can be displayed per line. If the project name does not fit on one line, it will scroll.

● Version

1 Select "Version" in the menu of CPU monitor and press the "ENTER" key.

The version will be displayed.

Reference

The version, serial number for expansion unit other than CPU unit can be confirmed by using the "▲" and "▼" keys.
(It will not appear to the version screen for expansion unit other than for KV-8000/7000 series.)

Alarm

After messages are registered in the alarm relays (CR3300 to CR3415) as device comment, the messages will be displayed in Access Window automatically when alarm relay is ON. In addition, alarm logs generated can also be confirmed or deleted.

■ How to use

Devices used during alarming are described below.

Control relay	R/W	Description
CR3300 to CR3415	R/W	Alarm relay (32). Device comment is registered in advance.
CR3500	R	Even if one alarm relay exists, CR3500 will also be ON. When all the alarms in ON state are closed, CR3500 will be OFF automatically.
CR3501	R/W	ON→OFF log is stored.
CR3502	R/W	The alarm relay log is cleared when this switches from OFF to ON

Point

Be sure to operate the alarm relays by the SET and RES instructions. Currently operating alarm relays can be reset and batch-cleared also by the ARES instruction.

● Register of alarm message

Device comments are registered into alarm relays (CR3300 to CR3415).

Point

The maximum message length that can be displayed in the Access Window is 8 full-width (16 half-width) characters. When no device comment is set, the alarm relay No. will be displayed.

● About alarm log

CPU unit can store up to 16 alarm logs.

In the event of excess of 16, the latest log will replace the earliest one on in sequence.

Each log will store alarm relay No., time (year/month/day/hour/minute/second) and ON/OFF State.

- When ON → OFF log relay (CR3501) is in OFF state.
  Alarm relay is only stored in OFF → ON state.
- When ON → OFF log relay CR3501 is in ON state.
  Logs will be stored when alarm relay is in OFF → ON and ON → OFF state.

The alarm relay log is cleared when CR3502 turns ON. When it has been cleared, CR3502 turns OFF.

■ The items that can be operated

The functions that can be operated from Access Window are as follows.

English	Japanese	Chinese	Contents to be set
Monitor	モニタ	报警监控器	Alarm relay in ON state can be monitored in real-time.
Monitor Record	履歴モニタ	日志监控器	Can monitor ON/OFF logs of alarm relay.
Clear Record	履歴クリア	日志清除	Clear ON/OFF logs of alarm relay .

■ How to operate

● Switch to alarm

1 Press the "MENU" key.

The menu will be displayed. (When other page is selected press the "MENU" key.)

2 Select "8. Alarm", press "ENTER" key.

Alarm menu is displayed.

● Monitor

1 Select "Monitor" in the Alarm menu, and then press "ENTER" key.

The alarm monitor is displayed.

When several alarm relays are in ON state, other alarms can be displayed by pressing "▲" and "▼".

(1) "01/03"

Alarm No. currently displayed/total number of alarms occurred.

When several alarms occur, the alarm with smaller alarm relay No. will be displayed first.

(2) "Abnormality of Upper Speed Limit"

Comment of the destination device of currently displayed alarms.

Reference

- After pressing a certain key, flashing stops.
- In case any alarm occurs when other screens are currently displayed, the alarm monitor will be displayed automatically.

● Log display

1 Select "log monitor" in the alarm menu and press the "ENTER" key.

Alarm logs will be displayed.

Press "▲"key or "▼" key to display previous/next log.

(1) "01/04"

Alarm log No./total number of logs currently displayed.

The latest log No. is "01".

(2) "Temperature warning"

Comment of the destination device of currently displayed logs.

(3) "↓"

It indicates the log when alarm relay switches from ON to OFF. The log when alarm relay switches from OFF to ON is indicated by "↑".

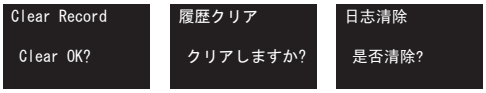
(4) "2015/3/13 12:13:14"

It indicates the log date and time.

● Log clear

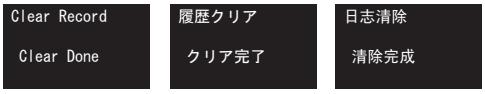
1 In the alarm menu, select "Clear Record"and then select "ENTER."

The confirmation message of cleared logs will be displayed.



2 Hold down the "ENTER" key for at least one second.

The following contents are displayed when the logs are cleared.



Press "MENU" key to return to alarm menu display.

Storage

In the Access Window, you can save data and read files from the memory card or CPU memory.

For details on memory card and CPU memory functions, refer to "5-2 Memory Card" (page 5-3), "5-3 CPU Memory (User Memory)" (page 5-27).

■ Items that can be operated

The functions that can be operated from Access Window are as follows.

English	Japanese	Chinese	Contents to be set
Load	ロード	加载	Read data from any project folder into CPU unit.
Save	セーブ	保存	Save CPU unit data to a memory card or the CPU memory. Save ladder program, device value, device comment, unit setting Information and log setting Information.
File Management	ファイル管理	文件管理	Display a list of directories and files on the memory card or in the CPU memory. You can also select Copy, Move or Delete from the list to process a file.
Free Space	空き容量	可用空间	Display the free space available on the memory card or in the CPU memory.

Point **"Load" is only possible during PRG mode.**

■ How to operate

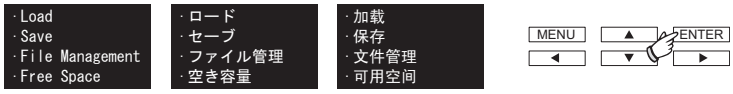
● Switch to Storage

1 Press the "MENU" key.

A menu is displayed. (If other units are selected, press the "MENU" key twice.)

2 Select "4. Storage" and press "ENTER" key.

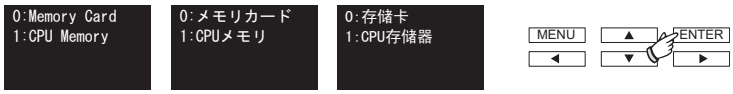
Storage menu is displayed.



● Load

1 Select "Load" in the storage menu, and then press the "ENTER" key.

A message appears prompting you to select a memory card or CPU memory storage device.



2 Use "▲" and "▼" to select a storage device, then press the "ENTER" key or the "▶" key.

The (project) folder in the root folder of the memory card or CPU memory appears.

(No file is displayed. Folder names are displayed as "<D>KVSAVE01"where the first three characters are displayed as "<D>.")

Reference Scroll to display long folder names.

3 Select a folder by pressing "▲" and "▼", then press the "ENTER" key or the "▶" key.

A message asking you to confirm that you want to perform the load operation is displayed.

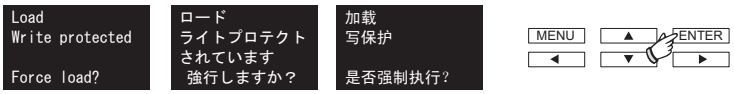
4 **Execute load by pressing the "ENTER" key for 1 second or longer.**

During loading, the following screen is displayed.

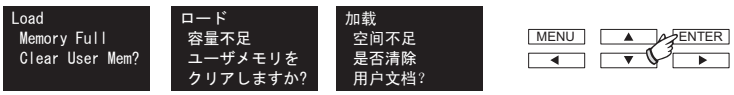
A load operation once started cannot be canceled.

Point

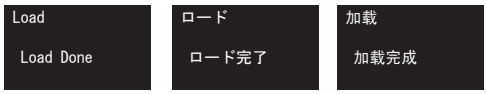
- **If the memory card is write-protected, the following screen is displayed.**
  The load will be executed by force when the "ENTER" key is pressed for 1 second.
  The load will be cancelled when the "MENU" key is pressed.



- **When there is little free space in CPU memory and it is not possible to change CPU memory capacity, the following screen appears.**
  The CPU memory clear will be executed when the "ENTER" key is pressed for 1 second.
  The load will be cancelled when the "MENU" key is pressed.



5 **The following screen is displayed after loading is completed.**

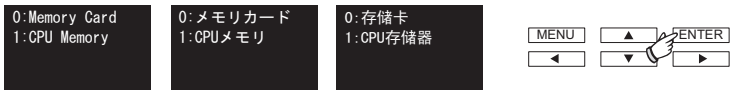


Press the "MENU", "ENTER", "◀", and "▶" keys to return to the storage menu.

● Save

1 Select "Save" in the storage menu and press the "ENTER" key.

A message appears prompting you to select a storage device, either a memory card or CPU memory.



2 Use "▲" and "▼" to select a storage device, then press the "ENTER" key or the "▶"key.

A message asking you to confirm that you want to save CPU data is displayed.

3 Execute save by pressing the "ENTER" key for 1 second or longer.

The software looks for a directory called KVSAVE?? (?? indicates 2-digit single-byte characters) and creates a directory with two unique initial characters to save the data.

During the save operation, the name of the file being saved and its progress are displayed.

4 After saving is finished, the name of the saved project folder will be displayed.

Press the "MENU," "ENTER," "◀ ," and "▶" keys to return to the storage menu.

The error code during the execution of load, save

Number	Message	Cause	Countermeasure	At occurrence
2	空き番号がありません Unused number not available 没有空编号	The write in was not possible since there was a file already.	Please configure an empty number to the folder number to save.	Save
3	オートローディング失敗 Auto Loading Failed 自动加载失败	The folder number designated at CM was out of range during the execution of auto load or run load.	Please designate number from 0 to 99.	Load/Save
4	ロードできません Cannot load data 无法加载	The folder of CM designated to memory card/CPU memory does not exist.	Please confirm that project data is inside.	Load
5	メモリカードがありません ---------- CPUメモリアンマウント Memory card not found ---------- CPU Memory unmounted 没有存储卡 ---------- CPU存储器卸载	The load or save was executed during the condition without being able to recognize memory card/ CPU memory.	Please insert the memory card when loading from the memory card, or saving to memory card. Please mount the CPU memory.	Load/Save
6	アクセス競合 Access conflict 访问冲突	The save/load was not possible since transfer or read out of the program was done from another route.	Please execute save/load after the transfer or read out of the program is complete.	Load/Save
7	オートローディング失敗 Auto Loading Failed 自动加载失败	The autoload or run load was executed towards PLC with write protect configured.	Please execute after the write protect of PLC is released.	Load
9	CPUユニット機種不一致 PLC model mismatch CPU单元机型不一致	It tried to read in to the memory card file of the separate device model.	Please confirm the CPU unit model of project.	Load
10	非対応のCPU機能バージョン Unsupported Function Version 不支持的功能版本	It tried to read in function version that does not correspond or the project of separate model.	Please confirm the CPU unit model of the project.	Load
11	PRGモード時のみ可能です PRG mode only 仅PRG模式时可执行	It tried to load with RUN mode.	Please switch to PRG mode to load.	Load
12	空き容量がありません No free space 没有可用空间	There are no free space in the memory card/CPU memory.	Please increase the free space.	Load/Save
15	プロジェクト領域サイズオーバー Project area size over 项目区域容量超出	The write in of CPU memory capacity configuration has failed (project domain is insufficient).	Please review the configuration of CPU memory capacity configuration or reduce data of project domain inside the CPU.	Load
16	ユーザメモリ領域サイズオーバー UserMem area size over 用户存储器区域容量超出	The write in of CPU memory capacity configuration has failed (user memory domain is insufficient).	Please review the configuration of CPU memory capacity configuration or reduce data of user memory domain inside the CPU.	Load
17	スイッチがPRG時のみ可能です Only when switch in PRG 仅开关处于PRG时可执行	Tried to upgrade the version of system program when the switch was in RUN mode.	Turn the switch to PRG mode then execute version upgrade of the system program.	Load

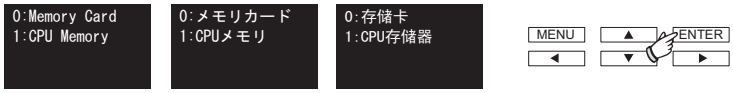
Number	Message	Cause	Countermeasure	At occurrence
18	ユニット種別不一致 Unit Type　Mismatch 单元类型不一致	YTC file has loaded which is different kind of unit compared to the connected unit.	Please check unit type of YTC file.	Load
19	オールクリアが必要 All Clear Reqred Before loading 加载前　需全部清空	An attempt was made to load a project that requires all clear in advance. If the version of the project in the PLC and the project to load do not match, all clear must be executed before loading the project.	Execute all clear and then load the project. If you are loading from the CPU memory, back up the data beforehand as the data to load will also be cleared.	Load
20	ユニット非対応 機能バージョン Unsupported Unit Function Version 不支持的单元功能版本	An attempt was made to read save data with a function version not compatible with the expansion unit.	Check the function version of the expansion unit in the project. Check the function version of the expansion unit.	Load
100	メモリカードカバーオープン Memory card cover open 存储卡外盖打开	The memory card cover is opened.	Please close the memory card cover.	Load/Save
101	メモリカードがLOCKされています Memory card locked 存储卡被锁定	The memory card is in LOCK status.	Please release LOCK or use memory card that is writable.	Save
102	ロードできません ---------- セーブできません Cannot load data ---------- Cannot save data 无法加载 ---------- 无法保存	The memory card is damaged.	Please check the file system of the memory card.	Load ---------- Save
103	ファイル数上限を超過しました Max no. of files exceeded 超过了文件数上限	The sum of number of files and folders inside the user memory has gone over the upper limit.	Please reduce the file or folder inside the user memory.	Load/Save
104	フォルダ階層が深すぎます Folder layer too deep 文件夹层数过多	The folder hierarchy inside the user memory is too deep.	Please confirm the folder hierarchy inside the user memory.	Load/Save
1000	ロードできません Cannot load data 无法加载	• The file is damaged. • The memory card is damaged. • The card has been extracted during the loading process. • It tried to load from CPU memory with the configuration "Load after clearing the user memory". • It tried to execute load from CPU memory with methods other than the autoload, run load with the configuration "Load after all clear".	Please confirm if the project and memory card is appropriate.	Load
1001	セーブできません Cannot save data 无法保存	• The memory card has been damaged. • The card has been extracted during save process.	Please confirm if the project and memory card is appropriate.	Save

Reference The load / save result can also be confirmed on the CM.

● File Management

1 Please select "File Management" from the menu of storage and press "ENTER" key.

A message appears prompting you to select storage device, either a memory card or CPU memory.



2 Use "▲" and "▼" to select a storage device, then press the "ENTER" key or the "▶"key.

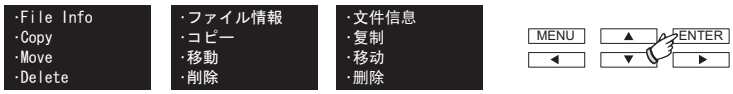
The (project) folder and files in the root folder appear.

Reference

- Long file names and folder names scroll across the display.
- Folder names are displayed as "<D>KVSAVE01"where the first three characters are displayed as "<D>."
- Please use "▶" key or the "◀" key when moving the folder.
- "<D>." represents the current folder and "<D>.." represents the folder that is one hierarchy above.

3 Select folder and file with "▲" "▼" and press "ENTER" key.

The Operation list appears.



4 Select an operation by pressing "▲" and "▼", then press the "ENTER" key or the "▶" key.

The Operation list display appears.

"File information" (page 5-145)
"Copy" (page 5-146)
"Move" (page 5-148)
"Delete" (page 5-150)

Press the "MENU," "ENTER," "◀ ," and "▶" key to return to the file list display.

- **File information**

1 Check file information.

Reference

Scroll to display long file names.

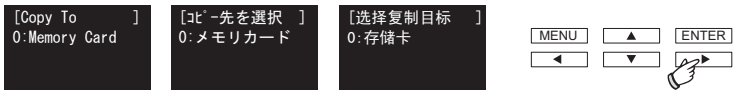
- **Copy**

1 Press the "ENTER" key or the "▶" key.

The operation guide screen will be displayed.

2 Press the "▶" key after confirming the operation.

The folder selection screen of the copy destination will be displayed.



- Please use "▶" key or the "◀" key when moving the folder.
- "<D>." represents the current folder and "<D>.." represents the folder that is one hierarchy above.

Reference You can only copy data between different storage devices, for example, between memory card and CPU memory, or vice versa. (You cannot copy data to the same storage folder.)

3 Press the "ENTER" key.

- Please use "ENTER" / "▶" key or the "MENU" / "◀" key when moving the folder.

"<D>." represents the current folder and "<D>.." represents the folder that is one hierarchy above.

4 Copy the file by pressing the "ENTER" key for 1 second or longer.

A message asking you to confirm that you want to perform the copy operation is displayed.

Point

- **A file/folder with the same name as the source is displayed in the screen shown below.**
  It will be overwritten when the "ENTER" key is pressed for 1 second.
  Copying is canceled by pressing the "MENU" key or "◀" key.

- **To abort an ongoing copy process, press the "MENU" key.**

In the screen shown below, hold down the "ENTER" key for 1 second or longer to cancel copying.

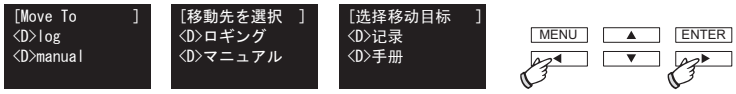
- **Move**

1 Press the "ENTER" key or the "▶" key.

The operation guide screen will be displayed.

2 Press the "▶" key after confirming the operation.

A screen for selecting a folder at a move destination appears.



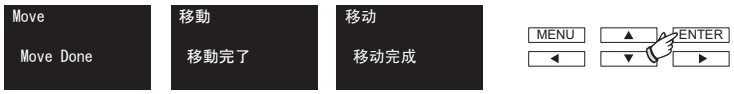
- Please use "▶" key or the "◀" key when moving the folder.
- "<D>." represents the current folder and "<D>.." represents the folder that is one hierarchy above.

Reference The copy can only be executed towards the storage that differs such as memory card→CPU memory, or CPU memory→memory card (It can not copy to the same storage folder).

3 Select a folder by pressing "ENTER"/"▶" key or the "MENU"/"◀" key.

4 Press the "ENTER" or "▶" for 1 second or longer and move the file.

A message asking you to confirm that you want to perform the move operation is displayed.



Point

- **A file/folder with the same name as the source is displayed in the screen shown below.**
  Press the "ENTER" key or "▶" key for 1 second or more to overwrite.
  The move is canceled by pressing the "MENU" key or "◀" key.

- **To abort an ongoing move process, press the "MENU" key.**

In the screen shown below, hold down the "ENTER" for 1 second or longer to cancel a move operation.

- **Delete**

1 **Keep on pressing the "ENTER" key for 1 second or longer.**

2 **The deletion has completed.**

Reference Deleting a selected folder will also delete all the files and folders it contains.

● Free space

1 **Select "Free space" in the storage menu, and then press the "ENTER" key.**
A message appears prompting you to select a storage device, either memory card or CPU memory.

2 **Use "▲" and "▼" to select a storage device, then press the "ENTER" key or the "▶" key.**
The free space on each storage device will be displayed.

Access Window Setup

The following two settings in the access window can be changed.

English	Japanese	Chinese	Setting
Contrast	コントラスト	对比	LCD contrast adjustment. (1 to 11)
Comment Number	コメント番号	注释编号	Specify the number of the comment you want to display. (Comment 1 to comment 8)

■ Contrast

1 Press the "MENU" key.

The MENU appears. (If another unit is selected, press the "MENU" key twice.)

2 Select "9. AW Setting" by pressing "▲" and "▼", then press the "ENTER" key.

The Access window setup menu appears.

3 Select a "Contrast" by pressing "▲" and "▼", then press the "ENTER" key.

4 Scroll the contrast values by pressing the "▲" and "▼" keys to set an appropriate contrast value.

Contrast can be set to between 1 and 11.

■ Comment Number

1 Press the "MENU" key.

The MENU appears. (If another unit is selected, press the "MENU" key twice.)

2 Please select "9.AW Setting" with "▲" "▼" key and press "ENTER" or "▶" key.

The Access window setup menu appears.

3 Select a "Comment number" by pressing "▲" and "▼", then press the "ENTER" key.

4 Hold down the "ENTER" key or the "▶" key for 1 second or more.

This allows you to switch comment numbers.

5 Select by pressing the "▲" and "▼" key, then hold down the "ENTER" key.

The comment number will switch.

CLEAR

There are RAM clear, all clear, user memory clear for clearing.

■ RAM clear

You can use RAM clear to clear device values, alarm logs, error logs, PLC event log, device value change log and current bank numbers of the file register at the same time. It also returns access window settings to their defaults. A RAM clear initializes R, MR, LR, B, CR, T/C (current values), CM, DM, EM, FM, ZF, W and Z. Ladder programs, unit setup information, comments, T/C (set values), and logging setup information will not disappear.

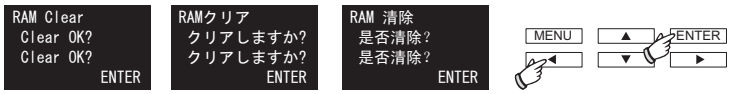
Point **The RAM clear can only be executed during PRG mode.**

1 Press the "MENU" key.
The MENU appears. (If another unit is selected, press the "MENU" key twice.)

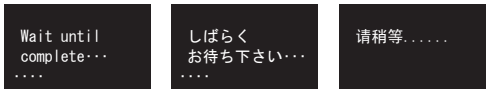
2 Select "10. Clear" by pressing "▲" and "▼", then press the "ENTER" key or the "▶" key.
The Clear menu appears.

3 Select "RAM Clear" by pressing the "▲" and "▼" keys.
A message asking you to confirm that you want to perform the RAM Clear operation is displayed.

4 Hold down the "◀" key while pressing the "ENTER" key for 1 second or more.
The confirmation message is displayed again.



5 When the second confirmation is displayed, hold down the "ENTER" key for 1 second to clear RAM.
During a RAM clear, the following screen is displayed. When the RAM clear operation ends, the unit is automatically restarted.



Point **If you do not hold down the "ENTER" key for 1 second or more, the message "Wait..." will not be displayed, and the operation will be ignored.**

■ All Clear

The all clear mode returns the CPU unit settings to their default values. (Ladder clear, device all clear)

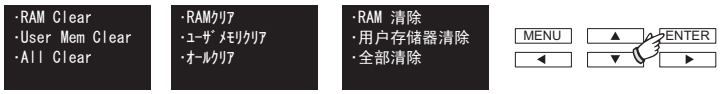
Point **The all clear can only be executed during PRG mode.**

1 Press the "MENU" key.

The MENU appears. (If another unit is selected, press the "MENU" key twice.)

2 Select "11. Clear" by pressing "▲" and "▼", then press the "ENTER" key or the "▶" key.

The Clear menu appears.

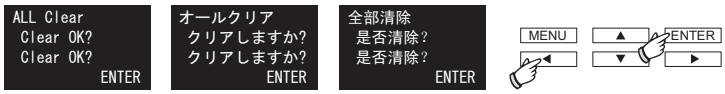


3 Select "All Clear" by pressing the "▲" and "▼" keys.

A message asking you to confirm that you want to perform the All Clear operation is displayed.

4 Hold down the "◀" key while pressing the "ENTER" key for 1 second or more.

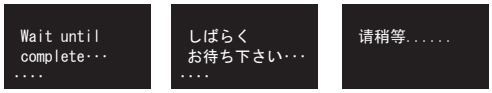
The all clear will be cancelled when the "MENU" key is pressed, and returns to the menu display.



The all clear will be cancelled when the "MENU" key is pressed, and returns to the menu display.

5 When the second confirmation is displayed, hold down the "ENTER" key 1 second or more to perform an All Clear operation.

During an All Clear, the following screen is displayed. When the All Clear operation ends, the unit is automatically restarted.



Point **If you do not hold down the "ENTER" key for 1 second or more, the message "Wait..." will not be displayed, and the operation will be ignored.**

6 The No Ladder Program of CPU error 40 will be displayed after restarting.

■ User Memory Clear

The data such as logging data, user document etc. stored inside the user memory will be cleared during user memory clear. In addition, ladder program, unit configuration information, comment, T/C (configured value), logging configuration information will not be deleted.

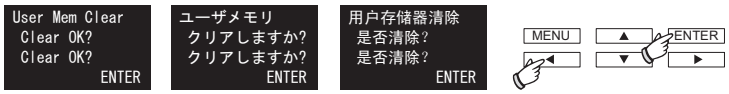
Point **The user memory clear can only be executed during PRG mode.**

1 Press the "MENU" key.
The MENU appears. (If another unit is selected, press the "MENU" key twice.)

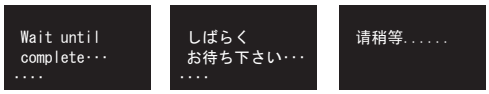
2 Select "10. Clear" by pressing "▲" and "▼", then press the "ENTER" key or the "▶" key.
The Clear menu appears.

3 It selects "User Memory Clear" with the "▲" "▲"key.
The user memory clear confirmation message will be displayed.

4 Hold down the "◀" key while pressing the "ENTER" key for 1 second or more.
The confirmation message is displayed again.



5 The user memory clear will be executed when the "ENTER" key is pressed for 1 second or longer, while the 2nd confirmation screen being displayed.
It will display the next screen while user memory clear is being executed. The completion message will appear, when the user memory clear is executed and completes.



Point **If you do not hold down the "ENTER" key for 1 second or more, the message "Wait..." will not be displayed, and the operation will be ignored.**

Access Window Settings Used by KV STUDIO

You can use window settings from KV STUDIO to make the following settings.

- Display language setting
- Display comment number setting
- Backlight OFF time setting
- User message setting

■ Display language settings

You can select Japanese, English or Chinese (simplified characters) as the display language of the access window.

Once this setting is enabled, the display language will use the KV STUDIO setting at power on.

Point

- **The access window allows you to change the language setting, but it will revert to the KV STUDIO setting the next time the system is turned on.**
- **You can change the default display language setting by changing the CM value and turning the power off and back on again.**
  (When the display language setting is made both in KV STUDIO and CM, the KV STUDIO setting takes precedence.)

■ Display comment number setting

You can set to display one device comment from comment 1 to comment 8 in the access window.

Once this setting is enabled, the display language will use the KV STUDIO setting at power on.

If you change the set value of the display comment number setting, the comments will change when the following are displayed.

- Device monitor comment display
- Alarm monitor comment display
- User message

Point

- **The access window allows you to change the display comment number setting, but it will revert to the KV STUDIO setting the next time the system is turned on.**
- **You can change the default display language setting by changing the CM value and turning the power off and back on again.**
  (When the display comment number setting is made both in KV STUDIO and CM, the KV STUDIO setting takes precedence.)

■ Backlight OFF time setting

This setting determines the period of inactivity in the access window after which the backlight will be turned off.

You can set the following time periods: 1 min, 3 min, 10 min, 30 min, 60 min and always ON (the default value).

Point

- **An interrupt screen that appears will temporarily turn on the backlight (if OFF). It will then go OFF after the set time has elapsed.**
- **Touching the keys when the backlight is OFF will turn it back ON.**
- **The backlight is not turned OFF in PRG mode.**
- **The backlight OFF time setting cannot be changed from the access window.**

■ User message setting

You can use a list in KV STUDIO to set the messages you want to display in the access window.

When a control relay (CR) for a user message is turned on, the message set by the message editor in the access window is displayed.

"User Message" (page 5-161)

Access Window Initial Menu Display Function

When this function is used, the access window menu displayed when powered on may be set via ladder diagram program.
If this function is not used, the menu displayed upon the previous power off will be displayed.

■ Display method

When powered on, value may be saved in control memory (CM) from the ladder diagram program, select initial menu of access window when powered on.
When access window initial menu display function is used, save 1 in CM1780, set the menu displayed on CM1781-CM1788.

Reference Decimal number is represented by #, hexadecimal number is represented by $.

■ Control memory list for access window initial menu display

Device No.	Function
CM1780	Select enable, disable of access window initial menu display function (0: disable, 1: enable)
CM1781	Select device type*1
CM1782 CM1783	Select display menu
CM1784 CM1785	Select device No.*2
CM1786	Select display format of the device*1
CM1787	Select device name representation*1 (0: device No., 1: device comment)
CM1788	Select key lock state (0: key lock is disabled, 1: key lock is enabled)

*1 Type of the display menu is valid only in device mode.
*2 Type of the display menu is valid only in device mode, or trimmer mode.

● Select enable, disable of access window initial menu display function

Select whether the function is used or not according to the values stored in CM1780.

CM1780 = 1: display the menu selected by CM1781-1789.

CM1780 = 0: display the menu displayed upon the previous power off.

● Selection scope of the device type (type of the display menu is valid only in device mode)

Select the device type to be displayed according to the values stored in CM1781.

Device type	CM1781(hexadecimal)	Device type	CM1781(hexadecimal)
R	$0000	LR	$0012
CR	$0001	EM	$0017
T	$0002	FM	$0019
C	$0003	B	$001B
DM	$0006	W	$001C
CM	$0007	ZF	$002C
TM	$0008	Z	$0030
MR	$0011		

● Type and selection scope of the display menu

Select the type of display menu according to the values stored in CM1782 and CM1783.

Menu	CM1782, CM1783 value (hexadecimal)
Top menu	$00000000
Device mode	$01000000
Trimmer	$03000000
Display clock	$05000000
CPU monitor menu	$06000000
CPU monitor project name	$06100000
CPU Monitor Version	$06200000
Alarm menu	$07000000
Alarm monitor	$07100000
Alarm log	$07200000
Alarm elimination	$07300000
Storage menu	$08000000
AW setting menu	$0A000000
AW setting comparison setup	$0A100000
Clear menu	$0C000000

● Selection method of device No. (type of the display menu is valid only in device mode or trimmer mode)

Select the device No. To be displayed according to the values stored in CM1784 and CM1785.

- Number across channels, select continuous number for R, MR, LR, CR.

Calculation expression is as follows.

aaa: Channel number (000 to 999)

aaabb

bb: Contact number (00 to 15)

N:Values stored in CM1784 and CM1785 (decimal numbers)

aaa:Channel No.

bb:Contact No.

N = 16 x aaa + bb

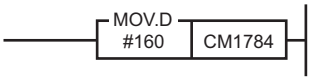
Examples

When R1000 is selected

aaa = 10, bb = 0

N = 16 x 10 + 0 = 160

The value of CM1784 and CM1785 is 160 ($A0).



When R10011 is selected

aaa = 100, bb =11

N = 16 x 100 = 11 = 1611

The value of CM1784 and CM1785 is 1611 ($064B).

- For B, W, use the displayed device No. value (hexadecimal number) to select.

Examples

When B1000 is selected

The value of CM1784 and CM1785 is 4096 ($1000).

When W1FFF is selected

The value of CM1784 and CM1785 is 8191 ($1FFF).

- For T, C, DM, CM, TM, EM, FM, ZF, use the displayed device No. value (decimal number) to select.

Examples

When DM1000 is selected

The value of CM1784 and CM1785 is 1000 ($3E8).

When DM8191 is selected

The value of CM1784 and CM1785 is 8191 ($1FFF).

● Device display format and selection scope

According to the values stored in CM1786, select display format of the device.

Display format	CM1786 value (hexadecimal)	Display format	CM1786 value (hexadecimal)
Default*1	$0000	Decimal Number 32BIT	$0006
Binary Digit 1BIT	$0001	±Decimal Number 32BIT	$0007
Decimal Number 16BIT	$0002	Hexadecimal 32BIT	$0008
±Decimal Number 16BIT	$0003	ASCII16BIT	$0009
Hexadecimal 16BIT	$0004	FLOAT	$000A
Binary Digit 16BIT	$0005	DOUBLE FLOAT	$000B

*1 Bit device is represented by ON/OFF, word device is represented by unsigned 16Bit-BASE10.

■ Default display language setting

Change the default language setting to suit the numeric values stored in CM1736.

1: English (default)

2: Japanese

3: Chinese (simplified characters)

Settings other than the above are disabled.

Point **A KV STUDIO access window setting takes precedence of other display language settings.**

■ Default display comment number setting

Enter the number of a comment stored in C1737 to change the default comment setting.

1 to 8: Comment 1 to comment 8

Settings other than the above are disabled.

Point **When setting the default display comment number in the access window setting of KV STUDIO, the setting of KV STUDIO takes precedence.**

■ Display examples

1. When power on is selected, top menu is displayed in the access window.
   Save the following settings in CM.

```
1.Device Mode
2.Error Clear
3.Unit Test
4.Storage
```

```
1.デバイスモード
2.エラークリア
3.ユニットテスト
4.ストレージ
```

```
1.软元件模式
2.错误清除
3.单元测试
4.存储
```

Device No.	CM value (hexadecimal)	Set content
CM1780	$0001	AW initial menu display is enabled
CM1781	-	-
CM1782 CM1783	$00000000	Top menu
CM1784 CM1785	-	-
CM1786	-	-
CM1787	-	-
CM1788	$0000	Key lock is invalid

2. When power on is selected, device mode, R200 ON/OFF are displayed in the access window.
   Save the following settings in CM.

```
R200
          OFF
R201
           ON
```

Device No.	CM value (hexadecimal)	Set content
CM1780	$0001	AW initial menu display is enabled
CM1781	$0000	R
CM1782 CM1783	$01000000	Device mode
CM1784 CM1785	$00000020	200
CM1786	$0000	Default
CM1787	$0000	Device numbering
CM1788	$0000	Key lock is invalid

User Message

The user message function is a function for displaying messages created by the message editor consisting of up to 64 single-byte (32 double-byte) characters in the access window.
User messages can use up to 63 control relays and character strings can be changed just by turning on a control relay. The displayed message can be changed simply by turning the CR on. Furthermore, you can switch between comments 1 to 8 and use the user messages in up to 8 languages using the display comment number settings.

■ How to Display User Message

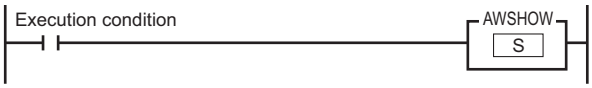
There are two ways of displaying user message.

(1) Using the AWSHOW instruction.

The user message can be easily displayed without worrying about control relay (CR) by using the exclusive instruction term.
When user messages with multiple IDs are ON simultaneously, the user message with the lowest numerical value is displayed.

For details on instructions, refer to the "KV-8000/7000/5000/3000/1000 and Nano Series Instruction Reference Manual", "5-14 Access Window Instructions."

Ladder program

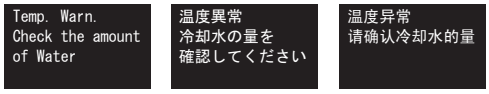


Entry method

(2) Create a ladder program using the control relay (CR).

To display the user message for ID 1, turn on CR 4601.
When CR4601 to CR4915 are ON, the user messages with corresponding IDs are displayed.

■ Display example

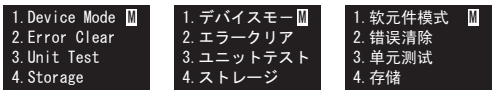


When displayed, rows set using CR4500 - CR4503 start to flash.
The screen will return to original display in the following cases.

- When the "ENTER" key in the Access Window is pressed. When pressed, the corresponding CR will turn OFF.
- When the corresponding CR turns OFF.
- When the AWHIDE instruction is executed.

Point

- **You can set the user message ID that is displayed at power on.**
- **You can determine whether or not pressing the "ENTER" key during a user message display will close the user message.**
- **You can use CM1720 to check the currently displayed user message ID.**
- **Pressing the "MENU" key when a user message is displayed will display "M" to the right of the access window.**



Menus can be accessed while a user message is displayed, but focus will automatically revert to the user message if you do not perform any key operations for 30 seconds.

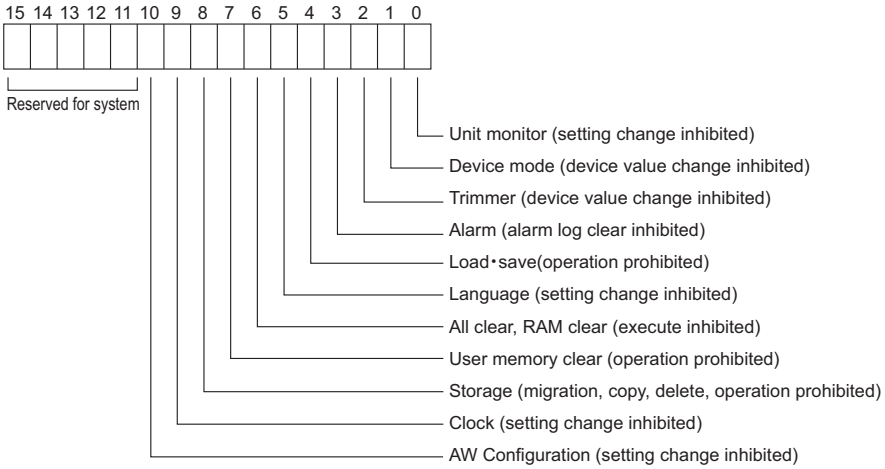
- **If a CR is set for multiple user message IDs, scroll buttons (▲ / ▼) will appear on the right side. While the scroll buttons are displayed you can press the "▲ " and " ▼" keys to display user messages for other IDs.**

Operation Inhibited Setting

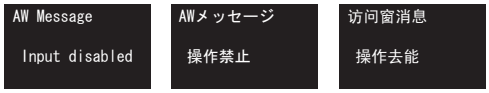
The operations on Access Window can be separately inhibited according to different functions.
The functions and projects of operation inhibited setting are as follows.

Function	Item
Unit monitor	Change settings
Device mode	Change device value
Trimmer	Change device value
Alarm	Alarm log clear
Load·Save	Load, save
Language	Change settings
All Clear/RAM Clear	Execute
User memory clear	Execute
Storage	Move/copy/delete file
Clock	Change settings
Switch comment	Change settings

Operation inhibited setting is used to assign 1 bit to CM1738 for each function, which will be inhibited when the bit is ON.

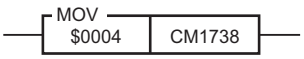


The following screen is displayed when an attempt is made to perform an inhibited operation.

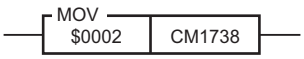


Example

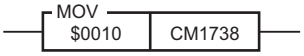
●Write inhibited setting of digital trimmer



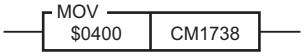
●Write disable setting in inhibited mode



●Load/Save Prohibited



●Setting change inhibited set by Access Window



Point **In device mode, when device write inhibited is set, it can not be released in the Access Window.**

Key Lock Function

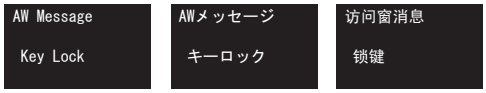
Key lock function is used to inhibited operations in Access Window. This is used when you intend to make the settings unchanged, or fix the display screen.

■ How to operate

Hold down the "◀" key and "MENU" key simultaneously for 3 seconds or longer to lock or unlock the keys.

● Incase of key unlock

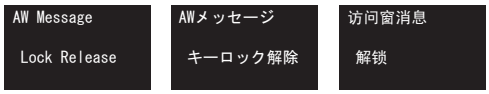
Key lock state is entered when the following screen is displayed.



Under the state of key lock, this display remains unchanged irrespective of any operation, unless you release key lock.

● Incase of key lock

Key lock will be released when the following screen is displayed.



