MONITOUCH

Macro Reference

Hakko Electronics Co., Ltd.

Record of Revisions

Reference numbers are shown at the bottom left corner on the back cover of each manual.

Date	Reference No.	Revised Contents
May, 2014	1071NE0	First edition

Preface

Congratulations on purchasing the MONITOUCH V series.
The “V Series Macro Reference” manual describes macro functions used on the drawing/editing software (V-SFT version 6) for the MONITOUCH V series. For a correct use of the product, read this manual thoroughly.

About Manuals

The following manuals are available for the MONITOUCH V series.
Refer to them as necessary.

Manual Name	Reference No.	Contents
V9 Series Macro Reference (this manual)	1071NEx	An overview of macros of V-SFT version 6 as well as macro editor operations and macro command descriptions are explained.
V9 Series Reference Manual [1]	1065NEx	The functions and instructions of the V9 series are explained.
V9 Series Reference Manual [2]	1066NEx	
V9 Series Troubleshooting/Maintenance Manual	1068NEx	Errors and operation procedures of the V9 series are explained.
V9 Series Hardware Specifications	2023NEx	Notes on usage and hardware specifications for the V9 series are explained.
V9 Series Connection Manual [1]	2210NEx	The connection and communication parameters for the V9 series and controllers are explained in detail.
V9 Series Connection Manual [2]	2211NEx	
V9 Series Connection Manual [3]	2212NEx	

V Series Models

The following V9 series models are available:

Generic Name	Series	Model
V9 series	V9 Standard	V9100iS
		V9080iS
	V9 Lite	V9010iC
		V9080iC

Please note that the V series model names are used as listed above in the manuals.

Notes on Safe Usage of MONITOUCH

In this manual, you will find various notes categorized under the following levels with the signal words “DANGER” and “CAUTION”.

⚠DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury and could cause property damage.

Note that there is a possibility that items listed with **⚠CAUTION** may have serious ramifications.

⚠DANGER

- Never use the output signal of the V9 series for operations that may threaten human life or damage the system, such as signals used in case of emergency. Please design the system so that it can cope with a touch switch malfunction. A touch switch malfunction may result in machine accidents or damage.
- Turn off the power supply when you set up the unit, connect new cables, or perform maintenance or inspections. Otherwise, electrical shock or damage may occur.
- Never touch any terminals while the power is on. Otherwise, electrical shock may occur.
- You must cover the terminals on the unit before turning the power on and operating the unit. Otherwise, electrical shock may occur.
- The liquid crystal in the LCD panel is a hazardous substance. If the LCD panel is damaged, do not ingest the leaked liquid crystal. If leaked liquid crystal makes contact with skin or clothing, wash it away with soap and water.
- Never disassemble, recharge, deform by pressure, short-circuit, reverse the polarity of the lithium battery, nor dispose of the lithium battery in fire. Failure to follow these conditions will lead to explosion or ignition.
- Never use a lithium battery that is deformed, leaking, or shows any other signs of abnormality. Failure to follow these conditions will lead to explosion or ignition.
- The power lamp flashes when the backlight has reached the end of its service life or when the backlight is faulty. Note that the switches on the screen remain operable when this occurs. Do not touch the screen when the screen becomes dark and the power lamp is flashing. Otherwise, a malfunction may occur and result in machine accidents or damage.

⚠CAUTION

- Check the appearance of the unit when it is unpacked. Do not use the unit if any damage or deformation is found. Failure to do so may lead to fire, damage, or malfunction.
- For use in a facility or as part of a system related to nuclear energy, aerospace, medical, traffic equipment, or mobile installations, please consult your local distributor.
- Operate (or store) the V9 series under the conditions indicated in this manual and related manuals. Failure to do so could cause fire, malfunction, physical damage, or deterioration.
- Observe the following environmental restrictions on use and storage of the unit. Otherwise, fire or damage to the unit may result.
  - Avoid locations where there is a possibility that water, corrosive gas, flammable gas, solvents, grinding fluids, or cutting oil can come into contact with the unit.
  - Avoid high temperatures, high humidity, and outside weather conditions, such as wind, rain, or direct sunlight.
  - Avoid locations where excessive dust, salt, and metallic particles are present.
  - Avoid installing the unit in a location where vibrations or physical shocks may be transmitted.

⚠CAUTION

- Equipment must be correctly mounted so that the main terminal of the V9 series will not be touched inadvertently. Otherwise, an accident or electric shock may occur.
- Tighten the mounting screw on the fixtures of the V9 series to an equal torque of 0.6 N·m. Excessive tightening may distort the panel surface. Loose mounting screws may cause the unit to fall down, malfunction, or short-circuit.
- Check periodically that terminal screws on the power supply terminal block and fixtures are firmly tightened. Loosened screws or nuts may result in fire or malfunction.
- Tighten the terminal screws on the power supply terminal block of the V9 series to an equal torque of 7.1 to 8.8 inch-lbf (0.8 to 1.0 N·m). Improper tightening of screws may result in fire, malfunction, or other serious trouble.
- The V9 series has a glass screen. Do not drop the unit or impart physical shocks to the unit. Otherwise, the screen may be damaged.
- Correctly connect cables to the terminals of the V9 series in accordance with the specified voltage and wattage. Overvoltage, overwattage, or incorrect cable connection could cause fire, malfunction, or damage to the unit.
- Always ground the V9 series. The FG terminal must be used exclusively for the V9 series with the level of grounding resistance less than 100 Ω. Otherwise, electric shock or a fire may occur.
- Prevent any conductive particles from entering the V9 series. Failure to do so may lead to fire, damage, or malfunction.
- After wiring is finished, remove the paper used as a dust cover before starting operation of the V9 series. Operation with the dust cover attached may result in accidents, fire, malfunction, or other trouble.
- Do not attempt to repair the V9 series yourself. Contact Hakko Electronics or the designated contractor for repairs.
- Do not repair, disassemble, or modify the V9 series. Hakko Electronics Co., Ltd. is not responsible for any damages resulting from repair, disassembly, or modification of the unit that was performed by an unauthorized person.
- Do not use sharp-pointed tools to press touch switches. Doing so may damage the display unit.
- Only experts are authorized to set up the unit, connect cables, and perform maintenance and inspection.
- Lithium batteries contain combustible material such as lithium and organic solvents. Mishandling may cause heat, explosion, or ignition resulting in fire or injury. Read the related manuals carefully and correctly handle the lithium battery as instructed.
- Do not press two or more positions on the screen at the same time. If two or more positions are pressed at the same time, the switch located between the pressed positions may be activated.
- Take safety precautions during operations such as changing settings when the unit is running, forced output, and starting and stopping the unit. Any misoperations may cause unexpected machine movement, resulting in machine accidents or damage.
- In facilities where the failure of the V9 series could lead to accidents that threaten human life or other serious damage, be sure that such facilities are equipped with adequate safeguards.
- When disposing of the V9 series, it must be treated as industrial waste.
- Before touching the V9 series, discharge static electricity from your body by touching grounded metal. Excessive static electricity may cause malfunction or trouble.
- Insert an SD card into the unit in the same orientation as pictured on the unit. Failure to do so may damage the SD card or the slot on the unit.
- The SD card access LED flashes red when the SD card is being accessed. Never remove the SD card or turn off power to the unit while the LED is flashing. Doing so may destroy the data on the SD card. Check that the LED has turned off before removing the SD card or turning off the power to the unit.

[General Notes]

- Never bundle control cables or input/output cables with high-voltage and large-current carrying cables such as power supply cables. Keep control cables and input/output cables at least 200 mm away from high-voltage and large-current carrying cables. Otherwise, malfunction may occur due to noise.
- When using the V9 series in an environment where a source of high-frequency noise is present, it is recommended that the FG shielded cable (communication cable) be grounded at each end. However, when communication is unstable, select between grounding one or both ends, as permitted by the usage environment.
- Be sure to plug connectors and sockets of the V9 series in the correct orientation. Failure to do so may lead to damage or malfunction.
- If a LAN cable is inserted into the MJ1 or MJ2 connector, the device on the other end may be damaged. Check the connector names on the unit and insert cables into the correct connectors.
- Do not use thinners for cleaning because it may discolor the V9 series surface. Use commercially available alcohol.
- If a data receive error occurs when the V9 series unit and a counterpart unit (PLC, temperature controller, etc.) are started at the same time, read the manual of the counterpart unit to correctly resolve the error.
- Avoid discharging static electricity on the mounting panel of the V9 series. Static charge can damage the unit and cause malfunctions. Discharging static electricity on the mounting panel may cause malfunction to occur due to noise.
- Avoid prolonged display of any fixed pattern. Due to the characteristic of liquid crystal displays, an afterimage may occur. If prolonged display of a fixed pattern is expected, use the backlight's auto OFF function.
- The V9 series is identified as a class-A product in industrial environments. In the case of use in a domestic environment, the unit is likely to cause electromagnetic interference. Preventive measures should thereby be taken appropriately.

[Notes on the LCD]

Note that the following conditions may occur under normal circumstances.

- The response time, brightness, and colors of the V9 series may be affected by the ambient temperature.
- Tiny spots (dark or luminescent) may appear on the display due to the characteristics of liquid crystal.
- There are variations in brightness and color between units.

Contents

Chapter 1 Outline

Chapter 2 Edit

1 Outline

1.1 Type of V Series Macros

Macros, created with V-series-specific commands, are used to process user programs.
Macro creation is made simple with easy-to-use commands.
Macros are executable for the following occasions:

- Screen
  - OPEN macro: Executes once when the screen is opened.
  - CLOSE macro: Executes once when the screen is switched.
  - CYCLE macro: Executes repeatedly while the screen is open.

- Multi-overlap
  - OPEN macro: Executes once when the multi-overlap is opened.
  - CLOSE macro: Executes once when the multi-overlap is closed.
    * OPEN and CLOSE macros cannot be used for call-overlaps.

- Switch
  - ON macro: Executes once when the switch is pressed.
  - OFF macro: Executes once when the switch is released.

- Function switch
  - ON macro: Executes once when the function switch is pressed.
  - OFF macro: Executes once when the function switch is released.

- Initial macro
  The specified macro block executes once before the V series starts communicating with the PLC. (Refer to page 1-6.)

- Global macro
  The specified macro block is executed once when the control device memory is changed from 0 to 1 (leading edge). (Refer to page 1-7.)

- Event timer macro
  The specified macro block executes at regular intervals, regardless of which screen is currently displayed. (Refer to page 1-8.)

- Interval timer
  While a screen equipped with the interval timer is displayed, the timer starts as preset. Each time the preset time has elapsed, the specified macro block is executed. (Refer to page 1-9.)

- Macro mode
  While a screen equipped with macro mode is displayed, macros are executed according to the status at the specified device memory addresses. (Refer to page 1-15.)
  - ON macro: Executes when the bit at the specified device memory address changes from 0 → 1 (leading edge).
  - OFF macro: Executes when the bit at the specified device memory address changes from 1 → 0 (falling edge).
- Alarm macro
  When a macro is set in the [Alarm Server] window, it is executed according to a change in the status of the device memory for errors. (Refer to page 1-17.)
  - Occurrence macro: To be executed at the time of alarm occurrence
  - Resetting macro: To be executed at the time of alarm reset

- Scheduler macro
  When a macro is set in the [Scheduler] window, it is executed at the timing specified for [Trigger]. (Refer to page 1-18.)

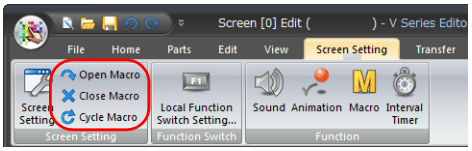
1.2 Screen Macro

This macro is registered for screens.
Registered commands are executed at the following timings:

- OPEN macro
  This macro is executed once when a screen is opened.
  Select [Screen Setting] → [Open Macro] and register the command to be executed.

- CLOSE macro
  This macro is executed once when a screen is closed.
  Select [Screen Setting] → [Close Macro] and register the command to be executed.

- CYCLE macro
  This macro is executed repeatedly while the screen is open.
  Select [Screen Setting] → [Close Macro] and register the command to be executed.

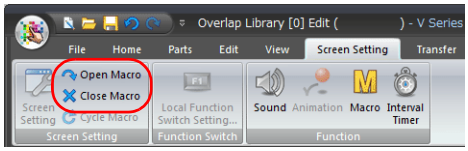


1.3 Multi-overlap Macro

This macro is registered for overlap displays.
Registered commands are executed at the following timings:

- OPEN macro
  This macro is executed once when a multi-overlap display is opened.
  Select [Screen Setting] → [Open Macro] in the overlap library window and register the command to be executed.

- CLOSE macro
  This macro is executed once when a multi-overlap display is closed.
  Select [Screen Setting] → [Close Macro] in the overlap library window and register the command to be executed.



1.4 Switch Macro

This macro is registered for switches.

- ON macro
  This macro is executed once when a switch is pressed.
  Set the command in the [Macro] window of the switch.

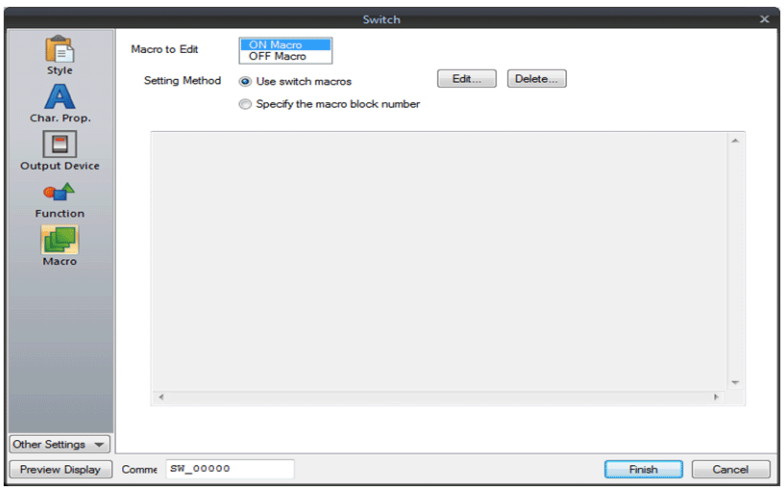
  - Editing with [Setting Method: Use switch macros] selected
    Macro commands are registered for switches.

  - Editing with [Setting Method: Specify the macro block number] selected
    Register macro commands in a macro block, and select a number of the macro block to execute.

- OFF macro
  This macro is executed once when a switch is released.
  Set the command in the [Macro] window of the switch.

  - Editing with [Setting Method: Use switch macros] selected
    Macro commands are registered for switches.

  - Editing with [Setting Method: Specify the macro block number] selected
    Register macro commands in a macro block, and select a number of the macro block to execute.

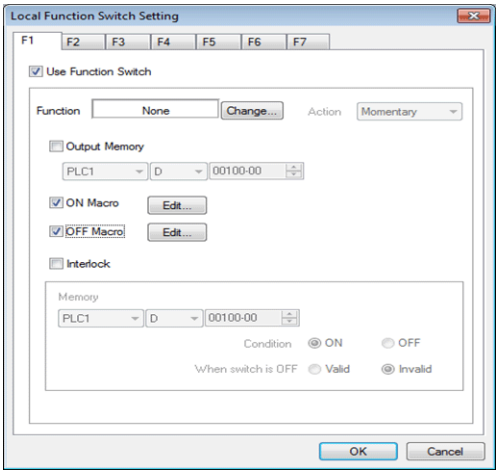


1.5 Function Switch Macro

This macro is registered for function switches.

- ON macro
  This macro is executed once when a switch is pressed.
  Set the command in the [Local Function Switch Setting] window.

- OFF macro
  This macro is executed once when a switch is released.
  Set the command in the [Local Function Switch Setting] window.

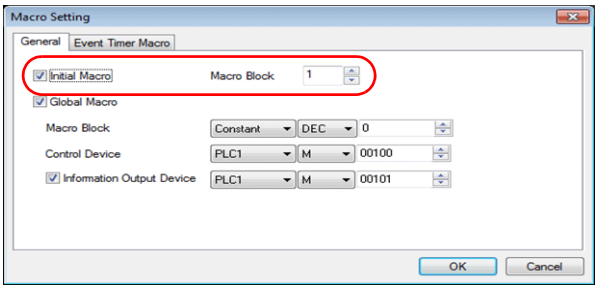


1.6 Initial Macro

An initial macro is executed once before the V series starts communicating with an external device.
Select [System Setting] → [Macro Setting] to make settings.
Register the command in [Macro Block].
Select [Home] → [Registration Item] → [Macro Block] to register a macro block.
For more information, refer to page 2-3.

Macro Setting

[General] tab window



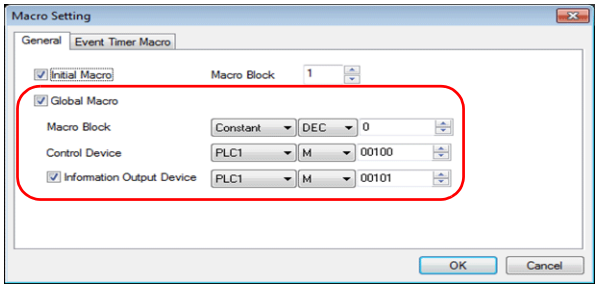
Initial Macro	Check this box to use an initial macro. Specify the macro block number to be executed before the V series starts communicating with the PLC. 0 - 1023: Macro block number

1.7 Global Macro

A global macro is executed when the bit is set (ON), regardless of the screen being displayed.
Select [System Setting] → [Macro Setting] to make settings.
Register the command in [Macro Block].
Select [Home] → [Registration Item] → [Macro Block] to register a macro block.
For more information, refer to page 2-3.

Macro Setting

[General] tab window



Global Macro	Check this box to use a global macro.
Macro Block	Specify the macro block number to be executed. It can also be specified by specifying a device memory address.
Control Device	Specify a macro start bit. The macro is executed when the specified bit changes from 0 → 1 (leading edge).
Information Output Device	This reflects the status of the control device memory.

Macro Execution Steps

1. Specify the number of the macro block for which commands to be executed are registered.
2. The control device memory is set ([0 → 1] leading edge).
   ↓
   Macro execution
   ↓
   The information output device memory is automatically set ([0 → 1]).
3. The control device memory is reset ([1 → 0] falling edge).

Supplemental Remarks

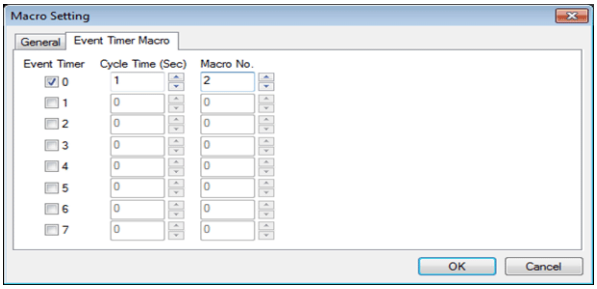
- By using the information output device memory, you can check the timing to reset (OFF) the control device memory.

1.8 Event Timer Macro

An event timer macro is executed at regular intervals, regardless of the screen being displayed.
Select [System Setting] → [Macro Setting] → [Event Timer Macro] to make settings.
Register the command in [Macro Block].
Select [Home] → [Registration Item] → [Macro Block] to register a macro block.
For more information, refer to page 2-3.

Macro Setting

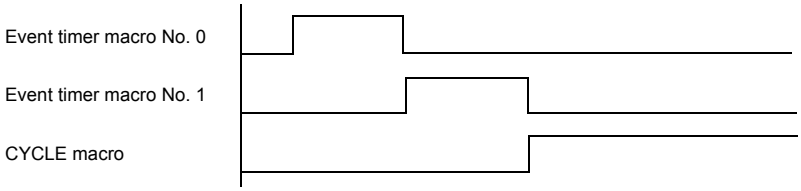
[Event Timer Macro] tab window



Event Timer	0 - 7 A maximum of eight event timer macro blocks can be set.
Cycle Time	0 - 3600 (sec) Specify a cycle time for the timer. The specified macro block is executed each time the specified time has elapsed.
Macro No.	0 - 1023 Specify the macro block number to be executed.