Point

- **During the process of key lock, operation is disabled even if direct access switch is pressed.**
- **The key lock is not possible during unit monitor, unit test.**

MEMO

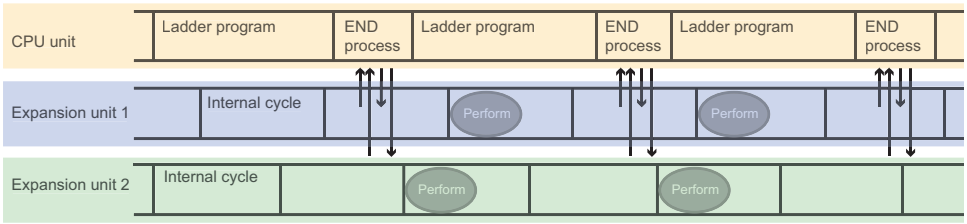
5-10 Inter-unit Synchronization Function

What is the Inter-unit Synchronization Function

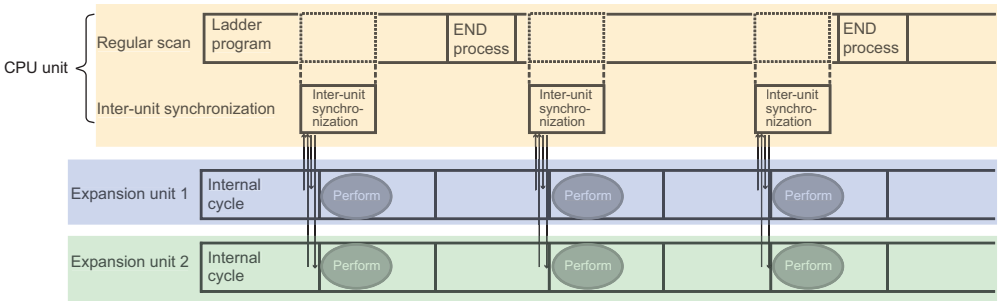
"Inter-unit synchronization function" is a function for adjusting the control timing between multiple units. By using this function, the input or output timing of the units subject to synchronization can be synchronized with the inter-unit synchronization cycle. The inter-unit synchronization cycle is the interval of the fixed inter-unit synchronization function cycle.
By using this function, simultaneous sampling between multiple units and simultaneous positioning can be undertaken.

Performance image for inter-unit synchronization

In case when inter-unit synchronizing function is not being used



In case when inter-unit synchronizing function is being used



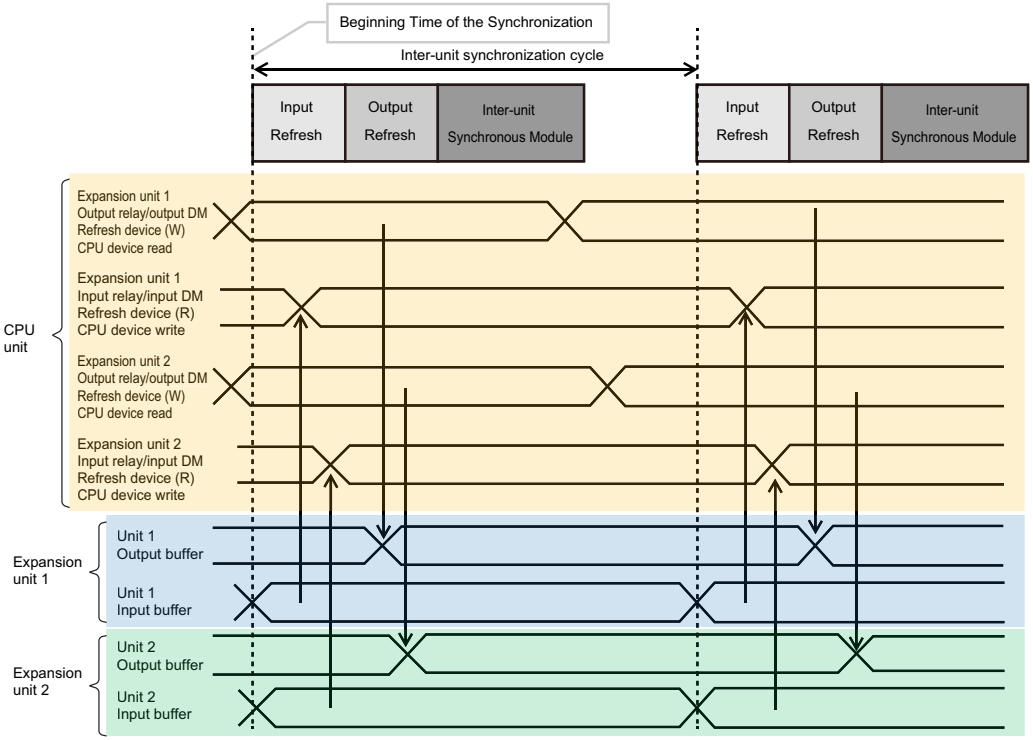
In order to adjust the start timing of the internal period by tuning internal period of each expansion unit, synchronized control in variation under 1 μs and synchronized sampling can be done between expansion units.
The method for adjusting beginning time and sampling time differs between each unit.
For more information, see the user manual for each unit.

Point

- **The expansion unit for the KV-8000/7000 (S-Unit, X-Unit) is the only unit supported by the inter-unit synchronization. The expansion unit for the KV-5000/3000 is not supported.**
- **As inter-unit synchronization function, output control is executed after inter-unit synchronization period. If high speed output is desired after operation, the direct refresh command should be used, not the inter-unit synchronization function.**
- **Inter-unit synchronization module or every-scan execution module (standby module) can be selected in "Scantime-related settings" of "CPU system settings".**

Inter-unit Synchronous Module

The inter-unit synchronous module performs an output refresh and input refresh for each unit after beginning inter-unit synchronization (there are cases when it does not, depending on the settings). The ladder program within the inter-unit synchronous module will be performed afterwards.
In case of not using direct updating command, items modified in inter-unit synchronization module will be reflected in the next execution of inter-unit synchronization.
Performance commands and such will be run based on the beginning time of the inter-unit synchronization.

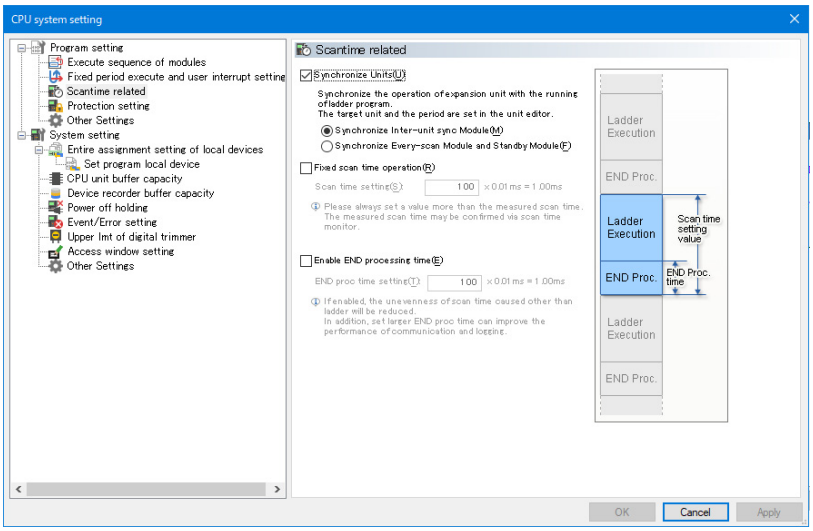


Point

- **The value read out at the time of input refreshing can read out the data of the same timing between units, based on the synchronization beginning timing. Please see the manual for each expansion unit, as the timing for reading out differs between units.**
- **Value (command) programmed at the time of output updating will be executed in the same timing based on synchronization start timing.**
- **Data refreshed by the inter-unit synchronization module input refresh and output refresh differs for each unit. Some expansion units have no refresh device (inter-unit synchronization refresh) or CPU device write/read function. For details, refer to the expansion unit manual.**
- **When the inter-unit synchronization refresh timing is set to "Batch (END processing)", it is not refreshed at inter-unit synchronization I/O refresh. However, even if the refresh timing is set to "Batch (END processing)", the units set with the refresh device (inter-unit synchronization refresh) or CPU device write/read function are always refreshed at the inter-unit synchronization I/O refresh.**

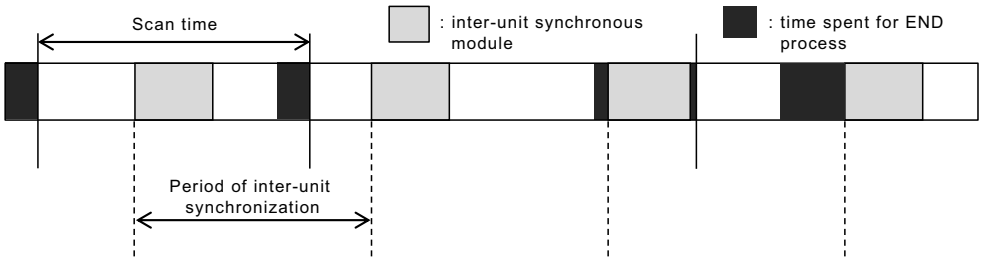
Operation When the Inter-Unit Synchronization Module is Not Used

If "Synchronize every-scan execution modules and standby modules" is checked in "Scantime related" in CPU system settings, inter-unit synchronization can be executed without using the inter-unit synchronization module. This is because every-scan execution modules/standby modules are executed in the inter-unit synchronization cycle.



Using the inter-unit synchronization module

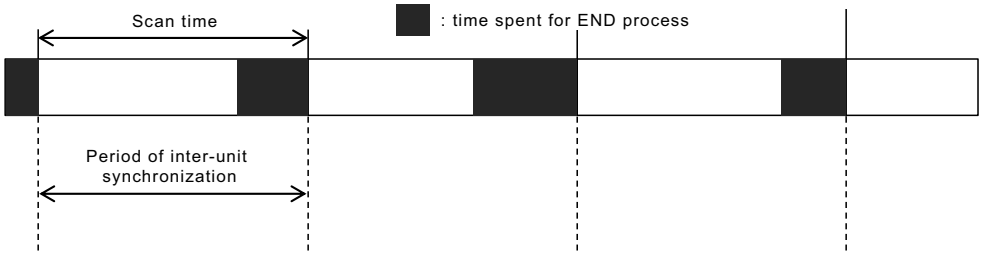
When "Synchronize inter-unit sync module" is selected



The inter-unit synchronization module is executed as interrupt processing in every inter-unit synchronization cycle.

Not using the inter-unit synchronization module

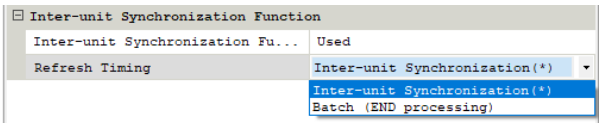
When "Synchronize every-scan execution modules and standby modules" is selected



Every-scan execution modules/standby modules are executed every inter-unit synchronization cycle. In other words, CPU unit program processing is executed with the scan time equal to the inter-unit synchronization cycle.

Refresh Timing Settings (For Expansion unit relays and DM only)

When using inter-unit synchronization, output refresh and input refresh for each unit's I/O relays and I/O DM can be executed in the normal program execution END processing by setting "Refresh timing" to "Batch (END processing)" in the unit settings for the expansion unit.

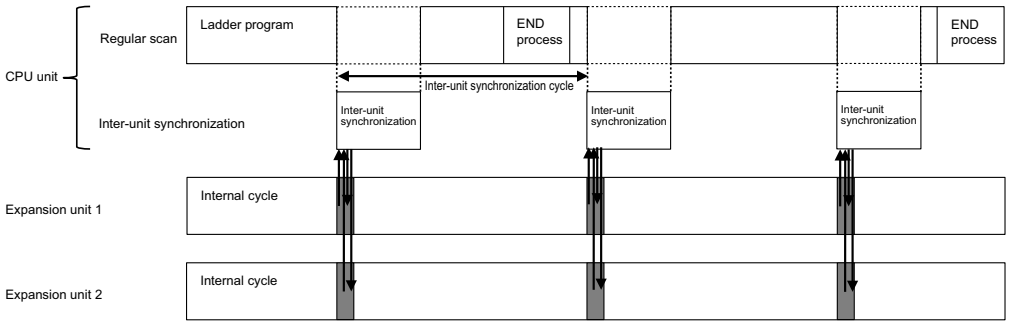


Point

Refresh timing settings are only valid for expansion unit relays/DM data. Units set with the refresh device (inter-unit synchronization refresh) or CPU device write/read function are always refreshed at inter-unit synchronization I/O refresh.

When the refresh timing is set to "Inter-unit synchronization"

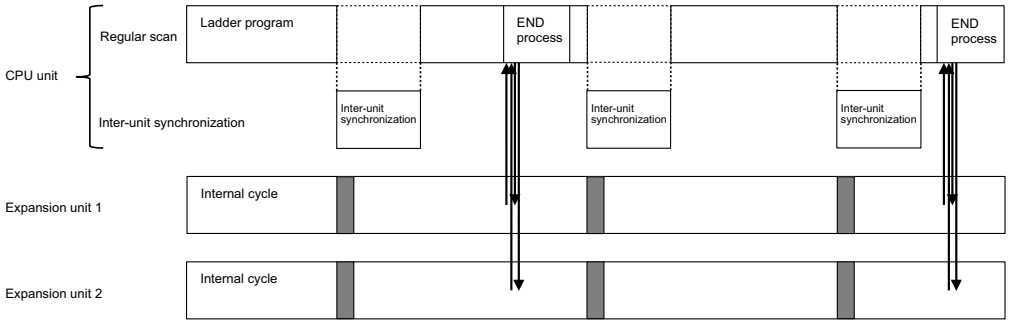
The expansion unit relays/DM are refreshed (written/read) when inter-unit synchronization execution starts.



The inter-unit synchronization module is executed as interrupt processing in every inter-unit synchronization cycle.

When the refresh timing is set to "Batch (END processing)"

The expansion unit relays/DM are refreshed (written/read) in END processing.



Point

The inter-unit synchronization module execution time can be shortened by setting the expansion unit setting for relays and DM that do not require refreshing when executing inter-unit synchronization to "Batch (END processing)". However, the refresh status of expansion unit relays and DM that are set to "Batch (END processing)" will stop matching the status of the inter-unit synchronization module status at execution.

Inter-Unit Synchronization Refresh Devices

Depending on the expansion unit, the required functions can be selected from the expansion unit buffer memory and these can be refreshed as the current value of the set data memory (DM) when inter-unit synchronization is executed.

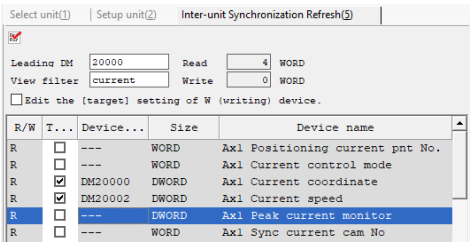
eg. The coordinates and speed of the motion unit can be read to the CPU at inter-unit synchronization, and conversely, data can be written from the CPU unit to the motion unit's sync control phase compensation value at inter-unit synchronization.

Setting the Inter-unit Sync Refresh Devices

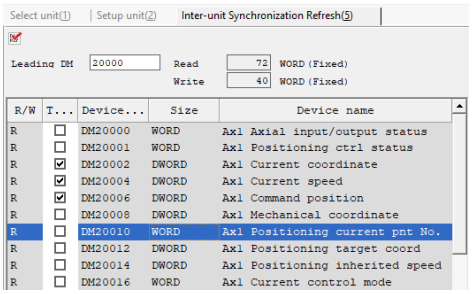
You can set the inter-unit sync refresh devices using the KV STUDIO unit editor.

If the "Inter-unit Synchronization Function" in the unit settings tab of each expansion unit in unit editor is set to "Enable", the "Inter-unit Sync Refresh (5)" tab is displayed. Select the contents you want to read or write to the CPU device in the "Inter-unit Sync Refresh (5)" tab.

For motion unit KV-XH16EC/XH16ML/XH04ML



For pulse positioning unit KV-SH04PL



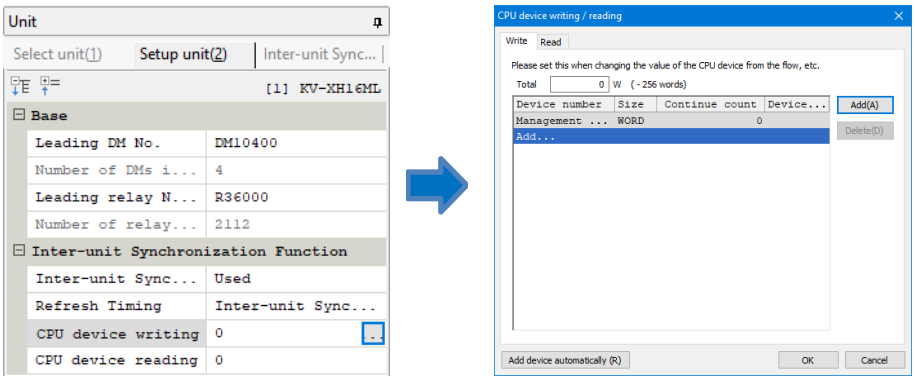
Refer to the manual for each expansion unit for details of the inter-unit synchronization refresh devices.

CPU Device Write/Read

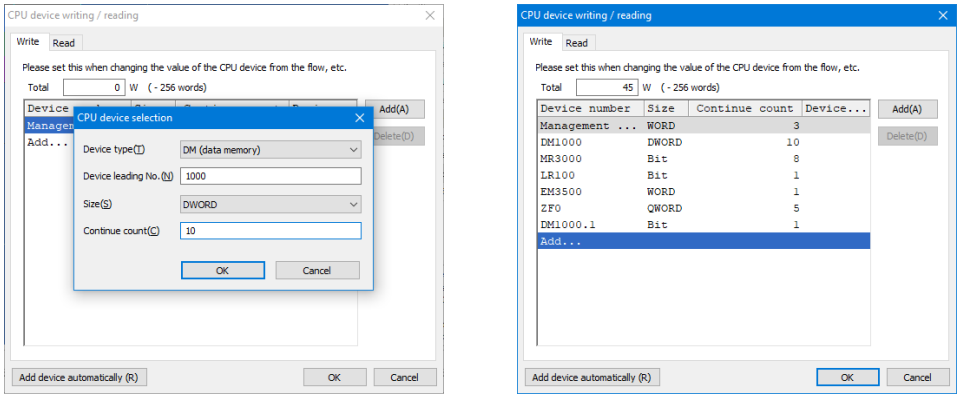
Depending on the expansion unit, you can write data to your chosen device in the CPU unit at inter-unit synchronization refresh, or read data from the CPU unit. You can use this to read data from the CPU unit and execute calculation in the unit program, or write calculation results to the CPU.

Setting CPU Device Write/Read

You can configure the devices that will perform CPU device write/read using the KV STUDIO unit editor. If the "Inter-unit sync function" in the unit settings tab of each expansion unit in unit editor is set to "Enabled", you will be able to select "CPU device write" and "CPU device read".



Click the "Add (A)" button of the "CPU device write/read" dialog box to display the "Select CPU device" dialog box. From there, you can select the device to read from or write to.



Refer to the manual for each expansion unit for details regarding CPU device write/read.

System Configuration of Inter-unit Synchronization

List of supported models

Product name		Model number	Function version
CPU unit		KV-8000(A)	After 1.0
S-Unit	High speed I/O unit	KV-SIR32XT	After 1.0
	High speed analog I/O unit	KV-SAD04/SDA04	After 1.0
	High speed counter unit	KV-SSC02	After 1.0
	High speed positioning unit	KV-SH04PL	After 1.0
X-Unit	Positioning/motion unit	KV-XH16ML/XH04ML/XH64EC/ XH32EC/XH16EC	After 1.0
	Ethernet Unit	KV-XLE02	After 1.0
	serial communication unit	KV-XL402/XL202	After 1.0

Cycle of Inter-unit Synchronization

Cycle that can be specified in the inter-unit synchronization is 40.0 µs to 10000.0 µs, 0.1 ms to 10.0 ms.
Setting is done by "Unit setting" tab of the CPU unit of the unit editor.
Inter-unit synchronization is executed by the specified cycle.

Point

- **If the updating timing of inter-unit synchronization set as "inter-unit synchronization", set the inter-unit synchronization cycle to 125 µs or greater.**
- **In some units, such as in the regular mode of KV-SH04PL or KV-XH16ML/XH04ML/ XH64EC/XH32EC/XH16EC, the inter-unit synchronization cycle must be set in unit control cycle units. In that case, the smallest cycle that can be set for inter-unit synchronization, complies with the control cycle of the unit with the largest control cycle.**

Performance Specifications of Inter-unit Synchronization

The following refresh times will be necessary when performing inter-unit synchronization.
For the inter-unit synchronization cycle, configure a cycle with enough time (5 times + 75 μs or higher) relative to the refreshing time.

N: Unit number
In_R[N]: Number of ch (assumed number of ch) of the interrupting input relay of unit number N
In_DM[N]: Number of DM words (assumed number of DM) of the interrupting input relay of unit number N
Out_R[N]: Number of ch (assumed number of ch) of the interrupting output relay of unit number N
Out_DM[N]: Number of DM words (assumed number of ch) of the interrupting output relay of unit number N
In_Buf[N]: Number of buffer memory words read out from unit number N
Out_Buf[N]: Number of buffer memory words written in from unit number N
In_CPU[N]: Number of CPU device words read out from unit number N
Out_CPU[N]: Number of CPU device words written in to unit number N
T_Ref[N]: Refresh time of the interrupted device of unit number N
T_Buf[N]: Refresh time of the buffer memory and CPU device of unit number N
T_Total: Synchronization refreshing time of all units

Calculation formula of T_Ref[N]

(a)T_Ref[N] = 0
(b)T_Ref[N] = 2.5 + (0.04 × (In_R[N] + In_DM[N] + Out_R[N] + Out_DM[N]))

*(a) If the refresh timing of the unit interruption device is being configured to END process
(b) If the refresh timing of the unit interruption device is being configured to inter-unit synchronization

Calculation formula of T_Buf[N]

(a)T_Buf[N] = 0
(b)T_Buf[N] = 0.04 × (In_Buf[N] + Out_Buf[N]) + 0.1 × (In_CPU[N] + Out_CPU[N])

*(a) If the buffer memory and CPU device are not refreshed
(b) If the buffer memory and CPU device are refreshed

T_Total = T_Ref[1]+T_Buf[1]+ T_Ref[2]+T_Buf[2]+・・・

Examples for using 2 sets of KV-SAD04

Calculation of T_Ref[N] (setting updating period to inter-unit synchronization)
(1st equipment)T_Ref[1]＝2.5＋(0.04 x (3＋21＋3＋32)) ＝4.9 μs
(2nd equipment)T_Ref[2]＝T_Ref[1]
Calculation of T_Buf[N] (not updating buffer memory)
(1st equipment)T_Buf[1]＝0 μs
(2nd equipment)T_Buf[2]＝T_Buf[1]

T_Total = 4.9＋4.9＋0＋0 = 9.8 μs

Point

- **The devices interrupted for in/output of each model are as follows. (Default Value)**

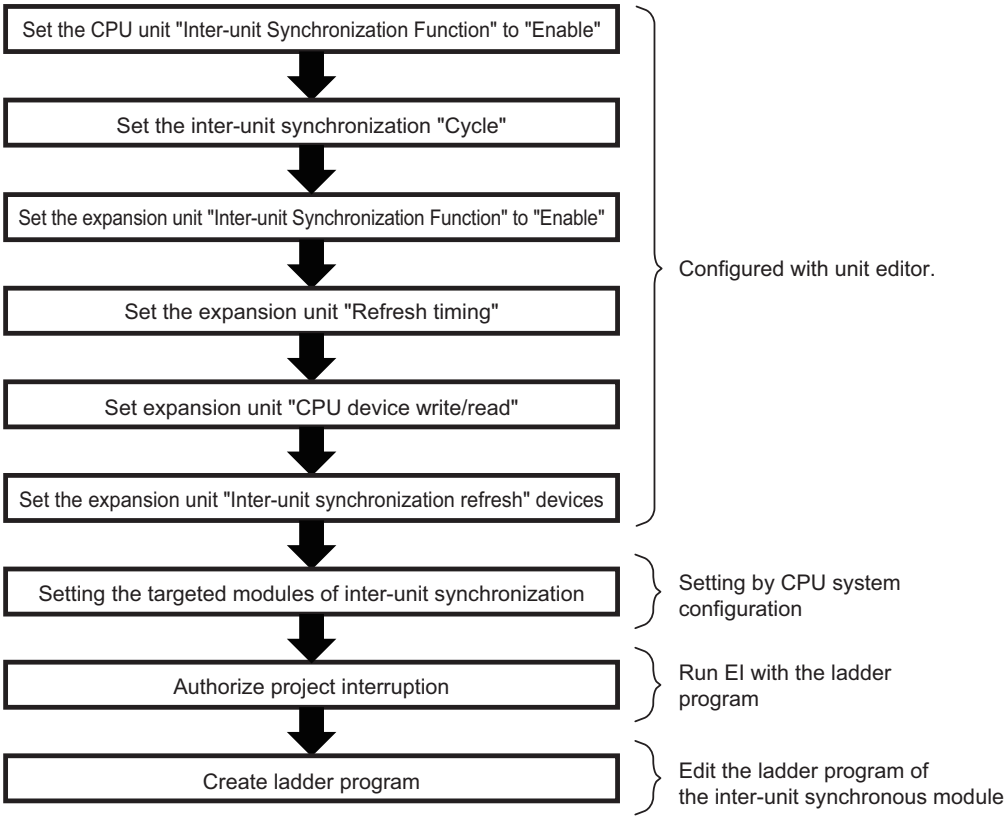
	In_R*1	In_DM*1	Out_R*1	Out_DM*1	In_Buf*2	Out_Buf*2	T_Ref
KV-SIR32XT	2ch (32 points)	0	2ch (32 points)	0	0	0	2.7 µs
KV-SAD04	3ch (48 points)	21	3ch (48 points)	32	0	0	4.9 µs
KV-SDA04	3ch (48 points)	5	3ch (48 points)	24	0	0	3.9 µs
KV-SSC02	2ch (32 points)	29	2ch (32 points)	40	0	0	5.6 µs
KV-SH04PL	10ch (160 points)	60	10ch (160 points)	0	0	0	5.7 µs
KV-XH04ML	18ch (288 points)	0	18ch (288 points)	0	0	0	3.9 µs
KV-XH16ML	66ch (1056 points)	0	66ch (1056 points)	0	0	0	7.8 µs
KV-XH16EC	66ch (1056 points)	0	66ch (1056 points)	0	0	0	7.8 µs
KV-XH32EC	130ch (2080 points)	0	130ch (2080 points)	0	0	0	12.9 µs
KV-XH64EC	258ch (4128 points)	0	258ch (4128 points)	0	0	0	23.1 µs
KV-XLE02*3	20 to 80ch (1280 points)	0	20 to 60ch (960 points)	0	0	0	4.1 to 8.1µs
KV-XL202*3	0 to 26ch (416 points)	0	0 to 26ch (224 points)	0	0	0	2.7 to 4.2µs
KV-XL402*3	0 to 26ch (416 points)	0	0 to 26ch (224 points)	0	0	0	2.7 to 4.2µs

*1 The default value if it is configured to refresh at the time of inter-unit synchronization
*2 Refreshing of the buffer memory is not supposed to be registered in the default value, but the settings can be changed.
*3 The number of occupied relay channels changes according to the unit editor setting. T_Ref changes according to the number of occupied channels.

- **If the settings below are used when configuring inter-unit synchronization, the time until refreshing is completed increases, and there is a possibility that the process will not be completed during the inter-unit synchronization cycle. When using with the following conditions, leave room for adjustment when configuring the cycle of inter-unit synchronization.**
  - **A unit for the KV-5000/3000 series is used (14 µs delay at most)**
  - **Interrupt using DI/DIC instructions is prohibited**

Setting Method for Inter-unit Synchronization

The following settings will be necessary to perform inter-unit synchronization.

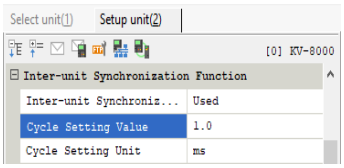


Setting Steps for Inter-unit Synchronization

This section describes the procedure for connecting two KV-XH16ML to CPU unit and executing Inter-unit synchronization.

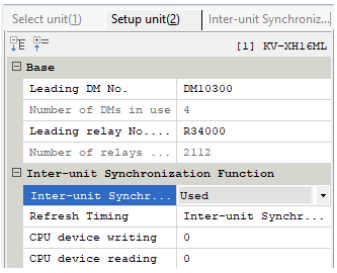
1 The KV-8000(A) will be configured using the unit editor.

Set the "Inter-unit Synchronization Function" in the KV-8000(A) unit settings to "Enable", and set the inter-unit synchronization "Cycle".
(For the next example, it will be configured to 1.0ms)



2 The KV-XH16ML will be configured using the unit editor.

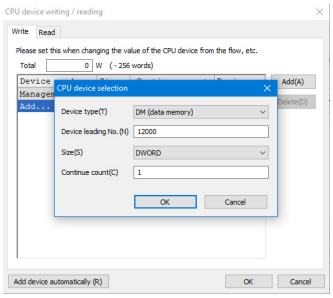
Set the inter-unit synchronizing function to "use this" by using KV-XH16ML unit setting (2) tab.



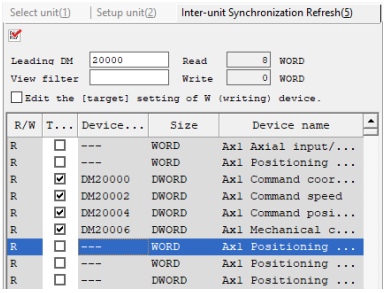
Reference You can also set the updating timing to "END processing", in addition to "inter-unit synchronization".

Set the CPU unit device write/read and inter-unit synchronization refresh devices as necessary.

"CPU unit device write/read" setting



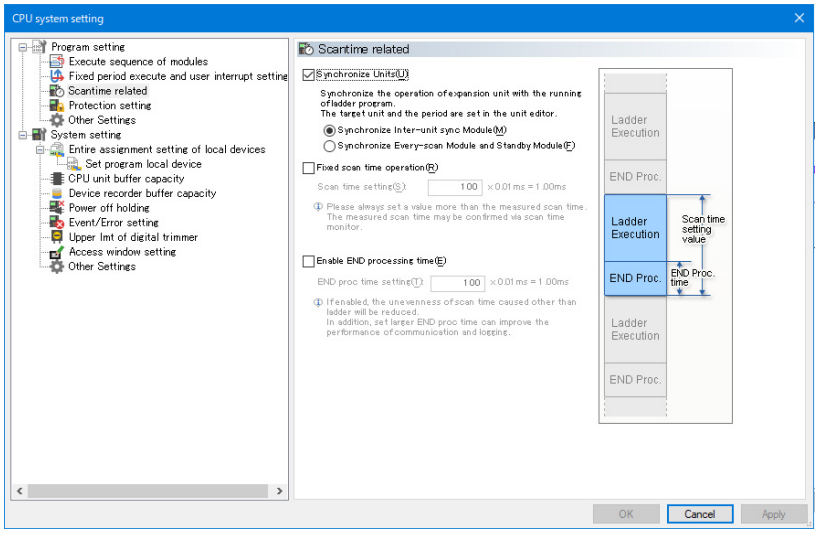
"Inter-unit synchronization refresh devices" setting



The second KV-XH16ML should be configured the same way.

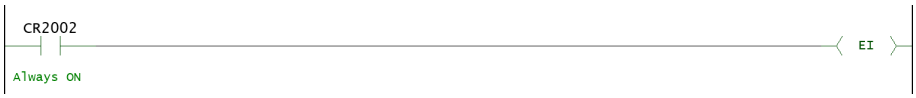
Point **Of the KV-8000/7000 series expansion units (S-Units, X-Units), the S-Unit does not have a "CPU unit device write/read" setting (function). Furthermore, some units do not have an "Inter-unit synchronization refresh" setting.**

3 **Setting target module which executes inter-unit synchronization by CPU system configuration.**



Point **No need for process 4 and 5 if the inter-unit synchronization target module is set as every-scan execution module or standby module.**

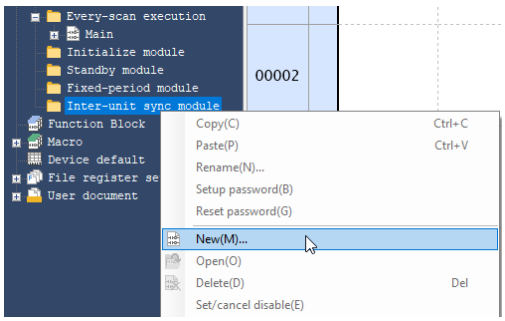
4 **Write an EI (enable interruption) instruction to every scan module or other ladder program.**



Point

- **Inter-unit synchronization refresh and inter-unit synchronization modules are executed as interrupt programs.**
- **If an interrupt with a higher priority than the inter-unit synchronization interrupt is being executed, it may not be possible to keep to the inter-unit synchronization cycle.**

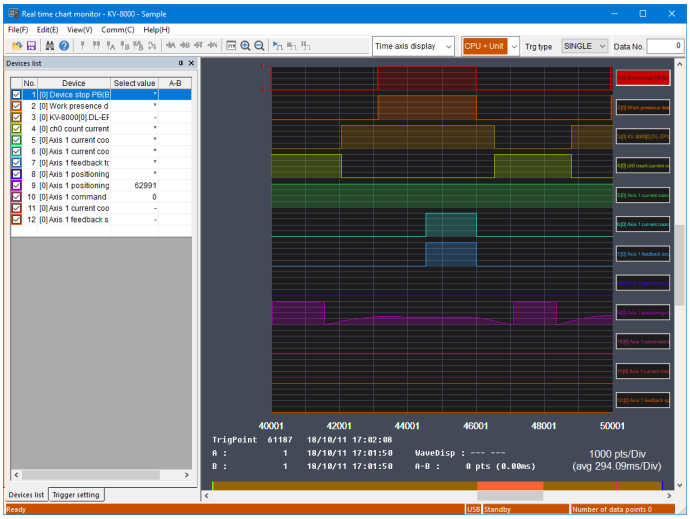
5 **The inter-unit synchronous module should be created, and written in the ladder program.**



6 **The program created should be transferred to PLC.**

Inter-Unit Synchronization Tracing

Inter-unit synchronization tracing is a function that simultaneously samples the internal data of KV-8000/7000 series expansion units (X-Units) that are executing inter-unit synchronization, and the device values of CPU units such as relays (R) and data memories (DM) at the inter-unit synchronization cycle, and monitors the waves.



Reference

The basic inter-unit synchronization tracing operation method is the same as for the real-time chart monitor. For details, refer to 📖 "9-6 Real-Time Chart Monitor" in the KV STUDIO User Manual.

Point

Only KV-8000/7000 series expansion units (X-Units) are compatible with inter-unit synchronization tracing. KV-8000/7000 series expansion units (S-Units), and KV-5000/3000 series expansion units are not compatible.

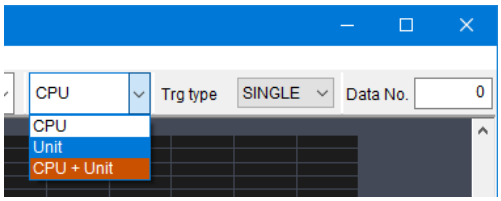
■ Start-up method

1 Click "Monitor/Simulator (N)" > "Real-time chart monitor (H)" from the menu.

The "real-time chart monitor" starts up

2 Change the operation mode to "CPU + Unit".

"Inter-unit synchronization tracing" starts up.



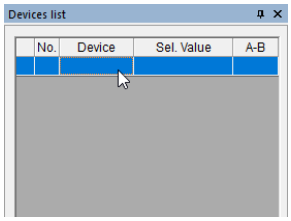
Point

- **Inter-unit synchronization tracing can only be started up from real-time charts started from KV STUDIO. It cannot be started up from a separately started real-time chart monitor.**
- **Inter-unit synchronization tracing can only be used when the CPU unit inter-unit synchronization function setting is set to "Enable". It is not displayed when set to "Disable".**

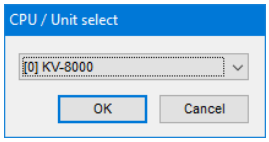
■ Registering Devices

1 Double click a cell in the "Devices list".

A "CPU/Unit Select" dialog box appears.



2 Select a unit.

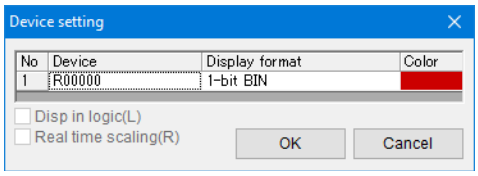


Point **Only CPU units or KV-8000/7000 series expansion units (X-Units) where the inter-unit synchronization function is set to "Used" in unit editor are displayed.**

3 Set the device to register.

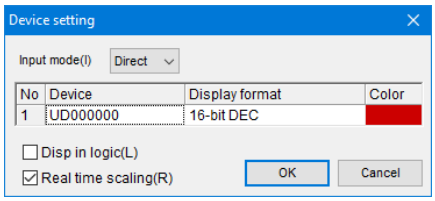
For CPU units

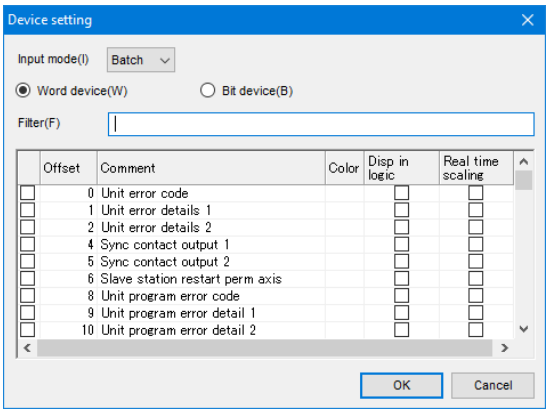
Directly enter the device and click "OK".



For KV-8000/7000 series expansion units (X-Units)

Select the input method from "Direct / Batch", specify the device to be registered, and click "OK".





Reference

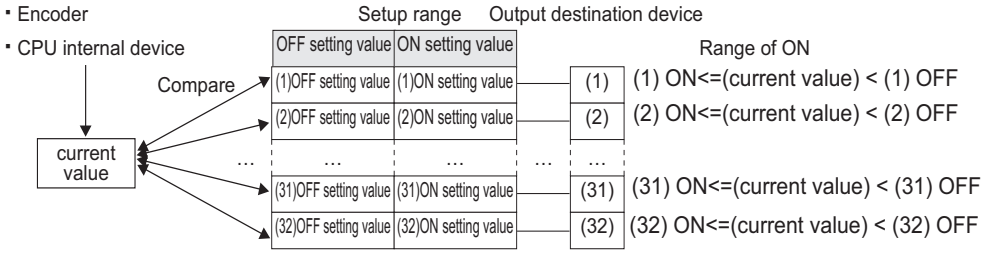
- Enter part of a comment in "Filter" to filter devices so only those with comments containing the filter term are displayed.
- Refer to the manual for each unit for details about the devices that can be registered and how to operate them.

5-11 Cam Switch

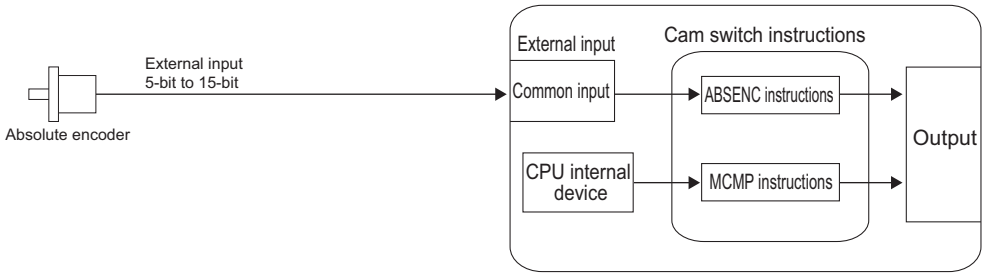
Angle conversion is performed on the value from encoder, and cam operation is simulated.

Cam Switch Function Overview

Current value obtained from acquisition encoder input is compared with preset setting of multiple ranges. The setting range in which the current value is located will be detected, and the output destination device corresponding to this setting range is set to ON.



Various special instructions corresponding to different input modes are available.



Cam Switch Instructions

There are the following 2 types of cam switch instructions.

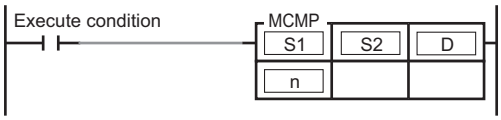
Instruction	mnemonics	Description	See
Multi-stage comparator	MCMP	Multi-stage comparator operation is performed.	Page 5-179
Absolute Encoder	ABSENC	Absolute encoder is used to perform cam operation.	Page 5-183

MCMP

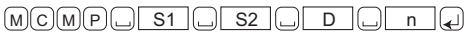
Multi-stage comparator

To perform multi-stage comparator operation.

Ladder program



Input mode



Operand	Description	Occupied area
S1	Specifies the leading device of data block for comparing. *1	32× n bits
S2	Specifies the device for comparing.*2	16 bits
D	Specifies the leading device for storing comparison result. *3	n bits
n	Specifies number of data for comparing or the device for storing number of data. (1 to 32)*4*5	16 bits

*1 If bit device is specified, then n x 32 continuous bits starting from channel will be processed.
If word device is specified, n x 2 words will be processed.

*2 If bit device is specified, then continuous 16 bits will be processed.
If relays other than channel leading relay (R002, R106, R1012, etc.) are specified, then 16 bits will be processed, straddling to the next channel.

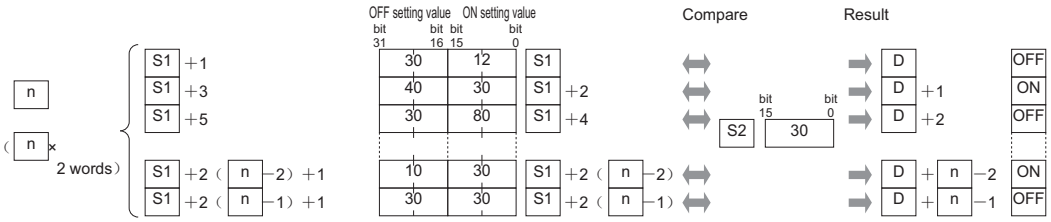
*3 When bit device is specified, n bits will be occupied.
When word device is specified, the result is stored in lowest bit of each word device. The bits other than the lowest bit is not changed.

*4 "$" cannot be used.

*5 When bit device is specified, continuous 16 bits will be occupied.
If word device is specified, 1 word will be occupied.

Description of Operation

When execution condition is ON, [n] 32-bit data (2 unsigned 16-bti BIN data: OFF setting and ON setting) starting from [S1] are compared with the unsigned 16-bit BIN data stored in [S2], and the result is stored in devices starting from [D].



When OFF setting > ON setting

If OFF setting > [S2] ≥ ON setting, then [D] is ON.

When OFF setting < ON setting

If OFF setting > [S2] or [S2] ≥ ON setting, then [D] is ON.

When OFF setting = ON setting

[D] is always OFF.

Point **Even if the execution condition is OFF, output will not be changed.**

Operation flag

CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	ON in the following cases; OFF in other cases. · When a value out of range is specified in [n]. · When a device other than channel leading device is specified in [S1]. · When range of indirect specifying and index modification is not proper. · When a device other than the channel leading device is specified in [D].

* If CR2012 is ON, the instruction will be not executed. When CR2012 is ON, the detailed error info will be stored in CM5100 to CM5176.

"Latest error details (CM)", Page 4-87

Sample Program

The value of data memory DM0 is compared with setting value, and output relay is ON/OFF according to the following conditions.

If #100≤DM0≤#200, R30000 is ON.

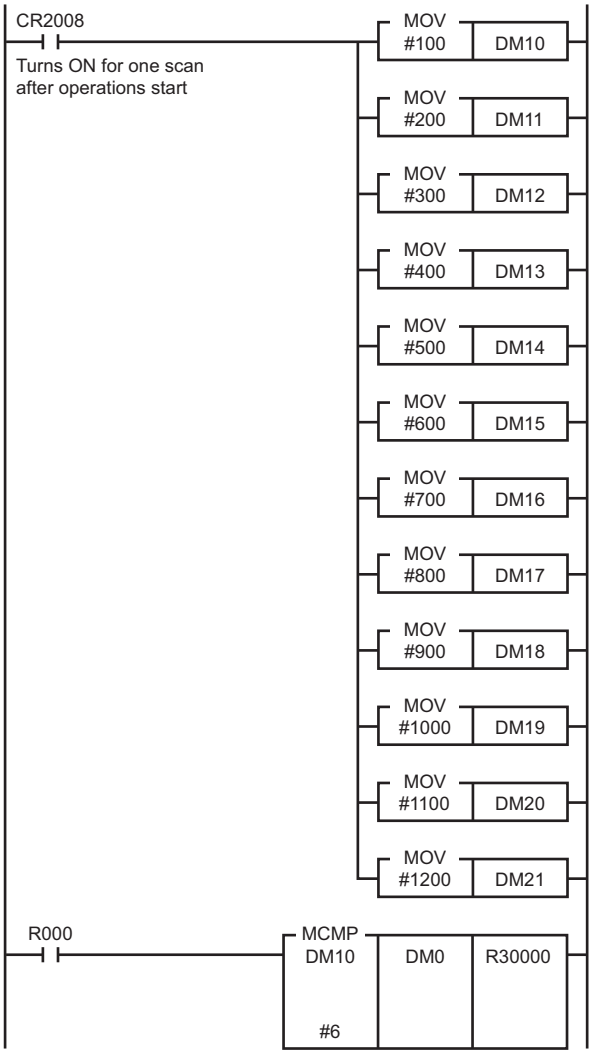
If #300≤DM0≤#400, R30001 is ON.

If #500≤DM0≤#600, R30002 is ON.

If #700≤DM0≤#800, R30003 is ON.

If #900≤DM0≤#1000, R30004 is ON.

If #1100≤DM0≤#1200, R30005 is ON.

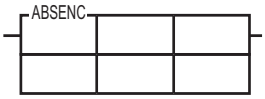


<Mnemonic List>

```
LD CR2008
MOV #100 DM10
MOV #200 DM11
MOV #300 DM12
MOV #400 DM13
MOV #500 DM14
MOV #600 DM15
MOV #700 DM16
MOV #800 DM17
MOV #900 DM18
MOV #1000 DM19
MOV #1100 DM20
MOV #1200 DM21
LD R000
MCMP DM10 DM0 R30000 #6
```

MEMO

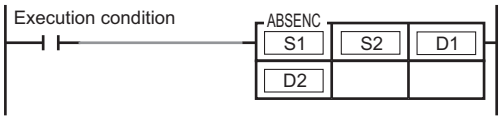
ABSENC



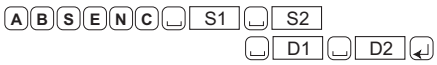
Absolute encoder

Performs cam switch operations using an absolute encoder

Ladder program



Entry mode



Operand	Description	Occupied area
S1	Specifies the leading device of the data block in which the angle value to compare (0 to 3599, in units of 0.1°) is stored*1	32 × N*5 bits
S2	Specifies the device in which the absolute encoder value is stored*2	16 bits
D1	Specifies the leading device in which the comparison results will be stored*3	N*5 bits
D2	Specifies the leading device in which parameters are stored *4	16×8 bits

*1 When you specify a bit device, the [D2 + 0] × 32 consecutive bits from the start of the channel will be used.
When you specify a word device, [D2 + 0] × 2 words will be used.

*2 When you specify a bit device, 16 continuous bits will be used. If you specify a relay other than the channel's leading relay (such as R002, R106, and R1012), the next channel will also be used to provide the necessary 16 bits.

*3 When you specify a bit device, [D2 + 0] bits will be occupied.
When you specify a word device, the result is stored in the lowest bit of the word device. The bits other than the lowest bit will not be changed.

*4 When you specify a bit device, 128 consecutive bits from the start of the channel will be occupied.
When you specify a word device, 8 words will be occupied.

*5 N is the number of comparison data specified by D2 +0.

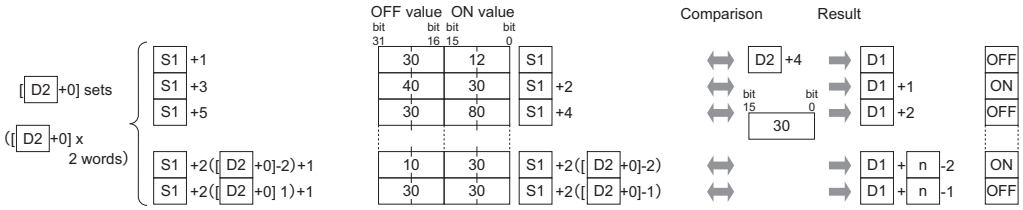
Operation Description

Parameters

Device		Details
D2 + 0	Amount of data to compare	Sets the amount of data to compare (1 to 32)
D2 + 1	Resolution	Sets the encoder resolution (32 to 32768)
D2 + 2	Residual gray code	Sets the residual gray code (0 to 16384) (Example) Resolution: 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 — 36, 360, 720 Residual gray code: 0 — 14, 76, 152
D2 + 3	Zero degree value	Sets the value to specify as zero degrees (0 to resolution - 1)
D2 + 4	Current angle (in units of 0.1°)	Stores the current angle after compensation (0 to 3599) • S2 - [D2 + 2] ≥ [D2 + 3]: (3600/[D2 + 1]) × (S2 - [D2 + 2] - [D2 + 3]) • S2 - [D2 + 2] < [D2 + 3]: (3600/[D2 + 1]) × ([D2 + 1] + S2 - [D2 + 2] - [D2 + 3])

Device		Details		
		D2 + 5	D1	D2 + 4
D2 + 5	Operation error	Bit 0: Output response error Conditions If, within a scan period, the ON and OFF values differ between two or more output destination devices that are changed or if there are differences between the ON values or OFF values, an error will occur. No error will occur in the following cases. • When the ON values and OFF values of the outputs that are changed at the same time are the same. • When the ON values of the outputs that are changed at the same time are the same and when the OFF values of the outputs that are changed at the same time are the same.	No change	Change
		Bit 1: Input value out of range error When the encoder's input value is outside of the setting range, this will be turned ON.	No change	No change
		Use a ladder program to reset the data.		
D2 + 6 D2 + 7	Work area	(This cannot be used by the user.)		

When the execution condition is turned ON, starting from S1, the first [D2 + 0] sets of 32-bit data (each comprised of two sets of 16-bit, unsigned BIN data indicating the OFF value and ON value) are compared with the current angle value stored in the device specified with [D2 + 4], and the results are stored in the bit devices starting from D1.



If the OFF value is greater than the ON value

When OFF value > [D2 + 4] ≥ the range of the ON value, D1 is turned ON.

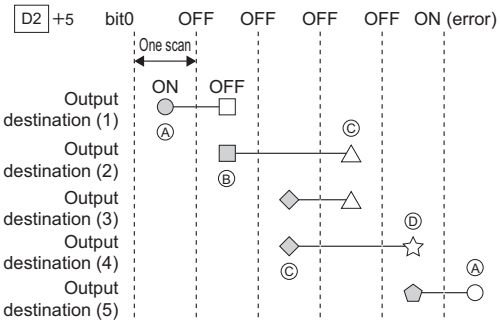
If the OFF value is less than the ON value

When OFF value > [D2 + 4] or [D2 + 4] ≥ the range of the ON value, D1 is turned ON.

If the OFF value equals the ON value

The operand D1 is always turned OFF.

■ Output response error (D2 + 5, Bit 0)



<Cases in which an error does not occur>

Ⓐ ..When one output destination device is changed within the scan period.

Ⓑ ..When the ON values and OFF values of two or more output destination devices that are changed within the scan period are the same.
- OFF value of output destination device (1) equals ON value of output destination device (2)

Ⓒ ..When the ON values of two or more output destination devices that are changed within the scan period are the same or when the OFF values of two or more output destination devices that are changed within the scan period are the same.
- OFF value of output destination device (2) equals OFF value of output destination device (3)
- ON value of output destination device (3) equals ON value of output destination device (4)

<Cases in which an error occurs>

If the ON values and OFF values of two or more output destination devices that are changed within the scan period are different. Alternatively, if the ON values are different or the OFF values are different.

Example) Ⓓ ... The ON values and OFF values of two or more output destination devices that are changed within the scan period are different.
OFF value of output destination device (4) does not equal ON value of output destination device (5).

Point

- **If you change the values during operation, the change will immediately be reflected in the output. If you do not want to reflect the changes in the output, turn the execution condition OFF.**
- **Do not use the CR2002 (Always ON) control relay as the execution condition.**
- **Even if the execution condition is turned OFF, there will be no change to the output.**
- **You cannot use this function in interrupt programs and initialization modules.**

Operation flags

CR2009	No change in status
CR2010	No change in status
CR2011	No change in status
CR2012	Turned ON when one of the following conditions is met. Otherwise, turned OFF. • If a value is specified for the encoder, the parameters, or the angle that is outside of the corresponding range. • S1 and D2: When you use these parameters to specify a bit device, if you specify a relay other than the channel's leading relay. • When the range specified for indirect specification or index modification is incorrect. • S1, D1, and D2: When you use these parameters to specify a timer or counter with indirect specification.

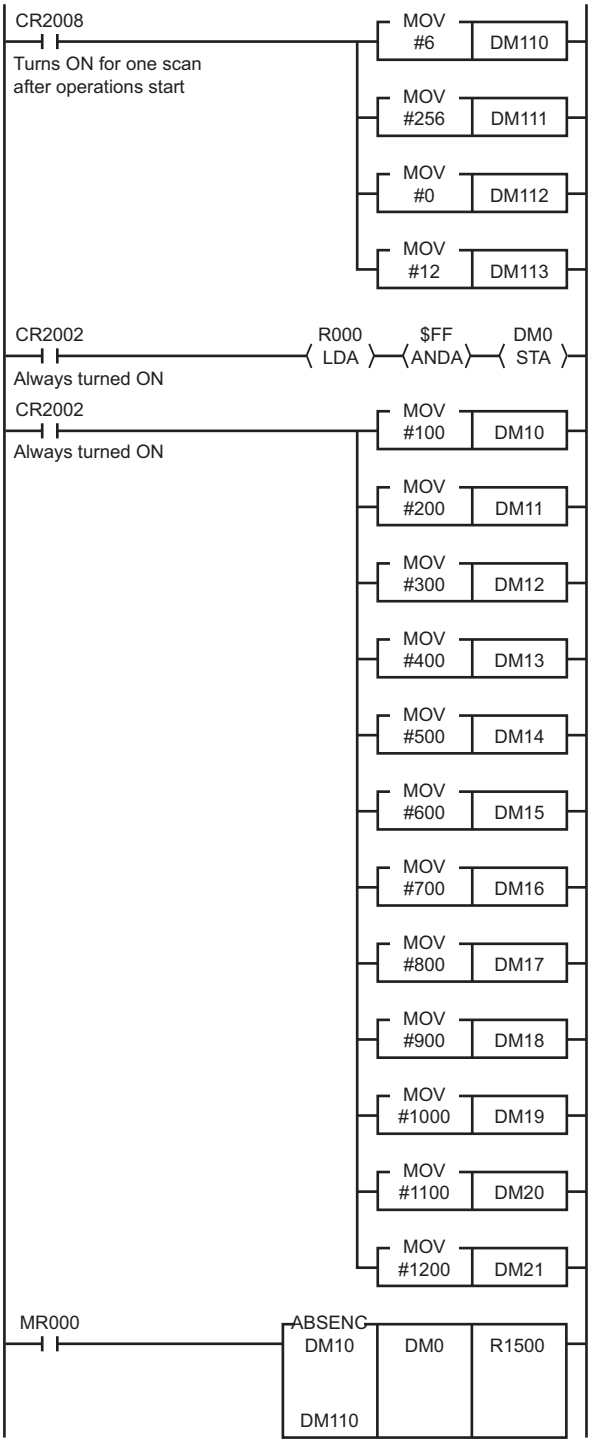
* If CR2012 is turned ON, the instruction will not be executed. When CR2012 is turned ON, the detailed error information is stored in CM5150 to CM5176.

"Error log", Page 4-86

Sample Program

The input values R000 to R007 from the absolute encoder are stored in DM0 and are compared with the set value. Output relays are then turned ON/OFF according to the following specifications.
Resolution: 256/Rev. Residual gray code: 0. Zero-degree teaching value: 12.

When DM0 is greater than or equal to 10.0° and is less than 20.0°, R1500 is turned ON.
When DM0 is greater than or equal to 30.0° and is less than 40.0°, R1501 is turned ON.
When DM0 is greater than or equal to 50.0° and is less than 60.0°, R1502 is turned ON.
When DM0 is greater than or equal to 70.0° and is less than 80.0°, R1503 is turned ON.
When DM0 is greater than or equal to 90.0° and is less than 100.0°, R1504 is turned ON.
When DM0 is greater than or equal to 110.0° and is less than 120.0°, R1505 is turned ON.



<Mnemonic List>

```
LD CR2008
MOV #6 DM110
MOV #256 DM111
MOV #0 DM112
MOV #12 DM113
LD CR2002
LDA R000
CON
ANDA $FF
CON
STA DM0
LD CR2002
MOV #100 DM10
MOV #200 DM11
MOV #300 DM12
MOV #400 DM13
MOV #500 DM14
MOV #600 DM15
MOV #700 DM16
MOV #800 DM17
MOV #900 DM18
MOV #1000 DM19
MOV #1100 DM20
MOV #1200 DM21
LD MR000
ABSENC DM10 DM0 R1500 DM110
```

5-12 Clock Function

On the CPU unit, "the year, month, day, hour, minute, second, and day of the week" are all managed by the built-in clock IC.

This is used to record the time and date of errors, the log of the power turning on and off, and the tracing function.

The clock accuracy at room temperature (+25°C) is ±60 s/month.

Reference

- Large-capacity capacitors are used to maintain the built-in clock, so if the device remains off for two weeks or more (at +25°C), the clock data will be cleared and then set to January 1, 2000.
- When the clock data is cleared, CR2311 is turned ON. It is turned OFF when the clock data is written.
- If you are using the KV-B1, you can back up the internal clock with a battery.

■ Reading clock data

Clock data is stored in control memory entries CM700 to CM706.

The format of the clock data is shown below.

	bit 15 … bit 0		
CM700	Year	0 to 99	The year indicates the last two digits of the year, assuming that "00" indicates "2000."
CM701	Month	1 to 12	All data is written in 16-bit binary data format.
CM702	Day	1 to 31	
CM703	Hour	0 to 23	
CM704	Minute	0 to 59	
CM705	Second	0 to 59	
CM706	Day of the week	0 to 6	The numbers correspond to the days of the week as follows: 0 indicates Sunday, 1 indicates Monday, 2 indicates Tuesday, 3 indicates Wednesday, 4 indicates Thursday, 5 indicates Friday, and 6 indicates Saturday.

- This data can be read directly by ladder programs (the data is read-only.)

■ Writing clock data (Setting the time and date)

● Using an access window to configure settings

You can display the clock data in the access window and use the access window to set the time and date.

"Date/Time" (page 5-132)

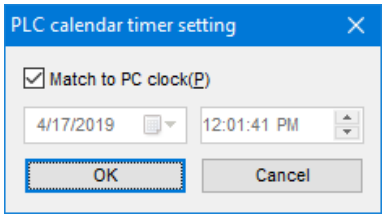
● Using ladder programs to configure settings

You can use the WTIME instruction to set the time and date. In addition, you can use the AJST instruction to perform 0-second adjustment.

For details, see the "clock processing instructions" in the "KV-8000/7000/5000/3000/1000 Series and KV Nano Series Instruction Reference Manuals."

● Using KV STUDIO to configure settings

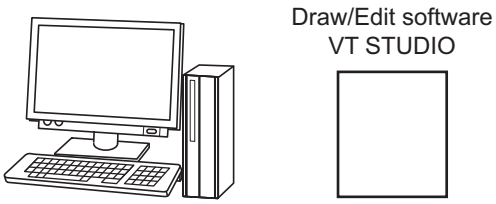
While using KV STUDIO to perform monitoring, on the "Debug" menu, click "Setup calendar timer."



See "Calendar Timer Setting" in the "KV STUDIO User's Manual."

● Using a touch panel (VT5/VT3 Series) to configure settings

If you use the PLC time and date synchronization function of the KEYENCE VT5/VT3 Series of touch panels, you can synchronize the date and time of the CPU unit with the date and time of the VT5/VT3 Series.



For details, see the "VT STUDIO Reference Manual."

Reference

Versions 4.6 and later of VT STUDIO have a function for restoring the time and date.
If the clock data of the connected CPU unit is cleared, it will automatically be synchronized to the time and date of the touch panel.

6

EXPANSION I/O UNIT FUNCTIONS

This section describes the functions provided by the KV-SIR32XT expansion I/O unit.

6-1 I/O Functions 6-2
6-2 Unit Interrupt Function 6-4
6-3 Inter-unit Synchronization Function 6-6
6-4 Input Capture Function 6-9
6-5 Buffering Input Function 6-10
6-6 PWM Output Function 6-19
6-7 Exclusive Instruction for the Unit for KV-SIR32XT 6-23
6-8 Exclusive Function for the Unit for KV-SIR32XT 6-30
6-9 Access Window 6-37
6-10 Unit Monitor 6-42
6-11 I/O Unit Reference Program 6-43
6-12 I/O Unit Error Mode 6-47

6-1 I/O Functions

The “KV-SIR32XT” can provide the functions listed below in addition to general I/O functions.

Item	Reference page
Interrupt Input	Unit Interrupt Function（page 6-4）
Input Capture	Input Capture Function（page 6-9）
Buffering Input Function	Buffering Input Function（page 6-10）
PWM Output Function	PWM Output Function（page 6-19）

General I/O

The interruption input, input capture, buffering input, general input can be used simultaneously.
PWM output and common output can not be used at the same terminal at the same time.
Please refer to "Wiring and Internal Circuit Diagrams" (page 2-16) for the terminal number of each input and output.
Useful methods for using I/O are described below.

■ I/O Refresh

Normally, during program execution, the status of I/O signals is updated (refreshed) only onceper scan (during ENDprocessing). You can use the following methods to update the I/O signals at arbitrary positions in the program (direct processing).

Direct I/O (DR)

Use the instruction operand to set the relay number to specify as "DR," and not "R."

I/O Refresh Instructions (RFSX, RFSY)

Use these instructions to refresh multiple bits of the I/O status at the same time.

For details, see the "RFSX and RFSY instructions" in the "KV-8000/7000/5000/3000/1000 Series and KV Nano Series Instruction Reference Manual".

■ Input range setting

Use the input range setting to select a 5 V or 24 V range for IN100 - IN115.
Use the unit editor to set the input range.
"Unit Editor" (page 6-3)
The editor allows you to change the IN100 - IN115 input range at one time.

Point **However, the editor cannot be used to change the IN000 - IN015 the input range.**

■ Constant configuration during input

The input time constant switching function can set the input time constant as the pulse width that is accurately acquired as signals from the input contact. If you make the input time constant small, you can obtain pulses with a narrower width. On the other hand, if you make the input time constant large, the input time constant can be used as a simple noise filter.

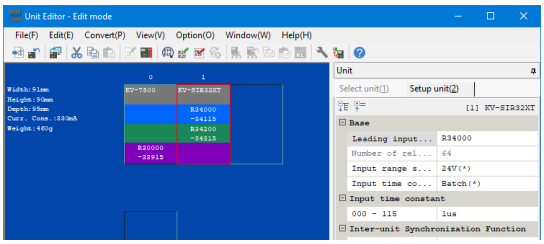
If you connect a non-contact signal in which chattering does not occur or when you are acquiring high-speed signals from an encoder or similar device, you can handle high-speed signals by reducing the input time constant.

Use the unit editor to set the input time constant. It can be set between 1 μs and 50 ms, and the default value is 10 μs.

See "Chapter 3 UNIT EDITOR" in the "KV STUDIO User's Manual".

● Unit Editor

You can use the Unit Editor to change the input time constant and the IN100-IN115 range (5 V/ 24 V) of "KV-SIR32XT" input relays.



■ Input Overvoltage Protection

If using 5V input mode, the overvoltage protection function operates when a voltage exceeding the rated voltage (maximum input voltage 6 V) is applied.

When overvoltage protection is activated, an "Input overvoltage error" (error code 101) is generated and all shared common inputs are turned OFF (I/O indicators are also turned OFF). (Shared common refers to IN100 - IN115.)

Point

Automatic recovery will not work when overvoltage protection is activated. Remove the cause of the problem, run PRG → RUN to effect a recovery and clear the error.

■ Output Overvoltage Protection

Overcurrent protection is activated in when an overcurrent is detected. When overcurrent protection is activated, an "Output overcurrent error" (202/203) is generated and all shared common outputs are also turned OFF. (Shared common refers to OUT200 - OUT215, OUT300 - OUT315; the generated error number vary with common. It is 202 for OUT200 - OUT215 and 203 for OUT300 - OUT315.)

Point

- **When overcurrent protection is activated, automatic recovery occurs when the cause of the problem has been eliminated.**
  ON/OFF occurs repeatedly until the cause of the problem has been eliminated.
- **When the PWM output function is used, PWM outputs take precedence over general outputs. It can be confirmed at #400 to #431 of buffer memory if the function is used or not.**

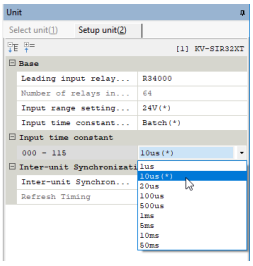
6-2 Unit Interrupt Function

S-Unit Function

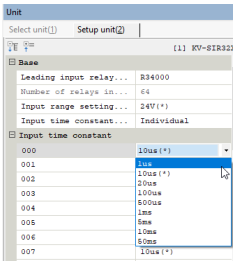
This function enables you to check the input status of the "KV-SIR32XT" and cause a unit interrupt to the CPU.
Use the IEDGE (input interrupt detection condition load) instruction to set the interrupt condition. Then execute the interrupt function using the INT (interrupt) instruction
You can select to perform the unit interrupt on the rising edge, falling edge or on both the rising and falling edge.

Setting Procedures

1 Use the unit editor to set the time constant of the input terminal that the interrupt will use.



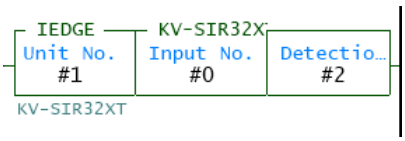
Reference When configuring individually



Point

- **Since the default value of the input time constant is 10 µs, it will take 10 µs or more for the interrupt to start when the input is turned ON.**
- **The input time constant can be set down to 1 µs.**

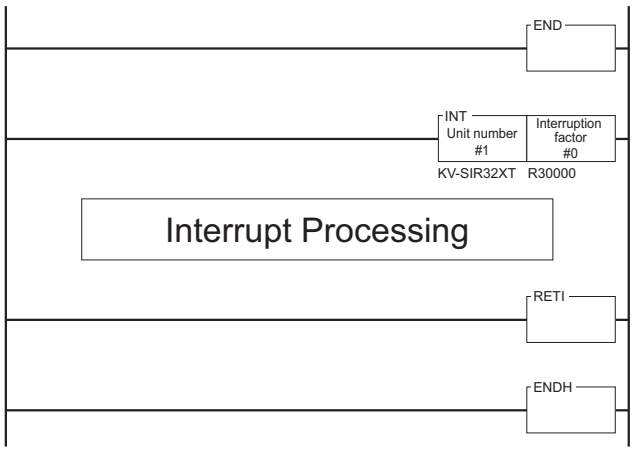
2 Use the IEDGE (input interrupt detection condition load instruction) to set the detection condition.
The interrupt detection condition can be set to detect on the rising edge, falling edge or on both the rising and falling edge.
The following is a setting example when detecting the falling edge of IN000 of unit number 1 and using the interrupt function.



Detection condition setting value	Detection condition
0	Rising edge
1	
2	Falling edge
3	Both rising edge and falling edge

Reference If not set, the condition will be detected at the rising edge.

3 Create a project to cause an interrupt between END - ENDH.



■ Unit interruption factor

The following factor number will be used for the unit interruption factor designated at INT instruction.

Factor number	Interruption factor of the factor number
0	IN000
1	IN001
2	IN002
:	:
15	IN015
16	IN100
:	:
30	IN114
31	IN115

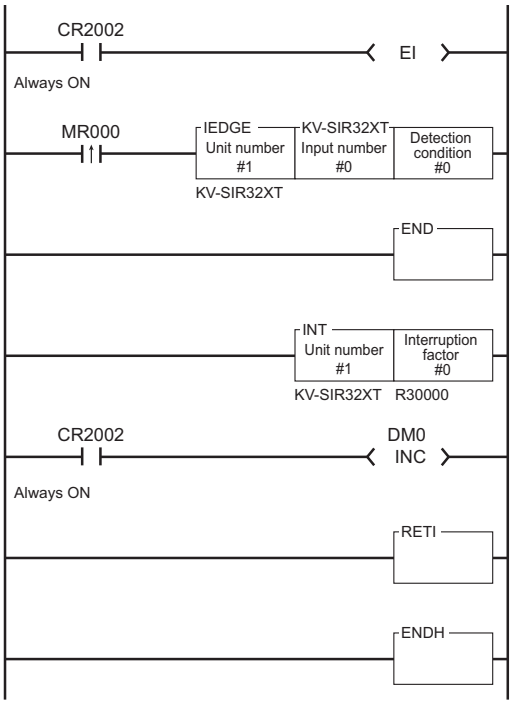
Reference

The detection condition of interruption can be selected from the rise, decline, or both rise/trail. Please use the IEDGE instruction in order to configure.

"KV-8000/7000/5000/3000/1000 series, KV Nano series instruction term reference manual" "IEDGE instruction"

(Usage example)

The diagram below shows an interrupt executed at the rising edge after unit number 1 input 000 goes on.



It allows interruption.

It configures the interruption condition.

It creates the interruption program and counts up DM0 within the program.

< Mnemonic List>

```
LD CR2002
EI
LDP MR000
IEDGE #1 #0 #0
END
INT #1 #0
LD CR2002
INC DM0
RETI
ENDH
```

6-3 Inter-unit Synchronization Function

S-Unit Function

The performance beginning time of the multiple “KV-SIR32XT“ that were connected to the CPU unit can be run simultaneously.
It can be controlled without being affected from scantime or mismatch of control period of units.

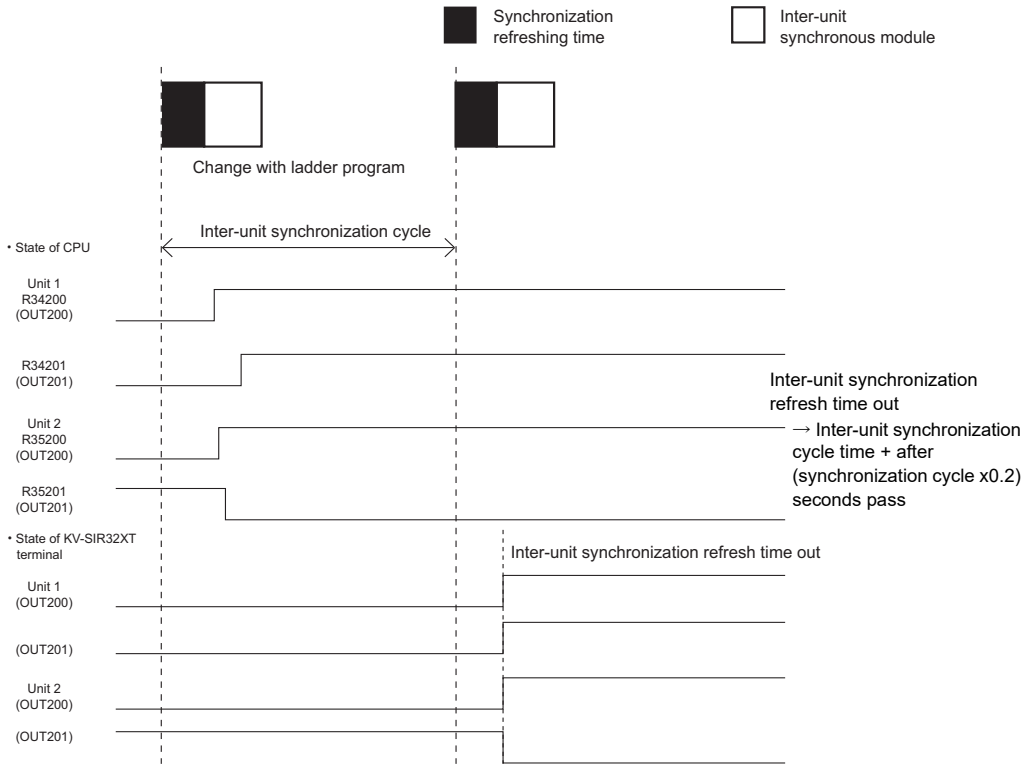
What is the Inter-unit Synchronization Function

The inter-unit synchronization function is a function that synchronizes the control timing by signal, between multiple units.
By using this function, simultaneous sampling between multiple units and simultaneous positioning can be undertaken.
For details on inter-unit synchronization, please see "5-10 Inter-unit Synchronization Function" (page 5-165).

Performance Overview when Running Inter-unit Synchronization Function

Here we will explain the performance when the inter-unit synchronization function is running.
If the refresh timing is being configured to inter-unit synchronization

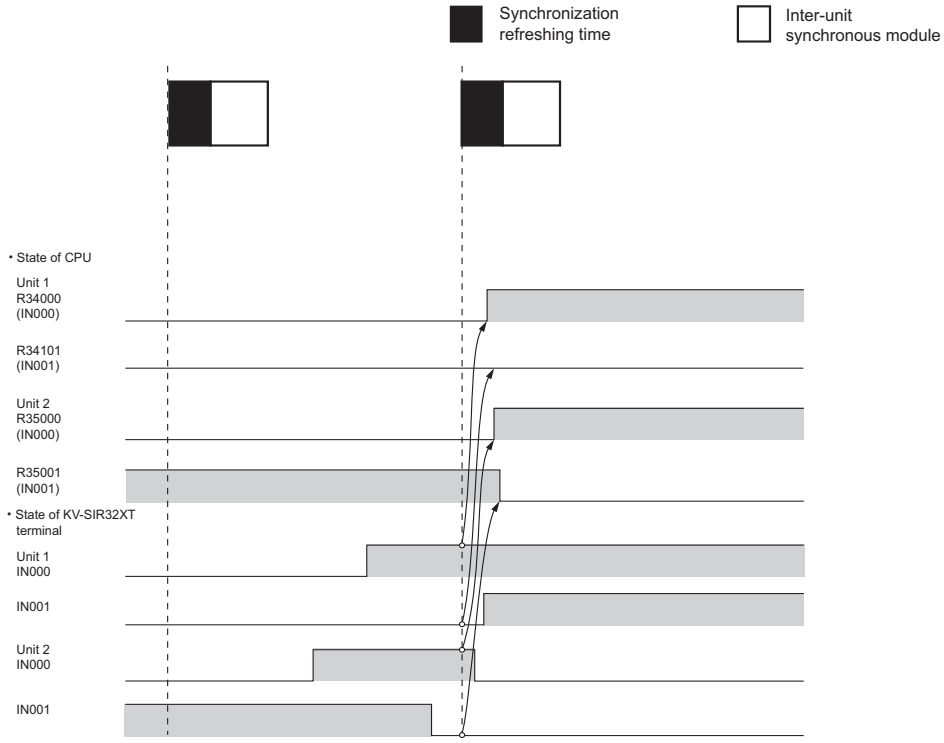
Output timing



Point

- **The state of the relay changed within the inter-unit synchronous module will be reflected in the output terminal from the beginning of the next inter-unit synchronization to after the inter-unit synchronization refresh time out.**
- **The comparator match read at inter-unit synchronization refresh and signals such as sync contact output can be simultaneously output for multiple units and multiple terminals at the inter-unit synchronization cycle x 0.2 (ms). This is because they are refreshed in the input/output order in inter-unit synchronization refresh.**
- **If a direct refresh such as DR is performed within the inter-unit synchronous module, it will immediately be reflected in the output.**
  If you run the direct updating, some mismatch can be occured in output timing between the units.

Input timing



Point

- **Data read in inter-unit synchronization refreshing is the value of synchronization startup and right before executing update synchronization. Moreover, values can be read during the same timing between the terminals and units.**
- **If a direct refresh command such as DR is used within the inter-unit synchronous module, the value when performing a direct refresh command can be read out, rather than the data when beginning synchronization.**

If the refresh timing is being configured to END process

The refreshing timing will be when END processing or when performing a direct refresh command, and sometimes there will not be synchronization between units and terminals.
However, regarding input, because the state of inter-unit synchronization will be latched to the [#240 IN000 to 015 synchronization capture, #241 IN100 to 115 synchronization capture] of the buffer memory every inter-unit synchronization cycle, by reading out within the inter-unit synchronous module, the state of inter-unit synchronization can be confirmed.

Unit Specification when Running Inter-unit Synchronization Function

Items		Parameter
Inter-unit synchronization cycle		40.0us～
Refresh device	Input R	2ch (32 points)
	Input DM	—
	Output R	2ch (32 points)
	Output DM	—
	Buffer memory	—

Point

The inter-unit synchronization cycle varies depending on the extension unit. The settable minimum cycle follows the control cycle of the unit with the largest control cycle.

6-4 Input Capture Function

When the "KV-SIR32XT" input goes ON/OFF, the input capture function captures the value of the free run counter (32-bit 0.1 μs) built into the "KV-SIR32XT". The result of an input capture is always updated and stored in buffer memory. As a result, when this function is used together with the buffer memory instruction, it is possible to measure input ON time. The input capture function can be used in all 32 input terminals.

Buffer memory address	Name	Attribute R: Read only
0000 to 0001	IN000_rising edge input capture	R
0002 to 0003	IN000_falling edge input capture	R
0004 to 0005	IN001_rising edge input capture	R
:	:	:
N*[1]×4 to N×4+1	IN***_rising edge input capture	R
N*[1]×4+2 to N×4+3	IN***_falling edge input capture	R
:	:	:
0124 to 0125	IN115_rising edge input capture	R
0126 to 0127	IN115_falling edge input capture	R
0128 to 0129	0.1μs 32-bit free run counter	R

* 1 0 to 31 (0:IN000, 1:IN001 to 15:IN015, 16:IN100,17:IN101 to 31:IN115) will be entered to N.