Supplemental Remarks

1. When the timers for multiple event timer macros are up at the same time:
   Event timer macro blocks are executed in ascending numeric order of [Event Timer]. After a macro block has been processed, execution proceeds to the next macro block.

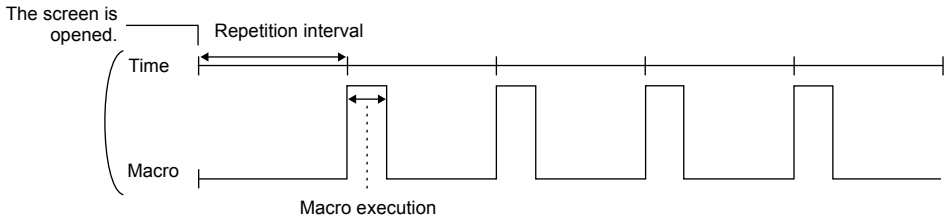


2. When accessing the same external device memory address in some event timer macros:
   The processing ability will be improved if you set the event timer macro No. 0 that reads the external device memory into the internal device memory and make other event timer macros refer to this internal device memory.
   In order to improve the overall processing ability, reduce the number of times that the external device memory is accessed.

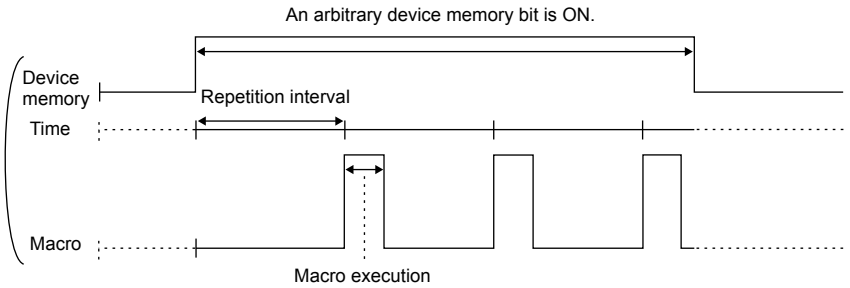
1.9 Interval Timer

The interval timer can be set for screens and multi-overlap displays.
Select [Screen Setting] → [Interval Timer] to make settings.
The interval timer has the following three functions.
Register the command in [Macro Block] for all cases.

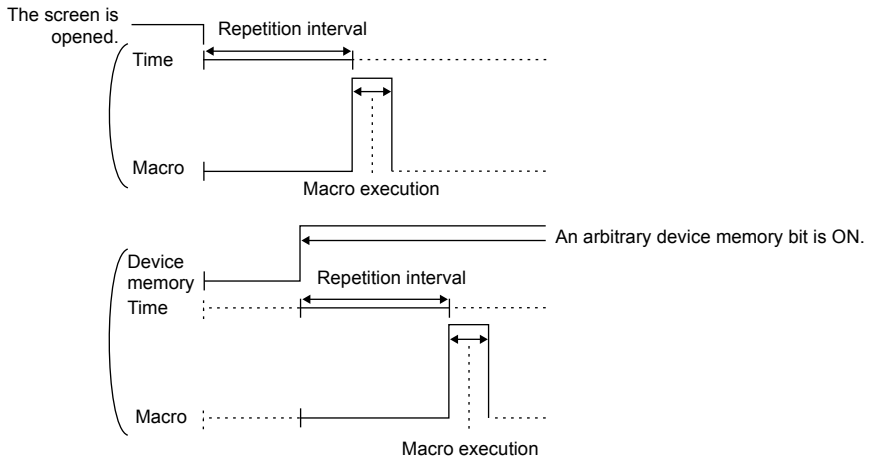
- The specified macro is executed at intervals specified for [Repeat Interval] from when the screen is opened.



- The specified macro is executed at intervals specified for [Repeat Interval] from when an arbitrary bit is set (ON). (This function is valid only while the bit is ON.)

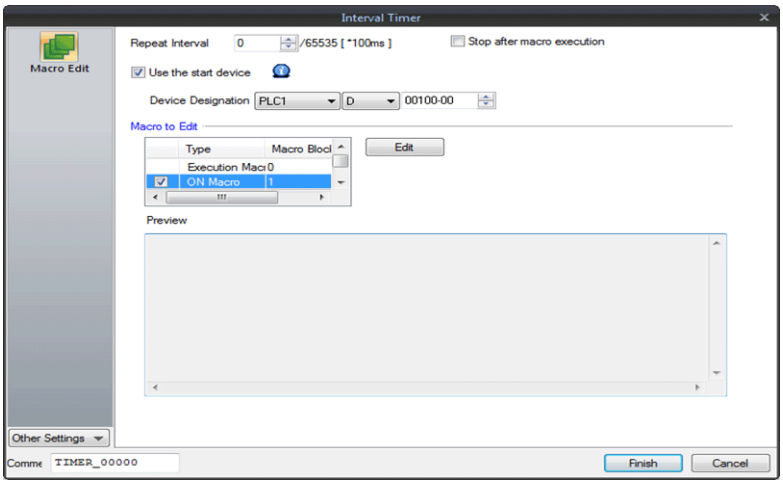


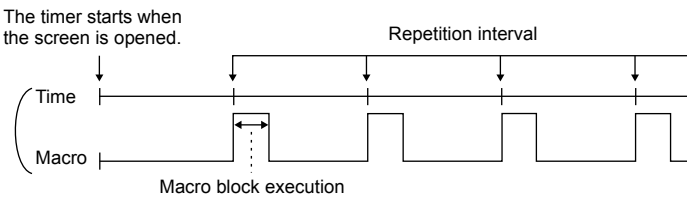
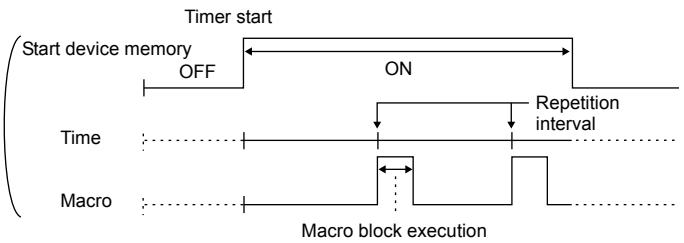
- The specified macro is executed once after the time specified for [Repeat Interval] has elapsed since the screen was opened or an arbitrary bit was set (ON).

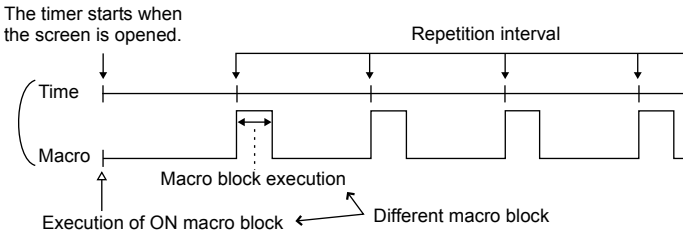
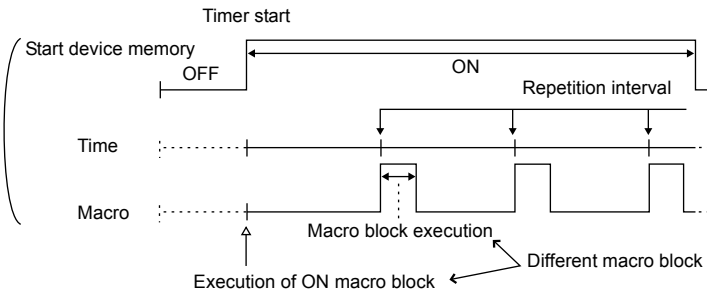
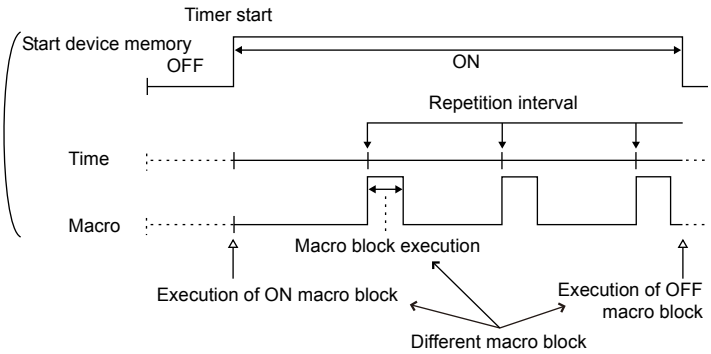


Setting Dialog

[Macro Edit] window

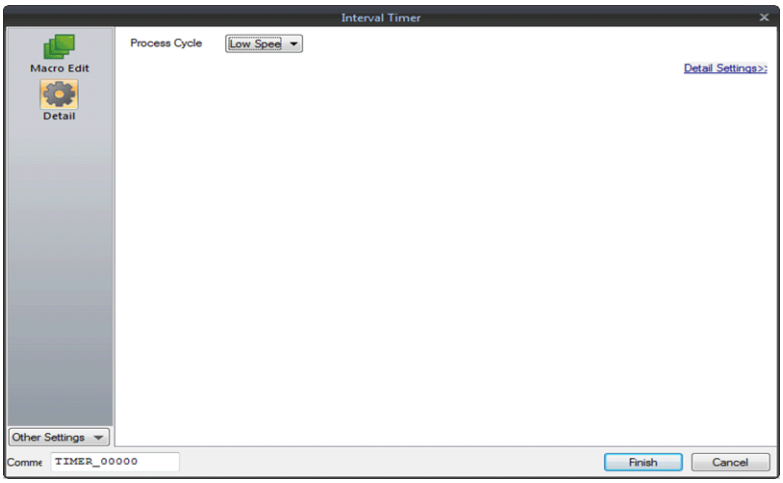


Repeat Interval *1	0 - 255 (× 100 msec) Specify a repetition interval to execute the macro. The macro is executed at the specified intervals. When “0” is specified, the macro is executed every cycle. The timing to start the timer depends on the setting for [Use the start device].
Stop after macro execution	Check this box to execute the macro only once. When the specified time has elapsed and the macro has been executed, the timer stops.
Use the start device	Check this box when specifying the start device memory address. • Unchecked  • Checked While the bit is “1” (ON): The macro is executed at intervals specified for [Repeat Interval]. While the bit is “0” (OFF): The macro is not executed. 
Execution Macro	Specify the macro block number to be executed at intervals specified for [Repeat Interval].

ON Macro	Specify the macro block number to be executed once when the timer starts. • [Use the start device] unchecked: When the screen is opened, the timer starts and the ON macro is executed once.  • [Use the start device] checked: The ON macro block is executed once when the start device memory bit is set to “1”. 
OFF Macro	This option is enabled, provided that [Use the start device] is checked. Specify the macro block number to be executed once when the start device memory bit is reset (1 → 0).  This option is useful for clearing the internal device memory that is used for a macro.
Edit	This button is used to open a macro block.
Preview	This area displays the contents of macros registered in the selected macro block number.

*1 The actual repetition interval may fluctuate according to the contents of the screen.

[Detail] window



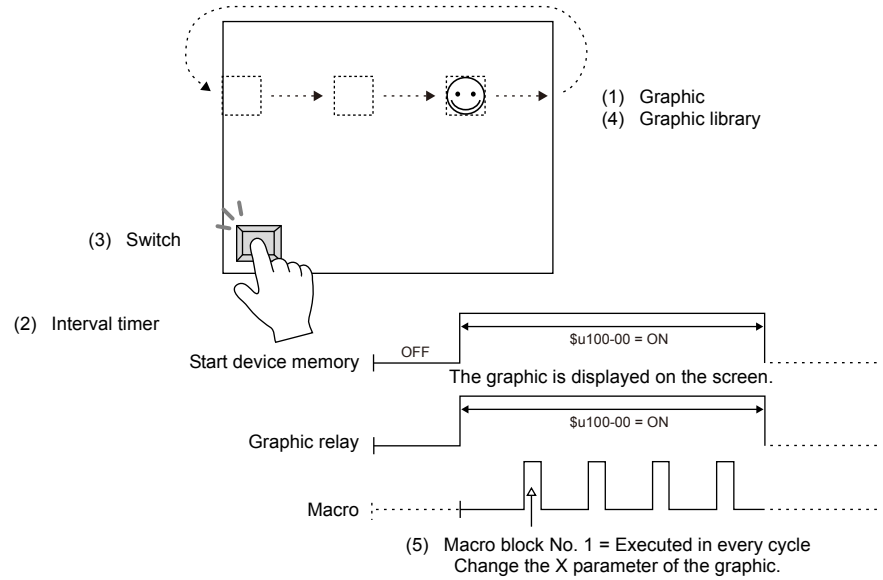
Process Cycle	Specify the cycle for the V series to read the PLC when they are communicating. For more information, refer to the V9 Series Reference Manual.
ID	Specify an ID.

Setting Example

Graphic movement on the screen

When the switch is pressed, a graphic from the graphic library is displayed. At the same time, the graphic placed on the left of the screen starts to move to the right.

Pressing the switch next clears the graphic. Pressing the switch again displays the graphic in the same position where it was displayed last. The graphic starts to move to the right.



Screen Edit

(1) Graphic
([Method: Device (Bit Designation)])

Number of Bits to Monitor: 1

Device Designation: $u100-00

Type: 1-Graphic

Mode: XOR

Start Graphic: GNo. 0 No. 0

Valid parameters No.: 1

(2) Interval timer

Repeat Interval: 0

☐Stop after macro execution

☑Use the start device: $u100-00

☑Execution Macro: Macro block No.: 1

☐ON Macro: Macro block No.

☐OFF Macro: Macro block No.

(3) Switch

Device to Output: $u100-00

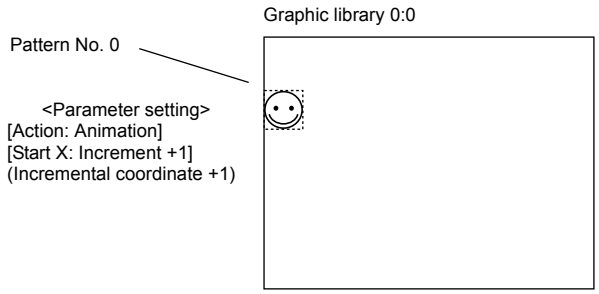
Output Action: Alternate

Lamp Device: $u100-00

Graphic Library Edit (4)

Example: GNo. 0 & No. 0

Place the following graphic on the screen, and specify the X parameter.



Macro Block Edit (5)

Example: Macro block No. 1

```
0   $u00101 = $u00101 + 1 (W)
1   IF ($u00101 = 640) LB00 (W)
2   RET
3   LB00:
4   $u00101 = 0 (W)
```

Macro to change the X parameter of the graphic start point

While the count on the X axis is increasing up to 640 (0 → 1 → ... → 640 → 0 → 1 → ... → 640), the graphic moves from the left to the right.

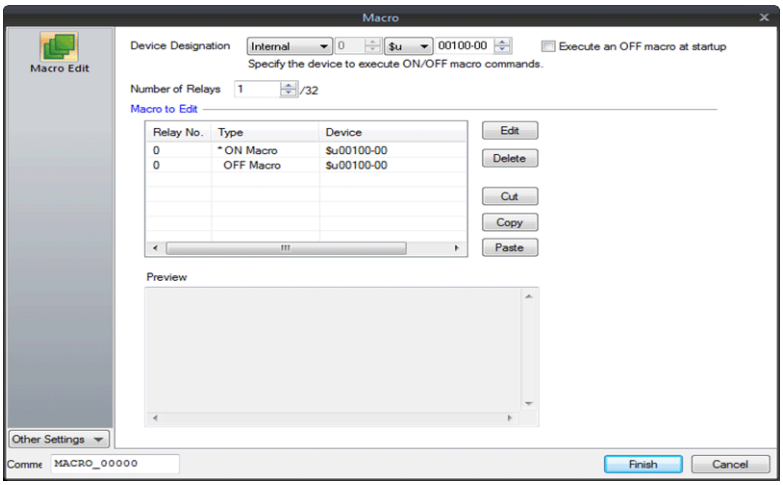
Transfer the above screen program to the V series for checking.

1.10 Macro Mode

Select [Screen Setting] → [Macro] to make settings.
The interval timer can be set for screens and multi-overlap displays.
Macro mode is used to execute an ON macro when the corresponding bit changes from 0 → 1 (leading edge) and an OFF macro when the corresponding bit changes from 1 → 0 (falling edge).
However, when the screen (multi-overlap) is opened, they are executed upon level recognition.
(Refer to [Execute an OFF macro at startup].)
Set the command in the [Macro Edit] window of the macro mode.
A maximum of 32 ON/OFF macros each can be set using the consecutive bits.

Setting

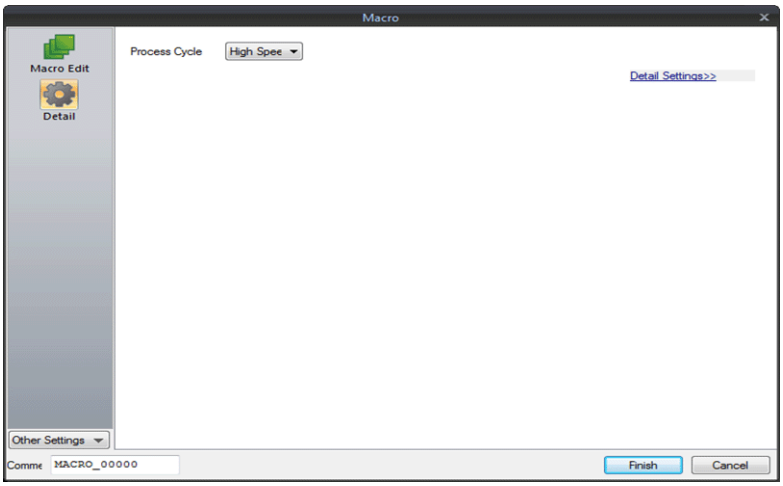
[Macro Edit] window



Device Designation	Specify the device memory address that triggers the macro.
Number of Relays	1 - 32 Specify the number of bits for triggering macros. The number specified here is common to both the ON macro and OFF macro. Example: “10” specified for [Number of Relays] - ON Macro: 10 maximum - OFF Macro: 10 maximum In this case, 10 bits must be allocated for [Device Designation].
Execute an OFF macro at startup	Set the operation to be performed when a screen or multi-overlap for which a macro mode is set is opened. • Checked While the bit specified for [Device Designation] is ON, the ON macro is executed; while it is OFF, the OFF macro is executed. • Unchecked The ON macro is executed while the bit specified for [Device Designation] is ON. While the bit is OFF, nothing is executed.
Macro to Edit	As many ON/OFF macros as the number for [Number of Relays] can be set.

Edit	The macro editor window corresponding to the selected relay number is opened.
Delete	The macro of the selected relay number is deleted.
Cut	The macro of the selected relay number is cut (copied and deleted).
Copy	The macro of the selected relay number is copied.
Paste	The copied macro is pasted to the selected relay number.
Preview	The macro of the selected relay number is shown.

[Detail] window



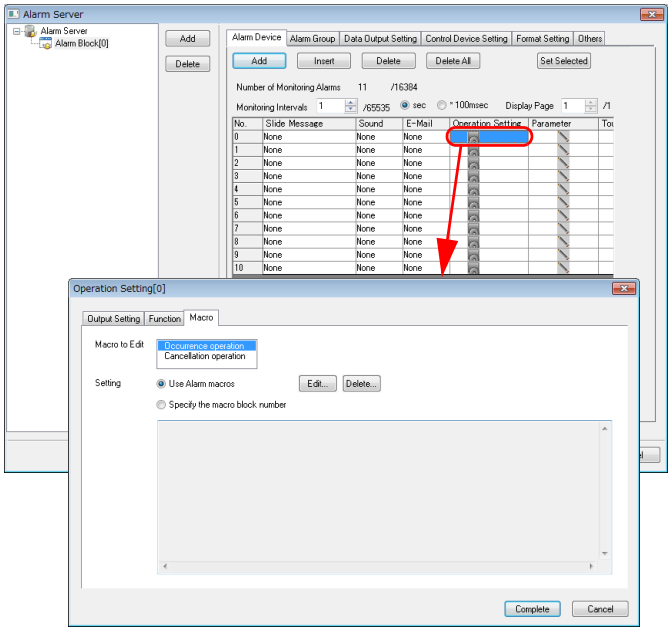
Process Cycle	Specify the cycle for the V series to read data in the PLC when they are communicating. For more information, refer to the V9 Series Reference Manual.
ID	Specify an ID.