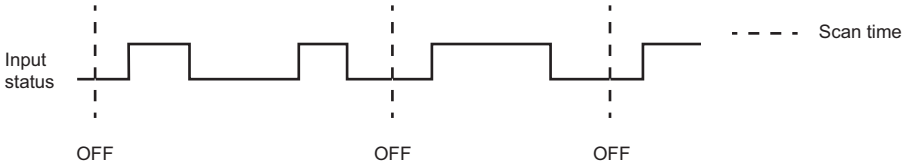
For reference examples of ON-time measurements using input capture, refer to "6-11 I/O Unit Reference Program".

6-5 Buffering Input Function

The buffering function samples the input status of the "KV-SIR32XT" down to 1 μs.
There are 2 types of mode in buffering, when continuous buffering is used, the acquisition of mass data is possible which exceeds the size of memory for buffering within the unit by reading out the data while buffering. When trigger buffering is used, it continues sampling until the trigger is entered and shuts down after it samples the amount of data configured before and after combined at the timing when the trigger is entered.

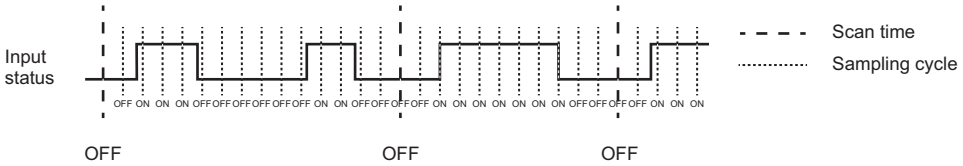
■ Buffering Operation Overview

- When acquired with each scan



Signals whose ON (OFF) time is shorter than the scan time make it impossible to correctly sample input status.

- When buffering is implemented

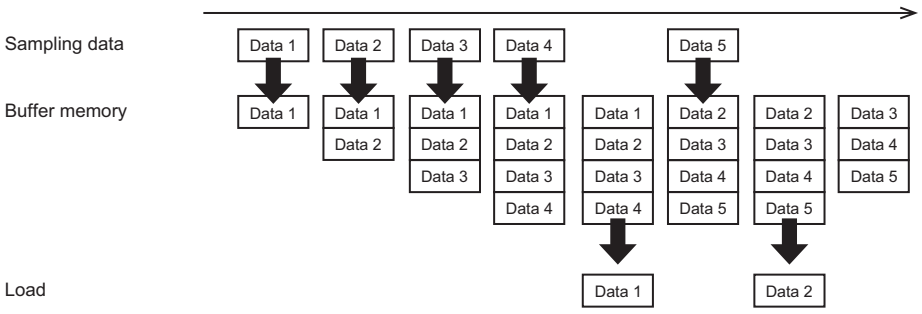


Correct sampling of input status is possible even when ON (OFF) time is shorter than the scan time.

■ Operating mode

● Continuous buffering

Sampled data is stored in order. It uses U_RDBF instruction and when the readout of buffering data is executed, it reads out data to the CPU device and the data of buffer memory will be deleted.
When data items in the buffer memory reaches 4096 items, the buffer overflows and buffering stops.

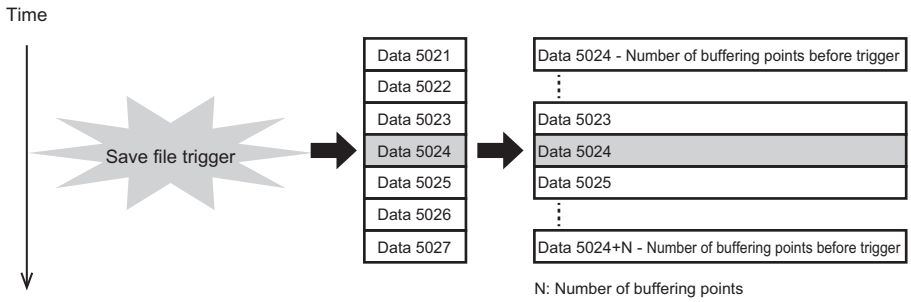


Point

- **Data that has been read out from the buffer is deleted.**
- **In the read-out, the data is read out starting in order from the oldest data. It can not read out the data in progress.**

● Trigger Buffering

Like continuous buffering, this function stores device status and values in an internal buffer. When files are saved, only the specified amount of data preceding and following is extracted and logged in buffer memory.



Point **Buffer overflow does not occur during trigger buffer execution.**

● Data storing procedures

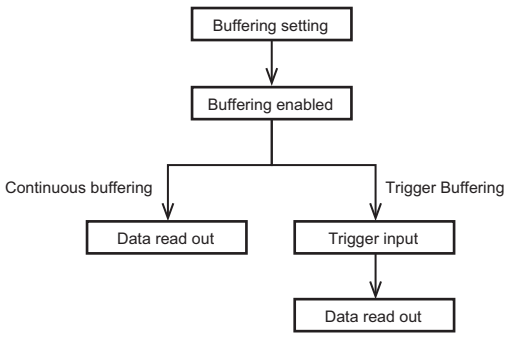
Sampled data is stored in buffer memory in 16-bit chunks.
Data is stored in IN000 - IN015 (or IN100 - IN115) order in bit 0 - bit 15.

Remarks	Data (when in decimal notation)
Oldest data	0010 0101 1101 0001 (9681)
Second oldest data	0010 0101 1101 0000 (9680)
:	:
Nth oldest data	0011 0001 1010 0101 (12709)
:	:
4095th oldest data	

Point **Input information for 16 points found during sampling is stored in one buffer memory address. To verify the time sequence data for each point, it is necessary to verify the time sequence for specific bit positions of the read out device.**

● Usage method

All settings are made in the ladder program.



● Used buffer memory address

Low : IN000 - IN015

Hi : IN100 - IN115 setting

Buffer memory Address	Name	Attribute R: Read only R/W: Read/write	Overview
#512	Low_Buffering mode configuration	R/W	0 : Consecutive 1 : Trigger (Initial value:0)
#513	Low_Buffering point configuration	R/W	1 to 4095（Initial value : 100）
#514	Low_Buffering cycle configuration	R/W	1～65535（μs）（Initial value : 10）
#515	(System reservation)	—	
#516	Low_Trigger condition configuration	R/W	0 : Writing 1 to trigger register 1 : Designated input ON edge 2 : Designated input OFF edge 3 : Interruption in inter-unit synchronization (Initial value : 0)
#517	(System reservation)	—	
#518	Low_Trigger terminal configuration	R/W	0 to 31（Initial value : 0）
#519	Low_ Buffering point configuration before trigger	R/W	0 to 4095（Initial value : 0）
#520	Low_ Select configuration of identification number	R/W	0 : None (Initial value) 1 : Serial number 2 : Separately elapsed time of inter-unit synchronization period 3 : Serial numbers of inter-unit synchronization period
#521～#527	(System reservation)	—	
#528	Low_Buffering mode configuration monitor	R	0 : Continuous 1 : Trigger（Initial value : 0）
#529	Low_Buffering point configuration monitor	R	1 to 4095（Initial value : 100）
#530	Low_Buffering cycle configuration monitor	R	1 to 65535（μs）（Initial value : 10）
#531	(System reservation)	—	
#532	Low_Trigger condition configuration monitor	R	0 : Writing 1 to the trigger register 1 : Designated input ON edge 2 : Designated input OFF edge 3 : Interruption in inter-unit synchronization （Initial value : 0）
#533	(System reservation)	—	
#534	Low_Trigger terminal configuration monitor	R	0～31（Initial value : 0）
#535	Low_Buffering point configuration monitor before trigger	R	0～4095（Initial value : 0）
#536	Low_Select configuration monitor of identification number	R	0 : None 1 : Serial number 2 : Time elapsed by the inter-unit synchronization cycle 3 : Serial number by the inter-unit synchronization cycle
#537	(System reservation)	—	
#538	Low_Buffering condition monitor	R	bit0 : Buffering in progress bit1 : Waiting for trigger bit2 : Buffering complete
#539	Low_ Current buffering point monitor	R	0 to 4095
#540	Low_Buffering overflow	R	0 : Normal 1 : Occurrence of overflow
#541	Low_Buffering point monitor before trigger	R	0 to 4095
#542～#543	(System reservation)	—	
#544～#575	(Hi buffering configuration equivalent to #512 to #543)	R/W	
#576～#639	(System reservation)	—	
#640	Low_Buffering enable	R/W	0 : Disabled 1 : Enabled
#641	Hi_Buffering enable	R/W	0 : Disabled 1 : Enabled
#642～#655	(System reservation)	—	
#656	Low_Trigger	R/W	1 : Trigger Input (Will automatically be updated to 0 after input)
#657	Hi_Trigger	R/W	1 : Trigger Input (Will automatically be updated to 0 after input)

■ Device (For Low:when IN000 to IN015 is used, For Hi:when IN100 to IN115 is used)

- **Buffering mode configuration (For Low: #512, For Hi: #544)**

  Select the buffering mode to execute: Continuous buffering or trigger buffering.

  Configured value 0: Continuous buffering (Initial value)
  1: Trigger Buffering

  The set value when "1" is written during buffering enable is applied to operation.

- **Buffering point configuration (For Low: #513, For Hi: #545)**

  For the continuous buffering mode, when the number of data sampled in buffer memory reaches the number configured, buffering completed bit will be ON.

  The buffering will continue.

  Configured value 1 to 4095（Initial value:100）

  For the trigger buffering mode, buffering will stop when sampling points before and after the trigger reaches the point of configured amount of buffering points.

  The set value when "1" is written during buffering enable is applied to operation.

- **Buffering cycle configuration (For Low: #514, For Hi: #546)**

  It configures the sampling cycle of buffering.

  Configured value 1 to 65535 (μs) (Initial value:10)

  The set value when "1" is written by the UWRIT instruction during buffering enable is applied to operation.

- **Trigger condition configuration (For Low: #516, For Hi: #548)**

  Sets trigger conditions during trigger buffering.

  Configured value 0 : When 1 is written to trigger (For Low: #656, For Hi: #657) (Initial value)
  1 : When input configured at trigger terminal configuration (For Low: #518, For Hi: #550) is ON
  2 : When input configured at trigger terminal configuration (For Low: #518, For Hi: #550) is OFF

  The configured value when 1 was written to buffering enable will be reflected to the operation.

- **Trigger terminal configuration (For Low: #518, For Hi: #550)**

  It configures input terminal used for trigger when 1 to 2 is configured for the configuration of trigger condition.

  Configuration value 0 to 31（IN000～IN115）（Initial value :0（IN000））

  The set value when "1" is written during buffering enable is applied to operation.

- **Configuration of buffering point before trigger (For Low: #519, For Hi: #551)**

  It configures sampling point before trigger when trigger buffering mode is used.

  Configured value 0 to 4095（Initial value:0）

  The set value when "1" is written during buffering enable is applied to operation.

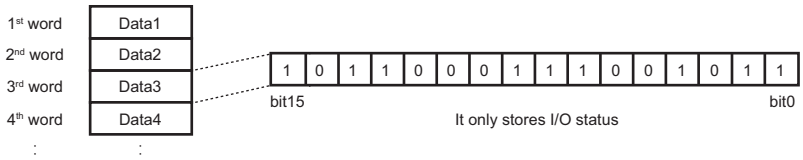
- **Configuration of identification number selection**

It configures the identification number given to the data.

Configured value 0: None (Initial value)
1 : Serial number
2: Time elapsed by the inter-unit synchronization cycle
3: Serial number by the inter-unit synchronization cycle

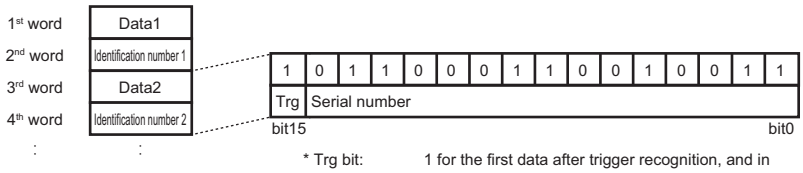
When configured to 0(None)

The identification number will not be given.



1 When configured to (Serial number)

It gives serial number and trigger recognition status to the 16 bit data and stores as 32 bit data.

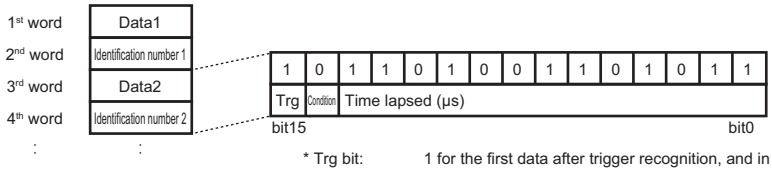


* Trg bit: 1 for the first data after trigger recognition, and in other cases 0 will be stored.

* Serial number: It will be divided by the sampling order within a range of 0 to 32767. (0 will be stored after 32767)

If configured to 2 (Time elapsed by inter-unit synchronization cycle)

The state of the inter-unit synchronization cycle and time elapsed (μs) within the synchronization cycle will be attached to 16 bit data, and stored as 32 bit data.



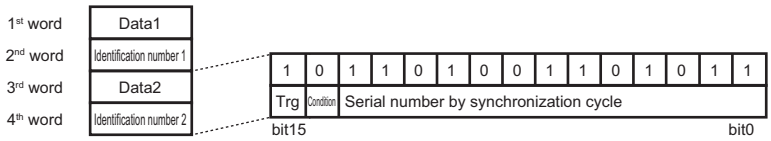
* Trg bit: 1 for the first data after trigger recognition, and in other cases 0 will be stored.

* Condition bit: 0 and 1 will switch for every synchronization cycle. For the same timing, the condition of 0 and 1 will be the same between expansion units.

* Time lapsed: The time elapsed (μs) from the timing of inter-unit synchronization operation will be stored in a range of 0-16383.

If configured to 3 (Different serial number by inter-unit synchronization cycle)

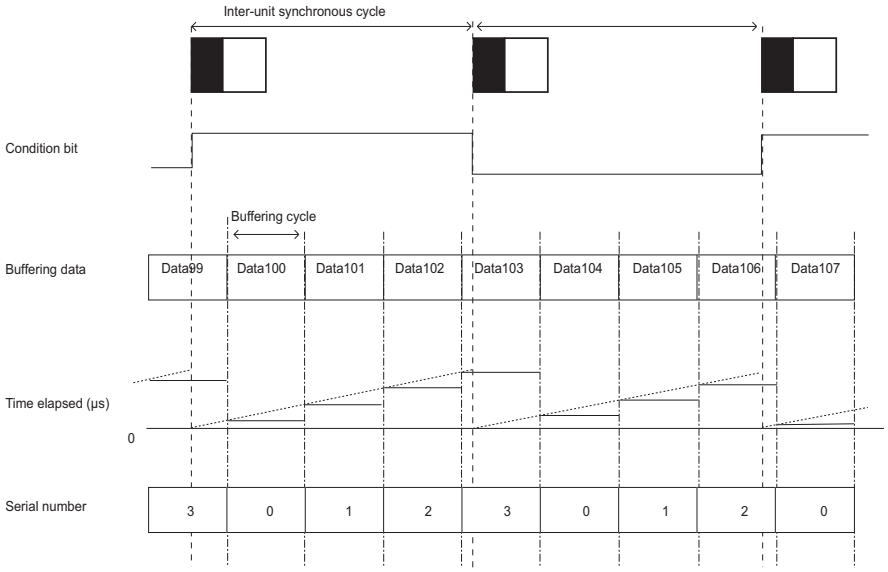
The state of the inter-unit synchronization cycle and serial number within the synchronization cycle will be attached to 16 bit data, and stored as 32 bit data.



* Trg bit: 1 for the first data after trigger recognition, and in other cases 0 will be stored.
* Condition bit: 0 and 1 will switch for every synchronization cycle. For the same timing, the condition of 0 and 1 will be the same between expansion units.
* Serial number synchronization cycle: The serial number (0~16383) from the timing of inter-unit synchronization performance will be stored.

Reference

Time elapsed by inter-unit synchronization cycle/serial number



- **Various monitor (Buffering mode configuration monitor/Buffering point configuration monitor/Buffering cycle configuration monitor/Trigger condition configuration monitor/ Trigger terminal configuration monitor/Buffering point configuration monitor before trigger/ Identification number selection configuration monitor)**
  (For Low: #528 to #536, For Hi: #560 to #568)
  Monitors set values used by buffering operations currently in progress.

Reference

The set value when "1" is written during buffering enable is applied to operation.
If a parameter changes after "1" has been written, the set value and the monitor value may differ.
Use the setting monitor to verify set values that are applied to actual operation.

- **Buffering condition monitor (For Low: #538, For Hi: #570)**

  Monitors buffering status.

  bit0 : Buffering in progress

  bit1 : Waiting for trigger

  bit2 : Buffering complete

  0 will always be stored while 0 is being written to buffering enable.

  1 will be stored until the buffering point before trigger is sampled, or until it reaches the buffering point after trigger input.

  3 will be stored if it is waiting for trigger in trigger mode while it is buffering in continuous mode.

  If it reaches the buffering point in trigger mode, 4 will be stored.

  If the buffering data of the unit side is more than the amount of points at buffering point configuration in continuous mode, 7 will be stored.

  Reference: The set value when "1" is written during buffering enable is applied to operation.
  If a parameter changes after "1" has been written, the set value and the monitor value may differ.

- **Current buffering point monitor (For Low: #539, For Hi: #571)**

  It monitors the data point within the buffer memory.

- **Buffering overflow monitor (For Low: #540, For Hi: #572)**

  In continuous buffering mode, the buffering status is monitored.

  0 : Normal operation

  1 : Occurrence of overflow

  Reference: It will be cleared to 0 when 0 is written to buffering enable.

- **Buffering point monitor before trigger (For Low: #541, For Hi: #573)**

  It monitors the buffering point before trigger with trigger buffering mode.

  Reference: If trigger is entered before buffering is implemented for the amount of buffering points configured before trigger after the buffering has started in trigger buffering mode, the buffering point before trigger will complete buffering with the status fewer than the configured value.

- **Buffering enable (For Low: #640, For Hi: #641)**

  Starts buffering execution.

  It clears the data of the data region when 1 is written (executes UWRIT instruction) and reads out the configured data and starts buffering.

  Reference: Please enter 0 once beforehand and write and execute 1 afterwards.

- **Trigger (For Low: #656, For Hi: #657)**

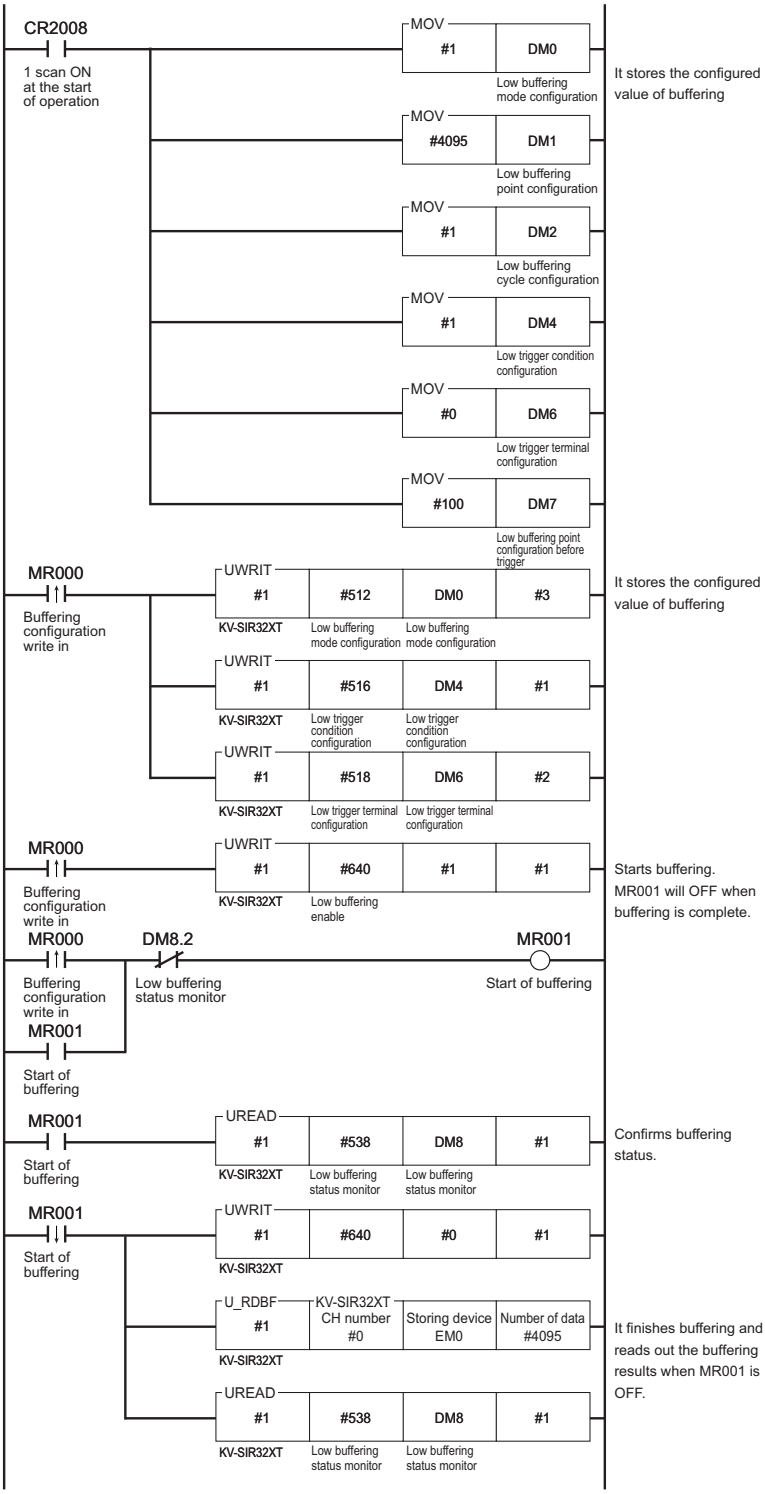
  When 0 is selected in trigger condition configuration (1 written to trigger) with trigger buffering mode, it will sample for number of times designated when 1 is written (executes UWRIT instruction) and finishes.

  Reference: There is no need to write in 0 beforehand.

■ Reference program

It buffers input condition from IN000 to IN015, 4095 points every 1μs.

It buffers 100 points before trigger and 3995 points after trigger with ON of IN000 as the trigger.

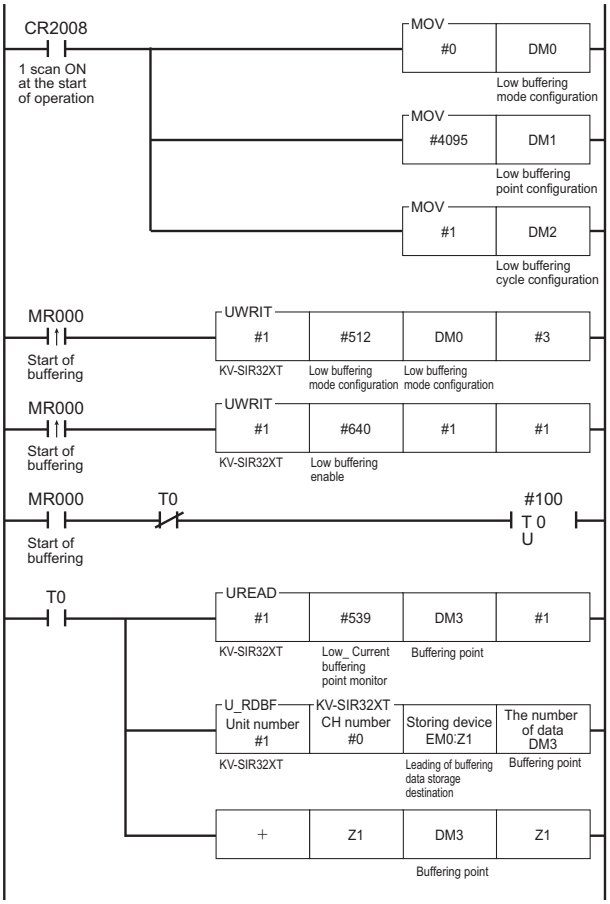


<Mnemonic List>

```
LD CR2008
MOV #1 DM0
MOV #4095 DM1
MOV #1 DM2
MOV #1 DM4
MOV #0 DM6
MOV #100 DM7
LDP MR000
UWRIT #1 #512 DM0 #3
UWRIT #1 #516 DM4 #1
UWRIT #1 #518 DM6 #2
LDP MR000
UWRIT #1 #640 #1 #1
LDP MR000
OR MR001
ANB DM8.2
OUT MR001
LD MR001
UREAD #1 #538 DM8 #1
LDF MR001
UWRIT #1 #640 #0 #1
U_RDBF #1 #0 EM0 #4095
UREAD #1 #538 DM8 #1
```

■ Reference program

Continuously buffers the input state of IN000 to IN015 every 1 µs. Reads the buffered data every 1ms with EM 0 leading.

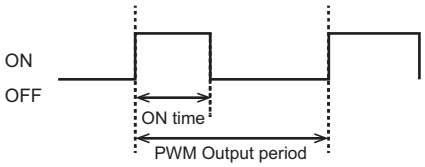


<Mnemonic List>

```
LD CR2008
MOV #0 DM0
MOV #4095 DM1
MOV #1 DM2
LDP MR000
UWRIT #1 #512 DM0 #3
LDP MR000
UWRIT #1 #640 #1 #1
LD MR000
ANB T0
TMU #0 #100
LD T0
UREAD #1 #539 DM3 #1
U_RDBF #1 #0 EM0:Z1 DM3
CAL+ Z1 DM3 Z1
```

6-6 PWM Output Function

This function performs PWM output specifying a pulse width in 0.1 μs increments up to 100 kHz using the "KV-SIR32XT" output terminal.
It outputs a pulse adjusting the pulse width by specifying PWM output period and ON time.



Point

- **PWM can be used simultaneously up to 8 points and can be set individually. They allow you to set and use the terminals, periods, pulse frequencies output by the settings.**
- **Up to 16 KV-SIR32XT can be connected to one KV-8000 series, so a maximum of 128 PWM output functions can be used.**

■ Operating mode

● Number of pulse setting mode

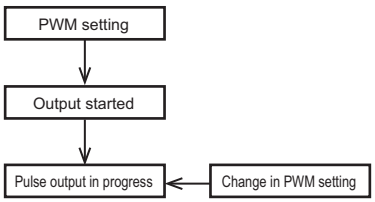
You can use PWM control to specify the number of pulses that will be output.

● Continuous pulse output mode

Pulses are continuously output until stop processing is performed.

■ Usage method

All settings are made in the ladder program.



Reference

Changing a setting during pulse output allows you to change the waveform output in a 1 x PWM period.

● Used buffer memory address

Buffer memory Address	Name	Attribute R: Read only R/W: Read/ write	Overview
#256	PWM0 output enable	R/W	0：Stop 1：PWM output （initial value：0）
#257	PWM0 output terminal	R/W	Output terminal selection （0 to 31（OUT 200 to OUT315))
#258～#259	PWM0 cycle	R/W	32 bits, 0.1μs unit 0.1 to 429496729.5（μs）
#260～#261	PWM0 ON time	R/W	32 bits, 0.1μs unit 0.0 to PWM0 cycle（μs）
#262～#263	PWM0 number of output pulse	R/W	32 bits, 0 to 4294967295（PLS）
#264～#271	PWM1 configuration	—	#256 to #263 equivalent
#272～#279	PWM2 configuration	—	
#280～#287	PWM3 configuration	—	
#288～#295	PWM4 configuration	—	
#296～#303	PWM5 configuration	—	
#304～#311	PWM6 configuration	—	
#312～#319	PWM7 configuration	—	
#320	PWM0 output status monitor	R	bit0：output enabled status monitor bit1：output complete bit2：output competition
#321	PWM0 output terminal monitor	R	Output terminal selection （0～31（OUT200～OUT315))
#322～#323	PWM0 monitoring number of output pulse	R	32 bits, 0 to 4294967295（PLS）
#328～#335	PWM1 output status monitor	R	#320 to #327 equivalent
#336～#343	PWM2 output status monitor	R	
#344～#351	PWM3 output status monitor	R	
#352～#359	PWM4 output status monitor	R	
#360～#367	PWM5 output status monitor	R	
#368～#375	PWM6 output status monitor	R	
#376～#383	PWM7 output status monitor	R	
#400	OUT200 function monitor	R	bit0：General output in progress bit1：PWM output in progress
#401	OUT201 function monitor	R	
:			
#415	OUT215 function monitor	R	
#416	OUT300 function monitor	R	
:			
#431	OUT315 function monitor	R	
#432	OUT200 to 215 monitor	R	Output status of each terminal
#433	OUT300 to 315 monitor	R	

■ Device

- **PWMn(n：0 - 7) output enabled**
  Select whether to execute PWM output.
  0: Not output
  1: PWM Output executed
  When the bit 0 is turned ON, the PWM output starts. Other bits are not used.

- **PWMn(n：0-7) output terminal**
  Select the output terminal that will execute PWM output.
  0 to 31: OUT200 to OUT315

- **PWMn(n：0 - 7) period**
  Set the PWM output period in the range 0.1 μs to 429496729.5 μs.
  It is recommended that the value be set to 10.0 μs or higher with the delay time of an actual circuit taken into account.
  Changes made during output will take effect when the PWM output period ends.

- **PWMn(n：0 - 7) ON time**
  Set the PWM output ON time in the range 0.0 μs to 429496729.5 μs.
  It will always be OFF when 0.0μs is configured to ON time.
  Changes made during output will take effect when the PWM output period ends.

- **PWMn(n：0 - 7) output pulse frequency**
  Set the number of output PWM pulses in the range 1 to 4294967295.
  When set to 0, pulses are output until output enabled is turned OFF.
  The configured value at the time when 1 was written to output enable will be reflected to operation. It is possible to rewrite during output enabled, but it will not be reflected.

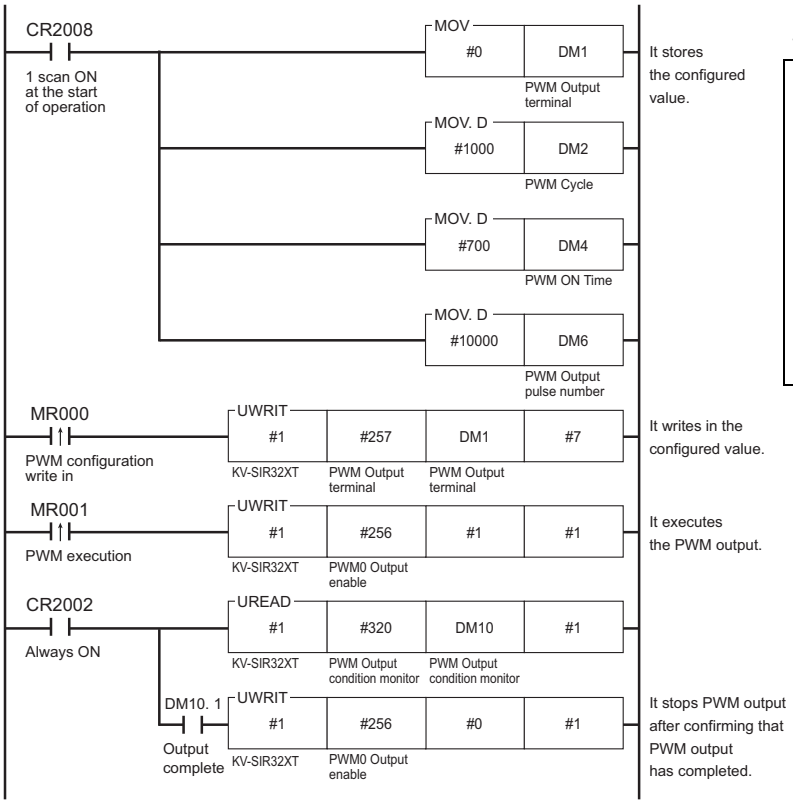
- **PWMn(n:0 - 7) output status monitor**
  Displays PWM output status. Storing 0 during PWM output enable turns off bits 0 - 2.
  Bit 0: ON: Output enabled status, OFF: Output disabled status
  Bit 1: ON: Output completed status, OFF: Output not completed status
  Bit 2: ON: Output is not possible because other output functions are in operation.
  OFF: Other output functions are not in operation.

- **PWMn(n:0 - 7) Output terminal monitor**
  Displays output terminals 0 - 31: OUT200 - OUT315 selected using PWMn(n：0 - 7).

- **PWMn(n:0 - 7)output pulse number monitor**
  PWM n(n: 0 - 7) is used to indicate the pulse number.
  Storing 0 when PWM output is enabled, clears the register.
- **Function monitor n（n:200 to 315）**
  It displays the function used for output.
  Bit 0: ON:General output in progress (Relay allocated to output is ON)
  Bit 1: ON: PWM output in progress
- **Output monitor**
  The output status of each terminal can be monitored.

■ Reference Program

The following is an example of a program where OUT200 generates PWM output.
Output 10000 pulses with a period of 100.0 µs (10 kHz) and an ON time of 70.0 µs.



<Mnemonic List>

```
LD CR2008
MOV #0 DM1
MOV.D #1000 DM2
MOV.D #700 DM4
MOV.D #10000 DM6
LDP MR000
UWRIT #1 #257 DM1 #7
LDP MR001
UWRIT #1 #256 #1 #1
LD CR2002
MPS
UREAD #1 #320 DM10 #1
MPP
AND DM10.1
UWRIT #1 #256 #0 #1
```

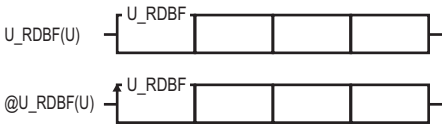
6-7 Exclusive Instruction for the Unit for KV-SIR32XT

It describes regarding the exclusive instruction for unit for KV-SIR32XT used within the ladder program. Please refer to "6-8 Exclusive Function for the Unit for KV-SIR32XT" (page 6-30) when using within the script program.

List of Exclusive Instructions for the Unit for KV-SIR32XT

Function	Instruction	Operation summary	Page
Read out of buffering data	U_RDBF	It reads out the status of I/O buffered at once from the buffer memory.	6-24
Disabling the input overvoltage protection function	U_DISOVP	It configures the activating/disabling of the input overvoltage protection function.	6-26
Disabling output overcurrent protection function	U_DISOCP	It configures activating/disabling of the output overcurrent protection function	6-28

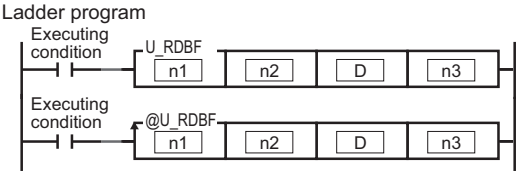
U_RDBF
@U_RDBF



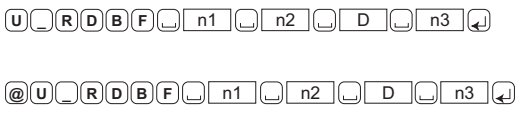
Reading out the buffering data

It reads out the buffered input data from the buffer memory.

Ladder program



Input method



Operand	Description	Occupied size
n1	Specifies the unit number.*1	—
n2	Specifies the channel number or its storage location. *2	16 bits
D	Specifies the start of the storage location for the read value.	n3 bits
n3	Specifies the number of data to be read (0~8192).*2	16 bits

*1 Only constants and variables set to constant can be specified.

*2 When bit data is specified, it occupies 16 continuous bits. When word data is specified, it occupies 1 word.

Description of Operation

U_RDBF When the executing condition is ON, it reads out n3 word of the buffering data of the channel designated at n2 of the unit number designated at n1 and stores to that places D at the top.

Point **When a value other than 0 is configured with the identification number selection of buffering, please enter the value that is doubled for the point to be read out since 1 data size becomes 2 words.**

@U_RDBF It executes only 1 scan at the rising edge of executing condition.

Computation Flag

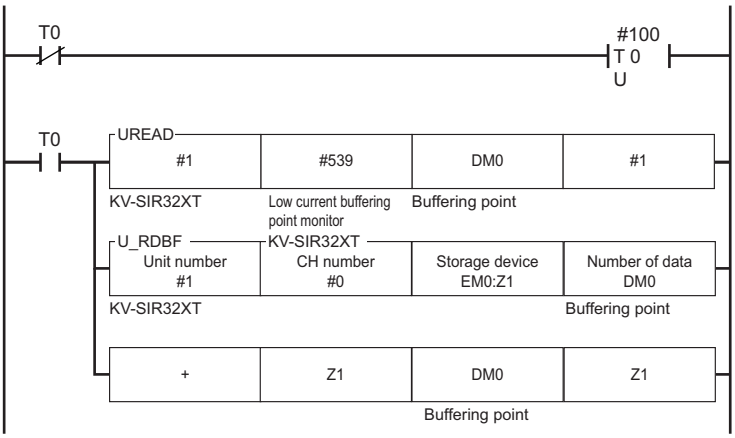
CR2009	No change
CR2010	No change
CR2011	No change
CR2012	ON for the following, otherwise it is OFF. • When the range of indirect designation, index modification is inappropriate. • When the unit number designated at n1 is equal to or greater than 64. • When the channel designated at n2 is out of range. • When with n3 word can not be obtained from designated at D. • When the number of data designated at n3 is greater than 8192. • When the unit number designated at n1 is not KV-SIR32XT.

* The instruction will not be executed when CR2012 turns ON.

KV-8000 Series The detailed information of the error will be stored to CM5150 to CM5176 when CR2012 is ON.

Sample Program

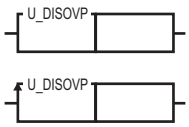
It reads out the buffered point for every 1ms and reads out the amount of points being buffered.



<Mnemonic List>

```
LDB T0
TMU #0 #100
LD T0
UREAD #1 #539 DM0 #1
U_RDBF #1 #0 EM0:Z1 DM0
CAL+ Z1 DM0 Z1
```

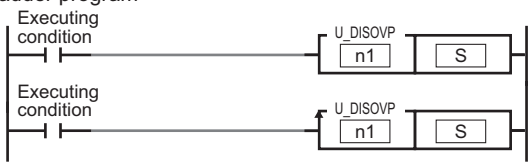
U_DISOVP
@U_DISOVP



Input overvoltage protection function configuration

It configures whether to activate of disable the input overvoltage protection function.

Ladder program



Input method

U _ D I S O V P ␣ n1 ␣ S ↵

@ U _ D I S O V P ␣ n1 ␣ S ↵

Operand	Description	Occupied size
n1	Specifies the unit number.*1	—
S	Specifies the value to be written or the storage location for the value to be written.*2 (0: Overvoltage protection enabled, 1: Overvoltage protection disabled)	16 bits

*1 Only constants and variables set to constant can be specified.

*1 When bit data is specified, it occupies 16 continuous bits. When word data is specified, it occupies 1 word.

Description of Operation

U_DISOVP When 1 is designated to S when the operating condition is ON, it deactivates overvoltage protection function of KV-SIR32XT of the unit number designated at n1.

Point

- **Setting of activating/disabling the Over voltage protection function is reflected when the instruction is executed.**
- **When powered on, 0: activating the Over voltage protection is active.**

@U_DISOVP It executes only 1 scan at the rising edge of executing condition.

Computation Flag

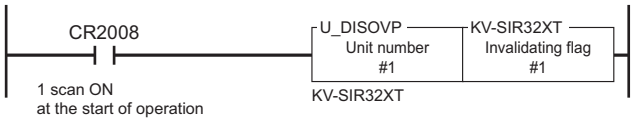
CR2009	No Change
CR2010	No Change
CR2011	No Change
CR2012	ON for the following, otherwise it is OFF. • When the range of indirect designation, index modification is inappropriate. • When the unit number designated at n1 is equal to or greater than 64. • When the unit number designated at n1 is not KV-SIR32XT.

* The instruction will not be executed when CR2012 is ON.

KV-8000 Series The detailed information of the error will be stored to CM5150 to CM5176 when CR2012 is ON.

Sample Program

It disables the overvoltage protection function of KV-SIR32XT of the unit number 1.



<Mnemonic List>

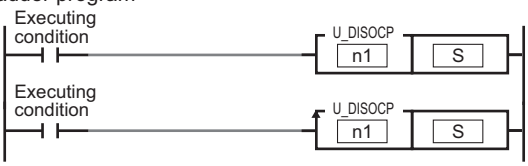
```
LD CR2008
U_DISOVP #1 #1
```

U_DISOCP
@U_DISOCP

Output overcurrent protection function configuration

It configures whether to activate of disable the output overcurrent protection function.

Ladder program



Input method

Operand	Description	Occupied size
n1	Specifies the unit number.*1	—
S	Specifies the value to be written or the storage location for the value to be written.*2 (0: Overcurrent protection enabled, 1: Overcurrent protection disabled)	16 bits

*1 Only constants and variables set to constant can be specified.

*2 When bit data is specified, it occupies 16 continuous bits. When word data is specified, it occupies 1 word.

Description of Operation

U_DISOCP It disables the overcurrent protection function of KV-SIR32XT of the unit number designated n1, when 1 is designated to S when executing condition is ON.

Point

- **Setting of activating/disabling the overcurrent protection function is reflected when the instruction is executed.**
- **When powered on, 0: activating the overcurrent protection is active.**

@U_DISOCP It executes only 1 scan at the rising edge of executing condition.

Computation Flag

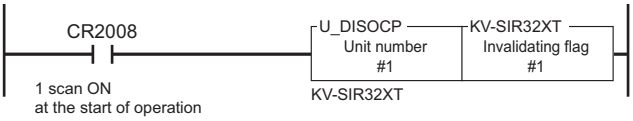
CR2009	No Change
CR2010	No Change
CR2011	No Change
CR2012	ON for the following, otherwise it is OFF. • When the range of indirect designation, index modification is inappropriate. • When the unit number designated at n1 is equal to or greater than 64. • When the unit number designated at n1 is not KV-SIR32XT.

* The instruction will not be executed when CR2012 is ON.

KV-8000 Series The detailed information of the error will be stored to CM5150 to CM5176 when CR2012 is ON.

Sample Program

It disables the overcurrent protection function of KV-SIR32XT of the unit number 1.



<Mnemonic List>

```
LD CR2008
U_DISOCP #1 #1
```

6-8 Exclusive Function for the Unit for KV-SIR32XT

It describes regarding exclusive function for the unit for KV-SIR32XT used within the script program. Please refer to "6-7 Exclusive Instruction for the Unit for KV-SIR32XT" (page 6-23), when using within the ladder program.

List of Exclusive Function for the Unit for KV-SIR32XT

■ ST Language function

Feature	Function	Summary of Operation	Page
Reading out the buffering data	U_RDBF	It reads out the status of I/O buffered at once from the buffer memory.	6-31
Disabling Input Overvoltage Protection Function	U_DISOVP	It configures the activating/disabling of the input overvoltage protection function.	6-32
Output overcurrent Protection Disabled	U_DISOCP	It configures the activating/disabling of the output overcurrent protection function.	6-33

■ KV Script function

Feature	Function	Summary of Operation	Page
Reading out the buffering data	U_RDBF	It reads out the status of I/O buffered at once from the buffer memory.	6-34
Disabling Input Overvoltage Protection Function	U_DISOVP	It configures the activating/disabling of the input overvoltage protection function.	6-35
Output overcurrent Protection Disabled	U_DISOCP	It configures the activating/disabling of the output overcurrent protection function.	6-36

ST Language function

U_RDBF Reads Buffering Data

Reads Buffering Data

U_RDBF ([EN,] Unit, Ch, Dst, DataCount)

Argument/ return value	Description	Type								Constant
		.U	.S	.D	.L	.F	.DF	.B	.T	
EN	Execution condition. Omissible. If omitted, always (every scan) is executed.	-	-	-	-	-	-	.B	-	○
Unit	Specifies the Unit No. of the unit to be read. (0 to 48)	.U	.U	.U	.U	-	-	-	-	○
Ch	Specifies the CH No. of the buffering data to be read (0 to 1)	.U	.U	.U	.U	-	-	-	-	○
Dst	Specifies the leading of the target data to store the read buffering data.	.U	.U	.U	.U	-	-	-	-	-
DataCount	Specifies the number of data points to be read. (1 to 8192)	.U	.U	.U	.U	-	-	-	-	○
R	None	-	-	-	-	-	-	-	-	-

Description of Operation

U_RDBF When the Execution Condition is ON, reads the buffering data of the channel specified by **Ch** of the unit number specified by **Unit** by the **DataCount** word, and stores in the device starting with **Dst**.

Please refer to "U_RDBF Instruction" (page 6-24) regarding the details.

● Example of Format

U_RDBF (bEnable,#1,#0,auData,#4000);

ST Language function

U_DISOVP Sets Input Overvoltage Protection Function

Sets Input Overvoltage Protection Function

U_DISOVP ([EN,] Unit, Flag)

Argument/ return value	Description	Type								Constant
		.U	.S	.D	.L	.F	.DF	.B	.T	
EN	Execution condition. Omissible. If omitted, always (every scan) is executed.	-	-	-	-	-	-	.B	-	○
Unit	Specify the target unit No. (0 to 48)	.U	.U	.U	.U	-	-	-	-	○
Flag	Specifies the target data to store the setting value. (0: Overvoltage protection enabled, 1: Overvoltage protection disabled)	.U	.U	.U	.U	-	-	-	-	○
R	None	-	-	-	-	-	-	-	-	-

Description of Operation

U_DISOVP If 1 is specified for **Flag** when ON is ON, disables the Overvoltage protection function of KV-SIR32XT with the unit No. specified in **Unit**.

Please refer to "U_DISOVP Instruction" (page 6-26) for details.

● Example of Format

U_DISOVP (bEnable,#1,#1);

ST Language function

U_DISOCP Sets Output Overcurrent Protection

Sets Output Overcurrent Protection

U_DISOCP ([EN,] Unit, Flag)

Argument/ return value	Description	Type								Constant
		.U	.S	.D	.L	.F	.DF	.B	.T	
EN	Execution condition. Omissible. If omitted, always (every scan) is executed.	-	-	-	-	-	-	.B	-	○
Unit	Specify the target unit No. (0 to 48)	.U	.U	.U	.U	-	-	-	-	○
Flag	Specifies the target data to store the setting value. (0: Overcurrent protection enabled, 1: Overcurrent protection disabled)	.U	.U	.U	.U	-	-	-	-	○
R	None	-	-	-	-	-	-	-	-	-

Description of Operation

U_DISOCP If 1 is specified for **Flag** when Execution Condition is 1, disables the Overcurrent protection function of KV-SIR32XT with the unit No. specified in **Unit**.

Please refer to "U_DISOCP Instruction" (page 6-28) regarding the details.

● Example of Format

U_DISOCP (bEnable,#1,#1);

KV Script function

U_RDBF Reading out the buffering data

Reading out the buffering data
U_RDBF ([Executing condition]*1, Unit number, CH number, Storage device, Number of data)

Parameter/Returned Value		Description	Type to Recognize								Constant	Device	Formula
			.U	.S	.D	.L	.F	.DF	.B	.T	#/$		
n1	Unit Number*2	It designates the unit number of the unit to read out.*2 (0 to 48)	.U	.U	.U	U	—	—	—	—	○*2	—	—
n2	CH Number*2	It designates CH number of the buffering data to read out. (0 to 1)	.U	.U	.U	U	—	—	—	—	○*2	○	○
D	Read Out Data Storage Device	It designates the leading device which stores the buffering data read out.	.U	.U	.U	U	—	—	—	—	—	○	○
n3	Number of Data to Read Out	It designates the data point to read out (1 to 8192).	.U	.U	.U	U	—	—	—	—	○*2	○	○

*1 [ ] can be omitted (When the operating condition is omitted, it will always (every scan) execute).

*2 $ (hexadecimal designated) can not be used.

Description of Operation

U_RDBF When the executing condition is ON, it reads out n3 word of the buffering data of the channel designated at n2 of the unit number designated at n1 and stores to the device that places D at the top.

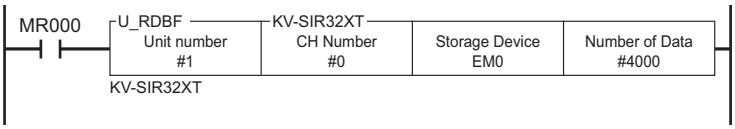
Please refer to "U_RDBF Instruction" (page 6-24) regarding the details.

● Example of Format

Script Description: U_RDBF (MR0,#1,#0,EM0,#4000)

Description of Operation: When MR0 is ON, it reads out 4000 words of the buffering data of KV-SIR32XT connected to the first unit starting with EM0.

Ladder Conversion



KV Script function

U_DISOVP Input overvoltage protection function configuration

Input overvoltage protection function configuration

U_DISOVP ([Executing Condition]*1, Unit number, Configured value)

Parameter/Returned Value	Description	Type to Recognize								Constant	Device	Formula
		.U	.S	.D	.L	.F	.DF	.B	.T	#/$		
n1 Unit number*2	Specifies a unit No.*2 (0 to 48)	.U	.U	.U	U	—	—	—	—	○*2	—	—
S Configured Value Storing Device	Specifies the Write value, or its Storage Destination. 2 (0: Overvoltage protection enabled, 1: Overvoltage protection disabled)	.U	.U	.U	U	—	—	—	—	○*2	○	○

*1 [ ] can be omitted (When the executing condition is omitted, it will always (every scan) execute).

*2 $ (Hexadecimal designation) can not be used.

Description of Operation

U_DISOVP When 1 is designated to S when the executing condition is ON, it disables overvoltage protection function of KV-SIR32XT of the unit number designated at n1.

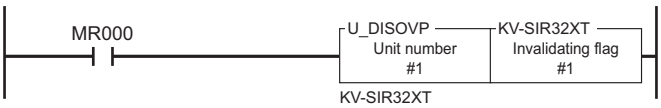
Please refer to "U_DISOVP Instruction" (page 6-26) for details.

● Example of Format

Script Description: U_DISOVP (MR0,#1,#1)

Description of Operation: When MR0 is ON, it will disable the input overvoltage protection function of KV-SIR32XT connected to the first unit.

Ladder Conversion



KV Script function

U_DISOCP Output overcurrent protection function configuration

Output Overvoltage Protection Function Configuration
U_DISOCP ([Executing Condition]*1, Unit number, Configured value)

Parameter/Returned Value	Description	Type to Recognize								Constant	Device	Formula
		.U	.S	.D	.L	.F	.DF	.B	.T	#/$		
n1 Unit number*2	Specifies a unit No.*2 (0 to 48)	.U	.U	.U	U	—	—	—	—	○*2	—	—
S Configured Value Storing Device	Specifies the Write value, or its Storage Destination. 2 (0: Overcurrent protection enabled, 1: Overcurrent protection disabled)	.U	.U	.U	U	—	—	—	—	○*2	○	○

*1 [ ] can be omitted (When the executing condition is omitted, it will always (every scan) execute).

*2 $ (Hexadecimal designation) can not be used.

Description of Operation

U_DISOCP It disables overcurrent protection function of KV-SIR32XT of the unit number designated at n1 when 1 is designated to S when the executing condition is ON.

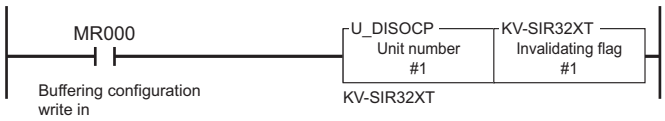
Please refer to "U_DISOCP Instruction" (page 6-28) regarding the details.

● Example of Format

Script Description: U_DISOCP (MR0,#1,#1)

Description of Operation: It disables the output overcurrent protection function of KV-SIR32XT connected to the first unit when MR0 is ON.

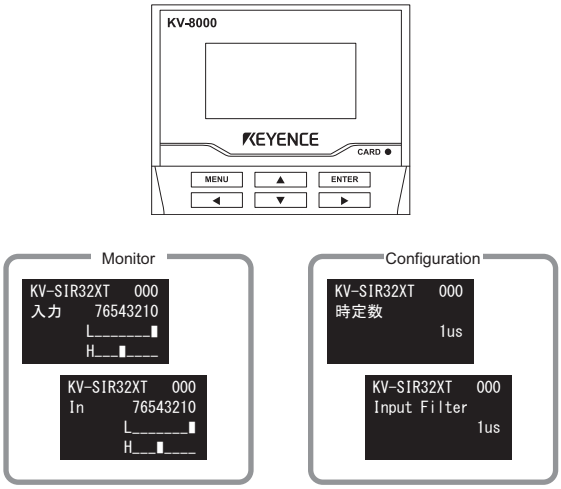
Ladder Conversion



6-9 Access Window

It describes the monitoring and modification procedure of the configured content of "KV-SIR32XT" by using the access window of the CPU unit.

Access Window



Regarding configuration modification with the access window

The part of configured content can be modified with the access window, but please use the unit editor in general.

Reference

When modifying configuration with the access window, execute with CPU unit in PROGRAM mode.
The actual modified configuration (modified content becomes active) is when mode of the CPU unit is switched to RUN mode.

Menu Structure

The menu structure of access window is as follows.

- Monitor
  - Input IN000 to IN015
  - Input IN100 to IN115
  - Output OUT200 to OUT215
  - Output OUT300 to OUT315
- Configuration Time Constant 000
- Configuration Time Constant 100
- Configuration Input Range

Operating Access Window

The access window operates with configuration operation key in the CPU unit.

Configuration Operation Key	In General	When Numerical Value is Modified
MENU	Menu Display	Menu Display
ENTER	It executes modification of configuration when pressed for 1 second or longer.	It executes modification of configuration when pressed for 1 second or longer.
▲	It switches page. It modifies select option when the configuration is modified.	It adds changed amount from the configured value.
▼	It switches page. It modifies select option when the configuration is modified.	It subtracts changed amount from the configured value.
◀	Menu Display	It modifies the digit of changed amount.
▶	Not Used.	

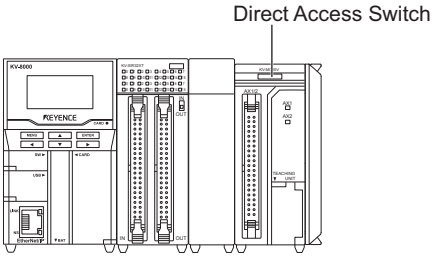
* The modification of configuration is possible only when CPU unit is in PROGRAM mode.

Migration to the menu screen

It describes the procedure of migrating to the menu screen.

1 Press the direct access switch of KV-SIR32XT to configure.

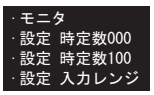
The direct access switch starts to blink and the monitor screen of IN000 to IN015 of the selected KV-SIR32XT will be displayed in the access window.

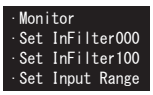


2 Press the "MENU" key of the CPU unit.

The menu screen will be displayed.

Scroll the menu screen with "▲" "▼".





Point

Press the "MENU" key again, or press the direct access switch that is blinking when returning to the menu screen of the CPU unit.

"5-9 Access Window"

Configuration Screen

It describes the modification procedure of the configured content of "KV-SIR32XT" by using the access window of the CPU unit.

Point **The modification of configuration in only possible when the CPU unit is in PRG mode.**

The method of modifying configuration

The part of configured content of "KV-SIR32XT" can be modified with the access window.
The procedure of modifying constant during input is as follows.

1 Select the category to configure from the menu screen with "▲" "▼" key and press the "ENTER" key.

The selected configuration screen will be displayed.

2 Press the "ENTER" key for 1 second or longer.

The configured value will blink.

3 Select with "▲" "▼" key and press the "ENTER" key for 1 second or longer.

Reference The category that can modify configuration and range of configuration is as follows.

Category	Range of Configuration
Constant during input 000	1μs/10μs/20μs/100μs/500μs/1ms/5ms/10ms/50ms
Constant during input 100	
Input range (100 to 115)	5V, 24V

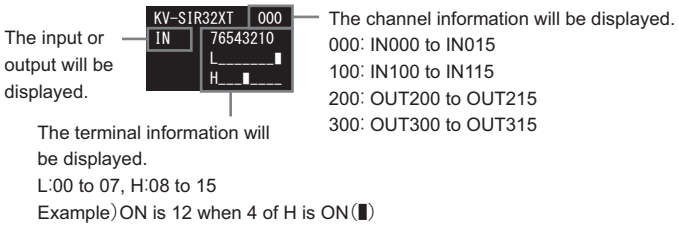
Monitor Screen

It describes the procedure of monitoring I/O status of "KV-SIR32XT" using the access window of CPU unit.

Method of Monitor

1 Press the "ENTER" key by selecting the channel to configure/monitor by using the "▲" "▼" key in the monitor screen from the menu screen.

The I/O status of selected channel will be displayed to the monitor screen.



Point

The modification of status is not possible from the monitor screen.
Please use the unit test function when there it is necessary to modify the output status.

Reference

The input, output status can be confirmed with the LED lamp of KV-SIR32XT.

Unit Test

It describes the procedure when testing the output status of "KV-SIR32XT" when using access window of the CPU unit.

Point **The unit test is only possible when the CPU unit is in PRG mode.**

Reference It can be confirmed even when the project that includes the configuration of unit structure that configures at KV STUDIO is not transferred to the CPU unit.

Test Method

1 **After pressing the "ENTER" key with "・Test" from the menu screen, select the channel to test with "▲" "▼" key and press "ENTER".**

The output test screen of the selected channel will be displayed.

2 **Select the terminal to test with "▲" "▼" "◀" "▶" key and press the "ENTER" key.**

The ON/OFF status of output of the selected terminal will switch.

6-10 Unit Monitor

The unit monitor function of "KV STUDIO" ladder support software is a function that optimally displays conforming to unit, the value of relay and DM used in the unit monitored.
The observation and operation method of unit monitor will be described as follows.

Summary of Unit Monitor

The unit monitor optimally displays conforming to unit, the value of relay and DM used in the unit monitored.

The following characteristic exist for unit monitor.
- The DM, relay allocated to each unit will be displayed in the initially registered status.
- The monitor is simple since the format of display content and layout is provided for each unit.
- The monitor of selected unit is possible just by selecting the unit to monitor, without worrying about the allocating status of relay and DM.

Point **The unit monitor can only be selected during monitor, online edit from the menu.**

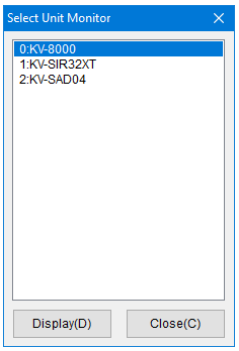
Displaying Unit Monitor

The unit to monitor can be displayed in the following procedure.

1 Select [monitor/simulator] [unit monitor] from the menu.

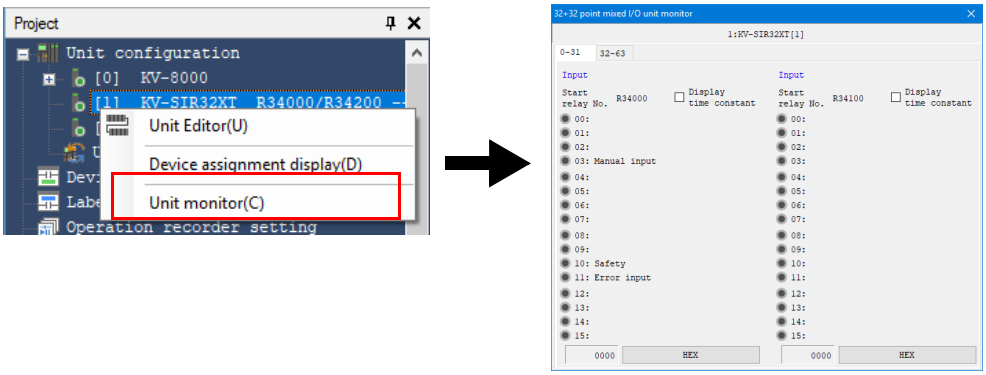
The “Unit Monitor Selection” dialog box will be displayed.

2 Click the "Display" button by selecting unit to monitor.



Alternative procedure

- Select unit to monitor with unit configuration of workspace and double click. Otherwise, select “Unit monitor” from right click menu.
- Select “Unit monitor” from the right click menu by selecting unit to monitor from the unit editor. Otherwise, select [Display] [Unit Monitor] from the menu.

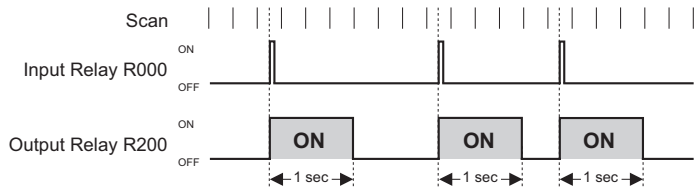


The following describes sample programs available to the "KV-SIR32XT."
The following program describes how input from R000 is applied to KV-SIR32XT.

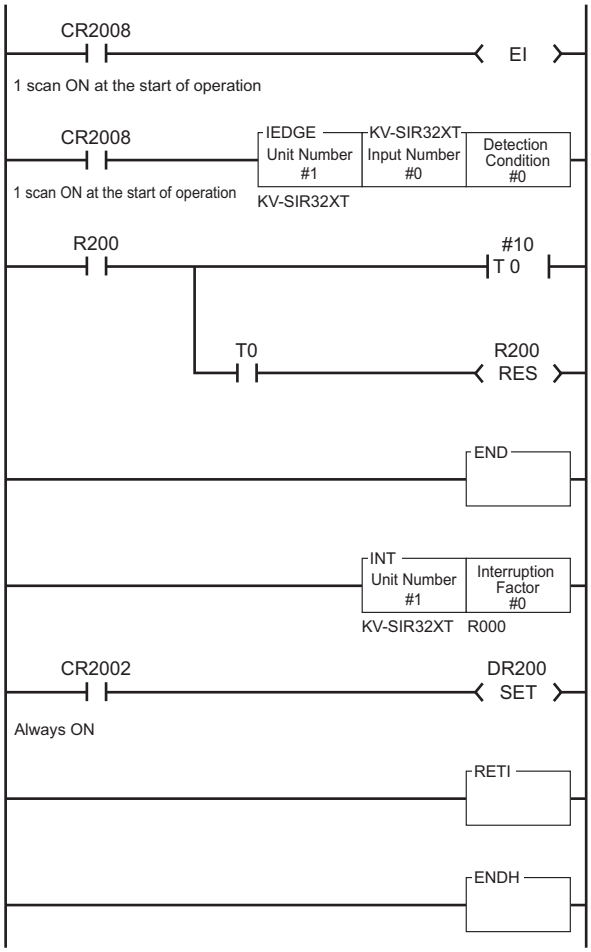
■ Application example of interrupts using a signal converter

In this example, interrupt processing is used to convert minute pulses into fixed width pulses, which are then output.

- The minute pulses applied to input relay R000 are converted to pulses with an ON width of 1 second, which are output from output relay R200.
- Output relay R200 is turned on for 1 second from the rising edge of input relay R000.



Interrupts Using a Signal Converter



When the power is turned on, interrupts are enabled.

The interrupt detection condition of input relay R000 is set to the rising edge.

Output relay R200 goes ON for 1 second only.

When input relay R000 goes ON, output R200 goes ON.

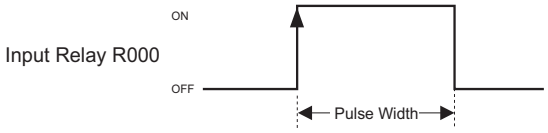
KV-SIR32XT setting for the Unit Editor
Leading DM No: -
Leading relay No: 000
IN000 input time constant: 1 μs

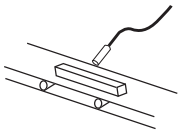
<Mnemonic List>

```
LD CR2008
EI
LD CR2002
IEDGE #1 #0 #0
LD R200
MPS
TMR #0 #10
MPP
ANB T0
RES R200
END
INT #1 #0
LD CR2002
SET DR200
RETI
ENDH
```

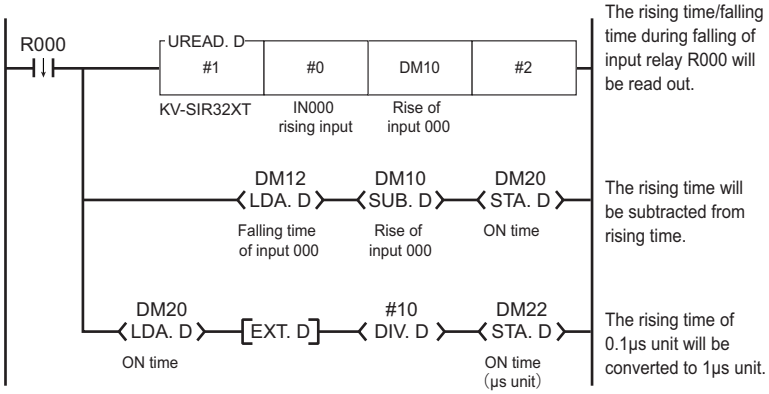
■ Application example of measuring the ON width of high-speed pulses

The pulse width of pulses applied to the input relay are measured.





- The input capture function is used to measure the pulse width in units of 0.1 μs.
- The rising time/falling time during falling of input relay R000 will be read out.
- It measures the ON time of the input relay by subtracting the rise time from the fall time.
- It converts ON time to μs unit.



The rising time/falling time during falling of input relay R000 will be read out.

KV-SIR32XT setting for the Unit Editor
Leading DM No: -
Leading relay No: 000
IN000 input time constant: 1 μs

The rising time will be subtracted from rising time.

The rising time of 0.1μs unit will be converted to 1μs unit.

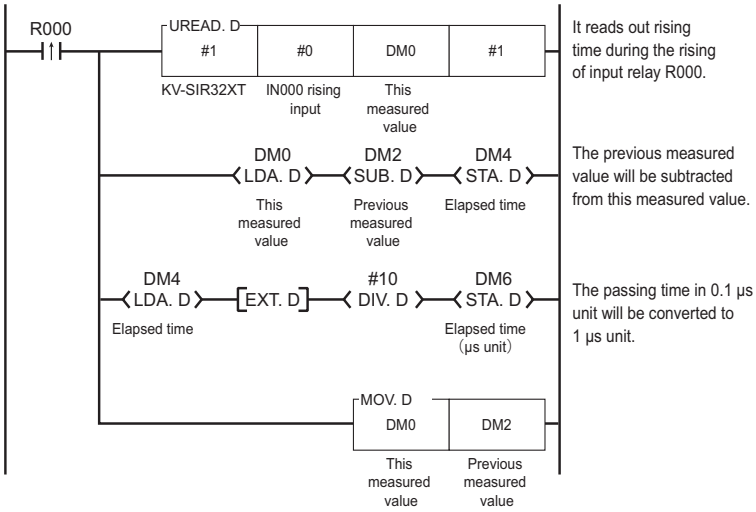
<Mnemonic List>

```
LDF R000
UREAD.D #1 #0 DM10 #2
MPS
LDA.D DM12
CON
SUB.D DM10
CON
STA.D DM20
MPP
LDA.D DM20
CON
EXT.D
CON
DIV.D #10
CON
STA.D DM22
```

■ Application example of measuring the ON width of high-speed pulses

The pulse width of pulses applied to the input relay are measured.

- It measures the passing interval from when the input relay R000 is turned ON until R000 is turned ON again.
- It reads out the rising time at the rise of R000.
- It stores the difference of this measured value and the previous measured value to data memory [DM6/DM7] in 1μs unit.
- It stores this measured value to the previous measured value.



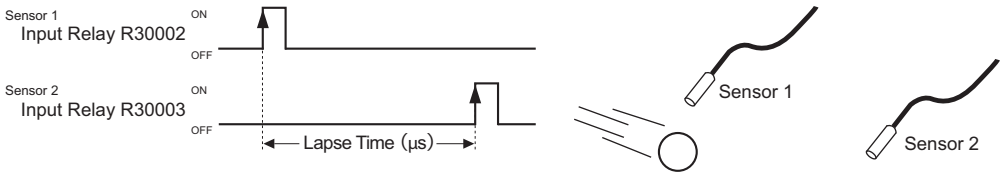
KV-SIR32XT setting for the Unit Editor
Leading DM No: -
Leading relay No: 000
IN000 input time constant: 1 μs

<Mnemonic List>

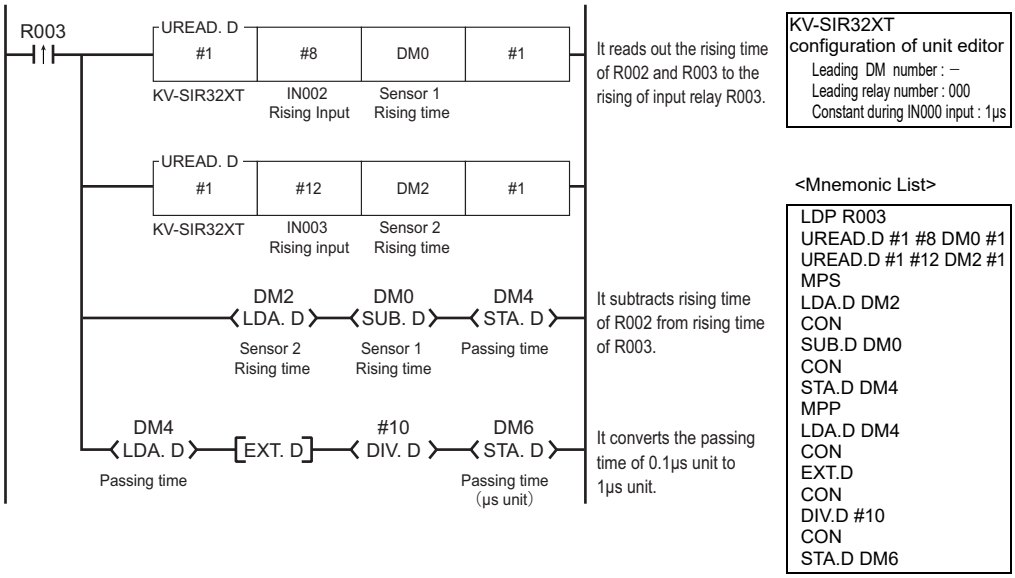
```
LDP R000
UREAD.D #1 #0 DM0 #1
MPS
LDA.D DM0
CON
SUB.D DM2
CON
STA.D DM4
MRD
LDA.D DM4
CON
EXT.D
CON
DIV.D #10
CON
STA.D DM6
MPP
MOV.D DM0 DM2
```

■ Example showing measurement of workpiece lapse interval

Measures the lapse interval when workpieces are detected.



- It measures the passing time from input relay R002 to input relay R003.
- It reads out the rising time of sensor 1 and sensor 2 with the rise of R003.
- It stores the difference of rising time of sensor 1 and sensor 2 in 1μs units to the data memory [DM6/DM7].



<Mnemonic List>

```
LDP R003
UREAD.D #1 #8 DM0 #1
UREAD.D #1 #12 DM2 #1
MPS
LDA.D DM2
CON
SUB.D DM0
CON
STA.D DM4
MPP
LDA.D DM4
CON
EXT.D
CON
DIV.D #10
CON
STA.D DM6
```

6-12 I/O Unit Error Mode

The error codes generated during “KV-SIR32XT” use are listed below.

No.	Message	Causes	Remedy
101	入力過電圧 IN100～IN115 Input over voltage IN100～IN115	The device uses IN100 - IN115 in the 5 V range.	Turn the power off, remove the cause of the problem, and then turn the power back on.
202	出力過電流 OUT200～OUT215 Output overcurrent OUT200～OUT215	It detects the overcurrent with either from OUT200 to OUT215.	Remove the cause of the problem.。
203	出力過電流 OUT300～OUT315 Output overcurrent OUT300～OUT315	It detects the overcurrent with either from OUT300 to OUT315.	Remove the cause of the problem.

MEMO

APPENDICES

The appendices include lists of the CPU unit control relays (CR) and control memory entries (CM), an error list, and a list of the instructions used in this manual.

1 List of PLC Unit Errors A-2
2 Lists of CR and CM Devices A-10
3 ASCII Code Table A-18
4 List of Instructions A-19
5 List of Unusable Instructions A-21
6 List of Abbreviated Mnemonics A-23
7 List of Shortcut Keys A-25
8 List of Characters That Cannot Be Used A-39
9 Precautions for Using the Existing KV Series Extension Unit A-42
10 Precautions When Changing From One KV-8000 Series Model to Another A-45
11 Index A-52

1 List of PLC Unit Errors

This appendix describes the errors that can occur on the CPU unit main unit.
You can use the following items to check for errors.

- Access Window
- KV STUDIO's error monitor
- CR/CM

Whether an error is a serious error or a minor error is determined by the error's group. When a serious error occurs, the ladder program stops, so it is not possible to clear the error from the ladder program.

Serious errors　Scan time over errors, system errors, and setting errors
Minor errors　Time errors, calculation errors, unit errors, memory card operation errors, memory card access errors, and other errors.

Reference
CR3905 turns ON when a serious error occurs, and the error code will be stored in CM5105.
CR3906 turns ON when a minor error occurs, and the error code is stored in CM5155.

Number	Group	Message	Cause	Remedy
10	Setting error	サブルーチン ネスト制限オーバー SBN Exceeds Nesting Limit 子程序嵌套超限	The nesting of subroutine (nesting) has gone over the limit.	The nesting limit of subroutine is 16 layers. Errors that occur during ladder execution will not occur during KV STUDIO conversion. Modify the ladder program, paying close attention to the proper usage of subroutines.
11	Setting error	FOR-NEXT ネスト制限オーバー FOR-NEXT Exceeds Nesting Limit FOR-NEXT 嵌套超限	The nesting (nesting) of FOR to NEXT instruction has gone over the limit.	The nesting limit of FOR~NEXT instruction is 16 layers. Errors that occur during ladder execution will not occur during KV STUDIO conversion. Modify the ladder program, paying close attention to the proper usage of FOR and NEXT instructions.
15	Setting error	コンバートエラー Failed To Convert 转换错误	The conversion operation failed.	Read the project from the main unit via KV STUDIO and convert it. Check the contents of the error and warning during conversion, and transfer the project to the main unit with the cause eliminated. * Some warnings may be cleared even if their causes are not eliminated.
22	Setting error	マクロ ネスト制限オーバー Macro Exceeds Nesting Limit 宏嵌套超限	The number of macros nested in multiple interrupts has exceeded the limit.	Reduce the number of macro instructions in the interrupt program.
29	System error	ラダースタック エラー Ladder Stack Error 梯形图堆栈错误	A system error has occurred.	Turn the power off and on again. If this error occurs frequently, contact your KEYENCE office.
30	Scan Time Over	スキャンタイム オーバー Scan Time Over 扫描超时	The scan time has reached or passed the configured time.	Modify the ladder program, checking to see if the scan time has been exceeded 300 ms because there are: • Repeatedly executed instructions such as FOR to NEXT instructions. • Instructions that cause processing to jump such as CJ or NCJ to LABEL. • Instructions that call programs such as CALL and MCALL. • Instructions with long processing times such as FMOV, WSR, and MAX. • Interrupt programs that are executed frequently.
31	Setting error	ラダースタック オーバーフロー Ladder Stack Overflow 梯形图堆栈溢出	A system error has occurred.	Turn the power off and on again. If this error occurs frequently, contact your KEYENCE office.

Number	Group	Message	Cause	Remedy
33	Setting error	プログラムネスト制限オーバー Program Exceeds Nesting Limit 程序 嵌套超限	The number of programs nested has exceeded the limit.	The nesting limit of program is 8 layers. Errors that occur during ladder execution will not occur during KV STUDIO conversion. Modify the ladder program, paying close attention to the proper usage of function block/function.
40	Setting error	ラダープログラムなし No Ladder Program 无梯形图程序	There is no program.	Transfer ladder programs to the main unit before starting operation.
50	Setting error	ユニット設定チェックサム異常 Unit Info Checksum Error 单元设定检验和异常	The configuration of expansion unit is damaged.	Please transfer the information of unit configuration to the main body. Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated.
51	Setting error	ユニット設定情報未設定 No Unit Info Set 单元设定信息未设定	The unit setting information has not been configured.	Transfer the unit setting information to the main unit.
52	Setting error	ユニット未接続 No Unit Connected 单元未连接	Expansion units that should be present according to the unit setting information are not actually connected.	Turn the power off, and then connect all the expansion units that have been set in the Unit Editor.
53	System error	拡張バスユニットタイムアウト Unit Timeout 扩展总线单元超时	The bus communication with the expansion unit was not possible for a period of time.	If a noise source (for example, a cable in which a large current flows such as a motor cable) is close to the units, distance the units from the noise source. If this error occurs frequently, contact your KEYENCE office.
54	Setting error	ユニット台数不一致 Number of Units Mismatch 单元台数不一致	The configuration of unit editor and the number of expansion unit connected actually differs.	Turn the power off, and then connect all the expansion units that have been set in the Unit Editor.
55	Setting error	ユニット種別不一致 Unit Type Mismatch 单元类型不一致	The configuration of unit editor and the type of actual expansion unit differs.	Turn the power off, and then connect all the expansion units that have been set in the Unit Editor.
56	System error	拡張バス通信エラー Expansion Bus Comm Error 扩展总线通信错误	The bus communication with the expansion unit was unsuccessful.	Make sure the expansion unit is connected properly. If a noise source (for example, a cable in which a large current flows such as a motor cable) is close to the units, distance the units from the noise source. If it occurs just before the power shutdown, it will not remain in the error log. If this error occurs frequently, contact your KEYENCE office.
57	Setting error	エンドユニット未接続 No End Unit 结束单元未连接	No end unit is connected.	Turn the power off and then connect an end unit.
59	Setting error	ユニットバージョンエラー Unit Version Error 单元版本错误	The function which can not be used with the version of expansion unit that is error connected is configured. Otherwise, the CPU unit that can not be connected to the expansion unit is connected.	Turn OFF power and connect the expansion unit configured with unit editor. Otherwise, if the unit corresponds to system version upgrade, update the system version to the latest.