1.11 Alarm Macro

An alarm macro can be set when the alarm server is used.
Select [Alarm Device] → [Operation Setting] to set macro commands.
When an alarm occurs, the occurrence macro is executed once. When it is reset, the resetting macro is executed once.
Select [Alarm Server] → [Alarm Device] → [Operation Setting] → [Macro], and set commands to be executed.

- Editing with [Setting: Use Alarm macros] selected
  Register macro commands directly in the [Operation Setting] window.

- Editing with [Setting: Specify the macro block number] selected
  Register macro commands in a macro block, and select a number of the macro block to execute.

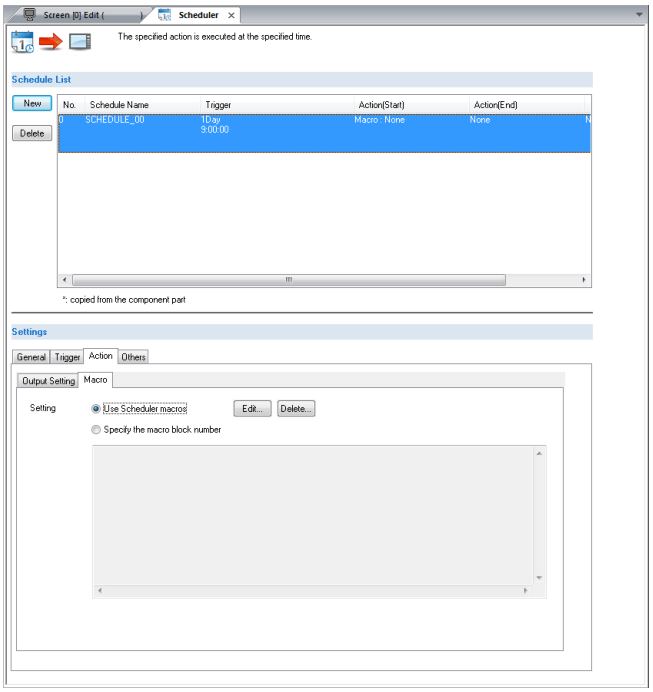


1.12 Scheduler Macro

A scheduler macro is executed by using the scheduler function.
Select [System Setting] → [Scheduler] to set macro commands.
The registered macro is executed once at the timing specified for [Trigger] in the [Scheduler] window.
Select [Action] → [Macro], and set commands to be executed.

- Editing with [Setting: Use Scheduler macros] selected
  Register macro commands directly in the [Action] tab window.

- Editing with [Setting: Specify the macro block number] selected
  Register macro commands in a macro block, and select a number of the macro block to execute.



1.13 Notes on Macros

- A maximum of 1,024 lines (instructions) can be set for one macro.
- The maximum of executable lines in macros is 160,000.
  If the maximum permissible number is exceeded by, for instance the repetition of the same macro with the use of a loop macro, macro execution is forcibly terminated.
  With the V9 series, if the maximum number of executions is exceeded “-1 (DEC)” is stored at $s1059.
- A maximum of 4096 words of data can be transferred per command. When creating macros, be careful not to exceed the maximum.
- When an external device memory is used with multiple MOV commands, the external memory is accessed each time so the processing speed is slowed down.
  Example:

  Line No. 0 PLC1 [D00200] = $u00200 (W)
  Line No. 1 PLC1 [D00201] = $u00201 (W)
  Line No. 2 PLC1 [D00202] = $u00202 (W)
  Line No. 3 PLC1 [D00203] = $u00203 (W)
  Line No. 4 PLC1 [D00204] = $u00204 (W)

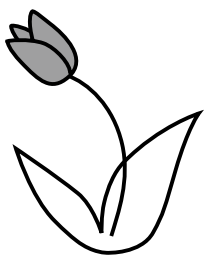
  In the above example, the V series goes and writes data to D200 as commanded in line No. 0, then goes and writes data to D201 as commanded in line No. 1, and so on. Communications that frequently occur will result in a prolonged processing time.
  To shorten the communications time, give a BMOV command as shown below. The contents of the macro using BMOV are the same as the above macro consisting of five lines, but the data writing takes place only once.

  Line No. 0 PLC1 [D00200] = $u00200 C:5 (BMOV) (W)

  The processing speed is increased and the number of macro commands is reduced.
  As described above, macros can be simplified when you plan to make their commands more efficient to use.

MEMO

Please use this page freely.



2 Edit

2.1 Macro Editor

This section describes the usage of the macro editor.

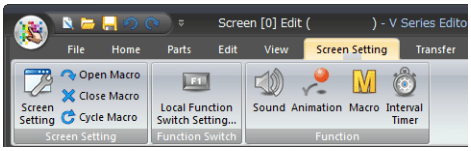
2.1.1 Start and Quit

Start

How to start the macro editor varies depending on the location where a macro command is registered.

Screen

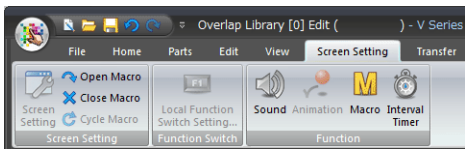
- OPEN macro
  [Screen Setting] → [Open Macro]
- CLOSE macro
  [Screen Setting] → [Close Macro]
- CYCLE macro
  [Screen Setting] → [Cycle Macro]



Overlap library

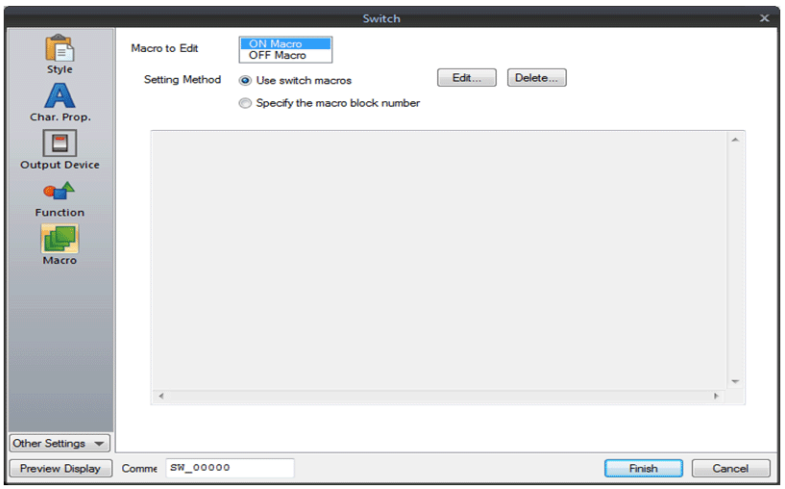
Select [Home] → [Registration Item] → [Overlap Library] to show the overlap display where a macro is to be registered.

- OPEN macro
  [Screen Setting] → [Open Macro]
- CLOSE macro
  [Screen Setting] → [Close Macro]



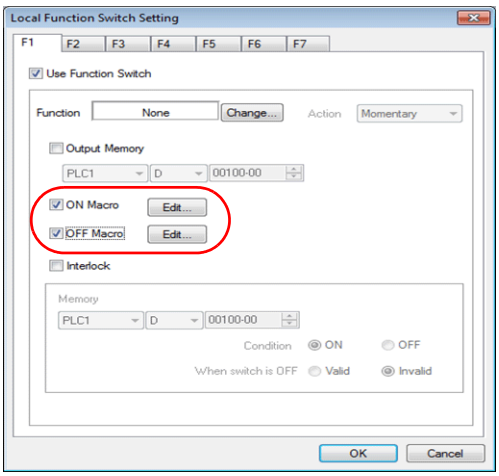
Switch

- ON macro
  [Macro] → [Macro to Edit: ON Macro] → [Edit]
- OFF macro
  [Macro] → [Macro to Edit: OFF Macro] → [Edit]



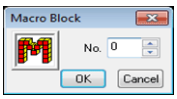
Function switch

- ON macro
  [Function Switch Setting] → [ON Macro] → [Edit]
- OFF macro
  [Function Switch Setting] → [OFF Macro] → [Edit]



Macro block

[Home] → [Registration Item] → [Macro Block]
Specify the number of the macro block where macro commands are to be registered, and click [OK].

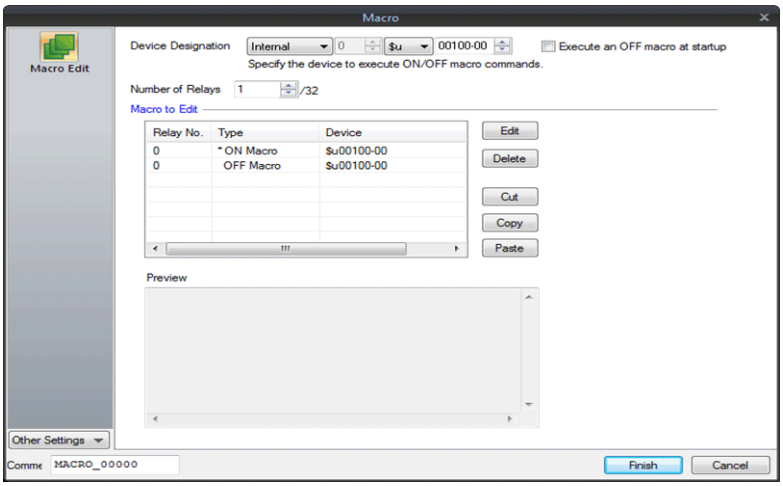


Macro mode

[Screen Setting] → [Macro]

- ON macro
  [Macro Edit] → [ON Macro] → [Edit]

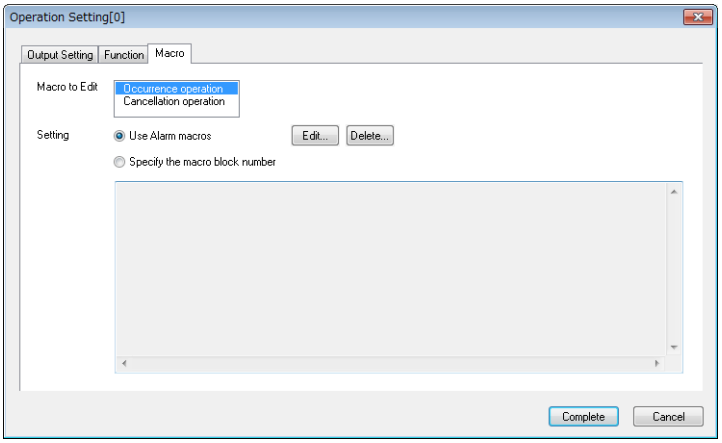
- OFF macro
  [Macro Edit] → [OFF Macro] → [Edit]



Alarm macro

[Alarm Server] → [Alarm Device] → [Operation Setting] → [Macro]

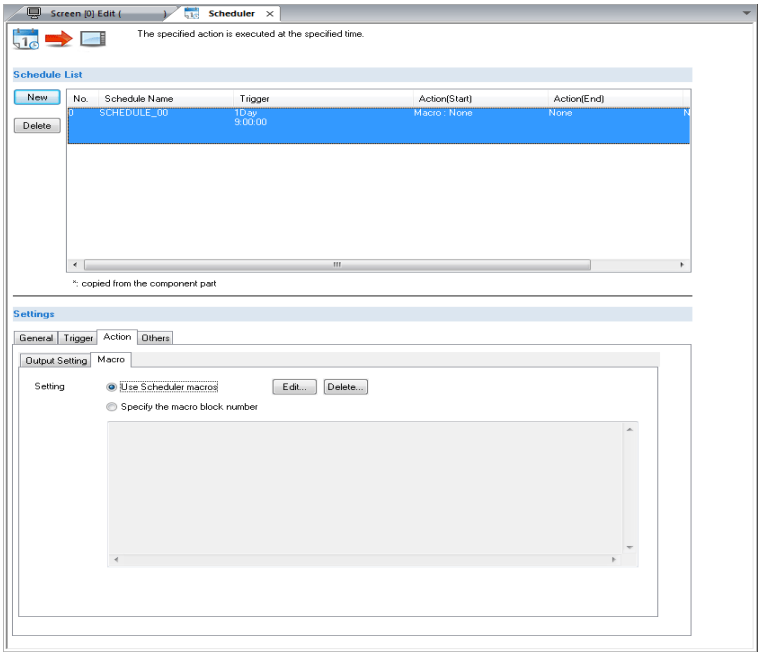
- Occurrence macro
  [Macro to Edit: Occurrence operation] → [Setting: Use Alarm macros] → [Edit]
- Resetting macro:
  [Macro to Edit: Cancellation operation] → [Setting: Specify the macro block number] → [Edit]



Scheduler macro

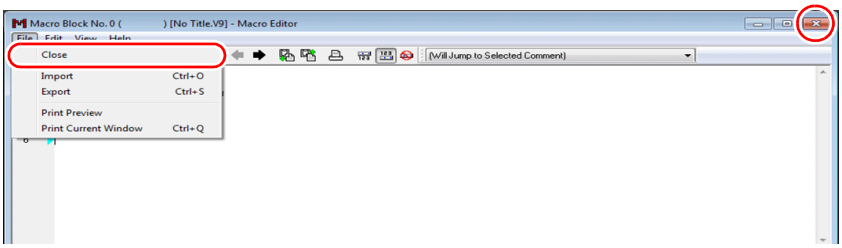
[Scheduler] → [Action] → [Macro]

[Setting: Use Scheduler macros] → [Edit]

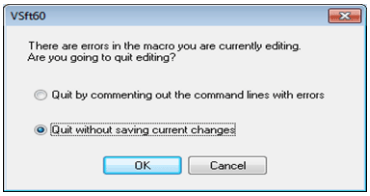


Quit

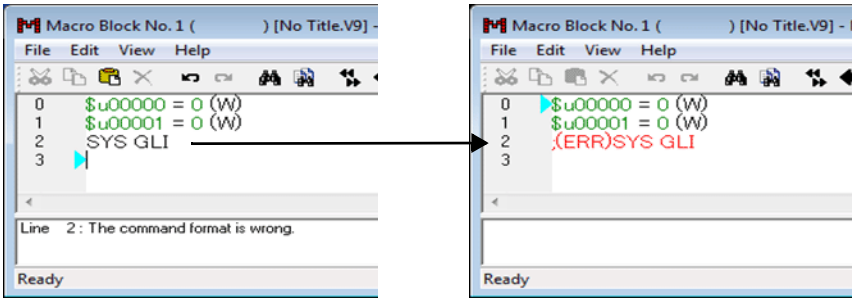
1. Select [File] → [Close], or click the close button in the upper right corner of the window.



2. When no error is detected, the macro editor ends normally. If detected, the following message appears.
   Select a countermeasure for the error, and quit the macro editor.



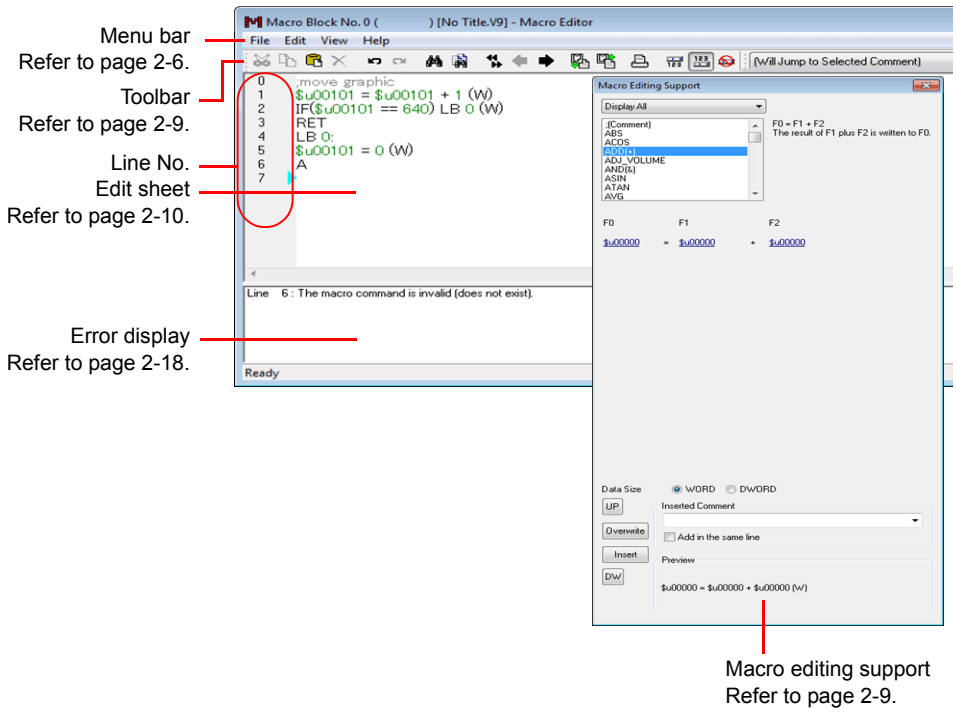
Example: When quitting by commenting out the error line



When the macro editor is opened again, the error line is changed to a comment in red with ";(ERR)" appended to the beginning of the line.

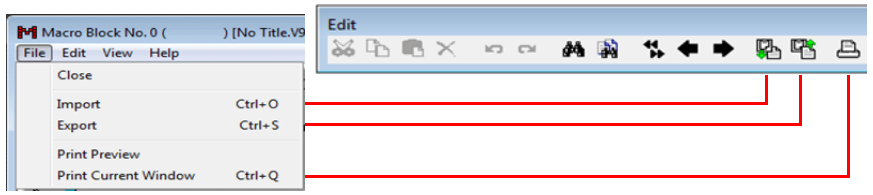
2.1.2 Screen Composition

The macro editor window is configured as follows:



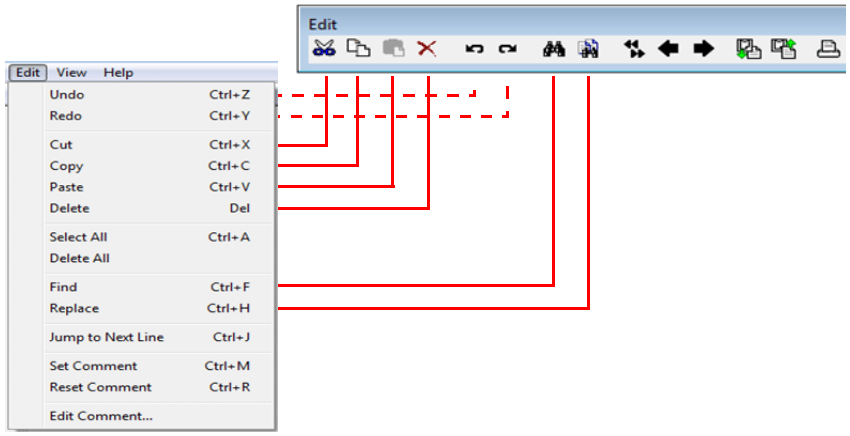
Menus

[File] menu



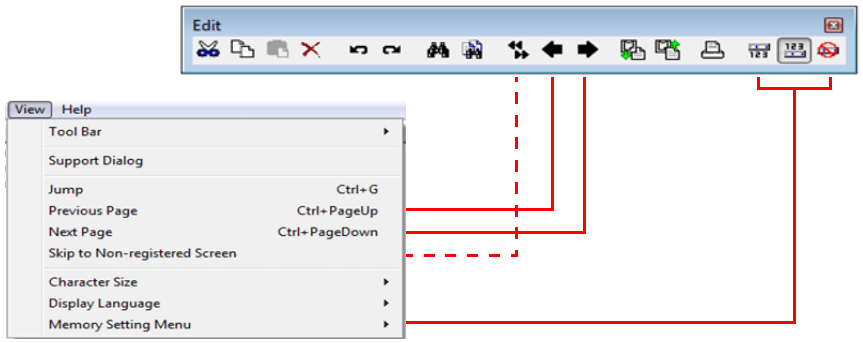
Close	Quits the macro editor.
Import	Reads text files.
Export	Saves the macro currently being edited to a text file.
Print Preview	Displays the printout image of the macro being edited.
Print Current Window	Prints the macro currently being edited.