Number	Group	Message	Cause	Remedy
63	Configuration error	電源再投入が必要 Power Restoring Needed 需要重新接通电源	The rebooting of power is necessary in order to reflect the modification of unit structure.	Please reboot power.
80	System error	FLASH ROM異常 Flash ROM Failure 闪存异常	The flash ROM is abnormal.	Please contact your KEYENCE office.
83	Time error	電池電圧の低下 Battery Voltage Degradation 电池电压降低	The voltage of KV-B1 is declining.	Please exchange KV-B1.
85	Time error	時計データ消失 Clock Data Lost 时钟数据缺失	The clock data has disappeared since the voltage of built-in condenser for backup has declined. The failure is due to no current flow for a long period of time or a hardware failure.	Please reconfigure the clock according to necessity. In order to retain the clock data, either charge the built-in condenser for backup by flowing current or attach the optional battery (KV-B1). If this error occurs frequently, contact your KEYENCE office.
86	System error	RTC 異常 RTC Failure RTC 异常	The clock data is abnormal.	If this error occurs frequently, contact your KEYENCE office.
87	Memory card access error	メモリカードアクセスエラー Memory Card Access Error 存储卡访问错误	Access to the memory card has failed due to the influence of external causes, such as memory card deterioration, abnormal contact point, abnormal insertion state or interference.	Check the insertion status of the memory card. Check the file system of the memory card. Use the chkdsk command on Windows command prompt to check the card. (Example) If the memory card is drive E C:\WINDOWS>CHKDSK E: Please keep it away from the device if noise origin (example: cable that flows large amount of current such as motor, etc.) is adjacent.
88	Memory card operation error	メモリカードが抜かれました Memory Card Removed 存储卡被取出	The memory card was removed while it was being accessed.	Do not remove the memory card while it is being accessed.
89	Memory card operation error	ストレージアクセス中電源OFF During Storage Access PowerOFF 存储体访问过程中断电	The power was cut off during storage access.	Do not turn off power during storage access.
90	Other error	ストレージアクセス完了待ち Waiting Storage Access 存储体访问完成等待	The switch to RUN mode was unsuccessful due to storage access in progress.	Please operate after the completion of storage access.
92	System error	アクセスウィンドウ異常 Access Window failure 访问窗异常	The communication with the access window has failed.	Clear the error and turn the power off and on again. If this error occurs frequently, contact your KEYENCE office.

Number	Group	Message	Cause	Remedy
95	Other error	ファイルアクセス中 File Access Busy 文件存取中	Functions that cannot be processed at the same time were executed.	During the execution of operation towards PLC transfer, load/save, all clear, operation towards user memory, please delay the timing of operation since PRG→RUN switch, unit test, all clear can not be implemented.
102	System error	POWER OFF エラー POWER OFF Error 断电错误	The data saving process during power cut off was not initialized successfully due to hardware failure, noise, etc.	If a noise source (for example, a cable in which a large current flows such as a motor cable) is close to the units, distance the units from the noise source. Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated. To clear the RAM: • Clear the PLC from KV STUDIO. • Use an expansion access window cassette. If this error occurs frequently, contact your KEYENCE office.
103	System error	電断処理超過 Shutdown Timeout 断电处理超时	A system error has occurred.	Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated. Contact your KEYENCE office.
106	System error	リセット異常 Abnormal Reset 复位异常	The data saving process during power cut off was not finished unsuccessful due to hardware failure, noise, etc.	If a noise source (for example, a cable in which a large current flows such as a motor cable) is close to the units, distance the units from the noise source. Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated. To clear the RAM: • Clear the PLC from KV STUDIO. • Access window If this error occurs frequently, contact your KEYENCE office.
107	Other error	システムエラー System Error 系统错误	A system error has occurred.	The system can continue operating, but this may lead to a serious error occurring. Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated. If this error occurs frequently, contact your KEYENCE office.
108	Convert Warning	コンバートワーニング Convert Warning 转换警告	A warning occurred during conversion.	The project information at the time of operation recorder setting creation is different from the project information of the main unit. The project should be read from the main unit via KV STUDIO, and transferred to the main unit after conversion. The content of the warning displayed during conversion should be checked when it cannot be cleared by the above method, and the project should be transferred to the main unit after eliminating the cause. * Some warnings may be cleared even if their causes are not eliminated.
118	System error	不正な伝送データ Invalid Data 非法数据传输	A system error has occurred.	Turn the power on again. If this error occurs frequently, contact your KEYENCE office.

Number	Group	Message	Cause	Remedy
120	System error	ウォッチドッグタイマ Watchdog Timer 看门狗定时器	A system error has occurred.	Turn the power on again. Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated. If this error occurs frequently, contact your KEYENCE office.
122	System error	システムエラー System Error 系统错误	A system error has occurred.	Turn the power on again. Keep away from the equipment when it is installed in a place where the strong electric field or strong magnetic field is generated. If this error occurs frequently, contact your KEYENCE office.
125	Other error	エラーバッファオーバーフロー ErrorBuffer Overflow 错误缓存溢出	The number of errors occurring simultaneously in the system has gone over the limit..	Make sure that no more than 16 errors occur at the same time. Clear the errors.
127	Setting error	オートローディング失敗 Auto Loading Failed 自动加载失败	The automatic loading operation failed.	Data cannot be loaded when no data is present or when the data is from a different model PLC. The automatic loading function cannot be used when the projects stored on the PLC are write protected. In situations other than those described above, check the error code during the execution of load, save. " The error code during the execution of load, save" (page 5-19)
128	Calculation error	ラダー演算エラー Ladder Calc 梯形图运算错误	A calculation error (an error that turns CR2012 ON) has occurred.	Check the ladder program, and correct the cause of the calculation error.
129	Unit error	ユニットエラー Unit Error 单元错误	The error has occurred in the expansion unit.	See the manuals of the units. For details on how to check the error unit, refer to " Error Clear" (page 5-127).
131	System error	FLASH ROMフォーマット FLASH ROM Format 闪存格式	The format of flash ROM is abnormal due to power cut off during all clear, hardware failure, noise, etc.	If FLASH ROM Format occurs, the FLASH ROM may be formatted for around 20 minutes. After 20 minutes have passed check that the error has been solved, execute all clear, and then send the project again. If this error occurs frequently, contact your KEYENCE office.
132	System error	ラダーファイル不正 Invalid Ladder File 梯形图文件不正确	The ladder stored in flash ROM is abnormal due to power cut off during all clear, hardware failure, noise, etc.	Perform an all clear operation, and then transfer the project to the PLC again. If this error occurs frequently, contact your KEYENCE office.
135	Memory card operation error	メモリカードカバーオープン Memory Card Cover Open 存储卡外盖打开	The memory card cover has been opened during memory card access.	Close the memory card cover.
136	Memory card operation error	メモリカードLOCKされています Memory Card Locked 存储卡被锁定	The memory card has been LOCK (write protect）during write in process of the memory card.	Remove the lock (write protection) or use a memory card that can be written to.

Number	Group	Message	Cause	Remedy
137	Memory card operation error	非対応の メモリカード種類 Unsupported Memory Card Type 不支持的存储卡种类	A memory card that is not supported was inserted.	Use an SD card or an SDHC card.
141	Other error	CPUメモリ 書込頻度オーバー CPU Memory Write Frequency Over CPU存储器 写入频度超出	The write in to CPU memory was executed in high frequency.	Reduce the frequency with which data is written to the CPU memory.
142	Other error	CPUメモリ 累積書込回数警告 CPU Memory Write Cycle Warning CPU存储器累积写入次数警告	The accumulated write in frequency to CPU memory is nearing its durability.	Please decrease the write in frequency to CPU memory. In addition, please prepare the CPU unit to exchange.
143	Other error	CPUメモリ 累積書込回数オーバー CPU Memory Write Cycle Over CPU存储器 累积写入次数超出	The accumulated write in frequency to CPU memory has passed its durability.	Read the data in the CPU unit memory and replace the CPU unit.
250	Other error	Start Up Error Please Reset Start Up Error Please Reset Start Up Error Please Reset	An error occurred during startup.	Please turn on the power again. (The error can not be cleared, or it will not remain in the error log.) If it occurs frequently, please contact the nearest sales office.
251	Other error	AW Timeout Error AW Timeout Error AW Timeout Error	Communication with the access window could not be performed normally due to hardware failure or noise.	If the vibration source or noise source (eg, a cable that carries a large current such as a motor) comes close to it, keep it away from the machine and turn the power on again. If it occurs frequently, please contact the nearest sales office.
—	Other error	Safe Mode Safe Mode Safe Mode	The system program is abnormal.	If a communication error occurs when updating the system program from KV STUDIO, update the system program again using another computer or another USB cable. If it occurs frequently, or cannot communicate with KV STUDIO, please contact your nearest sales office.

- Error number 128 (calculation error)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5157)	Error details 0 (Not used.) 1 Incorrect indirect specification. 2 The indirectly specified target is T/C. 3 Incorrect simple indirect specification. 4 Outside of operand range. 5 Floating-point real number overflow. 6 Incorrect device specification. 7 The ROOT instruction operand is negative. 8 No bits turned ON with the ENCO instruction. 9 Floating-point real number format error. 10 Incorrect trigonometric function instruction operand. 11 Data could not be converted. 12 Incorrect table specification. 13 Incorrect time comparison instruction operand. 14 Incorrect cam switch operand. 15 Incorrect frequency counter instruction operand. 16 Divide by zero. 17 Illegal logging ID. 18 Direct processing error. 19 Unit on which direct processing cannot be performed. 20 Jump destination subroutine does not exist. 21 Data being written during RUN mode, so instruction cannot be executed. 22 Error for expansion unit-dedicated instructions (unit number). 23 Reserved for the system. 24 PIDAT, outside of operation setting range. 25 PIDAT, outside of sampling period range. 26 PIDAT, outside of comparison constant range. 27 PIDAT, outside of integral constant range. 28 PIDAT, outside of differential constant range. 29 PIDAT comparison constant, two-degree freedom parameters outside of range. 30 PIDAT integral constant, two-degree freedom parameters outside of range. 31 PIDAT differential constant, two-degree freedom parameters outside of range. 32 PIDAT, outside of operation quantity upper limit or lower limit range. 33 PIDAT, outside of operation quantity change width limit range. 34 PIDAT, outside of measured value change width range. 35 PIDAT, AT state transition error. 36 PIDAT, outside of operation control flag range. 37 Operation period exceeded. 38 PIDAT, outside of AT adjustment parameter range. 39 PIDAT, outside of AT hysteresis range. 40 PIDAT, outside of AT timeout range. 41 PIDAT, outside of AT calculation control mode setting range. 42 PIDAT, AT timeout. 43 PIDAT, AT comparison constant exceeded. 44 PIDAT, AT integral constant exceeded. 45 PIDAT, AT differential constant exceeded. 46 Root of negative number cannot be calculated. 47 Incorrect ABS instruction operand. 48 Calculation overflow. 49 UR, UM, or UV unit not connected. 50 Unit device (R) not assigned. 51 Unit device (DM) not assigned. 52 Unit categories disagree. 53 operation recorder ID invalid 54 Stack overflow
Detailed information, piece 2 (CM5158)	Fixed to 00H.
Detailed information, piece 3 (CM5159)	Upper bits of the step number at which the error occurred.
Detailed information, piece 4 (CM5160)	Lower bits of the step number at which the error occurred.

Detailed Information, Piece n	Description
Detailed information, pieces 5 to 20 (CM5161 to CM5176)	The module name is saved according to settings made in the project language. (Data other than the module name is 00H.)

- Error numbers 50,53,55,56,58,59,63（Error relating to unit number）

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5107)	Unit number (the unit number will not be stored if it cannot be determined according to the occurrence conditions of the error)
Detailed information, pieces 2 to 20 (CM5108 to CM5126)	Fixed to 00H

- Error number 54 (number of units error)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5107)	The number of units as specified in the unit setup information
Detailed information, piece 2 (CM5108)	The number of units that are actually connected
Detailed information, pieces 3 to 20 (CM5109 to CM5126)	Fixed to 00H

- Error number 127 (auto loading failure error)

Detailed Information, Piece n	Description
Detailed information, piece 1 (CM5157)	Detailed content of the error Please refer to 📖 " The error code during the execution of load, save" (page 5-143) for details.
Detailed information, piece 2 (CM5158)	The unit number when detailed information piece 1 is 20 Fixed at 00H when the above do not apply
Detailed information, pieces 2 to 20 (CM5159 to CM5176)	Fixed to 00H

- Error number 129 (unit error)

Expansion Information, Piece n	Description
Expansion information, piece 1 (CM5157)	Unit number
Expansion information, piece 2 (CM5158)	Error number of each unit
Detailed information, pieces 3 to 20 (CM5159 to CM5176)	Fixed to 00H

- Error numbers other than those listed above

Expansion Information, Piece n	Description
Detailed information, pieces 1 to 20	Fixed to 00H

Control Relays (CR)

Control relays (CR0000 - CR7915) can be used to perform operations such as checking the operation status of the CPU unit and setting and monitoring the built-in functions.
Control relays reserved for the system cannot be used.

CR	Attribute R: Read only Blank: Read/write	Function	
CR0000		Logging or tracing function (ID0)	Logging and tracing execution (ON: enabled, OFF: disabled)
CR0001	R		Logging or tracing operation in progress
CR0002	R		Logging or tracing file writing completion
CR0003	R		Logging buffer less than 50% warning
CR0004	R		Logging buffer overflow
CR0005	R		Data to be written in RUN mode during logging present
CR0006 and CR0007	—		Reserved for the system
CR0008	R		Logging or tracing error
CR0009	R		Logging trace No memory space error
CR0010	R		Logging or tracing device or trigger setting error
CR0011	R		Tracing file save trigger monitor
CR0012	R		Tracing data capture completion
CR0013	R		Logging or tracing setting to be written present
CR0014 and CR0015	—		Reserved for the system
CR0100 to CR0915		Logging or tracing function (ID1 to ID9)	
CR1000 to CR1015		Real-time chart monitor (ID10)	
CR1100 to CR2001	—	Reserved for the system	
CR2002	R	Always turned ON	
CR2003	R	Always turned OFF	
CR2004	R	10 ms clock pulse (duty cycle: 50%)	
CR2005	R	100 ms clock pulse (duty cycle: 50%)	
CR2006	R	1 s clock pulse (duty cycle: 50%)	
CR2007	R	Turned OFF for just one scan when operation starts	
CR2008	R	Turned ON for just one scan when operation starts	
CR2009		Turned ON when the calculation result is negative or when an overflow occurs	
CR2010	R	Turned ON when the calculation result is zero	
CR2011	R	Turned ON when the calculation result is positive	
CR2012	R	Turned ON when a calculation execution error occurs	
CR2013 to CR2302	—	Reserved for the system	
CR2303		Fixed scan time operation (ON: enabled, OFF: disabled)	
CR2304	R	Turned ON for just one scan when the scan time setting (the fixed scan time operation and END processing time settings) is exceeded	
CR2305	—	Reserved for the system	
CR2306	R	Indicates backup battery errors (ON: error, OFF: normal)	
CR2307		Clears the minimum and maximum scan time values	
CR2308		Sets the END processing time (ON: enabled, OFF: disabled; if CR2303 is turned ON at the same time, CR2303 is given priority)	
CR2309 and CR2310	—	Reserved for the system	
CR2311	R	Indicates RTC low voltage error (ON: error has occurred, OFF: normal)	
CR2312	—	Reserved for the system	
CR2313		Forces switch of the VT sensor application screen (ON: switch, OFF: do not switch)	
CR2314	R	Project password authentication status (ON: authentication disabled, OFF: authentication enabled)	
CR2315 to CR2813	—	Reserved for the system	

CR	Attribute R: Read only Blank: Read/write	Function
CR2814		Zero suppression during execution of DASC and FASC instructions (all suffixes; ON: execute, OFF: do not execute)
CR2815		Omission of "+" during execution of DASC.S, DASC.L, and FASC instructions (ON: execute, OFF: do not execute)
CR2900 to CR3005	—	Reserved for the system
CR3100	R	Inter-unit synchronization refreshing time exceeded
CR3101 to CR3115	—	Reserved for the system
CR3200		CPU memory save save request
CR3201	R	CPU memory save being performed
CR3202	R	CPU memory save complete
CR3203	R	CPU memory save error
CR3204 to CR3206	—	Reserved for the system
CR3207		Instruction RAM mode operation The RAM mode activation of file register instructions, storage device instructions towards the CPU memory. (ON: enable, OFF: disable)
CR3208 to CR3209	—	Reserved for the system
CR3210	R	Arbitrary logging/trace operation
CR3211 to CR3213	—	Reserved for the system
CR3214	R	Storage instruction executing
CR3215	—	Reserved for the system
CR3300 to CR3415		Alarm relay
CR3500	R	Alarm operation relay (ON: one of CR3300 to CR3415 is turned ON)
CR3501		Record an entry in the log when an alarm relay switches from ON to OFF (ON: record, OFF: do not record)
CR3502		The alarm relay log is cleared when this switches from OFF to ON
CR3503 to CR3512	—	Reserved for the system
CR3513	R	Logging internal buffer overflow
CR3514		HKEY instruction - Disables multiple keys
CR3515	R	HKEY instruction - Scan completion
CR3600 to CR3615	R	HKEY information storage area
CR3700 to CR3815	—	Reserved for the system
CR3900		The serious error log is cleared when this switches from OFF to ON
CR3901		The minor error log is cleared when this switches from OFF to ON
CR3902	R	Indicates that the serious error log is full
CR3903	R	Indicates that the minor error log is full
CR3904	R	RUN-PRG selector switch status (ON: RUN, OFF: PROG)
CR3905	R	A serious error has occurred
CR3906	R	A minor error has occurred
CR3907 and CR3908		Reserved for the system
CR3909		Clears the current error (Set this relay in the ladder program)
CR3910	—	Clear all error and event log when OFF→ON
CR3911 to CR4000		Reserved for the system
CR4001 to CR4300	R	Expansion unit status (expansion units 1 - 48)
CR4301 to CR4315	—	Reserved for the system
CR4400	R	AW up key being pressed
CR4401	R	AW down key being pressed
CR4402	R	AW left key being pressed

CR	Attribute R: Read only Blank: Read/write	Function
CR4403	R	AW right key being pressed
CR4404	R	AW MENU key being pressed
CR4405	R	AW ENT key being pressed
CR4406 to CR4414	—	Reserved for the system
CR4415	R	AW user message being displayed
CR4500		AW line 1 blinking
CR4501		AW line 2 blinking
CR4502		AW line 3 blinking
CR4503		AW line 4 blinking
CR4504 to CR4600	—	Reserved for the system
CR4601 to CR4915		AW user message 1 to 63
CR5000 to CR5315	—	Reserved for the system
CR5400		Memory card mount request
CR5401		Memory card unmount request
CR5402	R	Memory card used
CR5403	R	Memory card recognized
CR5404	R	Memory card is available
CR5405	R	Memory card is write protected
CR5406 to CR5407	—	Reserved for the system
CR5408		CPU memory mount request
CR5409		CPU memory unmount request
CR5410	R	CPU memory used
CR5411	R	CPU memory recognized
CR5412 to CR5415	—	Reserved for the system
CR5500		Project load execution request
CR5501	—	Reserved for the system
CR5502		Project save execution request
CR5503	—	Reserved for the system
CR5504	R	Project load execution complete
CR5505	R	Project load execution failure
CR5506	R	Project save execution complete
CR5507	R	Project save execution failure
CR5508		Recipe load request
CR5509	—	Reserved for the system
CR5510	R	Recipe load complete
CR5511	R	Recipe load failed
CR5512		Recipe save request
CR5513	—	Reserved for the system
CR5514	R	Recipe save complete
CR5515	R	Recipe save failed

CR	Attribute R: Read only Blank: Read/write	Function	
CR5600		operation recorder function (ID0)	Operation record 0 Save request
CR5601	R		Operation record 0 Operating
CR5602	R		Operation record 0 Save trigger monitor
CR5603	R		Operation record 0 Saving
CR5604	R		Operation record 0 Collecting
CR5605	R		Operation record 0 Setting error
CR5606	—		System reservation
DR5607	R		Operation record 0 Save completed
CR5608	R		Operation record 0 Save error
CR5609	R		Device record 0 Buffer shortage
CR5610 to CR5613	—		System reservation
CR5614			Operation record 0 Buffer area full save mode
CR5615	—		System reservation
CR5700 to CR5919		operation recorder function(ID1 to ID3)	
CR6000 to CR7915		Reserved for the system	

Control Memory Entries CM

Control memory entries (CM0000 - CM7599) can be used to perform operations such as checking the operation status of the CPU unit and setting and monitoring the built-in functions.
Control memory entries reserved for the system cannot be used.

CM	Attribute R: Read only Blank: Read/ write	Function	Hold Yes: Hold
CM0000 and CM0699	–	Reserved for the system	
CM0700	R	Read from RTC (year)	
CM0701	R	Read from RTC (month)	
CM0702	R	Read from RTC (day)	
CM0703	R	Read from RTC (hour)	
CM0704	R	Read from RTC (minute)	
CM0705	R	Read from RTC (second)	
CM0706	R	Read from RTC (day of the week)	
CM0707 to CM709	–	Reserved for the system	
CM0710	R	File register current bank number	
CM0711	–	Reserved for the system	
CM0712	R	Computing error detailed information	
CM0713 to CM0899	–	Reserved for the system	
CM900	R	CPU function version ($0202 in hexadecimal notation indicates version 2.02)	
CM901 to CM906	–	Reserved for the system	
CM907	R	Project Version ($0202 in hexadecimal notation indicates version 2.02)	
CM908 to CM999	–	Reserved for the system	
CM1000 and CM1001	R	Scan time measuring value (1 μs) (32-bit)	
CM1002 and CM1003	R	END processing time measured value (1 μs unit) (32-bit)	
CM1004 and CM1005	R	Minimum fixed scanning time (1 μs unit) (32-bit)	
CM1006 and CM1007	R	Maximum fixed scanning time (1 μs unit) (32-bit)	
CM1008 and CM1009		Fixed scanning time run set value (1 μs unit) (32-bit)	
CM1010 and CM1011		END processing time set value (1 μs unit) (32-bit)	
CM1012 and CM1013	R	Scanning time when setting is exceeded (1 μs unit) (32-bit)	
CM1014 and CM1015	R	END processing time exceeds value (1 μs unit) (32-bit)	
CM1016 to CM1029	–	Reserved for the system	
CM1030 and CM1031	R	32-bit 1 ms Free Run counter	
CM1032 and CM1033	R	32-bit 10 μs Free Run counter	
CM1034 and CM1035	R	32-bit 1 μs Free Run counter	
CM1036 and CM1037	R	32-bit 0.1 μs Free Run counter	
CM1038 to CM1499	–	Reserved for the system	
CM1500 to CM1509	R	Log/trace (ID0 to ID9) File Save Counter	

CM	Attribute R: Read only Blank: Read/ write	Function	Hold Yes: Hold
CM1510 to CM1629	—	Reserved for the system	
CM1630 to CM1639	R	Log/trace Ring Buffer minimum free space (ID0 to ID9) (KB)	
CM1640 to CM1649	—	Reserved for the system	
CM1650	R	Number of sensor setting instruction executions	
CM1651 to CM1657	—	Reserved for the system	
CM1658	R	Number of consecutive project password authentication failures	Yes
CM1659	R	Total number of project password authentication failures	Yes
CM1660 to CM1669	R	Logging, trace ring buffer free capacity (ID0 - ID9) (in KB units)	
CM1670 to CM1679	—	Reserved for the system	
CM1680 to CM1689	R	Log/trace File Save Counter (ID0 to ID9) (Memory card use)	Yes
CM1690 to CM1699	—	Reserved for the system	
CM1700 and CM1701		Digital trimmer 0 upper limit (32-bit)	
CM1702 and CM1703		Digital trimmer 1 upper limit (32-bit)	
CM1704 and CM1705		Digital trimmer 2 upper limit (32-bit)	
CM1706 and CM1707		Digital trimmer 3 upper limit (32-bit)	
CM1708 and CM1709		Digital trimmer 4 upper limit (32-bit)	
CM1710 and CM1711		Digital trimmer 5 upper limit (32-bit)	
CM1712 and CM1713		Digital trimmer 6 upper limit (32-bit)	
CM1714 and CM1715		Digital trimmer 7 upper limit (32-bit)	
CM1716 to CM1719	—	Reserved for the system	
CM1720	R	AW user message display number	
CM1721 to CM1735	—	Reserved for the system	
CM1736		AW initial display language setting	
CM1737		AW initial display comment number setting	
CM1738		Access window operation disabled setting	
CM1739	—	Reserved for the system	
CM1740 to CM1772	R	Ladder project name (up to 32 characters + end code)	
CM1773 to CM1779	—	Reserved for the system	
CM1780		Access window initial screen setting, enabled	
CM1781		Access window initial screen setting, device type	
CM1782		Access window initial screen setting, screen ID	
CM1783		Access window initial screen setting, screen ID	
CM1784		Access window initial screen setting, device number	
CM1785		Access window initial screen setting, device number	
CM1786		Access window initial screen setting, display format	
CM1787		Access window initial screen setting, device name representation	
CM1788		Access window initial screen setting, key lock status	
CM1789	—	Reserved for the system	
CM1790		Automatic loading folder number (request)	
CM1791	R	Automatic loading complete code	
CM1792	R	Automatic loading folder number (complete)	
CM1793	—	Reserved for the system	
CM1794		Run load folder number (request)	

CM	Attribute R: Read only Blank: Read/ write	Function		Hold Yes: Hold
CM1795	R	Run load complete code		
CM1796	R	Run load folder number (complete)		
CM1797 to CM1868	—	Reserved for the system		
CM1869	R	Number of connected expansion units		
CM1870		Recipe load Drive number		Yes
CM1871		Recipe load Folder number		Yes
CM1872	—	Reserved for the system		
CM1873	R	Recipe load Completion code		Yes
CM1874	—	Reserved for the system		
CM1875		Recipe save Drive number		Yes
CM1876		Recipe save Reference folder number		Yes
CM1877		Recipe save Output folder number		Yes
CM1878	R	Recipe save Completion code		Yes
CM1879	—	Reserved for the system		
CM1880		Project load drive number (request)		Yes
CM1881		Project load folder number (request)		Yes
CM1882	R	Project load drive number (complete)		Yes
CM1883	R	Project load folder number (complete)		Yes
CM1884	R	Project load completion code		Yes
CM1875 to CM1877	—	Reserved for the system		
CM1888		Project save drive number (request)		Yes
CM1889		Project save folder number (request)		Yes
CM1890	R	Project save drive number (complete)		Yes
CM1891	R	Project save folder number (complete)		Yes
CM1892	R	Project save completion code		Yes
CM1893 to CM1999	—	Reserved for the system		
CM2000 to CM2001	R	operation recorder function (ID0)	Device record 0 Average sampling period	
CM2002 to CM2003	R		Device record 0 Average number of changed devices	
CM2004 to CM2005	R		Device record 0 Maximum collection period	
CM2006 to CM2009	—		Reserved for the system	
CM2010 to CM2039		operation recorder function (ID1 to ID3)		
CM2040 to CM2389	—	Reserved for the system		
CM2390	R	Error code for storage device instruction		
CM2391		Power OFF during storage access （Becomes value other than 0 during occurrence. Please change it to 0 from the program.）		
CM2392		Memory card retry occurrence times (still hold in case of power off)		
CM2393		Memory card check error times (still hold in case of power off)		
CM2394		Memory card timeout occurrence times (still hold in case of power off)		
CM2395		Reset delay time when error occurs on the memory card		
CM2396 to CM2399	—	Reserved for the system		
CM2400 to CM2401	R	Inter-unit synchronization refreshing beginning time (0.1μs)		
CM2402 to CM2403	R	Inter-unit synchronization refreshing finishing time (0.1μs)		
CM2404 to CM2405		Inter-unit synchronization refreshing beginning time maximum (0.1μs) (To reset, please change it to 0 from this program.)		Yes
CM2406 to CM2407		Inter-unit synchronization refreshing finishing time maximum (0.1μs) (To reset, please change it to 0 from this program.)		Yes
CM2408 to CM2409	R	Inter-unit synchronization refreshing operation time (0.1μs)		

CM	Attribute R: Read only Blank: Read/ write	Function		Hold Yes: Hold
CM2410 to CM5099	–	Reserved for the system		
CM5100	R	Details of the latest serious error	Year and month (Indicated as decimal values. The last two digits show Month, and previous digits from hundred show Year. Example: 1210 → October 2012)	Yes
CM5101	R		Day	Yes
CM5102	R		Hour	Yes
CM5103	R		Minute	Yes
CM5104	R		Second	Yes
CM5105	R		Error number	Yes
CM5106	R		Number of valid pieces of data	Yes
CM5107	R		Detailed information, piece 01	Yes
CM5108	R		Detailed information, piece 02	Yes
:	R			Yes
CM5126	R		Detailed information, piece 20	Yes
CM5127 to CM5149	–	Reserved for the system		
CM5150	R	Details of the latest minor error	Year and month (Indicated as decimal values. The last two digits show Month, and previous digits from hundred show Year. Example: 1210 → October 2012)	Yes
CM5151	R		Day	Yes
CM5152	R		Hour	Yes
CM5153	R		Minute	Yes
CM5154	R		Second	Yes
CM5155	R		Error number	Yes
CM5156	R		Number of valid pieces of data	Yes
CM5157	R		Detailed information, piece 01	Yes
CM5158	R		Detailed information, piece 02	Yes
:	R			Yes
CM5176	R		Detailed information, piece 20	Yes
CM5177 to CM5871	–	Reserved for the system		
CM5872 to CM5873	R	Used amount of CPU memory（32 bit, Byte unit）		Yes
CM5874 to CM5999	–	Reserved for the system		
CM6000		operation recorder function (ID0)	Operation record 0 Save completion code	
CM6001	–		Reserved for the system	
CM6002			Operation record 0 Save counter	
CM6003	–		Reserved for the system	
CM6004	R		Operation record 0 Save progress detail 1	
CM6005	R		Operation record 0 Save progress detail 2	
CM6006	R		Operation record 0 Save progress	
CM6007 to CM6009	–		Reserved for the system	
CM6010 to CM6171	R		Operation record 0 Save path	
CM6172 to CM6199	–		Reserved for the system	
CM6200 to CM6799		operation recorder function (ID1 to ID3)		
CM6800 to CM7599	–	Reserved for the system		

ASCII Code Table

Lower four bits \ Upper four bits	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p				ー	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
2	STX	DC2	”	2	B	R	b	r			「	イ	ツ	メ		
3	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
4	EOT	DC4	$	4	D	T	d	t			、	エ	ト	ヤ		
5	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
9	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A	LF	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C	FF	→	,	<	L	\	l	|			ャ	シ	フ	ワ		
D	CR	←	—	=	M	]	m	}			ュ	ス	ヘ	ン		
E	SO	↑	.	>	N	^	n	~			ョ	セ	ホ	゛		
F	SI	↓	／	?	O	_	o	DEL			ッ	ソ	マ	゜		

4 List of Instructions

This section gives a list of instructions and the processing outlines of those instructions. For details, see the pages noted for each instruction.
The following execution condition symbols are used: "⎍" indicates execution during each scan, "⎍" indicates execution for only one scan when a rising edge occurs, and "⎍" indicates execution for only one scan when a falling edge occurs.

Classification	Mnemonic	Symbol	Processing Outline	Execution Condition	Description Page
Logging instructions					
Logging	LOGE	n LOGE	Enables the execution of logging for the specified logging ID	⎍	5-75
	LOGD	n LOGD	Disables the execution of logging for the specified logging ID	⎍	5-75
	TRGD	n TRGD	Captures the logging data of the specified ID	⎍	5-77
operation recorder instructions					
Operation recorder	TRGR	n TRGR	Collect the operation recorder device data of the specified ID.	⎍	5-105
Cam switch instructions					
Cam switch	MCMP	MCMP S1 S2 D n	Performs multi-stage comparator operations	⎍	5-179
	ABSENC	ABSENC S1 S2 D1 D2	Performs cam operations using an absolute encoder	⎍	5-183
High-speed processing instructions					
Interrupts	DI	DI	Disables the execution of interrupt programs	⎍	4-218
	EI	EI	Enables the execution of interrupt programs	⎍	4-218
	DIC	D DIC	Sets the interrupt disabled range	⎍	4-220
	INT	INT	Executes the interrupt program from this instruction to the RETI instruction	—	4-222
	RETI	RETI	Indicates the end of the interrupt program	—	4-222
Module instructions					
Modules	MDSTRT	MDSTRT S	Starts the execution of the specified module	⎍	4-115
	MDSTOP	MDSTOP S	Stops the execution of the specified module	⎍	

Classification	Mnemonic	Symbol	Processing Outline	Execution Condition	Description Page
Function block instructions					
Function blocks	FBCALL	S2 S1 EN ENO Based on the function block to be executed	Executes the specified function block.	⎍	4-151
	FBSTRT	S2 S1 EN ENO Based on the function block to be executed	Executes the specified function block.	⎍	4-151
	FB	S Varies depending on the function block to be executed	Executes the specified function block.	⎍	4-148
	FEND	FEND	Stops execution of the function block.	⎍	4-148
	FUN	S Varies depending on the function block to be executed	Executes the specified function.	⎍	4-148
Macro instructions					
Macros	MCALL	MCALL Varies depending on the macro to be executed	Executes the specified subroutine macro	⎍	4-98
	MSTRT	MSTRT Varies depending on the macro to be executed	Executes the specified self-hold macro	⎍	
	MEND	MEND	Stops execution of the self-hold macro	⎍	

5 List of Unusable Instructions

This section describes the instructions that cannot be used in initialize modules, function blocks, interrupt programs, macros, and subroutine programs.

● Initialize module, interrupt program (fixed period module and inter-unit synchronization module are excluded)

Instruction Type		Instruction	
Common		@XXXX instruction	Differential execution type
Base instructions	Contact instructions	LDP and LDF	Load pulse and load pulse fall
		ANP and ANF	AND pulse and AND pulse fall
		ORP and ORF	OR pulse and OR pulse fall
		LDPB and LDFB	Load pulse bar and load pulse fall bar
		ANPB and ANFB	AND pulse bar and AND pulse fall bar
		ORPB and ORFB	OR pulse bar and OR pulse fall bar
	Output instructions	DIFU and DIFD	Differential up and differential down
		ONDL and OFDL	On delay and off delay
		SHOT	One shot
		FLIK	Flicker
		ALT	Alternate
	Timer and counter instructions	TMR, TMH, TMS, and TMU	Timer, high-speed timer, 1 ms high-speed timer, and 10 µs high-speed timer
		C	Counter
		OUTC	Out counter
		ITVL	Interval timer
		UDC	Up/down counter
		UDT	Up/down timer
	Connection/end instruction	MEP and MEF	Mep and mef
Application instructions	Shift instruction	SFT	Shift
	Stage processing instructions	W-ON and W-OFF	Wait on and wait off
		W-UE and W-DE	Wait rising edge and wait falling edge
	Flow instruction	SBN/RET/SCJ	Until the start of subroutine/End of subroutine/LABEL instruction
	File register instructions	FRSTM/FRLDM	File register saves in batches/file register reads in batches
Calculation instructions	Data control instructions	RAMP, TPOUT, and LLFLT	Ramp signal, time division proportion output, and advancement delay filter
	Character string instruction	RCOM	Read out of character string
Expansion instructions	Data processing instructions	HKEY/SORTN	Hexadecimal key input and binary data division sort
	Cam switch instructions	ABSENC	Absolute encoder and incremental encoder
	PID instructions	PID and PIDAT	PID and PID with automatic tuning
	Storage instruction	MWRIT/MREAD/ MFREE/MMKDIR/ MPRINT/MREADL/ MCOPY/MMOV/ MREN/MFREEK/ MSTAT	storage write/storage read/acquiring storage capacity/creating storage directory/writing in storage character string/storage 1 line read out/storage file copy/storage file migration/storage file name modification/acquiring free space of storage/acquiring storage file status
		MRMDIR/MDEL	storage folder deletion/ storage file deletion
	Sensor setting instruction	SPWR/SPRD/SSVC	Write sensor parameters/read sensor parametrs/execute sensor service
	Access window instructions	AWSHOW/AWHIDE	Display user message/cancel user message

● Function blocks

Instruction Type		Instruction	
Application instructions	Macro instructions	MCALL and MSTRT	Macro call and macro start

* The FEND (function block end) instruction is written in the function block. It cannot be used in other types of programs such as functions or modules.

● Functions

Instruction Type		Instruction	
Application instructions	Macro instructions	MCALL and MSTRT	Macro call and macro start
	Function block instructions	FBCALL/FBSTRT/FB	Function block

Macros

Instruction Type		Instruction	
Application instructions	Macro instructions	MCALL and MSTRT	Macro call and macro start
	Function block instructions	FBCALL/FBSTRT/FB/FUN	Function block / function

* Write the MEND (macro end) instruction in self-hold macros. It cannot be used in other types of programs such as sub-routine type macros or modules.

Subroutine programs

Instruction Type		Instruction	
Application instructions	Stage processing instructions	STG, JMP, and ENDS	Stage, jump, and end stage
Expansion instructions	High-speed processing instructions	INT and RETI	Interrupt and interrupt return

6 List of Abbreviated Mnemonics

This section describes the instructions and devices that can be abbreviated or that have representative input when entering mnemonics with KV STUDIO.

● Instructions

Instruction		Abbreviation
Contact instructions	LD and AND	A and X
	LDB and ANB	B, LDI, LDNOT, ANI, ANDNOT, and /*3
	ORB	ORI and ORNOT
	LDP and ANP	P
	LDF and ANF	F
Bit contact instructions	BLD and BAND	BA
	BLDB and BANB	BB
Comparison instructions	LD= and AND=	=
	LD< and AND<	<
	LD> and AND>	>
	LD<= and AND<=	<=
	LD>= and AND>=	>=
	LD<> and AND<>	<>
Output instructions	OUT	O and Y
	OUB	OB
	SET	S
	RES	R and RST
	DIFU	U and PLS
	DIFD	D and PLF
Bit output instructions	BOUT	BO
	BOUB	BOB
	BSET	BS
	BRES	BR
Timer and counter instructions*1	TMR	T, TIM, and OUTT
	TMH	TH, TIMH, and OUTH
	TMS	TS, TIMHH, and OUTS
	TMU	TU, TMUH, and OUTH
	C	CNT
	UDC	CNTR
Connection and end instructions	END	E and FEND
	ENDH	H
Stage instructions	W-ON	WON
	W-OFF	WOFF
	W-UE	WU and WUE
	W-DE	WD and WDE
Flow instructions	MC	IL
	MCR	ILC
Buffer memory instructions	UREAD	FROM
	UWRIT	TO
Arithmetic and comparison instructions	INC	++
	DEC	--
	ROOT	SQRT

Instruction		Abbreviation
Logic calculation instructions	EORA	XOR and XORA
	ENRA	XNORA
Data control instructions	APR	SCL
Floating-point instructions	DFLOAT	FLTD, DFLTD, DBL, and DBLL
	DINTG	INTD, DINTD, FIXD, and FIXLD
	FTODF	ECON
	DFTOF	EDCON
Text processing instructions	DASC	DSTR
	RDASC	DVAL
	HASC	HSTR
	RHASC	HVAL
	FASC	FSTR
	RFASC	FVAL
Simple calculation instructions	CAL+	+
	CAL-	- *2
	CAL*	*
	CAL/	/ *3
	CAL&	& and WAND
	CAL |	| and WOR
	CAL~	~ and WXOR
	CAL^	^, NOT, and CML
	CAL>>	>> and SFR
	CAL<<	<< and SFL
High-speed processing instructions	INT	I
	RETI	IRET

*1 You can enter timer and counter instructions without a space between the instruction and the device number.
(Example) TMR 0002 DM00001 → T2 D1

*2 If you enter "-" directly in the ladder editing window, a horizontal connecting line will be inserted.
If you enter "-" in the direct entry dialog box (Esc), "CAL-instruction" will be inserted.

*3 If you enter "/" directly in the ladder editing window, the "LDB" instruction will be inserted.
If you enter "/" in the direct entry dialog box (Esc), "CAL/instruction" will be inserted.