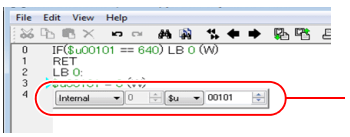
[Edit]/right-click menu



Undo	Returns you to the previous state by canceling the effect of the most recently executed command.
Redo	Returns you to the state before [Undo] is executed.
Cut	Cuts the selected area and saves it to the clipboard.
Copy	Copies the selected area and saves it to the clipboard.
Paste	Pastes the data from the clipboard.
Delete	Deletes the selected area.
Select All	Selects all macros currently being edited.
Delete All	Deletes all macros currently being edited.
Find	Searches for characters in the macro currently being edited.
Replace	Searches for characters in the macro currently being edited and replaces them.
Jump to Next Line	Jumps to the specified line.
Set Comment	Converts the line selected in the macro editor window to a comment (with ";" as the first character).
Reset Comment	Resets the comment conversion selected in the macro editor window (deletes the first character ";" from the comment).
Edit Comment	Allows you to edit comments on macro blocks during macro block editing.

[View] menu



Tool Bar	Selects whether to show/hide the toolbar.
Support Dialog	Selects whether to show/hide the [Macro Editing Support] dialog. For more information on the dialog, refer to page 2-9.
Jump	Opens the macro editor window for the number specified in [Macro Block].
Previous Page	Opens the previous page.
Next Page	Opens the next page.
Skip to Non-registered Screen	Skips the non-registered screens at the time of screen change.
Character Size	Allows you to select the size of characters to be displayed in the macro editor.
Display Language	Allows you to select the language to be displayed in the macro editor.
Memory Setting Menu (Upside Display, Downside Display, Hide)	Allows you to select the position where the memory setting pull-down menu appears in the macro editor. Example: [Downside Display] selected  Placing the cursor at a memory address brings up this underneath the address.

Toolbar

Edit

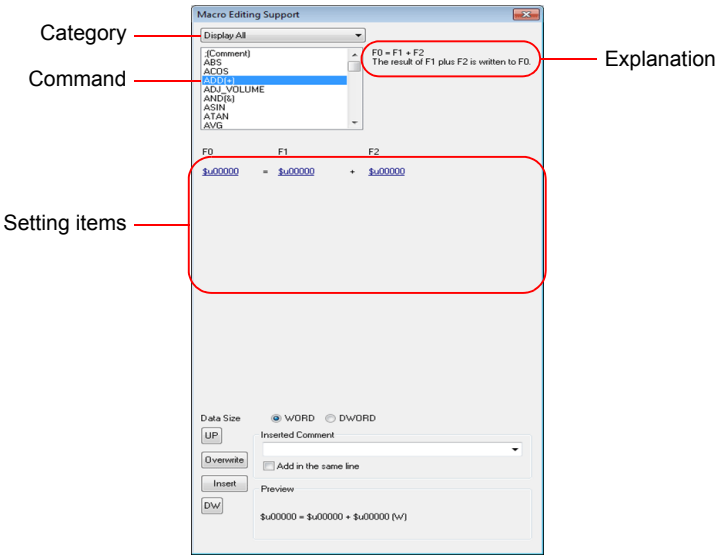
Refer to “Menus” (page 2-6).

Comment List

Comment List	Jumps to the selected comment line.

Macro Editing Support

To go to this dialog, select [Support Dialog] from the [View] menu.



Category	Macro category list
Command	The list of commands contained in the selected category
Setting items	Setting items required for the selected command
Inserted Comment	Comments can be registered together with commands.
UP/DW	Moves the selected line.
Overwrite	Overwrites the selected line with the contents of [Preview].
Insert	Inserts the contents of [Preview] into the position above the selected line.
Explanation	Explains the command selected from the list.
Preview	Displays the preview of macro editing.

2.1.3 Edit

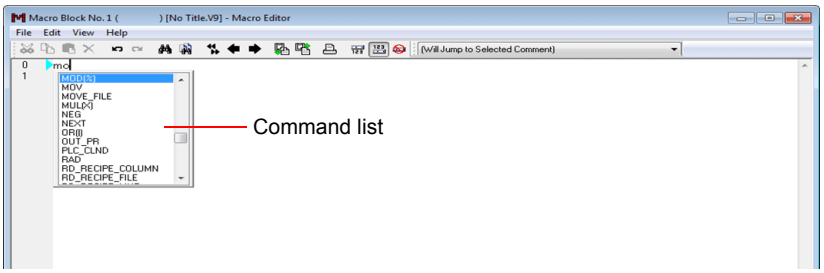
You can utilize the macro editor in several editing manners. Choose a desired one.

1: Command Entry
Editing is performed with the command list. This method is useful when you know the names of particular commands. (Refer to page 2-10.)
2: Direct Entry
Editing is performed by entering text through the keyboard of your computer. (Refer to page 2-12.)
3: Macro Editing Support
Editing is performed in the dialog that provides the explanation of individual commands. This method is best suited to beginners. (Refer to page 2-14.)
4: Text Entry
Editing is performed with a text editor (commercially available). Macro programming is enabled even in an environment without the editor. (Refer to page 2-16.)

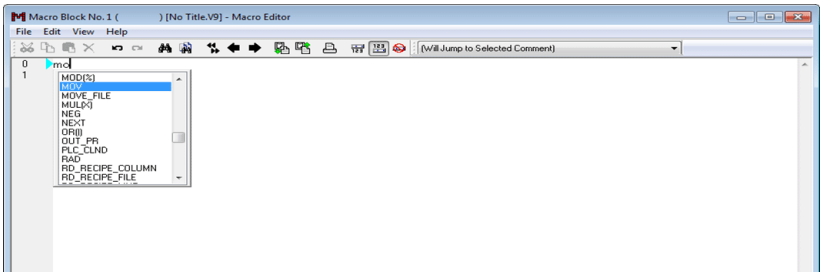
1: Command Entry

New registration

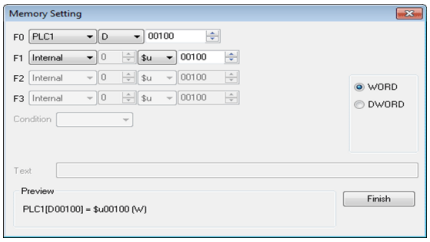
1. Select a line using the [UP] / [DW] button.
2. Enter a command. The command list appears.



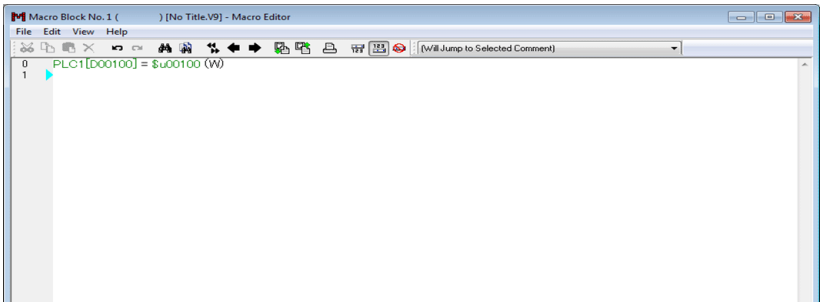
3. Choose the desired command from the list and double-click it.
Alternatively, choose the desired command using the [↑] / [↓] key on the keyboard and press the Enter key.



4. The [Device Setting] dialog appears. Make necessary settings, such as the address and data length, in the dialog, and click the [Finish] or [×] button.



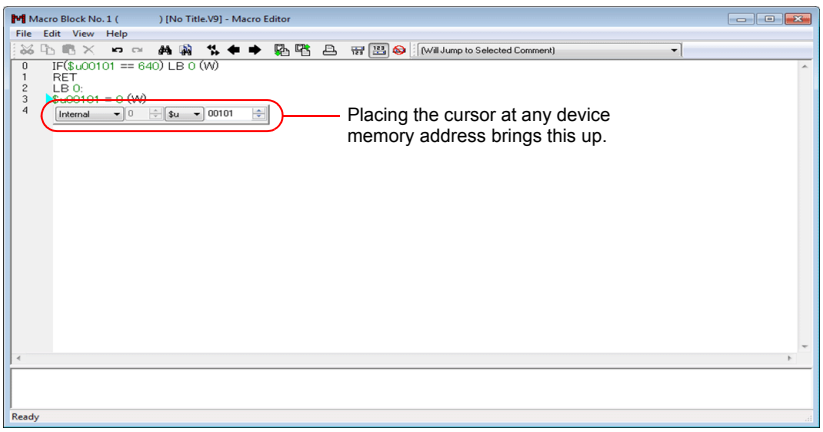
5. The line has been registered. To proceed with the next line registration, go back to step 1.



Device memory change

Device memory addresses (corresponding to [F0] / [F1] / [F2] / [F3]) are colored green. Follow the steps below when you wish to change any addresses:

- Select the desired memory address in green with the cursor. The device memory setting menu is displayed. Change the address as necessary.



- Select the desired device memory address in green with the cursor, and type an address change through your computer keyboard.

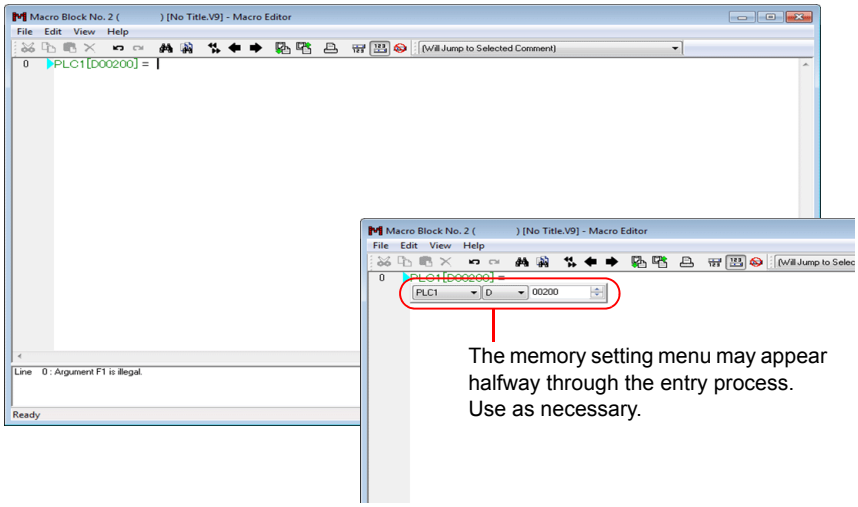
Command change

Choose the line you wish to change. Delete the line and register a new line.

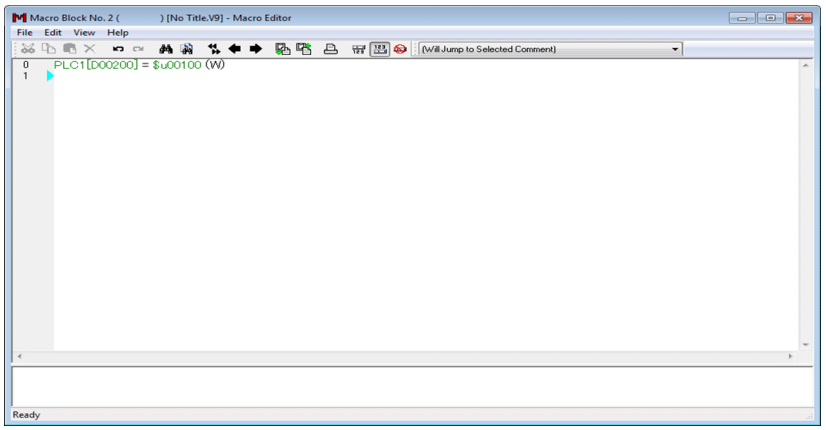
2: Direct Entry

New registration

1. Select a line using the [UP] / [DW] button.
2. Enter mnemonic codes through the keyboard.
   Example: MOV command
   PLC1 [D200] = $u100 (W)
   * For designating memory, refer to page 2-19.



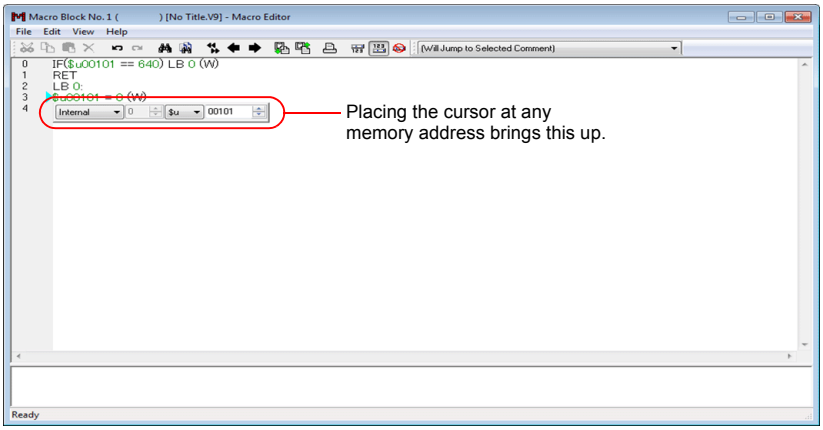
3. Press the Enter key to go to the next line. To proceed with the next line registration, go back to step 1.



Memory change

Memory addresses (corresponding to [F0] / [F1] / [F2] / [F3]) are colored green. Follow the steps below when you wish to change any memory addresses:

- Select the desired memory address in green with the cursor. The memory setting menu is displayed. Change the address as necessary.



- Select the desired memory address in green with the cursor, and type an address change through the keyboard.

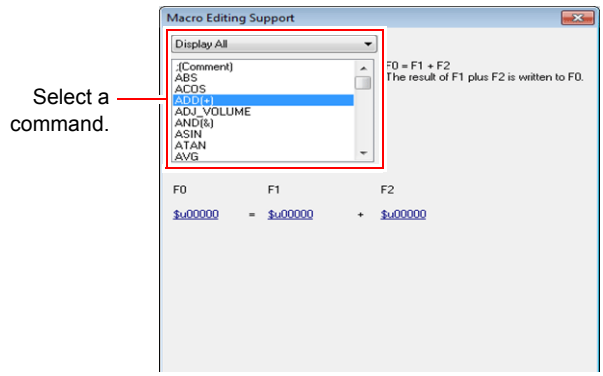
Command change

Choose the line you wish to change. Delete the line and register a new line.

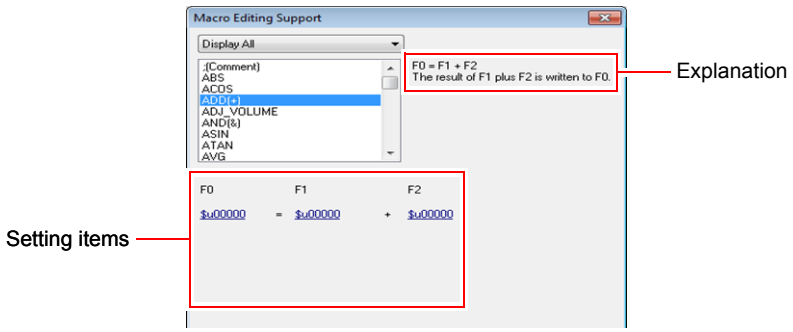
3: Macro Editing Support

New registration

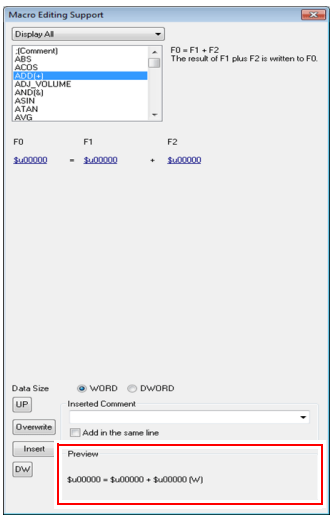
1. Select a line using the [UP] / [DW] button.
2. Select the desired command from the pull-down menu and the macro list.



3. The setting items required for the selected command are displayed. Specify the address, data length, etc.



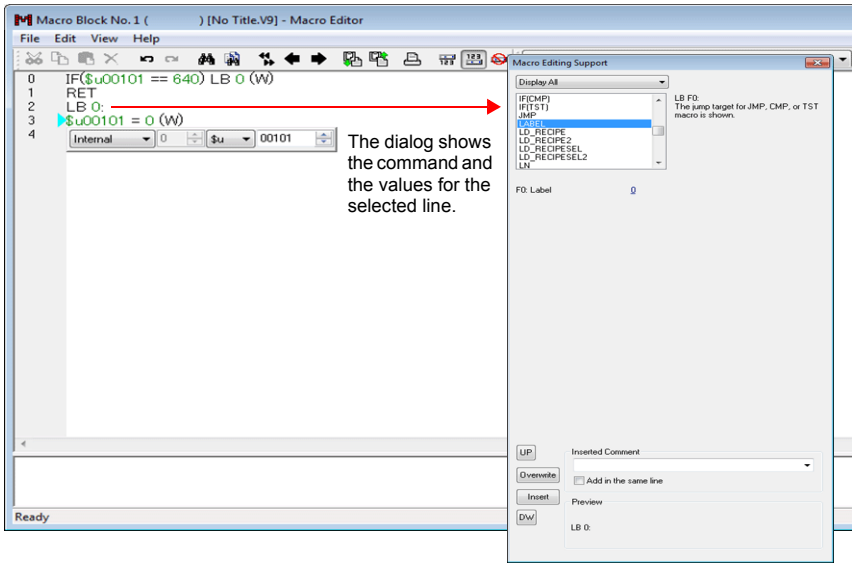
4. The settings made are displayed under [Preview].



5. If you wish to make a comment, enter it in the comment entry box.
6. To overwrite the selected line, press the [Overwrite] button. To insert a line into the position above the selected line, press the [Insert] button.
7. The line has been registered. To proceed with the next line registration, go back to step 1.

Device memory change

1. Select the line to be modified. The command and the values specified for the line are displayed in the [Macro Editing Support] dialog.



The dialog shows the command and the values for the selected line.

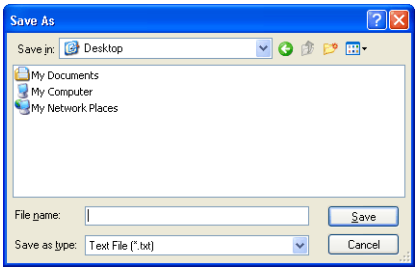
2. Change the device memory addresses as desired and click the [Overwrite] button. Clicking the [Insert] button inserts the changed setting into the position above the selected line.

4: Text Entry

The macro editor is capable of importing and exporting text files. Even if the editor is not installed on your computer, macros can be created with commercially available software.

Export

1. From the [File] menu, select [Export]. The [Save As] dialog is displayed.

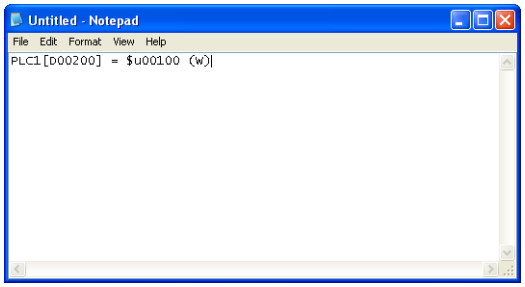


2. Enter a file name and click [Save]. A text file is created under the name.

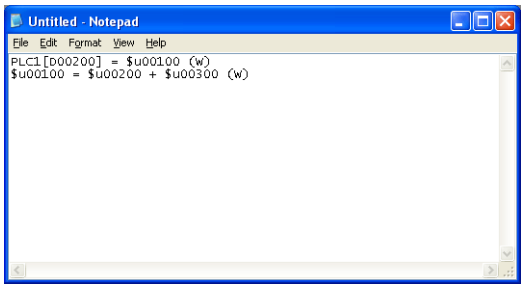
Text editing

Editing on Notepad

1. Open the text file on Notepad.



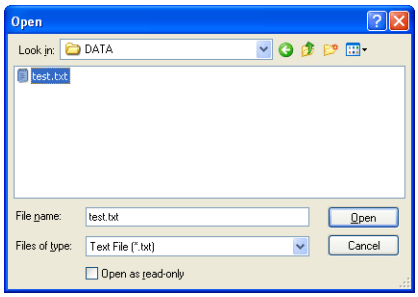
2. Select a line using the [UP] / [DW] button.
3. Enter mnemonic codes through the keyboard.
   Example: Addition command
   $u1000 = $u200 + $u300 (W)
   * For designating memory, refer to page 2-19.



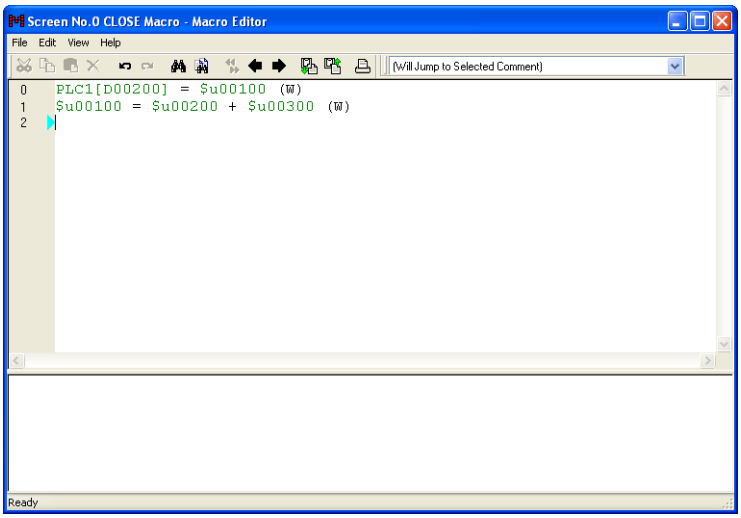
4. Save the file.

Import

1. Open the edit sheet, to which a text file will be imported.
2. From the [File] menu, select [Import]. The [Open] dialog is displayed.

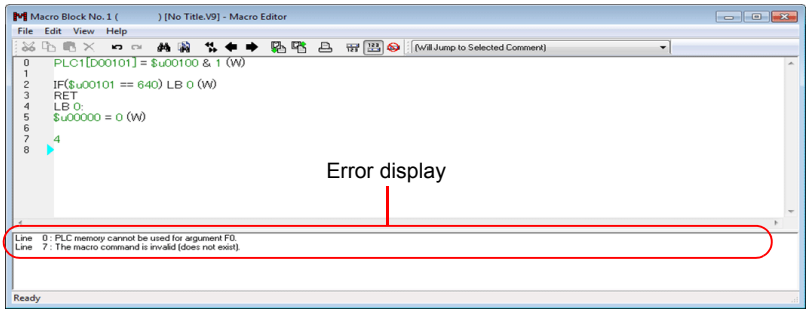


3. Select the desired file and click [Open]. The text file is imported.

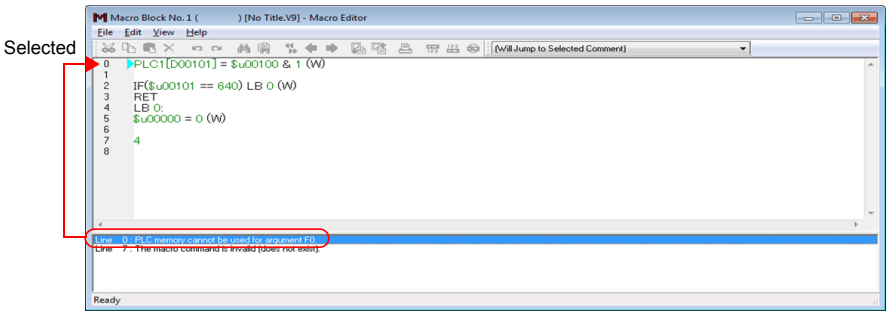


2.1.4 Error

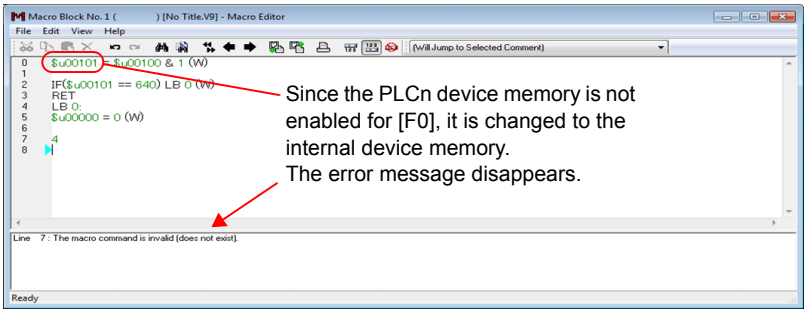
1. If the registered lines of a macro have any errors, error messages are displayed.



2. Double-clicking an error message selects the corresponding line.



3. Correct the error as needed in the message. Once finished, the message disappears.



2.2 Available Device Memory

2.2.1 Device Memory Types

The following device memory types can be used with macros:

Device Memory			Setting Range	Remarks
Internal device memory	$u		$u00000 - $u32767	
	$s		$s0000 - $s2047	
	$L		Depends on the setting *1	
	$LD		Depends on the setting *1	
	$T		$T0000 - $T1023	
	$P n : *2		$Pn:000 - $Pn:511	
	$M		$M0000 - $M2047	
	$MC		$MC0000 - $MC2047	
	$C		$C0000 - $C4095	
	Indirect device memory designation		For more information, refer to page 2-20.	$u/$T/$M only usable
Memory card	[File No.: Record No.] #address		[0:0] #0000 - [15:4094] #4095	
PLCn device memory	PLC n [xxxx] *2*3		(Example) PLC1 [D100]	1:1 communication
	PLCn [Port number: xxxx] *2*3		(Example) PLC1 [1:D100]	1:n communication
Constant	DEC	WORD	0U - 65535U	Add “U” to the extreme right position.
		DWORD	0U - 4294967295U	
	DEC-	WORD	-32768 - 32767	
		DWORD	−2147483648 - 2147483647	
	OCT	WORD	0o - 177777o	Add “o” to the extreme right position. (lower-case “o”)
		DWORD	0o - 37777777777o	
	HEX	WORD	0000H - FFFFH	Add “H” to the extreme right position.
		DWORD	00000000H - FFFFFFFFH	
	FLOAT	DWORD	−3.402823E+38 - −1.401298E-45 0 1.401298E-45 - 3.402823E+38	

*1 The available range varies depending on the settings set on the [SRAM/Clock Setting] dialog.

*2 For “n”, set the number of the connected device (1 to 8).

*3 The designation of [xxxx] varies depending on the type of the connected device. For more information, refer to the available device memory list in the V9 Series Connection Manual.

2.2.2 Indirect Device Memory Designation

Each device memory address can be indirectly designated.
The designation procedure varies depending on the device memory type and addresses.

Internal Device Memory, PLC (1 - 8) Device Memory

- Addresses 0 - 65535:

	15 MSB	8 7 LSB 0
n+0	Model	Device memory type
n+1	Device memory No. (address)	
n+2	Expansion code	Bit designation
n+3	00	Station number

- Addresses 65536 and above:

	15 MSB	8 7 LSB 0
n+0	Model	Device memory type
n+1	Device memory No. (address) lower-order	
n+2	Device memory No. (address) higher-order	
n+3	Expansion code	Bit designation
n+4	00	Station number

- Model, device memory type (hexadecimal)

Device memory			Model	Device memory type
Internal device memory	$u		00	00
	$s			01
	$L	0 - 65535	00	02
		65536 -	80	
	$LD	0 - 65535	00	03
		65536 -	80	
	$T		00	04
	$Pn*1		00	05
	$M		00	06
	$MC		00	07
	$C		00	08
PLC1 device memory	0 - 65535		01/11*2	The device memory type depends on the device memory used. Refer to the V9 Series Connection Manual or the PLC Connection Manual and set the type number of the device memory.
	65536 -		81/91*2	
PLC2 device memory	0 - 65535		03/12*2	
	65536 -		83/92*2	
PLC3 device memory	0 - 65535		13	
	65536 -		93	
PLC4 device memory	0 - 65535		14	
	65536 -		94	
PLC5 device memory	0 - 65535		15	
	65536 -		95	
PLC6 device memory	0 - 65535		16	
	65536 -		96	
PLC7 device memory	0 - 65535		17	
	65536 -		97	
PLC8 device memory	0 - 65535		18	
	65536 -		98	

*1 “n” treated as an expansion code
*2 The memory will work when specified with either model.

- Expansion code
  An expansion code should be designated, depending on the type of memory in use. For more information, refer to the description of indirect device memory designation relevant to the target device memory type in the V9 Series Connection Manual.

  Ex.: Mitsubishi Electric SPU device memory

  Unit No. 0: 00
  Unit No. 1: 01

- Station number
  1 : 1 or multi-link: Not used
  1 : n (multi-drop): Set the station number of the connected device.

Memory card

	15 MSB 8	7 LSB 0
n+0	02H	File No.
n+1	Word address in the record	
n+2	Record No.	

- File number, word address in the record, record number
  Refer to the memory card map in the V9 Series Reference Manual.

Example

- When accessing a word in the PLCn memory, “0” is specified for the “n + 2” word even in the case of device memory that does not use an expansion code.
  Ex.: Accessing D165 in a Mitsubishi PLC (PLC1)
  (Macro)
  $u100 = 0100H (W) Model: 01 (PLC1 memory) Memory type: 00
  $u101 = 0165 (W) Device memory No.: 165
  $u102 = 0000 (W) Expansion code: None
  $u200 = *$u100 (W)
  (Result of execution)
  Data at D165 is transferred to $u200.

- When accessing the bit-writable device memory, such as the Mitsubishi M Relay, the following setting is necessary.

  Device memory number = M (address)/16

  Ex.: Accessing M20
  (Macro)
  $u100 = 0106H (W) Model: 01 (PLC1 device memory) Device memory type: 06
  $u101 = 0001H (W) Device memory No. = 20 ÷ 16 = 1...4
  $u102 = 0004H (W) Expansion code: None Bit designation: 4
  *$u100 (ON)
  (Result of execution)
  The bit of M20 is set (ON).

2.3 CSV Format Setting (with Recipe or Sampling Macro Used)

Format settings are required for handling CSV files. Register data formats of CSV files in [Format Setting]. MONITOUCH will read/write the CSV files in accordance with these format settings.

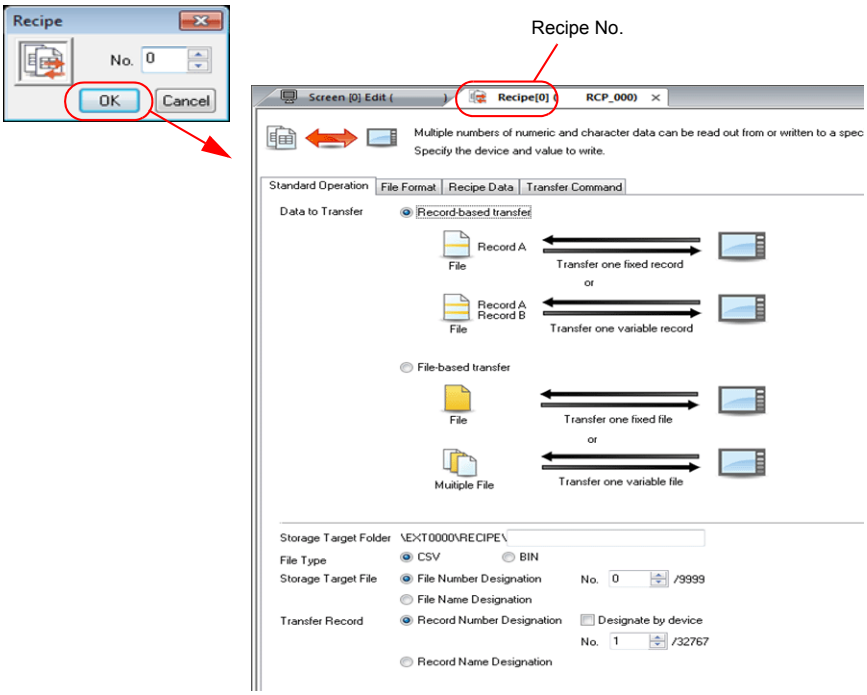
2.3.1 Applicable Macros

Function	Macro	CSV File Name	Setting Location	Refer to:
Recipe	LD_RECIPE	RECxxxx.CSV 0000 - 9999 (Designation of a number)	[Recipe] → [File Format]	page 4-97
	LD_RECIPE2			page 4-100
	LD_RECIPESEL			page 4-102
	LD_RECIPESEL2			page 4-105
	SV_RECIPE			page 4-109
	SV_RECIPE2			page 4-111
	SV_RECIPESEL			page 4-113
	SV_RECIPESEL2			page 4-116
	RD_RECIPE_FILE	xxxxxxxx.CSV 8 one-byte upper-case alphanumeric characters or less (Designation of a name)		page 4-120
	RD_RECIPE_LINE			page 4-122
	RD_RECIPE_COLUMN			page 4-124
	WR_RECIPE_FILE			page 4-126
	WR_RECIPE_LINE			page 4-128
	WR_RECIPE_COLUMN			page 4-130
Sampling	SMPL_CSV	xxxxxxxx.CSV Designation of a file name by the editor	• Alarm server [Alarm Block] → [Format Setting] • Logging Server [Logging Block] → [Format Setting]	page 4-136
	SMPL_CSV2	xxxxxxxx.CSV Designation of a file name by the device memory		page 4-138
	SMPLCSV_BAK	xxxxxxxx_YYYYMMDDHHMMSS.CSV Designation of a file name by the editor After _: Output time in year, month, day, hour, minute, and second		page 4-141
	SMPLCSV_BAK2	xxxxxxxx_YYYYMMDDHHMMSS.CSV Designation of a file name by the device memory After _: Output time in year, month, day, hour, minute, and second		page 4-144

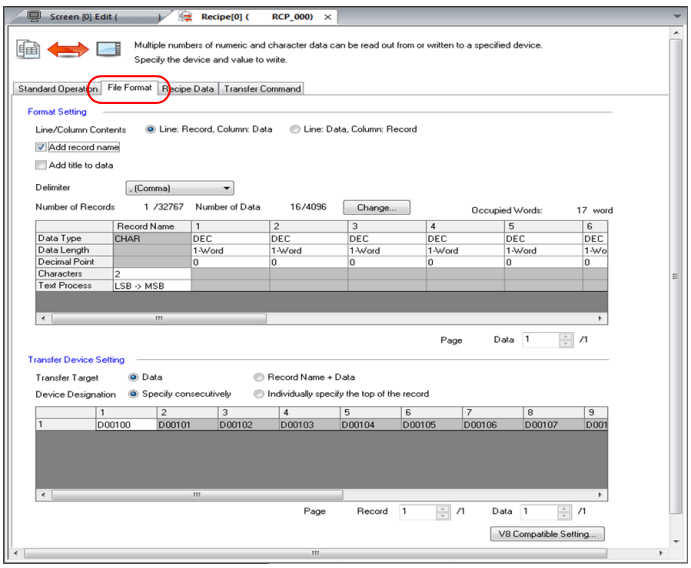
2.3.2 Recipe

Setting procedure

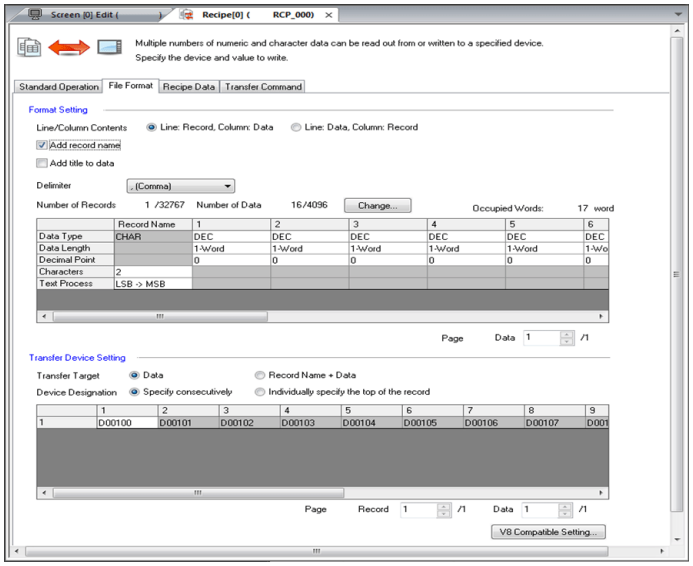
1. Select [System Setting] → [Recipe], and specify a recipe number.
   The [Recipe Edit] window is displayed.



2. In the [File Format] tab window, set and change settings of the selected recipe number.



Setting details



Format Setting	Line/Column Contents	Select an option according to the CSV file. • [Line: Record, Column: Data] CSV file Columns are in the same format. DEC CHAR DEC Record Line A 1 A 100 Line B 2 B 200 Line C 3 C 300 Line D 4 D 400 Data • [Line: Data, Column: Record] CSV file Record Line A Line B Line C Line D DEC → 1 2 3 4 Data CHAR → A B C D DEC → 100 200 300 400 Lines are in the same format.

<table>
<tr><td rowspan="5">Format Setting</td><td>Add record name *1</td><td>
Set how to treat the first column in the CSV file.
• Unchecked
The first column in the CSV file is treated as data.
CSV file
<table><tr><td>6000</td><td>15</td><td>200</td><td></td></tr><tr><td>6100</td><td>15</td><td>201</td><td></td></tr><tr><td>6200</td><td>20</td><td>202</td><td></td></tr><tr><td>6300</td><td>20</td><td>203</td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Display on MONITOUCH
<table><tr><td>..\</td><td>#1</td><td>#2</td><td>#3</td></tr><tr><td>#1</td><td>6000</td><td>15</td><td>200</td></tr><tr><td>#2</td><td>6100</td><td>15</td><td>201</td></tr><tr><td>#3</td><td>6200</td><td>20</td><td>202</td></tr><tr><td>#4</td><td>6300</td><td>20</td><td>203</td></tr></table>
• Checked
The first column in the CSV file is treated as a record name.
CSV file
<table><tr><td>ITEM1</td><td>6000</td><td>15</td><td>200</td></tr><tr><td>ITEM2</td><td>6100</td><td>15</td><td>201</td></tr><tr><td>ITEM3</td><td>6200</td><td>20</td><td>202</td></tr><tr><td>ITEM4</td><td>6300</td><td>20</td><td>203</td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Display on MONITOUCH
<table><tr><td>..\</td><td>#1</td><td>#2</td><td>#3</td></tr><tr><td>ITEM1</td><td>6000</td><td>15</td><td>200</td></tr><tr><td>ITEM2</td><td>6100</td><td>15</td><td>201</td></tr><tr><td>ITEM3</td><td>6200</td><td>20</td><td>202</td></tr><tr><td>ITEM4</td><td>6300</td><td>20</td><td>203</td></tr></table>
</td></tr>
<tr><td>Add title to data *1</td><td>
Set how to treat the first line in the CSV file.
• Unchecked
The first line in the CSV file is treated as data.
CSV file
<table><tr><td>6000</td><td>15</td><td>200</td><td></td></tr><tr><td>6100</td><td>15</td><td>201</td><td></td></tr><tr><td>6200</td><td>20</td><td>202</td><td></td></tr><tr><td>6300</td><td>20</td><td>203</td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>
Display on MONITOUCH
<table><tr><td>..\</td><td>#1</td><td>#2</td><td>#3</td></tr><tr><td>#1</td><td>6000</td><td>15</td><td>200</td></tr><tr><td>#2</td><td>6100</td><td>15</td><td>201</td></tr><tr><td>#3</td><td>6200</td><td>20</td><td>202</td></tr><tr><td>#4</td><td>6300</td><td>20</td><td>203</td></tr></table>
• Checked
The first line in the CSV file is treated as title.
CSV file
<table><tr><td>Title1</td><td>Title2</td><td>Title3</td><td></td></tr><tr><td>6000</td><td>15</td><td>200</td><td></td></tr><tr><td>6100</td><td>15</td><td>201</td><td></td></tr><tr><td>6200</td><td>20</td><td>202</td><td></td></tr><tr><td>6300</td><td>20</td><td>203</td><td></td></tr></table>
Display on MONITOUCH
<table><tr><td>..\</td><td>Title1</td><td>Title2</td><td>Title3</td></tr><tr><td>#1</td><td>6000</td><td>15</td><td>200</td></tr><tr><td>#2</td><td>6100</td><td>15</td><td>201</td></tr><tr><td>#3</td><td>6200</td><td>20</td><td>202</td></tr><tr><td>#4</td><td>6300</td><td>20</td><td>203</td></tr></table>
</td></tr>
<tr><td>Delimiter</td><td>Select a delimiter used in the CSV file.
, (comma) / <tab> (tab) / . (period)
* When “.” (period) is selected, a comma is used for a decimal point.</td></tr>
<tr><td>Number of Records</td><td>Set this option when transferring data on a file-by-file basis.
Specify the number of records to contain in one file.</td></tr>
<tr><td>Number of Data (1 - 4096)</td><td>Specify the number of data to contain in one record.
* The column of record names is not counted.</td></tr>
<tr><td>Occupied Words (1 - 65535)</td><td>The number of words used is automatically calculated.
• When transferring data on a record-by-record basis:
Total number of words in one record
• When transferring data on a file-by-file basis:
Total number of words in one file</td></tr>
</table>

Format Setting	Data Type	Specify the data format in the CSV file. • Record Name This option is enabled when [Add record name] is checked. Specify the number of characters and the order of text processing for a record name. • 1 - Specify the data format. - Data Type: DEC, DEC-, HEX, OCT, BIN, CHAR, BCD, FLOAT - Data Length: 1-Word, 2-Word - Decimal Point: 0 - 32 - Characters: 2 - 255 - Text Process: LSB → MSB, MSB → LSB
Transfer Device Setting	Transfer Target	This option is enabled when [Add record name] is checked. • [Data] Only data is transferred. • [Record Name + Data] Both record name and data are transferred.