● Device names

Device Name	Abbreviated Input
R	None (numbers only)
MR	M
LR	L
DM	D
EM	E
FM	F

● Device numbers and constants

You can use zero suppression.

7 List of Shortcut Keys

This section gives a list of the shortcut keys that are used to execute the various functions.

Shortcut Keys That Are Valid in the Unit Editor

● File

Import	Ctrl + I
Close	Alt + F4

● Edit

Cut	Ctrl + X
Copy	Ctrl + C
Paste	Ctrl + V
Delete	Delete
Unit comments	Ctrl + Shift + U
Block comment	Ctrl + Shift + B
System comments	Ctrl + Shift + S
Unit preset	Ctrl + R
Batch input relay No.	Ctrl + A

● Convert

Jump to error rung	F4
Auto-assign relay/DM	F5
Device check	F6

● Setup slave

Setup slave	Ctrl + N
Import slave information	Ctrl + K
Copy KL unit information	Ctrl + Alt + C
Paste KL unit information	Ctrl + Alt + V
KL address setting	Ctrl + E

● View

Display unit setting	↵
Built-in Function Monitor	Ctrl + Alt + U
Custom Monitor	Ctrl + Alt + M
Buffer memory monitor	Ctrl + Alt + R

● Window

Unit configuration area	Alt + U
Select unit	Alt + 1
Setup unit	Alt + 2
Refresh setting	Alt + 3
Message area	Alt + 4

● Help

How to use Unit Editor	F1

● Operations in the unit configuration area

Move cursor	← • → • ↑ • ↓
Move cursor to start of rung	Home
Move cursor to end of rung	End
Delete configured units	Delete
Multi select	Shift + ← • → • ↑ • ↓

● Operations in unit selection window

Move cursor	← • → • ↑ • ↓
Add selected unit	↵

● Operations in unit setup window

Move cursor	↑ • ↓
Move displayed unit	Ctrl + ← • → • ↑ • ↓
Input	Tab
Input/confirm	↵
Cancel	Esc

● Operations in message area

Move cursor	↑ • ↓

Shortcut keys that are valid on the Unit Editor slave setup screen

● File

Import slave information	Ctrl + K
Close	Alt + F4

● Edit

Cut	Ctrl + X
Copy	Ctrl + C
Paste	Ctrl + V
Delete	Delete
Input unit/slave comment	Ctrl + Shift + U
Reserve unit slave	Ctrl + R

● Convert

Jump to error rung	F4
Automatic address assignment	F5
Check addresses	F6

● View

Unit or slave settings	↵

● Window

Slave configuration area	Alt + C
Unit selection window	Alt + 1
Unit setup window	Alt + 2
Address setup window	Alt + 3
Message area	Alt + 4

● Help

How to use Unit Editor	F1

● Operations in slave configuration area

Move cursor	↑ · ↓
Display the Setup unit window of the selected unit	↵
Delete configured units	Delete

● Operations in unit selection window

Move cursor	← · → · ↑ · ↓
Add selected unit	↵

● Operations in unit setup window

Move cursor	↑ · ↓
Enter	Tab
Enter and confirm	↵
Cancel	Esc

● Operations in address setup window

Move cursor	↑ · ↓

Shortcut Keys That Are Valid in KV STUDIO

Items that are indicated with a "◆" can be assigned to other shortcut key sequences.

Shortcut keys that are valid on the initial screen

● File

New project	Ctrl + N
Open project	Ctrl + O

● View

Workspace	Alt + 1

● Monitor/Simulator

PLC monitor	Ctrl + F1
Read from PLC	Ctrl + F5

Shortcut keys that are valid in editor mode

● File

New project	Ctrl + N
Open project	Ctrl + O
Save project	Ctrl + S
Print	Ctrl + P

● Edit

Undo		Ctrl + Z
Redo		Ctrl + Y
Cut		Ctrl + X
Copy		Ctrl + C
Paste		Ctrl + V
Multi paste		Ctrl + Shift + V
Search		Ctrl + F
Replace operand		Ctrl + H
New cross reference		(Space)
Cross reference, Create rung cmnts list		Ctrl + L
Cross reference, Previous◆		Shift + F3
Cross reference, Next◆		F3
Jump	Rung or rung step number	Ctrl + Shift + G
	Top	Ctrl + Home
	Bottom	Ctrl + End
	Previous block	Ctrl + ↑
	Next block	Ctrl + ↓
	Previous symbol	Shift + Tab
	Next symbol	Tab
	Home	Home
	End	End
Jump	Prev. changed block	Ctrl + Alt + ←
	Next changed block	Ctrl + Alt + →
Insert	Cell	Ins
	Rung	Shift + ↵ / Shift + Ins
	Rung comment	Ctrl + ↵
	Loopback ladder	Ctrl + Shift + F8
Delete rung		Shift + Delete
Disable rung, Setup		Ctrl + Q
Disable rung, Cancel		Ctrl + W
Bookmark	Insert	Ctrl + Shift + Ins
	Set	Ctrl + Shift + ↵
	Remove	Ctrl + Shift + Delete
Rung selection		Ctrl + → (Shift + ←)*1 (Shift + →)*2
Ladder arrangement		Ctrl + Shift + F
Edit list		Ctrl + D
Change operand		↵
Change device cmnt		Ctrl + M
Edit connection line	Enter vert. connection line◆	F8
	Enter horiz. connection line◆	F9 (−)
	Horizontal connection line to end of rung	Alt + ↵ (Ctrl + Tab)

Edit connection line	Delete vert. connection line◆	Shift + F8
	Delete horiz. connection line◆	Shift + F9
	Enter/delete vert. connection line	Alt + ↑ / Alt + ↓
	Enter/delete horiz. connection line	Alt + ← / Alt + →
Display cross references of the device at the cursor position		(Space)
Delete instruction to the left of cursor position		Backspace
Delete instruction at cursor position		Delete
Directly enter instruction at cursor position		Esc
Move symbol at the cursor position to the right		Ins
Enter NO contact◆		F5
Enter NC contact◆		Shift + F5
Enter NO contact OR◆		F4
Enter NC contact OR◆		Shift + F4
Enter NO contact coil◆		F7
Enter NC contact coil◆		Shift + F7
SET◆		Alt + F1
RES◆		Alt + F2
TMR◆		Alt + F3
TMS◆		Alt + F5
C◆		Alt + 0
DIFU◆		Alt + F8
DIFD◆		Alt + F9
MOV◆		Alt + F10
LDA◆		Alt + F11
STA◆		Alt + F12
LDP◆		Alt + F7
LDF◆		Alt + Shift + F7
ORP◆		Alt + Shift + F8
ORF◆		Alt + Shift + F9
MEP◆		Alt + Shift + F5
MEF		Alt + Ctrl + F5
INV		Alt + Ctrl + F10

Switch between NO contact and NC contact, change suffix, enter NC contact	/
Switch between rising edge and falling edge	Ctrl + /
Switch between direct input and current value◆	F2
Next pane◆	Ctrl + Shift + Alt + F11 *
Previous pane◆	Ctrl + Shift + Alt + F12 *
Increment device number	Ctrl + Shift + ↑
Decrement device number	Ctrl + Shift + ↓

* This is only valid when the display of the ladder editing window is split.

● View

Workspace	Alt + 1
Output window	Alt + 2
Instruction palette◆	Shift + Ctrl + P
Operand dialog box	↵ *
Comment editing window	Ctrl + F7
Device use list	Ctrl + E
Zoom in	Ctrl + PageUp
Zoom out	Ctrl + PageDown
Display variable	Ctrl + Backspace
Display cmnts◆	Ctrl + (Space)
Display unit device	Shift + (Space)
Switch between editing windows	Alt + F6

* This is only valid when an instruction is input in the selected cell.

● ST/Script

Insert area script	Ctrl + R
Insert box script	Ctrl + B
Insert area ST	Ctrl + Shift + A
Insert box ST	Ctrl + Shift + B
Convert selected ST/Script	Ctrl + T
Display/hide converted ladder	Ctrl + G
ST/Script function list	Ctrl + (Space)
ST/Script function tip	Ctrl + Shift + (Space)

● Convert

Convert◆	Ctrl + F9

*1 This is only valid when the cursor is in the far-left position (the first column).

*2 This is only valid when the cursor is in the far-right position (the tenth column).

● Monitor/Simulator

Monitor mode◆	Ctrl + Shift + F3
PLC transfer → monitor mode◆	Ctrl + F8
PLC read → monitor mode	Ctrl + F1
PLC read◆	Ctrl + F5
Start Online Edit◆	F10
Simulator	Ctrl + F2

● Tool

Unit Editor	Ctrl + U

● Help

Instruction help	F1
Instruction Reference Manual	Ctrl + 1

Shortcut keys that are valid in monitor mode

● Edit

Search		Ctrl + F or ↵
Jump	Rung or rung step number	Ctrl + Shift + G
	Top	Ctrl + Home
	Bottom	Ctrl + End
	Previous block	Ctrl + ↑
	Next block	Ctrl + ↓
	Previous symbol	Shift + Tab
	Next symbol	Tab
	Home	Home
	End	End
	Prev. changed block	Ctrl + Alt + ←
	Next changed block	Ctrl + Alt + →
Contact/Coil		F12
New cross reference		(Space)
Cross reference, Previous		Shift + F3
Cross reference, Next		F3
Cross reference, Create rung cmnts list		Ctrl + L
Rung selection		Ctrl + →
Switching contact between ON and OFF		/
Register set		Ctrl + Shift + J
Register reset		Ctrl + Shift + K
Cancel		Ctrl + Shift + L

● View

Workspace	Alt + 1
Output window	Alt + 2
Zoom in	Ctrl + PageUp
Zoom out	Ctrl + PageDown
Display variable	Ctrl + Backspace
Display cmnts◆	Ctrl + (Space)
Display unit device	Shift + (Space)
Change device value	F2
Switch between editing windows	Alt + F6

● Monitor/Simulator

Return to Editor◆		Ctrl + F1
Start Monitor		F4
Stop Monitor		Shift + F4
Online Editing	Start Online Edit	F10
	Setup Online Edit	F12
Simulator		Ctrl + F2
Display/hide watch window		Alt + 3

● Debugging

Execute 1 scan	F9
Execute 1 step	F8

● Help

Instruction help	F1
Instruction Reference Manual	Ctrl + 1

Shortcut keys that are valid in simulator mode

● Edit

Search		Ctrl + F (↵)
Rung selection		Ctrl + →
Jump	Top	Ctrl + Home
	Bottom	Ctrl + End
	Previous block	Ctrl + ↑
	Next block	Ctrl + ↓
	Previous symbol	Shift + Tab
	Next symbol	Tab
	Home	Home
	End	End
	Prev. changed block	Ctrl + Alt + ←
	Next changed block	Ctrl + Alt + →
	Contact/Coil	F12
New cross reference		(Space)
Cross reference, Previous		Shift + F3
Cross reference, Next		F3
Cross reference, Create rung cmnts list		Ctrl + L
Switching contact between ON and OFF		/

● View

Workspace	Alt + 1
Output window	Alt + 2
Zoom in on ladder display	Ctrl + PageUp
Zoom out from ladder display	Ctrl + PageDown
Display variable	Ctrl + Backspace
Display cmnts◆	Ctrl + (Space)
Display unit device	Shift + (Space)
Change device value	F2

● Monitor/Simulator

Return to Editor◆	Ctrl + F1
Monitor mode◆	Ctrl + Shift + F3
Start Online Edit	F10

● Help

Instruction help	F1
Instruction Reference Manual	Ctrl + 1

Shortcut keys that are valid during online editing

● Edit

Undo	Ctrl + Z
Redo	Ctrl + Y
Cut	Ctrl + X
Copy	Ctrl + C
Paste	Ctrl + V
Multi paste	Ctrl + Shift + V
Search	Ctrl + F
Replace operand	Ctrl + H
Online edit Search for edit place	Ctrl + . , Ctrl + O

Jump	Rung or rung step number	Ctrl + Shift + G
	Top	Ctrl + Home
	Bottom	Ctrl + End
	Previous block	Ctrl + ↑
	Next block	Ctrl + ↓
	Previous symbol	Shift + Tab
	Next symbol	Tab
	Home	Home
	End	End
	Prev. changed block	Ctrl + Alt + ←
	Next changed block	Ctrl + Alt + →
	Contact/Coil	F12
New cross reference		(Space)
Cross reference, Previous◆		Shift + F3
Cross reference, Next◆		F3
Cross reference, Create rung cmnts list		Ctrl + L
Insert cell		Ins

Insert empty rung		Shift + ↵ or Shift + Ins
Delete rung		Shift + Delete
Insert rung comment		Ctrl + ↵
Disable rung, Setup		Ctrl + Q
Disable rung, Cancel		Ctrl + W
Bookmark	Insert	Ctrl + Shift + Ins
	Set	Ctrl + Shift + ↵
	Remove	Ctrl + Shift + Delete
Rung selection		Ctrl + →
Edit connection line	Enter vert. connection line◆	F8
	Enter horiz. connection line◆	F9 (−)
	Horizontal connection line to end of rung	Alt + ↵ (Ctrl + Tab)
	Delete vert. connection line◆	Shift + F8
Edit connection line	Delete horiz. connection line◆	Shift + F9 (−)
	Enter/delete vert. connection line	Alt + ↑ / Alt + ↓
	Enter/delete horiz. connection line	Alt + ← / Alt + →
Insert loopback ladder		Ctrl + Shift + F8
Ladder arrangement		Ctrl + Shift + F
Edit list		Alt + D
Change operand		↵
Change device cmnt		Alt + M
Delete instruction to the left of cursor position		Backspace
Delete instruction at cursor position		Delete
Directly enter instruction at cursor position		Esc
Enter NO contact◆		F5
Enter NC contact◆		Shift + F5
Enter NO contact OR◆		F4
Enter NC contact OR◆		Shift + F4
Enter NO contact coil◆		F7
Enter NC contact coil◆		Shift + F7
SET◆		Alt + F1
RES◆		Alt + F2
TMR◆		Alt + F3
TMS◆		Alt + F5
C◆		Alt + 0
LDP◆		Alt + F7
DIFU◆		Alt + F8
DIFD◆		Alt + F9
LDF◆		Alt + Shift + F7
ORP◆		Alt + Shift + F8
ORF◆		Alt + Shift + F9
MOV◆		Alt + F10
LDA◆		Alt + F11
STA◆		Alt + F12
MEP◆		Alt + Shift + F5
MEF		Alt + Ctrl + F5
INV		Alt + Ctrl + F10
Switch between NO contact and NC contact, change suffix, enter NC contact		/
Switch between rising edge and falling edge		Ctrl + /
Switch between direct input and current value		F2
Register set		Ctrl + Shift + J
Register reset		Ctrl + Shift + K
Cancel		Ctrl + Shift + L

● View

Workspace	Alt + 1
Output window	Alt + 2
Instruction palette◆	Shift + Ctrl + P
Operand dialog box	↵ *
Comment editing window	Ctrl + F7
Device use list	Ctrl + E
Zoom in	Ctrl + PageUp
Zoom out	Ctrl + PageDown
Display variable	Ctrl + Backspace
Display cmnts◆	Ctrl + (Space)
Display unit device	Shift + (Space)

* This is only valid when an instruction is input in the selected cell.

● ST/Script

Insert area script	Ctrl + R
Insert box script	Ctrl + B
Insert area ST	Ctrl + Shift + A
Insert box ST	Ctrl + Shift + B
Convert selected ST/Script	Ctrl + T

● Monitor/Simulator

Return to Editor◆	Ctrl + F1
Monitor mode◆	Ctrl + Shift + F3
Online edit transfer	F11
Simulator	Ctrl + F2
Display/hide watch window	Alt + 3

● Help

Instruction help	F1
Instruction Reference Manual	Ctrl + 1

Shortcut keys valid in the replay mode

● Edit

Search		Ctrl + F / ↵
Jump	Rung or rung step number	Ctrl + Shift + G
	Top	Ctrl + Home
	Bottom	Ctrl + End
	Previous block	Ctrl + ↑
	Next block	Ctrl + ↓
	Previous symbol	Shift + Tab
	Next symbol	Tab
	Home	Home
	End	End
	Prev. changed block	Ctrl + Alt + ←
	Next changed block	Ctrl + Alt + →
	Contact/Coil	F12
New cross reference		(Space)
Cross reference, Previous		Shift + F3
Cross reference, Next		F3
Cross reference, Create rung cmnts list		Ctrl + L
Rung selection		Ctrl + →

● View

Workspace	Alt + 1
Output window	Alt + 2
Zoom in	Ctrl + PageUp
Zoom out	Ctrl + PageDown
Zoom in and zoom out	Alt + wheel button Ctrl + wheel button
Display variable	Ctrl + Backspace
Display cmnts	Ctrl + (Space)
Display unit device	Shift + (Space)
Switch between editing windows	Alt + F6

● Monitor/Simulator

Return to Editor	Ctrl + F1
Start Monitor	F4
Stop Monitor	Shift + F4
Start Online Edit	F10
Simulator	Ctrl + F2
Display/hide watch window	Alt + 3

● Help

Instruction help	F1
Instruction Reference Manual	Ctrl + 1

List of GPPQ and GPPA Format Shortcut Keys

You can change the key assignments of each shortcut.

Enter NO contact	F5	
Enter NC contact	GPPQ format	F6
	GPPA format	Shift + F5
Enter NO contact OR	GPPQ format	Shift + F5
	GPPA format	F6
Enter NC contact OR	Shift + F6	
Enter coil	F7	
Enter NC contact coil	Not assigned	
Enter vert. connection line	GPPQ format	Shift + F9
	GPPA format	F10
Delete vert. connection line	Ctrl + F10	
Enter horiz. connection line	F9	
Delete horiz. connection line	Ctrl + F9	
SET	Not assigned	
RES	Not assigned	
TMR	Not assigned	
TMS	Not assigned	
C	Not assigned	
LDP	Shift + F7	
LDF	Shift + F8	
ORP	Alt + F7	
ORF	Alt + F8	
DIFU	Not assigned	
DIFD	Not assigned	
MOV	Not assigned	
LDA	Not assigned	
STA	Not assigned	
MEP	Alt + F5	
Draw and erase connecting line (right)	Alt + →	
Draw and erase connecting line (left)	Alt + ←	
Draw and erase connecting line (down)	Alt + ↓	
Draw and erase connecting line (up)	Alt + ↑	
Convert	Ctrl + Alt + F4 / F4	
Switch between direct input and current value	Not assigned	

Cross reference, Previous	Not assigned
Cross reference, Next	Not assigned
PLC transfer → monitor mode	F3
PLC read	Not assigned
Start Online Edit	Shift + F3
Return to Editor	F2
Next pane	Ctrl + Shift + Alt + F11
Previous pane	Ctrl + Shift + Alt + F12
Monitor mode	F3
Display cmnts	Ctrl + F5
Instruction palette	Ctrl + Shift + P

List of CX Format Shortcut Keys

You can change the key assignments of each shortcut.

Enter NO contact	C
Enter NC contact	Not assigned
Enter NO contact OR	W
Enter NC contact OR	X
Enter coil	O
Enter NC contact coil	Q
SET	Shift + F8
RES	Shift + F9
TMR	Ctrl + Alt + O
TMS	Not assigned
C	Not assigned
LDP	Not assigned
LDF	Not assigned
ORP	Not assigned
ORF	Not assigned
DIFU	Not assigned
DIFD	Not assigned
MOV	Not assigned
LDA	Not assigned
STA	Not assigned
MEP	Shift + Alt + F5
Draw and erase connecting line (right)	H
Draw and erase connecting line (left)	Not assigned
Draw and erase connecting line (down)	V
Draw and erase connecting line (up)	U
Convert	F7
Switch between direct input and current value	I
Cross reference, Previous	B
Cross reference, Next	N
PLC transfer → monitor mode	F3
PLC read	Ctrl + F5
Start Online Edit	F10

Return to Editor	F2
Next pane	Ctrl + Shift + Alt + F11
Previous pane	Ctrl + Shift + Alt + F12
Monitor mode	Ctrl + Shift + F3
Display cmnts	Alt + Y
Instruction pallet	Ctrl + Shift + P

List of Logix Format Shortcut Keys

You can change the key assignments of each shortcut.

Enter NO contact	F5
Enter NC contact	Shift + F5
Enter NO contact OR	F4
Enter NC contact OR	Shift + F4
Enter coil	F7
Enter NC contact coil	Shift + F7
SET	Alt + F1
RES	Alt + F2
TMR	Alt + F3
TMS	Alt + F5
C	Alt + 4
LDP	Alt + F7
LDF	Alt + Shift + F7
ORP	Alt + Shift + F8
ORF	Alt + Shift + F9
DIFU	Alt + F8
DIFD	Alt + F9
MOV	Alt + F10
LDA	Alt + F11
STA	Alt + F12
MEP	Alt + Shift + F5
MEF	Ctrl + Alt + F5
Convert selected script	Ctrl + T
Edit list	Ctrl + Shift + D
Draw and erase connection line (right)	Alt + →
Draw and erase connection line (left)	Alt + ←
Draw and erase connection line (down)	Alt + ↓
Draw and erase connection line (up)	Alt + ↑
Horizontal connection rung to end of line (-)	Ctrl + Alt + ↵

Convert	Ctrl + F9
Switch between direct input and current value	F2
Cross reference	Ctrl + E
Cross reference, Previous	Shift + F3
Cross reference, Next	F3
PLC transfer → monitor mode	Ctrl + F8
PLC read	Ctrl + F5
Start Online Edit	F10
Return to Editor	Ctrl + F1
Instruction/Macro/ Pack Pallet	Alt + Ins
Insert rung comment	Ctrl + D
Switch between editing windows	Ctrl + Tab
Switch between editing windows in reverse order	Ctrl + Shift + Tab
Variable Edit Window	Ctrl + W
Jump to Rung/ step No.	Ctrl + G
Insert Empty rung	Ctrl + R
Disabling Rungs Setup	Ctrl + Q
Disabling Rungs Cancel	Ctrl + Shift + W
Switch to display/hide all workspace windows	Alt + 0
Switch to display/hide all output windows	Alt + 1
Switch to display/ hide Search1 result windows	Alt + 2
Device use list	Ctrl + Shift + E
Insert area script	Ctrl + Alt + A
Insert box script	Ctrl + Alt + B
Insert area ST	Ctrl + Shift + A
Insert box ST	Ctrl + Shift + B

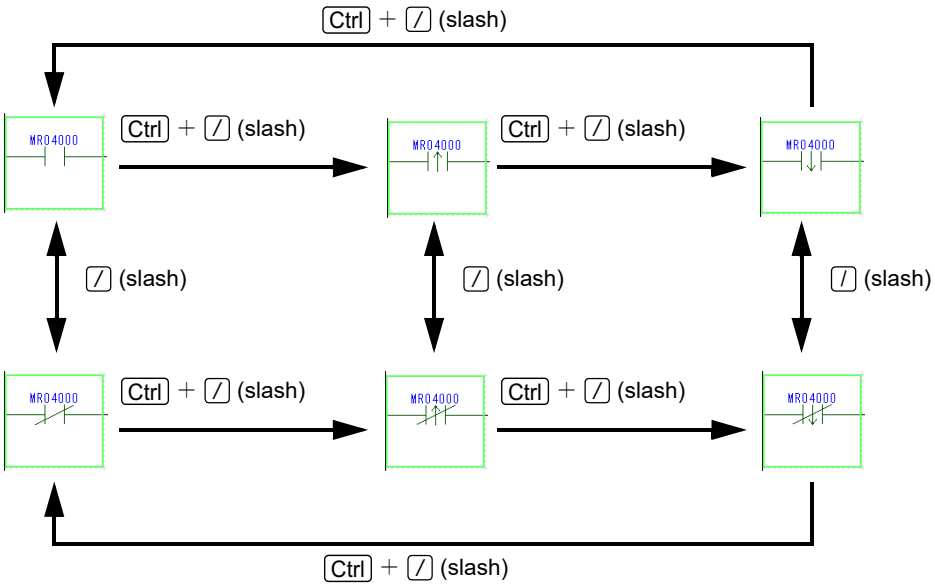
Convert selected ST/Script	Ctrl + Shift + T
Display/hide converted ladder	Ctrl + Shift + G
Execute continuous steps	Shift + F8
Execute 1 step	F8
Execute reverse continuous steps	Shift + F7
Execute reverse 1 step	F7
Pause	Shift + F6

How to Use the / Key

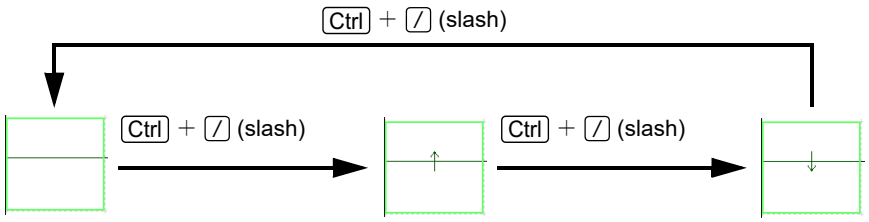
● **When you press /, the setting at the cursor position changes.**

The values that you can select are shown below.

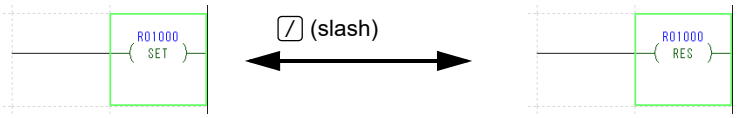
• (1) NO contact or NC contact and (2) rising edge or falling edge



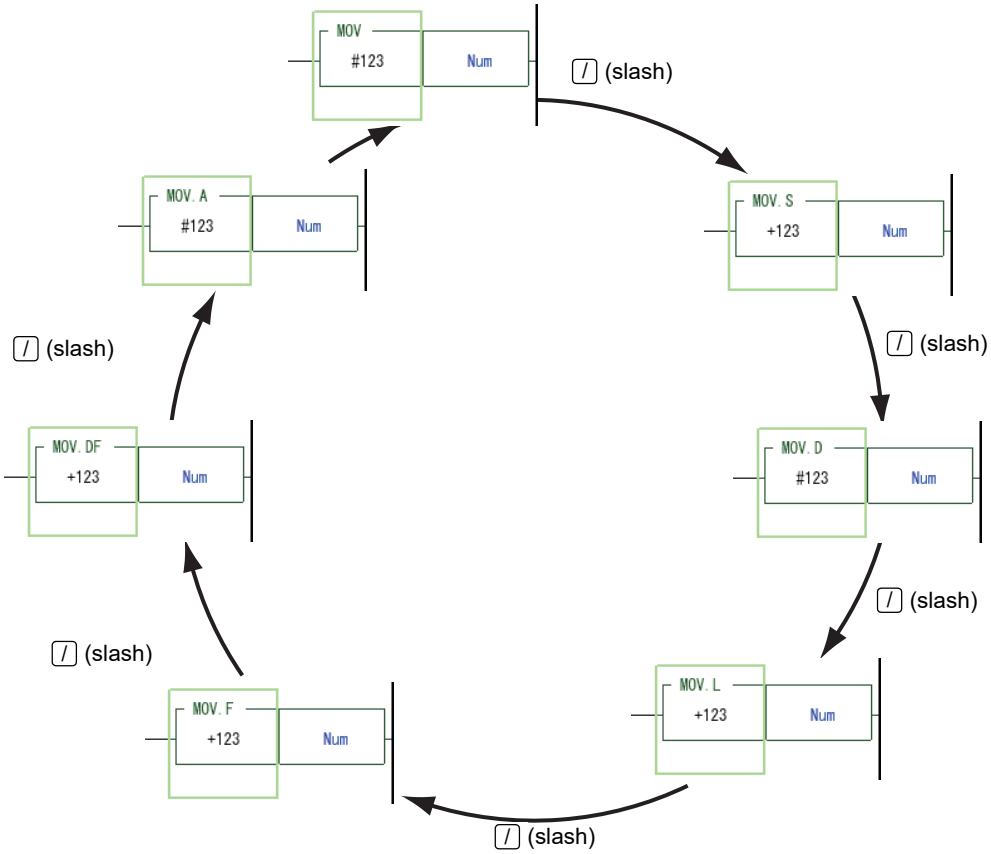
Point **Press Ctrl + / to change the values as shown below.**



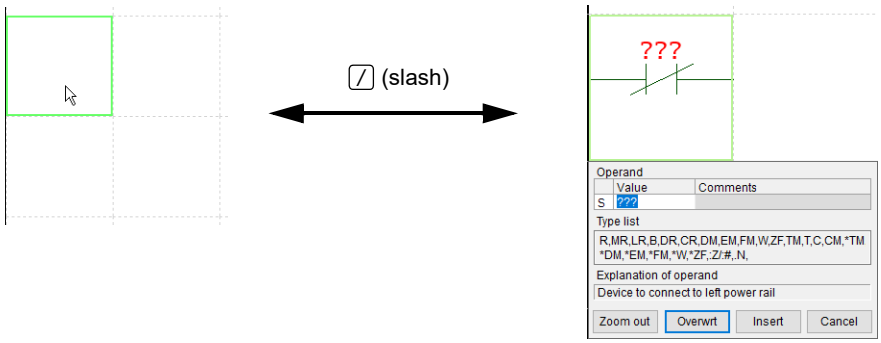
• SET and RES



• Suffix



● **If you press this key when a cell with no instruction is selected, an NC contact will be entered.**



*This can only be entered in the direct entry dialog box.

8 List of Characters That Cannot Be Used

This section describes the characters that cannot be used in project names, module names, function blocks, macro names, variable, Structure and device comments.

● Characters that cannot be used in project names

Characters that cannot be used	! ″ # $ % & ‘ * , . / : ; < = > ?
	@ ¥ ^ ` | ~ (space) _

● Characters that cannot be used in program names

Characters that cannot be used	! ″ # $ % & ’ ( ) * , . / : ; < = > ?
	@ [¥] ^ ` { | } ~ (space)*1 + −
Characters that cannot be used as the first character	0 1 2 3 4 5 6 7 8 9 _
Character and character strings that cannot be used as the last character or characters	_ _0 _1 _2 _3 _4 _5 _6 _7 _8 _9
Character string that cannot be used	Global グローバル Global 全局

*1 Can be used on KV-8000 series with CPU function version 2.1 or later.

● Characters that cannot be used in function / function block / structure names

Characters that cannot be used	! ″ # $ % & ‘ * , . / : ; < = > ? @ ¥
	^ ‘ | ~ (space) (__) + −
Characters that cannot be used at the start	0 1 2 3 4 5 6 7 8 9 _ *1
Characters or strings that cannot be used at the end*2	_ _0 _1 _2 _3 _4 _5 _6 _7 _8 _9
Strings that cannot be used	Global グローバル Global 全局 全域 글로벌 • TIMER 定时器 定時器 타이머 COUNTER 计数器 計數器 카운터 FUN_ENO • Keywords specified in IEC61131-3 (For the contents of keyword, refer to "Characters that cannot be used in variable names".)

*1 Can be used on KV-8000 series with CPU function version 2.1 or later.
*2 The _ (number) at the end can be used on KV-8000 series with CPU function version 2.1 or later.

● Characters that cannot be used in variable

Characters that cannot be used	! ″ # $ % & ’ ( ) * + , − . / : ; < =
	> ? @ [¥] ^ ` { | } ~ (space)
Characters that cannot be used as the first character	0 1 2 3 4 5 6 7 8 9 _
Character strings that cannot be used	Global グローバル Global 全局
	The same names as instructions
	Device names
	• (1) One alphabet character + R or (2) M + a number • One alphabet character + a number • X + (A to F) • Y + (A to F)

	(Same name as ST language function)
	(Same name as KV script function)
	• The following keywords, which are prescribed in KV script !!,#PRAGMA,',(,),*,*=,+,+=,-,-=,.B,.D,.DF,.F,.L,.S,.T,.U,/,/=,<,<=,<>,=,>,>=,AND,AS,B,BREAK,C,CASE,CM,CR,CTC,CTH,DIM,DM,DO,DOUBLE,DR,ELSE,ELSE IF,EM,END,END IF, END SELECT,END TYPE,END WHILE, FALSE,FM,FOR,IF,IS,LR,MC,MCR,MOD,MR,NEXT,NOT,OFF,ON,OR,P,R,REM,SELECT,SELECT CASE,STEP,STRWORK,T, THEN,TM,TO,TRUE,TYPE,UM,UNTIL,UR,UV,V,W,WHILE,XOR, Z,ZF,[,\\,\\A,\\B,\\F,\\N,\\R,\\T,\\V,]
Character strings that cannot be used	• The following keywords, which are prescribed in IEC61131-3 TRUE, FALSE, TIME, DATE,TIME_OF_DAY, TOD, DATE_AND_TIME, DT, BOOL, SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL, STRING, WSTRING, BYTE, WORD, DWORD, LWORD, ANY, ANY_DERIVED, ANY_ELEMENTARY, ANY_MAGNITUDE, ANY_NUM, ANY_REAL, ANY_INT, ANY_BIT, ANY_STRING, ANY_DATE, ACTION, END_ACTION, ARRAY, OF, AT, CASE, END_CASE, CONFIGURATION, END_CONFIGURATION, CONSTANT, EN, ENO, EXIT, • F_EDGE, FOR, TO, BY, DO, END_FOR, FUNCTION, END_FUNCTION, FUNCTION_BLOCK, END_FUNCTION_BLOCK, IF, THEN, ELSIF, ELSE, END_IF, INITIAL_STEP, END_STEP, NOT, MOD, AND, XOR, OR, PROGRAM, WITH, END_PROGRAM, R_EDGE, READ_ONLY, READ_WRITE, REPEAT, UNTIL, END_REPEAT, RESOURCE, END_RESOURCE, RETAIN, NON_RETAIN, RETURN, STEP, STRUCT, END_STRUCT, TASK, TRANSITION, FROM, END_TRANSITION, TYPE, END_TYPE, VAR,VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR_TEMP, VAR_EXTERNAL, VAR_ACCESS, VAR_CONFIG,VAR_GLOBAL, END_VAR, WHILE, DO, END_WHILE • Other character strings (see the IEC61131-3 specifications) • A variable name cannot have two or more consecutive underscore characters. The string "a_b_c" is acceptable, but "a__bc" cannot be used.

● Characters that cannot be used in structure member names

Characters that cannot be used	( ) * / + - < > = & ! ¥ ″ # $ % % & | ' ~ ` @ [ ] { } : ; , . ? ^ (Half-width space) (Full-width space) (__)*1
Characters that cannot be used at the start	0 1 2 3 4 5 6 7 8 9 _ *1
Characters or strings that cannot be used at the end*2	_ _0 _1 _2 _3 _4 _5 _6 _7 _8 _9

Strings that cannot be used	Global グローバル Global 全局 全域 글로벌
	• Strings that cannot be distinguished from numeric values
	  • K
	  • K + value
	  • K + value + suffix(U/S/D/L/F/DF)
	  • K + value + E
	  • K + value + E + value
	  • K + value + E + value + suffix(F/DF)
	  • H + hexadecimal value (decimal value or ABCDEF string)
	• The following key words, which are prescribed in KV script
	(For the contents of keyword, refer to "Characters that cannot be used in variable names".)
	• Keywords specified in IEC61131-3
	(For the contents of keyword, refer to "Characters that cannot be used in variable names".)

*1 Can be used on KV-8000 series with CPU function version 2.1 or later.
*2 The _ (number) at the end can be used on KV-8000 series with CPU function version 2.1 or later.

● Character that cannot be used in device comments

Character that cannot be used	,

List of Compatible Equipment

The table below lists existing KV Series expansion units compatible with the KV-8000 Series.

Category	Device	Compatibility Yes: Compatible -: Not compatible
AC power unit	KV-U7	– (Can be used to supply power to the KV-EB1)
Input unit	KV-B16XA/KV-B16XC/ KV-C32XA/KV-C32XC/ KV-C64XA/KV-C64XB/ KV-C64XC	Yes
Output unit	KV-B8RC/KV-B16RA/KV-B16RC/ KV-B16TA/KV-B16TC/KV-B16TD/ KV-B16TAP/KV-B16TCP/ KV-C32TA/KV-C32TC/KV-C32TD/ KV-C32TAP/KV-C32TCP/ KV-C64TA/KV-C64TC/KV-C64TD/ KV-C64TAP/KV-C64TCP	Yes
I/O unit	KV-B8XTD/KV-C16XTD/KV-C32XTD	Yes
Analog unit	KV-AD40/KV-AD40V/KV-AD40G/ KV-DA40/KV-DA40/ KV-AM40V	Yes
Temperature unit	KV-TF40/KV-TP40	Yes
Positioning unit	KV-H20/KV-H20S/KV-H40S/KV-H20G	Yes
Positioning unit, motion control unit	KV-ML16V/KV-MC20V/KV-MC40V/KV-MX1	Yes
Communication type positioning unit	KV-LH20V	Yes
High-speed counter unit	KV-SC20/KV-SC20V	Yes
Communication unit	KV-XLE02	Yes
	KV-XCM02	Yes
	KV-XL202/XL402	Yes
	KV-L20/KV-L20R	–
	KV-L20V/KV-L21V	Yes
	KV-EP21V	Yes
	KV-LE20/KV-LE20A	–
	KV-LE20V/KV-LE21V	Yes
	KV-FL20	–
	KV-FL20V	Yes
	KV-LM20	–
	KV-LM20V/KV-LM21V	Yes
	KL-N20V/MV-L40/KV-CL20/KV-DN20	Yes
System distribution unit	KV-EB1	Yes
Error output unit	KV-DR1	–
Bluetooth unit	KV-BT1	–

Usage Restrictions

The following restrictions apply when existing KV series expansion units are used with the KV-8000 Series.

■ KV-XLE02

- Only units of the function version 1.2 or later can be connected.
- For KV-8000A, the Mail send/receive function cannot be used.

■ KV-XCM02

- For KV-8000A, the mail sending and receiving functions are not available.

■ KV-XL202/XL402

- Only units of the function version 1.1 or later can be connected.

■ KV-XH16ML/XH04ML/XH64EC/XH32EC/XH16EC

- For KV-8000A, the Sync contact output cannot be used.

■ KV-L21V/KV-L20V

- When using the host link mode, the range of R that can be read/written via KV-L21V/KV-L20V is R00000 to R99915, and the range of MR is MR00000 to MR99915. (In XYM mode, M00000 to M64000 can be used.)
- When host link mode is used, the expansion unit buffer memory that can be read and written to via KV-L21V and KV-L20V is 0 to 32767.
- When protocol mode is used, the expansion unit buffer memory that can be read and written to via KV-L21V and KV-L20V is 0 to 32767.
- When using the link mode, the range of R that can be read/written via KV-L21V/KV-L20V is R00000 to R99915, and the range of MR is MR00000 to MR99915.

■ KV-LE21V/KV-LE20V

- VM65536 or later cannot be accessed via KV-LE21V/KV-LE20V.
- When using the host link mode, the range of R that can be read/written via KV-LE21V/KV-LE20V is R00000 to R99915, and the range of MR is MR00000 to MR99915. (In XYM mode, M00000 to M64000 can be used), and the range of VM is VM00000 to VM63999.
- When MC protocol mode is used, the expansion unit buffer memory range that can be read and written to via KV-LE21V and KV-LE20V is 0 to 32767.
- When using the FTP server function, device files of R100000 or higher will not be displayed.
- When FTP server functions are used, MR indicated with 6 digits or more (for example, MR100000 or higher devices) the last digit cannot be indicated.
- The read/write to CPU memory is not possible when FTP server function is used.
- The read/write to CPU memory is not possible when FTP client function is used.
- For KV-8000A, the Mail send/receive function cannot be used.