*1 When both title and record name are used:

CSV file

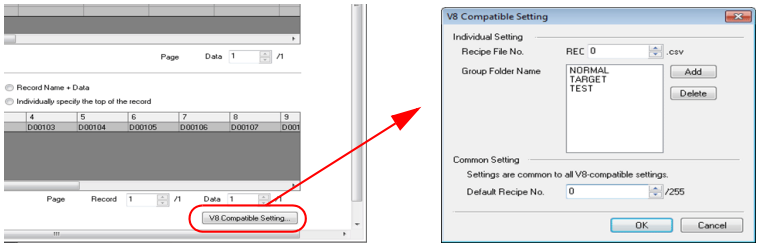
-	Title1	Title2	Title3
ITEM1	6000	15	200
ITEM2	6100	15	201
ITEM3	6200	20	202
ITEM4	6300	20	203

Display on MONITOUCH

..\	Title1	Title2	Title3
ITEM1	6000	15	200
ITEM2	6100	15	201
ITEM3	6200	20	202
ITEM4	6300	20	203

V8-compatible settings

When using a recipe macro, the following settings are necessary:



Individual Setting	Recipe File No.	Specify a number when using a LD_RECIPE, LD_RECIPESEL, SV_RECIPE, or SV_RECIPESEL macro. Set the CSV file number (REC0000.CSV to REC9999.CSV) that corresponds to the format of the recipe setting. Location of the CSV file Storage \ (access folder) \ RECIPE folder
	Group Folder Name (8 one-byte upper-case alphanumeric characters or less)	Set a group folder name when executing a recipe macro by randomly specifying CSV file names. • [Add] Creates a group folder in which CSV files are to be stored. The folder name can be changed as desired. • [Delete] Deletes a group folder. * All CSV files contained in the group folder use the same format settings.
Common Setting	Default Recipe No.	This is common to all recipe settings. Format settings of the default recipe number take effect in the following cases: • There is no recipe setting that corresponds to the file named “RECxxxx.csv”. • A group folder that does not exist in the recipe setting is added to the storage via Explore.

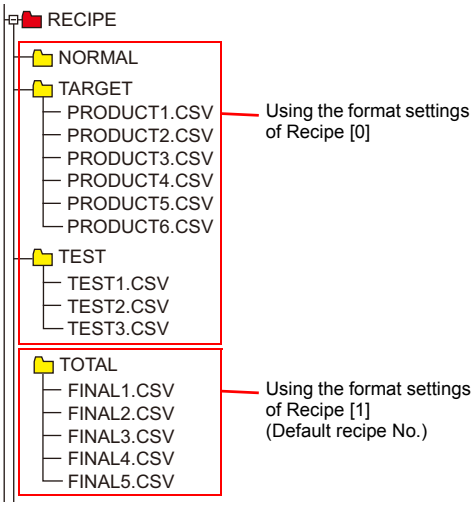
• Relationship between the recipe setting and the CSV file

- Recipe setting

Recipe No.	Group Folders
Recipe [0]	NORMAL TARGET TEST
Recipe [1]	

Default Recipe No. 1 = Recipe [1]

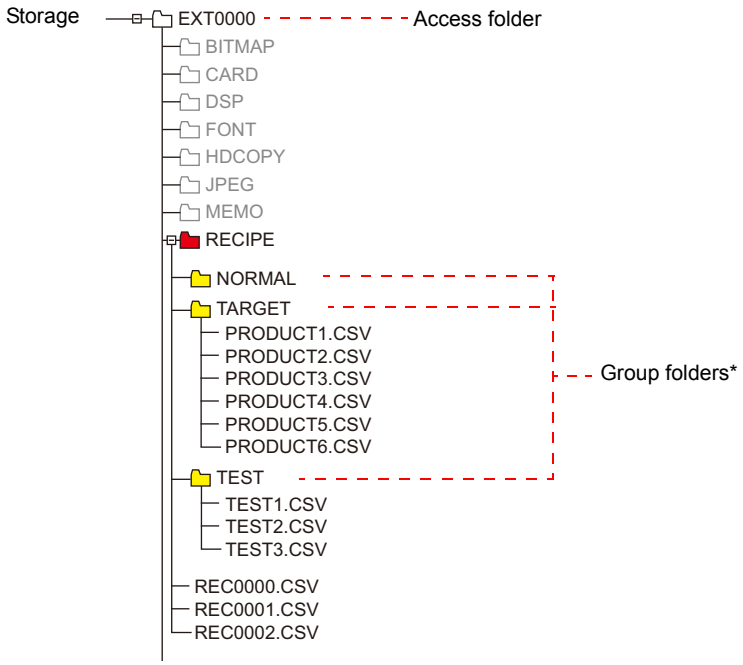
- Storage



CSV File Name and Storage Target

Depending on the name of a CSV file, its location and file designation vary. Create a file according to your purpose.

File name	Store target
RECxxxx.CSV 0000 - 9999	Access folder\RECIPE\ See the following:
xxxxxxxx.CSV 8 one-byte upper-case alphanumeric characters or less	Access folder\RECIPE\(group folder)\ 8 one-byte upper-case alphanumeric characters or less See the following:



* Group folders are defined in [Format Setting] → [V8 Compatible Setting]. They are automatically created when MONITOUCH recognizes the storage.

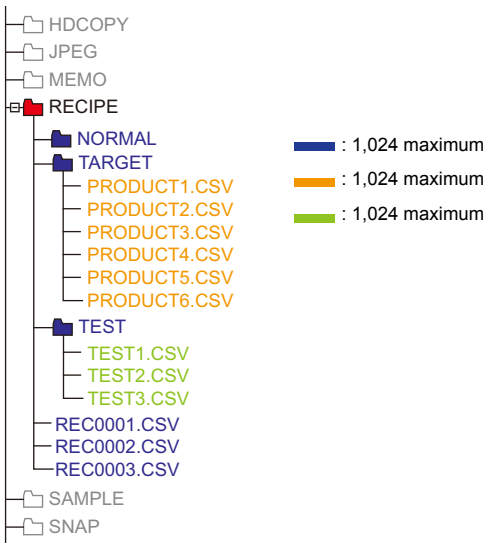
Total Number of CSV Files

There is a limitation on the number of group folders and CSV files that can be handled in the recipe mode.

- The total of group folders and CSV files in the RECIPE folder: 1,024 maximum
- The number of CSV files in a group folder: 1,024 maximum

Any more folders and files than 1,024 are not recognized in the recipe mode.

* When access to CSV files is made by a macro command, this limitation is not imposed. The time for accessing increases proportionately with the number of files.



Data in CSV File

- The number of words to be transferred
  A maximum of 65536 words can be read and written at one time in recipe mode or a macro. If you attempt to transfer data exceeding capacity, 65536 words are transferred, but extra words will not be transferred.

- Lines and columns
  The number of lines/columns to be handled varies, depending on the format setting.

	Line: Record, Column: Data	Line: Data, Column: Record
Number of lines	1 - 32767	1 - 4096 *3
Number of columns *1	1 - 4096 *2	1 - 4096

*1 Excel is capable of handling a maximum of 256 columns.
*2 The maximum number of words per column: 4,096 words
*3 The maximum number of words per line: 4,096 words
*4 File size: 1 MB or less

- Number of bytes for record
  64 bytes maximum per record
  * This setting can be made in [Format Setting].

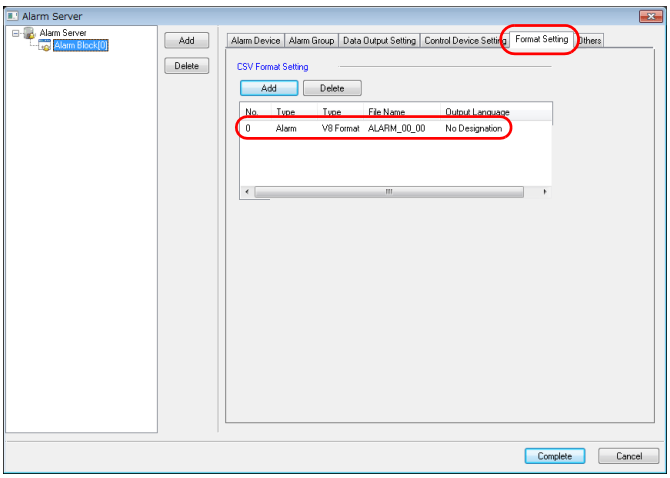
- Number of bytes for a title name
  64 bytes maximum per title

2.3.3 Sampling

Alarm Server

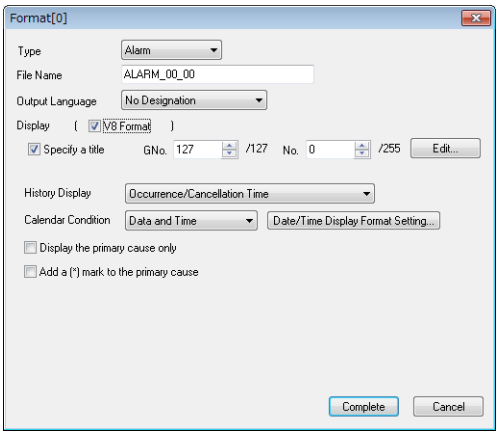
Setting procedure

1. Select [Alarm Server] → [Alarm Block].
   The [Alarm Block] window is displayed.
2. In the [Form Setting] tab window, double-click on the block number to be output in CSV format. The [Format] dialog appears.



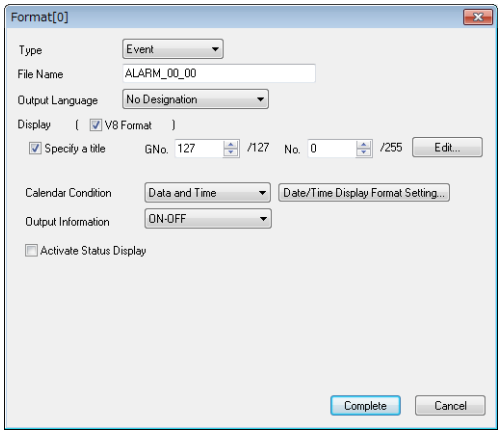
Setting details

- [Alarm]



Specify a title	Specify a title to be added to the header in the CSV file.
History Display	Set the display order of alarm history.
Calendar Condition	Select a format of date display to be output in the CSV file.
Display the primary cause only	Only error messages of primary causes are output in the CSV file.
Add a (*) mark to the primary cause	Primary cause messages are output with (*) marks appended in the CSV file.

- [Event]

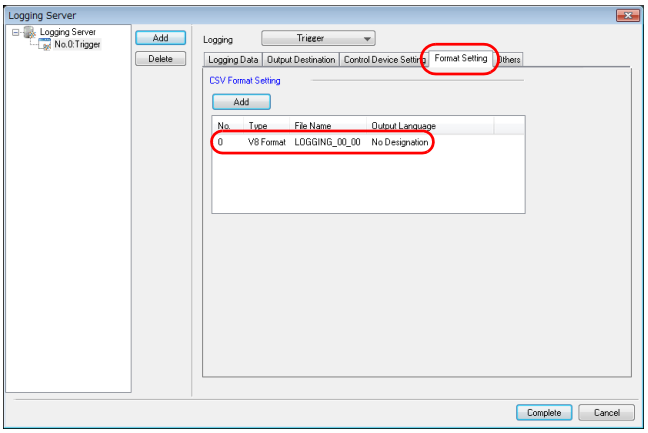


Specify a title	Specify a title to be added to the header in the CSV file.
Calendar Condition	Check this box to output the date in the CSV file.
Output Information	Select a status to output.
Activate Status Display	Select a format of status display to be output in the CSV file.

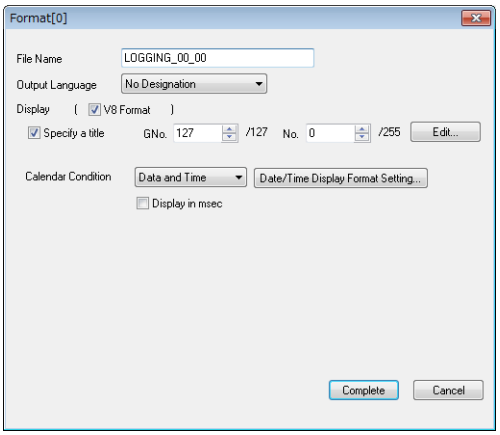
Logging Server

Setting procedure

1. Select [Logging Server] → [Logging Block].
   The [Logging Block] window is displayed.
2. In the [Form Setting] tab window, double-click on the block number to be output in CSV format. The [Format] dialog appears.



Setting details



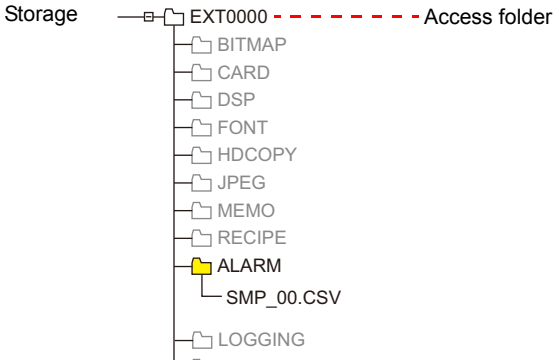
Specify a title	Specify a title to be added to the header in the CSV file.
Calendar Condition	Select a format of date display to be output in the CSV file.
Display in msec	Check this box to output the time data in the unit of msec in the CSV file.

CSV File Name and Storage Target

For “SMPL_CSV”

File Name	Storage Target
(User-specified name).CSV * A maximum of 64 one-byte characters can be used as a user-specified name.	• Alarm server (access folder)\ALARM\ • Logging Server (access folder)\LOGGING\

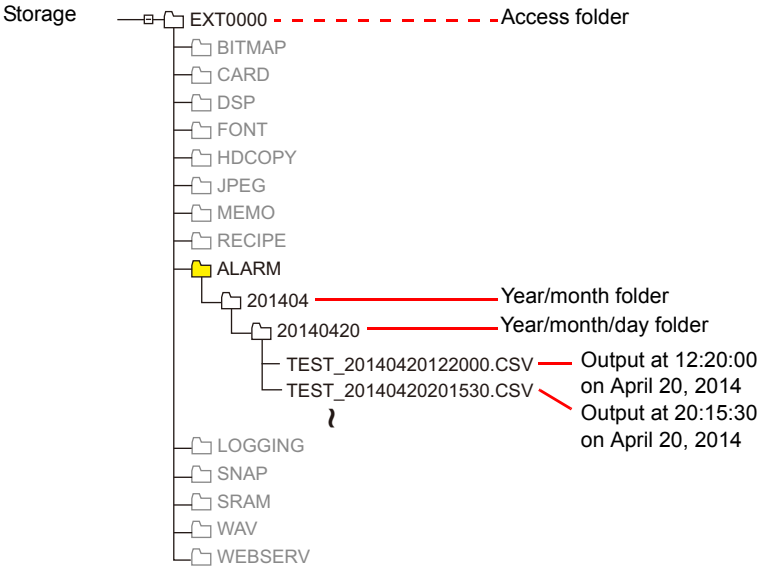
Example: Alarm server



For “SMPLCSV_BAK/SMPLCSV_BAK2”

File Name	Storage Target
(User-specified name)_YYYYMMDDHHMMSS.CSV Output time in year, month, day, hour, minute, and second * A maximum of 64 one-byte characters can be used as a user-specified name.	• Alarm server (access folder)\ALARM\(year/month folder)\(year/month/day folder) • Logging Server (access folder)\LOGGING\(year/month folder)\(year/month/day folder)

Example: Alarm server



3 Command

3.1 Macro Command List

Category	Command Name	Mnemonic	Contents	Refer to:
Arithmetical Operation	ADD(+)	F0 = F1 + F2 (W) F0 = F1 + F2 (D)	Addition	page 4-2
	SUB(-)	F0 = F1 – F2 (W) F0 = F1 – F2 (D)	Subtraction	page 4-4
	MUL(X)	F0 = F1 F2 (W) F0 = F1 F2 (D)	Multiplication	page 4-6
	DIV(/)	F0 = F1 / F2 (W) F0 = F1 / F2 (D)	Division	page 4-8
	MOD(%)	F0 = F1 % F2 (W) F0 = F1 % F2 (D)	Remainder of division	page 4-9
Logical Operation	AND(&)	F0 = F1 & F2 (W) F0 = F1 & F2 (D)	Logical product	page 4-10
	OR(|)	F0 = F1 | F2 (W) F0 = F1 | F2 (D)	Logical add	page 4-11
	XOR(^)	F0 = F1 ^ F2 (W) F0 = F1 ^ F2 (D)	Exclusive OR	page 4-12
	SHL(<<)	F0 = F1 << F2 (W) F0 = F1 << F2 (D)	Left shift	page 4-13
	SHR(>>)	F0 = F1 >> F2 (W) F0 = F1 >> F2 (D)	Right shift	page 4-14
Statistic	MAX	F0 = MAX (F1 C:F2) (W) F0 = MAX (F1 C:F2) (D)	Maximum	page 4-15
	MIN	F0 = MIN (F1 C:F2) (W) F0 = MIN (F1 C:F2) (D)	Minimum	page 4-16
	AVG	F0 = AVG (F1 C:F2) (W) F0 = AVG (F1 C:F2) (D)	Average	page 4-17
	SUM	F0 = SUM (F1 C:F2) (W) F0 = SUM (F1 C:F2) (D)	Sum	page 4-18
Mathematics/ trigonometric	EXP	F0 = EXP (F1) (F)	Exponent	page 4-19
	EXPT	F0 = EXPT (F1,F2) (F)	Powers	page 4-20
	LN	F0 = LN (F1) (F)	Natural logarithms	page 4-21
	LOG	F0 = LOG (F1) (F)	Common logarithms	page 4-22
	SQRT	F0 = SQRT (F1) (F)	Square roots	page 4-23
	ABS	F0 = ABS (F1) (W) F0 = ABS (F1) (D) F0 = ABS (F1) (F)	Absolute value	page 4-24
	NEG	F0 = NEG (F1) (W) F0 = NEG (F1) (D) F0 = NEG (F1) (F)	Sign inversion	page 4-25
	SIN	F0 = SIN (F1) (F)	Sine	page 4-26
	COS	F0 = COS (F1) (F)	Cosine	page 4-27
	TAN	F0 = TAN (F1) (F)	Tangent	page 4-28
	ASIN	F0 = ASIN (F1) (F)	Arcsine	page 4-29
	ACOS	F0 = ACOS (F1) (F)	Arccosine	page 4-30
	ATAN	F0 = ATAN (F1) (F)	Arctangent	page 4-31
	DEG	F0 = DEG (F1) (F)	Convert radians → degrees	page 4-32
	RAD	F0 = RAD (F1) (F)	Convert degrees → radians	page 4-33

Category	Command Name	Mnemonic	Contents	Refer to:
Bit Operation	BSET	F0 (ON)	Bit set (ON)	page 4-34
	BCLR	F0 (OFF)	Bit reset (OFF)	page 4-35
	BINV	F0 (INV)	Bit inversion	page 4-36
Conversion	BCD	F0 = F1 BCD (W) F0 = F1 BCD (D)	Conversion to BCD	page 4-37
	BIN	F0 = F1 BIN (W) F0 = F1 BIN (D)	Conversion to BIN	page 4-38
	CWD	F0 = F1 D <- W	Convert one-word → double-word	page 4-39
	CVP	F0 = F1 PLC <- (W) F0 = F1 PLC <- (D)	Convert DEC → PLC1	page 4-40
	CVPFMT	F0 = F1 (W) PLC F2 <- F0 = F1 (D) PLC F2 <-	Convert DEC → PLCn	page 4-41
	CVB	F0 = F1 (W) <- PLC F0 = F1 (D) <- PLC	Convert PLC1 → DEC	page 4-42
	CVBFMT	F0 = F1 (W) <- PLC F2 F0 = F1 (D) <- PLC F2	Convert PLCn → DEC	page 4-43
	SWAP	F0 C:F1	Swap MSB with LSB	page 4-44
	CHR	F0 = ‘ ‘	Convert text → code (PLC1 code fixed)	page 4-45
	STRING	F0 = ‘ ‘ (STRING)	Convert text → code	page 4-46
	CVFD	F0 (D) <- F1 (F) F2 (D)	Convert real number → BIN	page 4-47
	CVDF	F0 (F) <- F1 (D) F2 (D)	Convert BIN → real number	Page 4-49
	CLND_TO_GRE	CLND_TO_GRE F0 F1 F2	Convert calendar data → GMT-based UNIX time	page 4-51
	GRE_TO_CLND	GRE_TO_CLND F0 F1 F2	Convert GMT-based UNIX time → calendar data	page 4-53
	FORMAT_DATA	FORMAT_DATA F0 F1 F2	Convert string → numerical data	page 4-55
	FORMAT_STR	FORMAT_STR F0 F1 F2	Convert numerical data → string	page 4-59
Transfer	MOV	F0 = F1 (W) F0 = F1 (D)	Transfer	page 4-63
	BMOV	F0 = F1 C:F2 (BMOV) (W) F0 = F1 C:F2 (BMOV) (D)	Block transfer	page 4-64
	CVMOV	F0 = F1 C:F2 (CVMOV) (W) F0 = F1 C:F2 (CVMOV) (D)	(With data conversion) Block transfer	page 4-66
	CVSMOV	F0 = F1 C:F2 (CVSMOV) (W) F0 = F1 C:F2 (CVSMOV) (D)	(With text conversion) Block transfer	page 4-69
	FILL	F0 = F1 C:F2 (FILL)	Transfer all	page 4-71

Category	Command Name	Mnemonic	Contents	Refer to:
Comparison	CMP	IF (F0 = F1)F2 (W) IF (F0 = F1)F2 (D)	Comparison	page 4-72
	TST	IFZ (F0 & F1) F2 (W) IFZ (F0 & F1) F2 (D)	Logical product comparison	page 4-74

	IF
	ELSE
	ENDIF
		IF (F0 (condition) F1) (W)
		IF (F0 (condition) F1) (D)
		IF ((condition) F0) (B)
		ELSE
		ENDIF
			Conditional branch	page 4-75

Macro Operation Control	CALL	CALL F0	Macro block call	page 4-77
	JMP	JMP F0	Jump	page 4-79
	LABEL	LB F0:	Label	page 4-80
	FOR/NEXT	FOR F0 / NEXT	Loop between FOR and NEXT	page 4-81
	RET	RET	Finish macro processing	page 4-83
	SWRET	SWRET	Execute switch function	page 4-84
	EN_INT	EN_INT	Interruption enabled	page 4-85
FROM Backup	FROM_WR	FROM_WR F0 F1	Write to FROM	page 4-86
	FROM_RD	FROM_RD F0 F1	Read from FROM	page 4-87
PLC	PLC_CLND	PLC_CLND F0 PLC F1 F2 F3	Calendar control for PLCn	page 4-88
	PLC_CTL	PLC_CTRL PLC F0 F1 F2	PLCn control	page 4-90
	TBL_READ	TBL_READ F0 <- TABLE : PLC F1 : F2	Read from device memory map	page 4-92
	TBL_WRITE	TBL_WRITE TABLE : PLC F1 : F0 <- F2	Write to device memory map	page 4-93
Ethernet	SEND	SEND F0 C:F1 TO F2	Transfer on the network	page 4-94
	EREAD	EREAD F0 = F1 C:F2 F3	Read on the network	page 4-95
	EWRITE	EWRITE F0 F1 = F2 C:F3	Write on the network	page 4-96

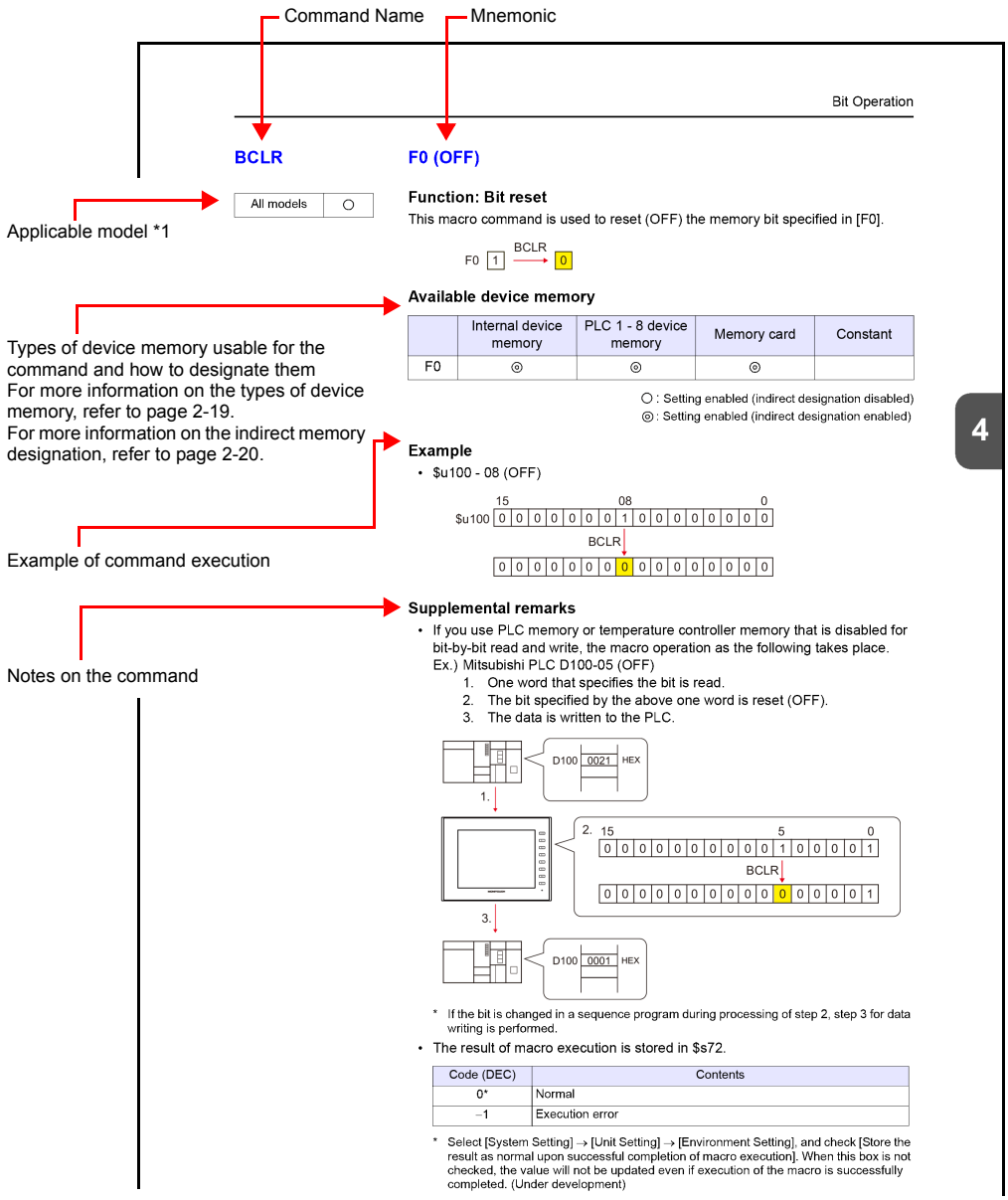
Category	Command Name	Mnemonic	Contents	Refer to:
Storage (Recipe)	LD_RECIPE	LD_RECIPE F0 F1	Read CSV file	page 4-97
	LD_RECIPE2	LD_RECIPE2 F0 F1 F2		page 4-100
	LD_RECIPESEL	LD_RECIPESEL F0 F1		page 4-102
	LD_RECIPESEL2	LD_RECIPESEL2 F0 F1 F2		page 4-105
	SV_RECIPE	SV_RECIPE F0 F1 F2	Save to CSV file	page 4-109
	SV_RECIPE2	SV_RECIPE2 F0 F1 F2 F3		page 4-111
	SV_RECIPESEL	SV_RECIPESEL F0 F1		page 4-113
	SV_RECIPESEL2	SV_RECIPESEL2 F0 F1 F2		page 4-116
	SET_ RECIPEFOLDER	SET_RECIPEFOLDER F0	Folder designation	page 4-118
	RD_RECIPE_FILE	RD_RECIPE_FILE F0 F1	Read CSV file	page 4-120
	RD_RECIPE_LINE	RD_RECIPE_LINE F0 F1 F2 F3		page 4-122
	RD_RECIPE_ COLUMN	RD_RECIPE_COLUMN F0 F1 F2 F3		page 4-124
	WR_RECIPE_FILE	WR_RECIPE_FILE F0 F1	Save to CSV file	page 4-126
	WR_RECIPE_LINE	WR_RECIPE_LINE F0 F1 F2 F3		page 4-128
	WR_RECIPE_ COLUMN	WR_RECIPE_COLUMN F0 F1 F2 F3		page 4-130
	GET_RECIPE_ FILEINFO	GET_RECIPE_FILEINFO F0 F1 F2	CSV file information	page 4-132
Storage (Sampling)	SMPL_BAK	SMPL_BAK F0	Save backup	page 4-134
	SMPL_CSV	SMPL_CSV F0	Create CSV file	page 4-136
	SMPL_CSV2	SMPL_CSV2 F0 F1	Create CSV file (file name designation)	page 4-138
	SMPL_SAVE	SMPL_SAVE	Save logging/alarm data stored in SRAM	page 4-140
	SMPLCSV_BAK	SMPLCSV_BAK F0	Save backup (CSV file)	page 4-141
	SMPLCSV_BAK2	SMPLCSV_BAK2 F0 F1	Save backup (CSV file, file name designation)	page 4-144
Storage (Others)	HDCOPY	HDCOPY	Hardcopy	page 4-146
	HDCOPY2	HDCOPY2 F0	Hardcopy	page 4-147
	HDCOPY3	HDCOPY3 F0	Hardcopy (file name designation)	page 4-148
	SET_DRIVE	SET_DRIVE F0	Select drive	page 4-149
	COPY_FILE	COPY_FILE F0 F1	Copy file	page 4-150
	MOVE_FILE	MOVE_FILE F0 F1 F2	Move file	page 4-152
	READ_FILE	READ_FILE F0 F1 F2 F3	Read universal file	page 4-154
	WRITE_FILE	WRITE_FILE F0 F1 F2	Write to universal file	page 4-156
Real No. Arithmetical Operation	F_ADD(+)	F0 = F1 + F2 (F)	Real number addition	page 4-158
	F_SUB(-)	F0 = F1 − F2 (F)	Real number subtraction	page 4-159
	F_MUL(X)	F0 = F1 × F2 (F)	Real number multiplication	page 4-160
	F_DIV(/)	F0 = F1 / F2 (F)	Real number division	page 4-161

Category	Command Name	Mnemonic	Contents	Refer to:
Real No. Statistics	F_SUM	F0 = F_SUM (F1 C:F2) (F)	Sum of real number data	page 4-162
	F_AVG	F0 = F_AVG (F1 C:F2) (F)	Average of real number data	page 4-163
	F_MAX	F0 = F_MAX (F1 C:F2) (F)	Maximum of real number data	page 4-164
	F_MIN	F0 = F_MIN (F1 C:F2) (F)	Minimum of real number data	page 4-165
Others	;(Comment)	;	Comment	page 4-166
	BRIGHT	BRIGHT F0	Brightness adjustment	page 4-167
	GET_MSGBLK	GET_MSGBLK F0 F1	Message acquisition	page 4-168
	PLC_ULR	PLC_ULR F0 F1	Read user log	page 4-169
	RECONNECT	RECONNECT F0	Multi-drop reconnection (PLC1)	page 4-171
	RECONNECT_EX	RECONNECT_EX PLC F0 F1	Restart	page 4-172
	SAMPLE	SAMPLE F0 F1 F2	Acquire logging/alarm data	page 4-173
	SEARCH_FILE	SEARCH_FILE F0 F1	Search for JPEG files	page 4-176
	ADJ_VOLUME	ADJ_VOLUME F0 F1 F2	Adjust volume	page 4-177
	SAVE_VOLUME	SAVE_VOLUME	Save volume adjustment value	page 4-178
	TREND REFRESH	TREND REFRESH F0 F1	Refresh trend data display	page 4-179
	SYS	SYS (SET_SCRN) F1	Screen number designation	page 4-180
		SYS (SET_MOVLP) F1	Multi-overlap/global overlap setting	page 4-181
		SYS (OVLP_SHOW) F1	Overlap ON/OFF	page 4-183
		SYS (OVLP_POS) F1	Overlap relocation	page 4-184
		SYS (GET_MSG) F1	Message acquisition	page 4-185
		SYS (GET_XY) F1	Acquisition of X and Y coordinates on circumference	page 4-186
		SYS (SET_BZ) F1	Buzzer control	page 4-188
		SYS (GET_TIME) F1	System time acquisition	page 4-189
		SYS (STA_TIME) F1	Timer setting	page 4-190
		SYS (GET_CLND) F1	Calendar acquisition	page 4-192
		SYS (SET_CLND) F1	Calendar setting	page 4-193
		SYS (SET_BUFNO) F1	Logging information	page 4-194
			Alarm log information	page 4-196
		SYS (GET_SMPL) F1	Acquire logging/alarm data	page 4-197
		SYS (GET_SCUR) F1	Cursor point acquisition	page 4-199
		SYS (DSP_DATA) F1	Show/hide numerical data display	page 4-201
		SYS (CHG_DATA) F1	Change numerical data display property	page 4-202
		SYS (STA_LIST) F1	Data sheet print	page 4-204

Category	Command Name	Mnemonic	Contents	Refer to:
Others	SYS	SYS (SET_BKLT) F1	Backlight control	page 4-206
		SYS (RESTART) F1	Restart	page 4-207
		SYS (CHG_LANG) F1	Language change	page 4-208
		SYS (RESET_SCRN) F1	Redisplay screen	page 4-210
		SYS (OUT_ENQ) F1	Universal serial	page 4-211
			A-link + Net10	page 4-212
		SYS (SET_SYS_CLND) F1	System calendar setting	page 4-213

4 Details of Macro Commands

4.1 Guide to Chapter 4



*1 Refer to “V Series Models” in this manual.

4.2 Arithmetical Operation

ADD(+)

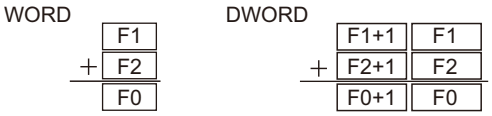
F0 = F1 + F2 (W) . . . . . WORD

F0 = F1 + F2 (D) . . . . . DWORD

All models	○

Function: Addition

This macro command is used to write the result of [F1] plus [F2] to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	◎	◎		○
F2	◎	◎		○

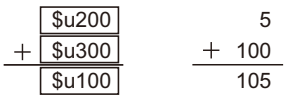
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

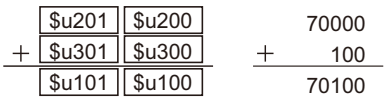
	WORD	DWORD
F0	−32768 - +32767 (Decimal system with signs)	−2147483648 - +2147483647 (Decimal system with signs)
F1		
F2		

Example

- $u100 = $u200 + $u300 (W)



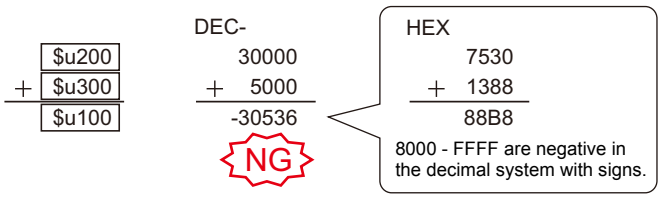
- $u100 = $u200 + $u300 (D)



Supplemental remarks

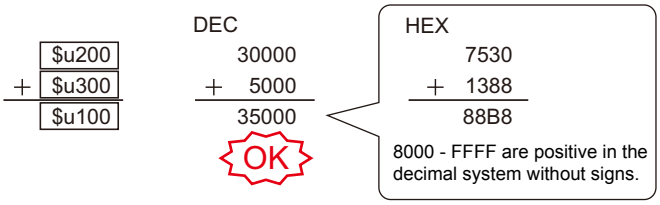
- Operation is performed in the decimal system with signs. Be sure that the result [F0] falls within the permissible range.

  $u100 = $u200 + $u300 (W)



* The execution result in the example above is an overflow.

  If an operation results in “65535” in the decimal system (WORD) or less, it matches the result in the decimal system without signs.



* The execution result in the example above is an overflow.

- In a case where [F1] and [F2] are specified in the following ranges, they are treated as negative values −1 to −32768.
  32768 - 65535 (DEC)
  100000 - 17777 (OCT)
  8000 - FFFF (HEX)

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SUB(-)

All models	○

F0 = F1 - F2 (W) . . . . . WORD

F0 = F1 - F2 (D). . . . . DWORD

Function: Subtraction

This macro command is used to write the result of [F1] minus [F2] to [F0].

WORD

	F1
−	F2
	F0

DWORD

	F1+1	F1
−	F2+1	F2
	F0+1	F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	◎	◎		○
F2	◎	◎		○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	−32768 - +32767 (Decimal system with signs)	−2147483648 - +2147483647 (Decimal system with signs)
F1		
F2		

Example

- $u100 = $u200 − $u300 (W)

	$u200
−	$u300
	$u100

	100
−	40
	60

- $u100 = $u200 − $u300 (D)

	$u201	$u200
−	$u301	$u300
	$u101	$u100

	70000
−	100
	69900

Supplemental remarks

- Operation is performed in the decimal system with signs. Be sure that the result [F0] falls within the permissible range.

	$u200
−	$u300
	$u100

DEC-

	-30000
−	5000
	30536

NG

HEX

	8AD0
−	1388
	7748

0000 - 7FFF are positive in the decimal system with signs.

* The execution result in the example above is an underflow.

- In a case where [F1] and [F2] are specified in the following ranges, they are treated as negative values −1 to −32768.
  32768 - 65535 (DEC)
  100000 - 17777 (OCT)
  8000 - FFFF (HEX)

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

MUL(X)

All models	○

F0 = F1 x F2 (W) . . . . . WORD

F0 = F1 x F2 (D) . . . . . DWORD

Function: Multiplication

This macro command is used to write the result of [F1] multiplied by [F2] to [F0].