■ KV-EP21V

- VM65536 or later cannot be accessed via KV-EP21V.
- When using the host link mode, the range of R that can be read/written via KV-LE21V/KV-LE20V is R00000 to R99915, and the range of MR is MR00000 to MR99915.（M00000 to M64000 can be used when XYM mode is used）, and the range of VM is VM00000 to VM63999.
- When the MC protocol mode is used, the expansion unit buffer memory which can read/write via KV-LE21V/KV-LE20V is from 0 to 32767.
- When using the FTP server function, device files of R100000 or higher will not be displayed.
- When FTP server function is used, the last digit of MR（for example, the device MR100000 or higher）with 6 or more digits can not be displayed.
- When FTP server function is used, the read/write to CPU memory is not possible.
- When the FTP client function is used, read/write to CPU memory is not possible.
- When a message communication function is used, the expansion unit buffer memory that can be read and written to via KV-EP21 is 0 to 32767.
- For KV-8000A, the Mail send/receive function cannot be used.

■ KV-LM21V/KV-LM20V

- When megalink mode is used, the expansion unit buffer memory that can be read and written to via KV-LM21V and KV-LM20V is 0 to 32767, and VM is 0 to 63999.
- When serial PLC link mode is used, the link bit devices and link registers that can be read and written to via KV-LM21V and KV-LM20V is B0000 to B1FFF and W0000 to W3FFF, respectively.

■ KV-FL20V

- The link bit device B0000 to B3FFF, link register W0000 to W3FFF, R1000 to R99915 can be read/written with KV-FL20V.

■ KV-CL20

- When the transient transfer function is used, write operations to T, C and CR are not available.

■ KL-N20V

- When starting up in the communication status confirmation mode, keep pressing the direct access switch until the direct access switch turns green. If the direct access switch is released early, it can not be started normally and CPU error E 63 [Need to turn on power again] may occur.

The correspondence status of model modification from the existing KV series to KV-8000 series is as follows.

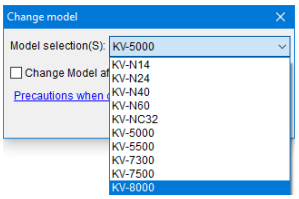
Original model	Change to the KV-8000 Series
KV-7500/7300	Yes
KV-5500/5000/3000	Yes
KV-1000	Yes
KV-700	Yes
KV Nano	No (first change to the KV-3000 model before changing to the KV-8000.)
KV-10/16/24/40	Yes

The list below describes precautions when changing from a KV-7000 Series and KV-5000/3000 Series model to a KV-8000 Series model.

Item	Description
Hardware functions	Cannot be replaced. Functions that cannot be used by the model are deleted or initialized when you change models. After changing models, check the specifications and use the model within the specified range. "Differences Related to Functions Such as CPU Special Hardware Functions" (page A-46)
Device	Devices other than control relays and control memory entries (CR/CM) cannot be automatically replaced. Only control relays and control memory entries that can be replaced are automatically replaced. "Differences between Control Relays and Control Memory Entries" (page A-49)
Expansion units	KV-7000 series extension unit It can be used directly. KV-5000/3000 series extension unit It can be used, but there are precautions. Check the following contents for precautions. "Precautions for Using the Existing KV Series Extension Unit" (page A-42)
Instructions	Depending on the instructions: • Variable may be converted to unregistered labels. • Instructions and operands may be changed to the appropriate values automatically. "Instructions That Have Differences" (page A-49)

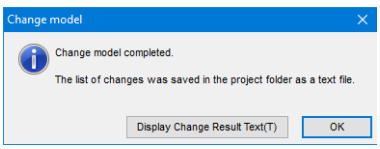
Model Replacement Procedure

1 In KV STUDIO, on the "File" menu, click "Change model" to execute the model change.



Reference

The results of changing the model are saved as the "conversion result text." View the "conversion result text" to check whether problems such as labels being changed to unregistered variable have occurred.



Differences Related to Functions Such as CPU Special Hardware Functions

Item				KV-8000 Series	KV-7000 Series		KV-5000/3000 Series	
				KV-8000(A)	KV-7500	KV-7300	KV-5500 KV-5000	KV-3000
Base specifications	Power supply	DC		24 V DC (-15% +20%)*5	24 V DC (-15% +20%)*5		24 V DC (±10%)	
		AC (power unit)		100 to 240 V AC (+10%,-15%), (50/60Hz)	100 to 240 V AC (+10%,-15%), (50/60Hz)		100 to 240 V AC ±10% (50/60Hz)	
	Battery			Sold separately (KV-B1)	Sold separately (KV-B1)		Supplied	
	Communication port	USB	1.1 (12 Mbps)	Yes	Yes		Yes	
			2.0 (480 Mbps)	Yes	Yes		-	
		RS-232C		-	-	Yes	-	Yes
		Ethernet*6	Ethernet	Yes*7	Yes	-	Yes	-
			FL-net	-	-	-	Yes (KV-5000)	-
			EtherNet/IP	Yes	Yes	-	Yes (KV-5500)	-
			Socket communication	Yes (Up to 16)	Yes (Ver. 1.1 or later) (Up to 16)	-	Yes (KV-5000) (Up to 8)	-
	Memory	CPU memory capacity*1		64 MB total	64 MB total	21 MB total	-	
		Breakdown	Program capacity*1	1500 k steps	1500 k steps	160 k steps	260 k steps	160 k steps
			Device comment capacity*3	224,000	224,000	102,000	96,000	30,000
			User memory capacity*1	40 MB	40 MB	10 MB	-	
		Device comment	Number of comments	8	8		1*2	
			Universal comments	Yes	Yes		-	
		Without memory card	Logging	Yes	Yes		-	
			File register	Yes	Yes		-	
			Load/Save	Yes	Yes		-	
	Local work			589824	51000		51000	
	Maximum number of I/O points			3072	3072		3096	
	Number of connectable units	KV-8000 Series Expansion Unit		16 units maximum (including KV-7000 Series expansion units)	-		-	
		KV-7000 Series Expansion Unit (S-Unit, X-Unit)		16 units maximum (including KV-8000 Series expansion units)	16 units maximum		-	
		KV-5000/3000 Series Expansion Unit		48 units maximum (including KV-8000/7000 Series expansion units)	48 units maximum (including KV-7000 Series expansion units)		48 units maximum	
	Instruction execution speed	Base instruction execution speed		0.96 ns	0.96 ns		10.0 ns	
	Power Failure Holding	Device		Backup using nonvolatile RAM	Backup using nonvolatile RAM		SRAM backup using the battery	
		Variable / Label		Yes	-		-	
		Calender timer (RTC)		Backup with condenser for backup（approx. 15 days） When using a battery (approx. 5 years (@ 25°C))	Backup with condenser for backup（approx. 15 days） When using a battery (approx. 5 years (@ 25°C))		Battery backup	
		Program		Flash ROM (can be rewritten 10,000 times)	Flash ROM (can be rewritten 10,000 times)		Flash ROM (can be rewritten 100,000 times)	
I/O	Inputs			-	-		16 (line driver supported)	
	Outputs			-	-		8	
	Connector			-	-		MIL connector	
	Extra power supply output			-	-		+5 V	
Access window	Access window	Fullwidth/halfwidth		Fullwidth/halfwidth	Fullwidth/halfwidth		Halfwidth	
		Size		Vertical: 4 characters, horizontal: 16 halfwidth characters	Vertical: 4 characters, horizontal: 16 halfwidth characters		Vertical: 4 characters, horizontal: 12 halfwidth characters	
		Language		Japanese, English, Chinese	Japanese, English, Chinese		Katakana or alphabetic characters	
		User message		Yes	Yes		-	

Item			KV-8000 Series	KV-7000 Series		KV-5000/3000 Series	
			KV-8000(A)	KV-7500	KV-7300	KV-5500 KV-5000	KV-3000
Memory card	Slot		Yes (SDHC supported)	Yes (SDHC supported)		Yes (SDHC supported)	
	Access from ladder programs		Possible (storage instruction)	Possible (storage instruction)		Possible（storage instruction）	
	Logging and tracing functions		Yes	Yes		Yes	
	Name of the folder when using the FTP server function		0_CARD	0_CARD		MMC	
CPU memory	Access from ladder programs		Yes	Yes		-	
	Logging and tracing functions		Yes	Yes		-	
	User documents		Yes	Yes		-	
Built-in functions	High-speed counter		-	-		Yes	
	Built-in positioning		-	-		Yes	
	operation recorder function		Yes	-		-	
	Frequency counter		-	-		Yes	
	Specified frequency pulse output		-	-		Yes	
	Cam switch function		-	-		Yes	
	Recording / tracking function	Buffer capacity	10240KB	2048KB		10240KB	
		Maximum number of devices	512	128		128	
	Serial communication	Operation mode	-	KV mode (host link) KV STUDIO mode		KV mode (host link) KV mode (text transmission) KV STUDIO mode	
	Interrupts	Interrupt conditions (input relays)	-	-		10 points (R000 to R009)	
		Interrupt conditions (high-speed counter comparators)	-	-		4 points (CTC0 to CTC3)	
		Interrupt condition (expansion unit interrupt)	Yes	Yes		-	
		Maximum number of interrupt conditions	64	64		14	
		Direct inputs	-	-		16 points (R000 to R015)	
		Direct outputs	-	-		8 points (R500 to R507)	
Recipe load / save			Memory card CPU memory	-		-	
Saving and reading project data			Memory card CPU memory	Memory card CPU memory		Memory card	
Saving and reading operation record			-	-		-	
Program	Supported languages		Ladder/KV script/ mnemonic	Ladder/KV script/mnemonic		Ladder/KV script/mnemonic	
	Fixed period execution	Period	50 μs	50 μs		200 μs	
		Number	4	4		1	
	Password	Module passwords	Up to 32 characters	Up to 32 characters		-	
		Function block passwords	Up to 32 characters	Up to 32 characters		-	
		Macro passwords	Up to 32 characters	Up to 32 characters		Up to 6 characters	
		Project passwords	Up to 32 characters	Up to 32 characters		Up to 6 characters	
	Number of programs		4,000	4,000		950	
Other functions	Break		No	No		During monitoring and simulating	
	Built-in Free Run counter		32-bit 0.1 μs 32-bit 1 μs 32-bit 10 μs 32-bit 1 ms	32-bit 0.1 μs 32-bit 1 μs 32-bit 10 μs 32-bit 1 ms		32-bit 1 μs	
	Input refreshing is prohibited		-	-		Yes	
	Bus communication	V-BUS RT	Yes	Yes		Yes	
		CPU inner bus	Yes	Yes		-	
	System error detection		Yes (when using KV-PU1)	Yes (when using KV-PU1)		Yes (when using KV-DR1)	

*1 The volume of the project area (programs, device comments, etc), and CPU memory area can be set by the user. The breakdown is for the default settings.
*2 Up to two can be set on KV STUDIO, but only one can be saved on the PLC.
*3 Estimate when the whole project area is used for device comments.
*4 Unit interrupt for the KV-7000 Series, and internal input interrupt for the KV-5000/3000 Series.
*5 24 ±10% V DC when using KV-5000/3000 Series expansion units.
*6 Refer to "EtherNet/IP Function User's Manual" for details.
*7 The Mail communication function cannot be used on KV-8000A.

Device Range Comparison Table

Item		KV-8000(A)	KV-7500/7300		KV-5500/5000/3000
			(Ver.2.3 or later)	(Ver.2.2 or earlier)	
CPU unit device	CPU input relays	-			R000 to R015
	CPU output relays	-			R500 to R507
	Internal auxiliary relays	R00000 to R199915	R00000 to R199915	R00000 to R99915	R01000 to R99915
		MR000000 to MR399915			MR00000 to MR99915
	Latch relays	LR00000 to LR99915			
	Control relays	CR0000 to CR7915			CR0000 to CR3915
	Timers	T0000 to T3999			
	Counters	C0000 to C3999			
	Data memory entries	DM00000 to DM65534			
	Expansion data memory entries	EM00000 to EM65534			
	File registers	FM00000 to FM32767 (×16) (ZF00000 to ZF524287)			FM00000 to FM32767 (×4) (ZF00000 to ZF131171)
	Temporary data memory entries	TM000 to TM511			
	Control memory entries	CM0000 to CM7599	CM0000 to CM5999		
	High-speed counters	-			CTH0 to CTH1
	High-speed counter comparators	-			CTC0 to CTC3
	Link relays	B0000 to B7FFF			B0000 to B3FFF
	Link data memory entries	W0000 to W7FFF			W0000 to W3FFF
	Index registers	Z01 to Z12			
	Digital trimmers	TRM0 to TRM7			

Instructions That Have Differences

The different instructions when changing the model from the KV-5000/3000 series to the KV-8000 series are as follows.

* The instructions are the same when changing the model from the KV-7000 series to the KV-8000 series.

Instruction		When Changing Model	When Converting KV STUDIO	When Executed
Storage Instruction	MFREE	Replace to MFREE D1 D2	–	Normal Operation
	MFREEK	Replace to MFREE D1 D2	–	Normal Operation
Access Window Instruction	AWNUM	Change to unregistered variable	Occurrence of Error	–
	AWMSG	Change to unregistered variable	Occurrence of Error	–
Positioning Instruction	PLSX	Change to unregistered variable	Occurrence of Error	–
	PLSY	Change to unregistered variable	Occurrence of Error	–
	JOGX	Change to unregistered variable	Occurrence of Error	–
	JOGY	Change to unregistered variable	Occurrence of Error	–
	ORGX	Change to unregistered variable	Occurrence of Error	–
	ORGY	Change to unregistered variable	Occurrence of Error	–
	HOMEX	Change to unregistered variable	Occurrence of Error	–
	HOMEY	Change to unregistered variable	Occurrence of Error	–
	CHGSPX	Change to unregistered variable	Occurrence of Error	–
	CHGSPY	Change to unregistered variable	Occurrence of Error	–
	RFSPSX	Change to unregistered variable	Occurrence of Error	–
	RFSPSY	Change to unregistered variable	Occurrence of Error	–
High Speed Processing Instruction	HSP	Change to unregistered variable	Occurrence of Error	–
	CTH	Change to unregistered variable	Occurrence of Error	–
	CTC	Change to unregistered variable	Occurrence of Error	–
	RFSCTH	Change to unregistered variable	Occurrence of Error	–
Interruption Instruction	INT	Change to unregistered variable	Occurrence of Error	–
Fiel Register Instruction	FRSTM	Replace to FRSTM #0 n S D	–	Normal Operation
	FRLDM	Replace to FRLDM #0 n S D	–	Normal Operation
Cam-operated Switch Instruction	INCENC	Change to unregistered variable	Occurrence of Error	–
Frequency Counter Instruction	FCNT	Change to unregistered variable	Occurrence of Error	–
	RCNT	Change to unregistered variable	Occurrence of Error	–
	PLSOUT	Change to unregistered variable	Occurrence of Error	–

* When converting to KV-8000A, the unit-specific instructions related to the Mail send/receive function will be changed to unregistered variables.

Differences between Control Relays and Control Memory Entries

For details on the difference between CR (control relays) and CM (control memory entries), in KV STUDIO, on the "Help" menu click "Handy-Nav(reference programs/tech materials)," and then see "CR_CM_ConvertTableInChangeModel.xls."

Differences between variable and label

Item		Variable	Label
Supported CPU unit		• KV-8000(A)	• KV-7500/7300 • KV-5500/5000/3000
Specifications	Maximum number of definitions*1	100000*2	About 285000
	Maximum length of name	128 characters (regardless of full-width / half-width)	32 characters
	Automatic assignment to work area	Both global and local variables supported	Only local labels supported
	Initial value	○	—
	Constant	○	○
	Hold	○(Up to 30000 definitions)*3	—
Data type	BOOL(bit)	○	○
	INT(1-word signed integer)	○	○
	UINT(1-word unsigned integer)	○	○
	DINT(2-word signed integer)	○	○
	UDINT(2-word unsigned integer)	○	○
	REAL(Single-precision floating point)	○	○
	LREAL(Double-precision floating point)	○	○
	STRING(String)	○	○
	TIMER(Timer)	○	○
	COUNTER(Counter)	○	○
	Array	○	○
	Structure	○	—
	Function block	○	—
Others	Comment8 types	8 type	1 type
	Linked with VT STUDIO*4	○	—

*1 Number of definitions is the number of names registered as variables or labels. For example, if an array type variable or label with 10000 elements is registered, the number of definitions will be counted as 1.
*2 Depending on the number of characters in the variable name, 100000 may not be set.
*3 Depending on the number of characters in the variable name, 30000 may not be set.
*4 When using variables in VT STUDIO, VT STUDIO should be upgraded to 8.0 or later, and the VT model should be set to VT5 series or Soft-VT.

MEMO

11 Index

An index of definitions used in this manual. They are arranged in the alphabetical sequence.

Numerical Value

100ms On-delay 4-167
10ms On-delay 4-169
10µs On-delay 4-173
16-Bit Binary Data 4-75
1ms On-delay 4-171
32-Bit Binary Data 4-75

A

About CSV File 5-71
About Maintenance 3-18
About the Connector 3-13
About the Memory Card 5-6
About the Restoration Method 3-20
ABSENC instruction 5-183
AC power supply unit/power supply terminal block wiring 3-10
AC Power Unit
How to Wire 2-12
Specification 2-12
Access Window 5-111, 6-37
Names and Functions of Parts 5-114
Operations that Can be Implemented Through Access Window 5-111
Access window
Function list 5-112
Access Window Initial Menu Display Function 5-157
Access Window Settings Used by KV STUDIO 5-156
Access Window Setup 5-151
ACNT instruction 4-66
Alarm 5-136
All Clear 5-154
Applications 4-198
Argument Abbreviation Settings 4-138
Argument Privacy Settings 4-139
Argument types 4-132
Array 4-57
ASCII code 4-81
Auto Load Function 5-45

B

Bank Switchover 4-226
Battery 3-18
Binary BIN 4-76
Binary Coded Decimal BCD 4-77
Binary Data 4-75
Bistable function block (Set priority) 4-155
Bit device ON/OFF 4-273
Bit device processing of word devices 4-52
Bit Devices 4-27
Bit that is turned OFF for only one scan when function block execution starts, "@CR2007" 4-143
Bit that is turned OFF for only one scan when macro execution starts, "@CR2007" 4-188
Bit that is turned OFF for only one scan when module execution starts, "@CR2007" 4-113
Bit that is turned ON during execution, "_(macro name)_(ID number)" 4-188
Bit that is turned ON during execution, "_(module name)" 4-113
Bit That Is Turned ON during Macro Execution 4-189
Bit that is turned ON for only one scan when function block execution starts, "@CR2008" 4-143
Bit that is turned ON for only one scan when macro execution starts, "@CR2008" 4-188
Bit that is turned ON for only one scan when module execution starts, "@CR2008" 4-113
BSIZE instruction 4-70
Built-In Functions 5-2
Bus Connection Unit
Names and Functions of Parts 2-40

C

C 4-35, 4-42
Callable position 4-196
Cam Switch Function Overview 5-178
Cam Switch Instructions 5-178
Candidate instruction 4-239
Changes to values when the value is greater than the maximum value or smaller than the minimum value 4-80
Changing the current value and status of devices 4-273
Character Codes 4-81
Characters that cannot be used in labels A-39
Characters that cannot be used in program names A-39
Characters that cannot be used in project names A-39
CLEAR 5-153
Clock Function
Reading clock data 5-187
Writing clock data (Setting the time and date) 5-187
Condition control 4-4
Configuration by processes 4-111
Configuration Screen 6-39
Connecting line editing 4-242
Connecting to Communication Devices 1-5
When communicating with a PC 1-5

When communicating with a touch panel 1-5
Connecting Units 3-2
Connector Assembly and Wiring 3-12
Connector wiring 3-14
Constants 4-23, 4-73
Constructing Units 3-3
Contact protection 3-9
Control memory 2-3, 4-43
Control Memory CM 4-43
Control relay (contact) CR 4-36
Convert result 4-14
Correct device value window 4-273
Correlations between Value Notations 4-78
Counter 2-3, 4-35, 4-42
Counter maximum counting speed 4-36
CPU Memory (user memory) 5-27
CPU Monitor 5-134
CPU Unit 2-5
CPU Unit Operation 4-82
CSV files 5-71
CTD 4-163
CTU 4-161
CTUD 4-165

D

Data memory 2-3
Date/Time 5-132
0 second adjustment 5-133
Date/time setting 5-133
Functions of setup key 5-132
How to operate 5-132
Switching clock 5-132
Decimal constant 4-55
Decimal Constants 4-73
Decimal DEC 4-76
Default status of differential execution instructions 4-197
Device Comments 4-252
Device List for Logging/Tracing 5-79
Device Mode 5-118
Changing device values 5-123
Changing the device value display format .5-120
Changing type of device to be monitored ..5-119
Device name/comment display switching ..5-122
Functions of Setup key 5-118
Operation method 5-119
When display format are not ON/OFF (T, C, TRM) 5-125
When displaying format is other than ON/OFF (except T, C, TRM) 5-124
Device notation for relay (R) units 4-28
Device notation method 4-24
Device orientation 3-6
Devices used by the CPU 5-32
Devices using memory cards 5-10
DI instruction 4-218
DIC instruction 4-220
Differences between Macros and Subroutines 4-195
Digital Trimmer 5-128
Functions of setup key 5-128
How to operate 5-128
Dimensions 2-17
Expansion I/O unit (KV-SIR32XT) 2-17
Displaying the assignment destinations of local devices and local labels 4-275
Displaying Unit Monitor 6-42
Double precision floating-point constants 4-74
Double precision floating-point real numbers DOUBLE-FLOAT 4-78
Down counter 4-163

E

Edit list 4-8
EI instruction 4-218
EN 4-122
ENO 4-122
Entry assistance 4-239
Error Clear 5-127
Ethernet/EtherNet/IP Communication (KV-7500 only) 5-110
The Role of CPU Built-In EtherNet/IP Communication Function 5-110
Ethernet/IP communication 5-110
Every-scan execution type 4-100
Exclusive Instruction for the Unit for KV-SIR32XT 6-23
Expansion I/O Unit (KV-SIR32XT)
Names and Functions of Parts 2-14
Specifications 2-15
Wiring and Internal Circuit Diagrams 2-16
Expansion input unit 2-18
Derate 2-19
Name and function of each part 2-18
Outline Dimensions 2-24
Specification 2-19
Expansion output unit
Name and function of each part 2-25
Outline Dimensions 2-33
Specification 2-26
Wiring and Internal Circuit Diagrams
Derate 2-26
Expansion unit 2-42
Name and function of each part 2-42
Outline Dimensions 2-45
Specification 2-43
Extension I/O unit 2-34
Overall dimension drawing 2-39

F

Falling edge detector 4-160
FB instruction 4-148
FBCALL instruction 4-151
FBSTRT instruction 4-151
Feedback control 4-2
FEND instruction 4-148
File Register 4-226
File Register ZF/FM 4-39
Fixed character strings 4-74
FRLDM 命令 4-234
FRSET instruction 4-230
FRSTM 命令 4-232
F_TRIG 4-160
FUN instruction 4-148
Function block instance 4-121
Function block management 4-117
Function Block Nests 4-140
Function Block Program Properties 4-144
  Disable program 4-144
  Perform OFF Process When Stopping Execution 4-144
  Read Only 4-144
Function Block System Devices 4-143
Function Block Type 4-120
Function block usage example 4-137
Function blocks 4-120, A-21
Functions 4-128, A-21

G

General I/O 6-2
General Specifications 2-2
Global devices 4-200
Global devices notation 4-200
Grounding precautions 3-8

H

HEX 4-76
Hexadecimal 4-76
Hexadecimal Constants 4-73
How to Connect to a PC 5-109
How to handle the set value and current value of a counter 4-36
How to handle the set value and current value of a timer 4-34
How to program counter operations 4-35
How to program Timer operations 4-33
How to search for rung comments 4-255
How to set the bit that Is turned ON during execution 4-189
How to Write Local Devices 4-200

I

I/O unit wiring precautions 3-8
ID No. 4-189
IEDGE instruction 4-224
Index constants 4-45
Index Modification 4-44, 4-265
Index register 2-3, 4-43, 4-44
Indirect Specification 4-46, 4-265
Infinity 4-80
Initialize module 4-100, A-21
Input relay 2-3, 4-27
Inspection 3-18
Inspection and Maintenance 3-18
Installation Environment 3-6
Installation Position 3-6
Installing the memory card 5-8
Installing the USB Driver 5-109
Installing Units on DIN Rails 3-5
Instance 4-120
Instruction list display 4-239
Instruction operand simple explanation display 4-239
Instructions A-23
INT instruction 4-222
Internal auxiliary relay 2-3, 4-30
Internal registers 4-23, 4-53
Internal work consumption 4-15
Interrupt programs 4-96
Interrupts 4-206
  Instructions That Cannot Be Used in Interrupt Programs 4-210
Inter-unit Synchronization Function
  CPU Device Write/Read 5-170
  Cycle of Inter-unit Synchronization 5-171
  Inter-Unit Synchronization Refresh Devices 5-169
  Inter-Unit Synchronization Tracing 5-176
  Inter-unit Synchronous Module 5-166
  Operation When the Inter-Unit Synchronization Module is Not Used 5-167
  Performance Specifications of Inter-unit Synchronization 5-172
  Refresh Timing Settings (For Expansion unit relays and DM only) 5-168
  Setting CPU Device Write/Read 5-170
  Setting Method for Inter-unit Synchronization 5-173
  Setting Steps for Inter-unit Synchronization 5-174
  Setting the Inter-unit Sync Refresh Devices 5-169
  What is the Inter-unit Synchronization Function 5-165
Inter-unit synchronization function (S-Unit)
  Performance Overview when Running Inter-unit Synchronization Function 6-6

Unit Specification when Running Inter-unit Synchronization Function 6-8
What is the Inter-unit Synchronization Function 6-6

K

Key Lock Function 5-163
KV Script 4-7

L

Ladder Program 4-5
Latch relay 2-3, 4-31
Link register W 4-41
Link relay B 4-29
List of Abbreviated Mnemonics A-23
List of argument devices 4-186
List of Characters That Cannot Be Used A-39
Character that cannot be used in device comments A-41
Characters that cannot be used in function / function block / structure names A-39
Characters that cannot be used in labels A-39
Characters that cannot be used in program names A-39
Characters that cannot be used in project names A-39
Characters that cannot be used in structure member names A-41
List of Exclusive Function for the Unit for KV-SIR32XT 6-30
List of Exclusive Instructions for the Unit for KV-SIR32XT 6-23
List of PLC Unit Errors A-2
List of Shortcut Keys A-25
How to Use the / Key A-37
List of CX Format Shortcut Keys A-34
List of GPPQ and GPPA Format Shortcut Keys A-33
List of Logix Format Shortcut Keys A-35
Shortcut Keys That Are Valid in KV STUDIO A-27
Shortcut Keys That Are Valid in the Unit Editor A-25
List of Unusable Instructions A-21
List of Variables and Data Types 4-56
Lists of CR and CM Devices A-10
Lists of Devices Bit devices 4-20
Word devices 4-21
Local Device Assignment 4-202
Local Devices 4-200
Log Instructions 5-73
LOGD instructions 5-75
LOGE instructions 5-75
Logging function 5-57
Logging/Tracing Function Precautions 5-83
LR (Latch relay) 4-31

M

Macro argument devices 4-22
Macro management 4-178
Macro Passwords 4-191
Macro System Devices 4-188
Macro Type 4-181
Macro Usage Procedure 4-180
Macros A-22
Main routine program 4-94
Maintenance 3-18
Maximum number of connected units 3-2
MCALL instruction 4-192
MCMP instruction 5-179
MDSTOP instruction 4-115
MDSTRT instruction 4-115
MEND instruction 4-192
Mnemonics 4-8
Module Configuration Examples 4-110
Module Execution Sequence 4-109
Module System Devices 4-113
Module Type 4-100
Monitor Screen 6-40
Monitoring expansion units 4-277
Monitoring instructions 4-276
Monitoring rising and falling edges of devices 4-276
Monitoring the execution time in program and interrupt program units (Performance monitor) 4-277
MR 4-31
MSTRT instruction 4-192

N

Names and Functions of Parts 2-6, 2-11
Nesting 4-95

O

Object creation 4-195
Online edit 4-269
Operation during Power Outages 2-4
Operation Inhibited Setting 5-162
Operation when the execution condition is turned OFF 4-197
Operation when the program is called between the MC (master control) and MCR (master control reset) instructions 4-197
Option 5-67
Outline Dimensions 2-10
Bus connection unit 2-10

CPU Unit 2-10
End unit 2-10
Output relay 2-3, 4-27
Overview of Recipe Load Function / Recipe Save Function 5-50

P

Package Contents 2-5
Password 4-18
Performance Specifications 2-3
Placing Function Blocks 4-135
Precautions on wiring (OP-22184/51404) 3-12
Precautions When Connecting Units 3-2
Precautions When Using a memory card 5-9
Precautions When Using the USB Port or Serial Communication Port to Communicate with a PC or Other Peripheral 3-16
Precautions when using Timer instructions 4-34
Procedure for generating / reading recipe data 5-52
Program capacity 4-14
Program Configuration 4-93
Programming Language 4-5
Programming without considering modules ...4-112
Project Configuration 4-11
Project property 4-18
Project Protection Settings 4-16

R

R 4-27, 4-30
RAM clear 5-153
RAM mode 5-47
Recipe load / save procedure 5-54
Recipe load completion code / recipe save completion code 5-56
Register and batch monitors 4-274
Relation mapping 4-246
Removing the memory card 5-8
Reserved area 4-202
Restoration Method 3-21
How to clear errors 3-21
How to clear the PLC status 3-21
How to set clock data 3-20
RETI instruction 4-222
RS 4-157
R_TRIG 4-159
Rung comments 4-254
RUN-PRG Selector Switch 2-8

S

Scan Time 4-90
Self-hold macros 4-183
Separate usages for index modification and indirect specification 4-265
Sequence control 4-4
Series Common Specifications 2-2
Set display mode 4-275
Setting a Function Block Password 4-141
Setting Arguments 4-131
Setting Function Blocks to Read Only 4-142
Shortcut keys that are valid during online editing A-30
Edit A-30
Help A-32
Monitor/Simulator A-32
ST/Script A-31
View A-31
Shortcut keys that are valid in editor mode A-27
Convert A-28
Edit A-27
File A-27
Help A-29
Monitor/Simulator A-29
ST/Script A-28
Tool A-29
View A-28
Shortcut keys that are valid in monitor mode .. A-29
Debugging A-29
Edit A-29
Help A-29
Monitor/Simulator A-29
View A-29
Shortcut keys that are valid in simulator mode A-30
Edit A-30
Help A-30
Monitor/Simulator A-30
View A-30
Shortcut Keys That Are Valid in the Unit Editor A-25
Convert A-25
Edit A-25
File A-25
Help A-25
Operations in message area A-25
Operations in the unit configuration area A-25
Operations in unit selection window A-25
Operations in unit setup window A-25
View A-25
Window A-25
Shortcut keys that are valid on the initial screen A-27
File A-27
Monitor/Simulator A-27
View A-27

Shortcut keys that are valid on the Unit Editor slave setup screen A-26
Shortcut keys valid in the replay mode A-32
Signed 4-76
Simple indirect specification #TM 4-49
Single precision floating point type constants ..4-73
Single precision floating-point constants 4-74
Single precision floating-point real numbers FLOAT 4-77
Space surrounding the device 3-7
Special-purpose crimping tool (OP-21734) 3-12
Split 4-251
SR 4-155
ST Language 4-6
Standby module 4-101
Storage 5-139
Storage Instructions 5-26, 5-49
Structure 4-59
Structure Definition 4-65
Subroutine macros 4-181
Subroutine programs 4-94, 4-195, A-22
Suffixes 4-9
Summary of Unit Monitor 6-42
System Configuration 1-2
System Configuration Overview 1-2
(1) Power Units 1-3
(2) CPU Units 1-3
(3) KV-7000 Series Expansion Units 1-3
(4) Bus Connection Units 1-4
(5) KV-5000/3000 Series Expansion Units 1-4
(6) End Units 1-4
System Function Block 4-146
System Program Update 3-23

T

T 4-32, 4-42
Temporary data memory TM 4-42
Temporary memory 2-3
Time limit control 4-4
Timer 2-3, 4-32, 4-42
Timer error 4-33
TON_100MS 4-167
TON_10MS 4-169
TON_10US 4-173
TON_1MS 4-171
Total steps 4-15
TRGD instructions 5-77
TRGR instruction 5-105
Types of Sequence Control 4-4

U

U_DISOCP instruction 6-28
U_DISOVP 6-32
U_DISOVP instruction 6-26
UNICODE 4-81
Unit Installation 3-6
Unit Monitor 5-115
Functions of setup key 5-115
Method of Operation (example: on I/O Unit) 5-116
Unit Test 5-130, 6-41
Unit Wiring Precautions 3-8
Up counter 4-161
Up/Down counter 4-165
U_RDBF 6-31
U_RDBF instruction 6-24
Usage example together with index modification 4-48
Usage Procedure 4-119
USB Communication 5-109
User Memory Clear 5-155
User Message 5-161
Display example 5-161
How to Display User Message 5-161

V

Value Notation 4-76
Variable 4-55
Variable Setting 4-63
Variable Types 4-55

W

What are arguments? 4-186
What You Can Do with the Built-In Functions 5-2
What You Can Do with the Memory Card 5-3
Wiring and Internal Circuit Diagram 2-8
Ethernet communication specifications (KV-7500) 2-9
KV-8000(A) 2-8
Wiring and Internal Circuit Diagrams 2-16
KV-SIR32XT I/O wiring diagram 2-16
KV-SIR32XT internal circuit diagram (input circuit) 2-16
KV-SIR32XT internal circuit diagram (output circuit) 2-16
Word Devices 4-37
Writing clock data (Setting the time and date) 5-187
Writing/Reading Project Data 5-35
WSIZE instruction 4-68

Z

Zero Suppression 4-27

Revision log

Printing Date	Version	Details of Revision
Aug 2019	1st version	Official release
Sep 2019	2nd version	
May 2020	1st revision 1st version	
Apr 2021	2nd revision 1st version	
Sep 2021	3rd revision 1st version	
Nov 2021	5th revision 1st version	
May 2022	6th revision 1st version	
Sep 2022	8th revision 1st version	

WARRANTIES AND DISCLAIMERS

(1) KEYENCE warrants the Products to be free of defects in materials and workmanship for a period of one (1) year from the date of shipment. If any models or samples were shown to Buyer, such models or samples were used merely to illustrate the general type and quality of the Products and not to represent that the Products would necessarily conform to said models or samples. Any Products found to be defective must be shipped to KEYENCE with all shipping costs paid by Buyer or offered to KEYENCE for inspection and examination. Upon examination by KEYENCE, KEYENCE, at its sole option, will refund the purchase price of, or repair or replace at no charge any Products found to be defective. This warranty does not apply to any defects resulting from any action of Buyer, including but not limited to improper installation, improper interfacing, improper repair, unauthorized modification, misapplication and mishandling, such as exposure to excessive current, heat, coldness, moisture, vibration or outdoors air. Components which wear are not warranted.

(2) KEYENCE is pleased to offer suggestions on the use of its various Products. They are only suggestions, and it is Buyer's responsibility to ascertain the fitness of the Products for Buyer’s intended use. KEYENCE will not be responsible for any damages that may result from the use of the Products.

(3) The Products and any samples ("Products/Samples") supplied to Buyer are not to be used internally in humans, for human transportation, as safety devices or fail-safe systems, unless their written specifications state otherwise. Should any Products/Samples be used in such a manner or misused in any way, KEYENCE assumes no responsibility, and additionally Buyer will indemnify KEYENCE and hold KEYENCE harmless from any liability or damage whatsoever arising out of any misuse of the Products/Samples.

(4) **OTHER THAN AS STATED HEREIN, THE PRODUCTS/SAMPLES ARE PROVIDED WITH NO OTHER WARRANTIES WHATSOEVER. ALL EXPRESS, IMPLIED, AND STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY RIGHTS, ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL KEYENCE AND ITS AFFILIATED ENTITIES BE LIABLE TO ANY PERSON OR ENTITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, PUNITIVE, SPECIAL OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, ANY DAMAGES RESULTING FROM LOSS OF USE, BUSINESS INTERRUPTION, LOSS OF INFORMATION, LOSS OR INACCURACY OF DATA, LOSS OF PROFITS, LOSS OF SAVINGS, THE COST OF PROCUREMENT OF SUBSTITUTED GOODS, SERVICES OR TECHNOLOGIES, OR FOR ANY MATTER ARISING OUT OF OR IN CONNECTION WITH THE USE OR INABILITY TO USE THE PRODUCTS, EVEN IF KEYENCE OR ONE OF ITS AFFILIATED ENTITIES WAS ADVISED OF A POSSIBLE THIRD PARTY’S CLAIM FOR DAMAGES OR ANY OTHER CLAIM AGAINST BUYER.** In some jurisdictions, some of the foregoing warranty disclaimers or damage limitations may not apply.

BUYER'S TRANSFER OBLIGATIONS:

If the Products/Samples purchased by Buyer are to be resold or delivered to a third party, Buyer must provide such third party with a copy of this document, all specifications, manuals, catalogs, leaflets and written information provided to Buyer pertaining to the Products/Samples.

Specifications are subject to change without notice.

KEYENCE CORPORATION

www.keyence.com/glb

1-3-14, Higashi-Nakajima, Higashi-Yodogawa-ku, Osaka, 533-8555, Japan PHONE: +81-6-6379-2211

AUSTRIA
Phone: +43 (0)2236 378266 0

BELGIUM
Phone: +32 (0)15 281 222

BRAZIL
Phone: +55-11-3045-4011

CANADA
Phone: +1-905-366-7655

CHINA
Phone: +86-21-3357-1001

CZECH REPUBLIC
Phone: +420 220 184 700

FRANCE
Phone: +33 1 56 37 78 00

GERMANY
Phone: +49-6102-3689-0

HONG KONG
Phone: +852-3104-1010

HUNGARY
Phone: +36 1 802 7360

INDIA
Phone: +91-44-4963-0900

INDONESIA
Phone: +62-21-2966-0120

ITALY
Phone: +39-02-6688220

KOREA
Phone: +82-31-789-4300

MALAYSIA
Phone: +60-3-7883-2211

MEXICO
Phone: +52-55-8850-0100

NETHERLANDS
Phone: +31 (0)40 206 6100

PHILIPPINES
Phone: +63-(0)2-8981-5000

POLAND
Phone: +48 71 368 61 60

ROMANIA
Phone: +40 (0)269 232 808

SINGAPORE
Phone: +65-6392-1011

SLOVAKIA
Phone: +421 (0)2 5939 6461

SLOVENIA
Phone: +386 (0)1 4701 666

SWITZERLAND
Phone: +41 (0)43 455 77 30

TAIWAN
Phone:+886-2-2721-1080

THAILAND
Phone: +66-2-078-1090

UK & IRELAND
Phone: +44 (0)1908-696-900

USA
Phone: +1-201-930-0100

VIETNAM
Phone: +84-24-3772-5555

B5WW1-MAN-2062

Copyright (c) 2022 KEYENCE CORPORATION. All rights reserved.

084755GB 2092-1 D42GB Printed in Japan

D42GB-1