WORD

	F1
×	F2
	F0

DWORD

	F1+1	F1
×	F2+1	F2
	F0+1	F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	◎	◎		○
F2	◎	◎		○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	−32768 - +32767 (Decimal system with signs)	−2147483648 - +2147483647 (Decimal system with signs)
F1		
F2		

Example

- $u100 = $u200 × $u300 (W)

	$u200
×	$u300
	$u100

	100
×	40
	4000

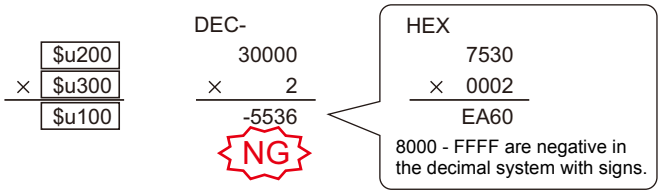
- $u100 = $u200 × $u300 (D)

	$u201	$u200
×	$u301	$u300
	$u101	$u100

	40000
×	2
	80000

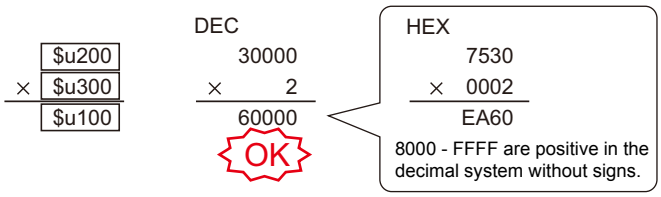
Supplemental remarks

- Operation is performed in the decimal system with signs. Be sure that the result [F0] falls within the permissible range.



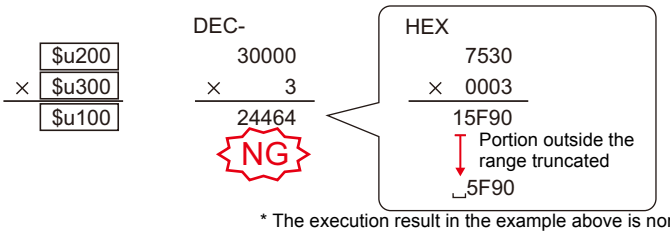
* The execution result in the example above is an overflow.

If an operation results in “65535” (WORD) or less, it matches the result in the decimal system without signs.



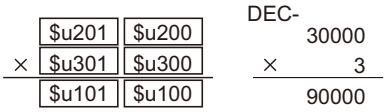
* The execution result in the example above is an overflow.

- If the result [F0] is outside the permissible range, the extra portion is truncated.



* The execution result in the example above is normal.

In this case, operation is performed in DWORD.



- In a case where [F1] and [F2] are specified in the following ranges, they are treated as negative values −1 to −32768.
  32768 - 65535 (DEC)
  100000 - 17777 (OCT)
  8000 - FFFF (HEX)

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

DIV(/)

All models	○

F0 = F1 / F2 (W) . . . . . WORD

F0 = F1 / F2 (D) . . . . . DWORD

Function: Division

This macro command is used to write the result of [F1] divided by [F2] to [F0].

WORD

	F1
÷	F2
F0	··· Remainder

DWORD

	F1+1	F1
÷	F2+1	F2
F0+1	F0	··· Remainder

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	◎	◎		○
F2	◎	◎		○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	−32768 - +32767 (Decimal system with signs)	−2147483648 - +2147483647 (Decimal system with signs)
F1		
F2		

Example

- $u100 = $u200 / $u300 (W)

	$u200
÷	$u300
$u100	··· Remainder

	100
÷	40
2	··· 20

- $u100 = $u200 / $u300 (D)

	$u201	$u200
÷	$u301	$u300
$u101	$u100	··· Remainder

	80000
÷	30000
2	··· 20000

Supplemental remarks

- Operation is performed in the decimal system with signs. Be sure that the [F1] value falls within the permissible range.
- In a case where [F1] and [F2] are specified in the following ranges, they are treated as negative values −1 to −32768.
  32768 - 65535 (DEC)
  100000 - 17777 (OCT)
  8000 - FFFF (HEX)
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
3	Calculation operation execution error
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

MOD(%)

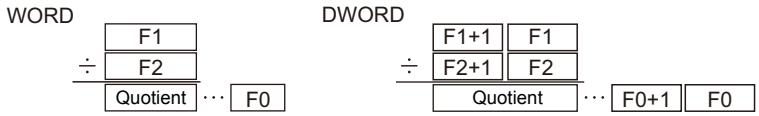
All models	○

F0 = F1 % F2 (W) . . . . . WORD

F0 = F1 % F2 (D) . . . . . DWORD

Function: Remainder of division

This macro command is used to write the remainder of [F1] divided by [F2] to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	◎	◎		○
F2	◎	◎		○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

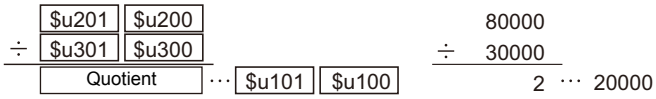
Setting range

	WORD	DWORD
F0	–32768 - +32767 (Decimal system with signs)	–2147483648 - +2147483647 (Decimal system with signs)
F1		
F2		

Example

- $u100 = $u200 % $u300 (W)

- $u100 = $u200 % $u300 (D)



Supplemental remarks

- Operation is performed in the decimal system with signs. Be sure that the [F1] value falls within the permissible range.
- In a case where [F1] and [F2] are specified in the following ranges, they are treated as negative values –1 to –32768.
  32768 - 65535 (DEC)
  100000 - 17777 (OCT)
  8000 - FFFF (HEX)
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
3	Calculation operation execution error
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.3 Logical Operation

AND(&)

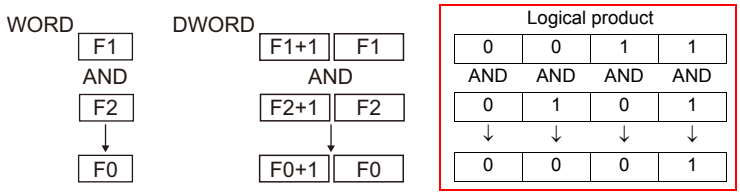
All models	○

F0 = F1 & F2 (W) . . . . . WORD

F0 = F1 & F2 (D) . . . . . DWORD

Function: Logical product

This macro command is used to write the result of [F1] ANDed with [F2] bit by bit to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

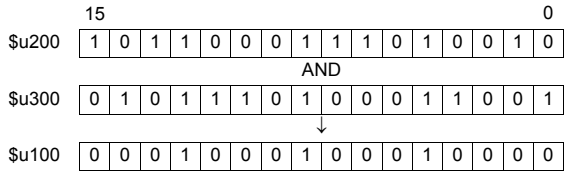
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

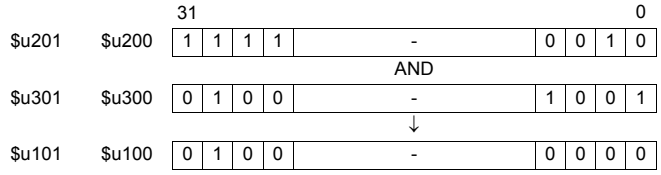
	WORD	DWORD
F0 F1 F2	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)

Example

- $u100 = $u200 & $u300 (W)



- $u100 = $u200 & $u300 (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

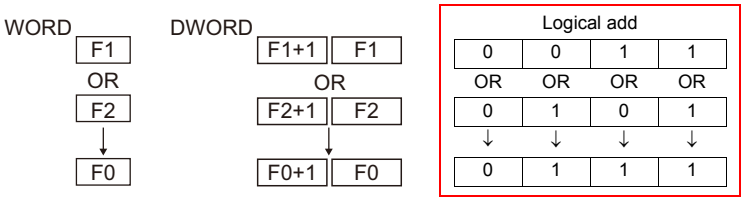
OR(|)

All models	○

F0 = F1 | F2 (W) . . . . . WORD
F0 = F1 | F2 (D) . . . . . DWORD

Function: Logical add

This macro command is used to write the result of [F1] ORed with [F2] bit by bit to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

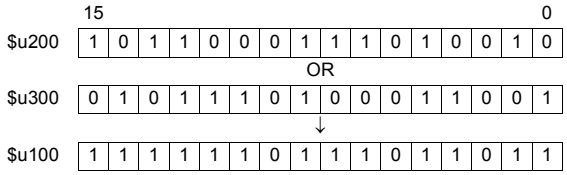
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

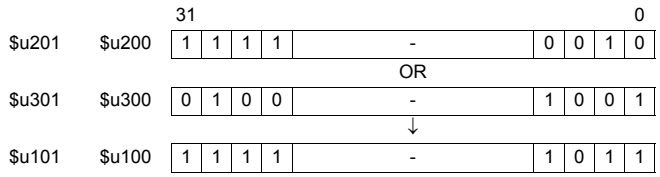
	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2		

Example

- $u100 = $u200 | $u300 (W)



- $u100 = $u200 | $u300 (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

XOR(^)

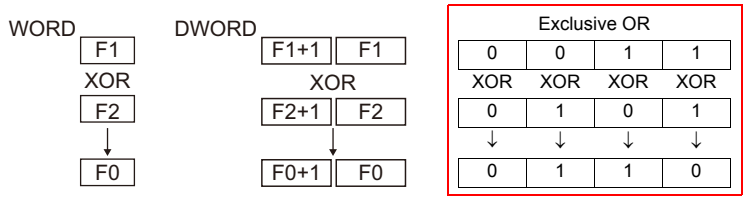
All models	○

F0 = F1 ^ F2 (W) . . . . . WORD

F0 = F1 ^ F2 (D) . . . . . DWORD

Function: Exclusive OR

This macro command is used to write the result of [F1] XORed with [F2] bit by bit to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

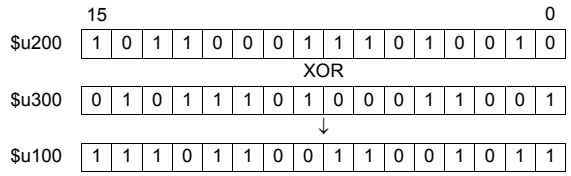
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

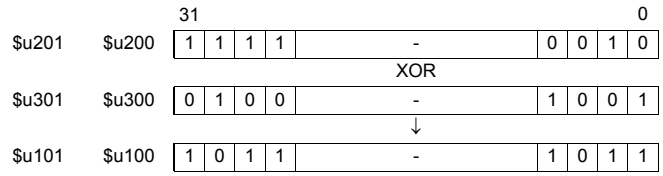
	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2		

Example

- $u100 = $u200 ^ $u300 (W)



- $u100 = $u200 ^ $u300 (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

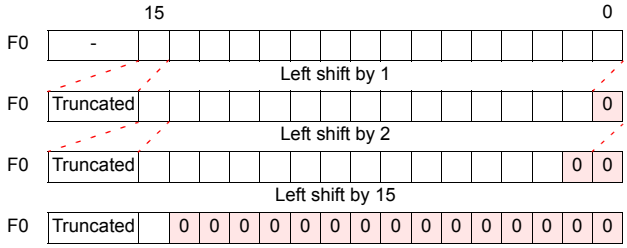
SHL(<<)

All models	○

F0 = F1 << F2 (W) . . . . . WORD
F0 = F1 << F2 (D) . . . . . DWORD

Function: Left shift

This macro command is used to perform logical shift of [F1] to the left by the number of bits specified in [F2] and write the result to [F0]. The higher-order bits (by the number in [F2]) are truncated. “0” is assigned to the lower-order bits (by the number in [F2]).



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	○			○

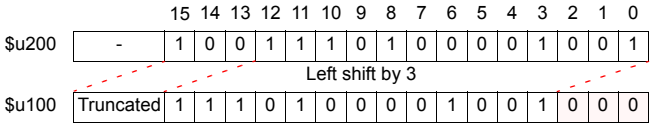
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

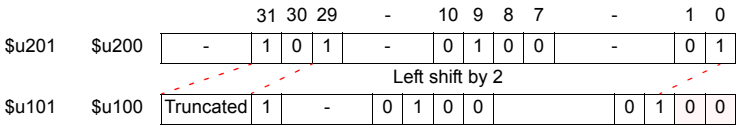
	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2	0 - 15	0 - 31

Example

- $u100 = $u200 << 3 (W)



- $u100 = $u200 << 2 (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SHR(>>)

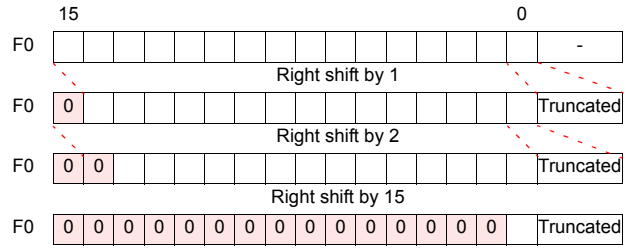
All models	○

F0 = F1 >> F2 (W). . . . . WORD

F0 = F1 >> F2 (D) . . . . . DWORD

Function: Right shift

This macro command is used to perform logical shift of [F1] to the right by the number of bits specified in [F2] and write the result to [F0]. The lower-order bits (by the number in [F2]) are truncated. “0” is assigned to the higher-order bits (by the number in [F2]).



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	○			○

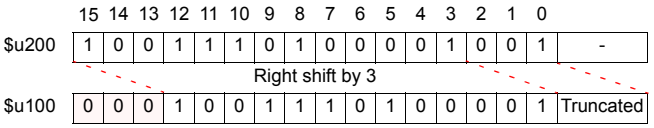
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

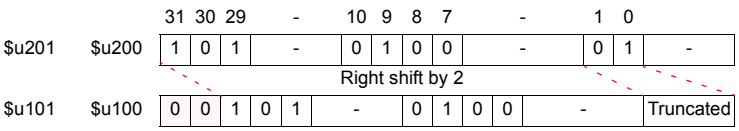
	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2	0 - 15	0 - 31

Example

- $u100 = $u200 >> 3 (W)



- $u100 = $u200 >> 2 (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.4 Statistic

MAX

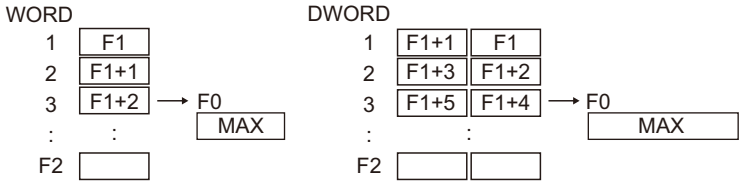
All models	○

F0 = MAX (F1 C : F2) (W) . . . . . WORD

F0 = MAX (F1 C : F2) (D) . . . . . DWORD

Function: Maximum

This macro command is used to find the maximum data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

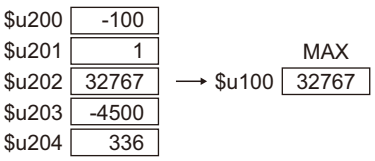
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

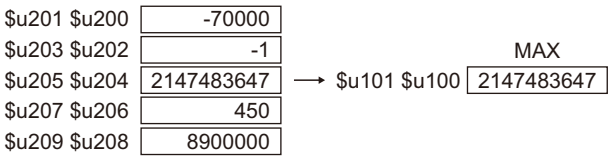
	WORD	DWORD
F0	–32768 - +32767 (Decimal system with signs)	–2147483648 - +2147483647 (Decimal system with signs)
F1		
F2	0 - 512	0 - 512

Example

- $u100 = MAX ($u200 C : 5) (W)



- $u100 = MAX ($u200 C : 5) (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

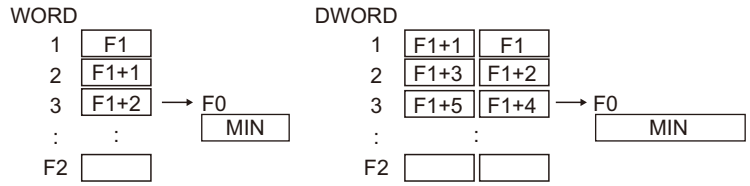
MIN

All models	○

F0 = MIN (F1 C : F2) (W) . . . . . WORD
F0 = MIN (F1 C : F2) (D) . . . . . DWORD

Function: Minimum

This macro command is used to find the minimum data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

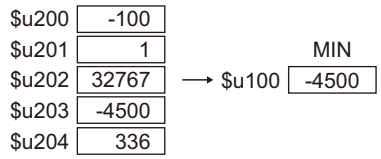
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

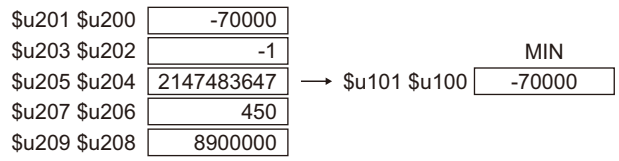
	WORD	DWORD
F0	−32768 - +32767 (Decimal system with signs)	−2147483648 - +2147483647 (Decimal system with signs)
F1		
F2	0 - 512	0 - 512

Example

- $u100 = MIN ($u200 C : 5) (W)



- $u100 = MIN ($u200 C : 5) (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

AVG

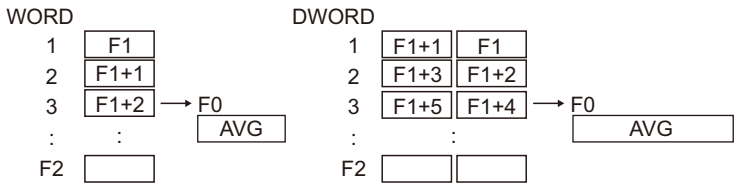
All models	○

F0 = AVG (F1 C : F2) (W) . . . . . WORD

F0 = AVG (F1 C : F2) (D) . . . . . DWORD

Function: Average

This macro command is used to average the data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

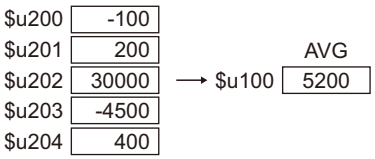
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

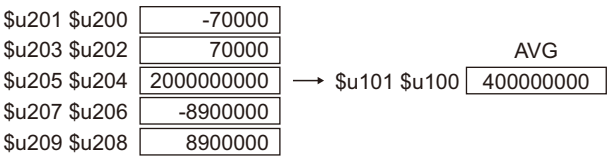
	WORD	DWORD
F0	–32768 - +32767 (Decimal system with signs)	–2147483648 - +2147483647 (Decimal system with signs)
F1		
F2	0 - 512	0 - 512

Example

- $u100 = AVG ($u200 C : 5) (W)



- $u100 = AVG ($u200 C : 5) (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
3	Calculation operation execution error
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SUM

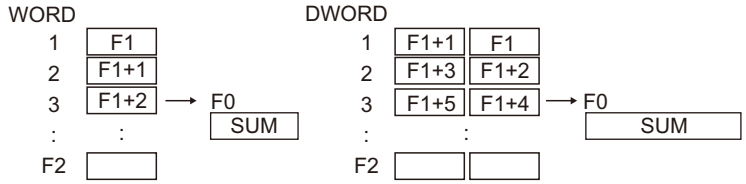
All models	○

F0 = SUM (F1 C : F2) (W) . . . . . WORD

F0 = SUM (F1 C : F2) (D) . . . . . DWORD

Function: Sum

This macro command is used to determine the sum of the data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

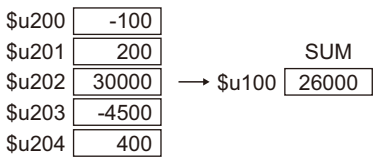
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

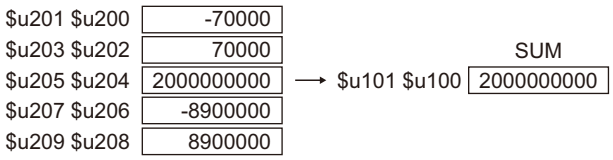
	WORD	DWORD
F0 F1	–32768 - +32767 (Decimal system with signs)	–2147483648 - +2147483647 (Decimal system with signs)
F2	0 - 512	0 - 512

Example

- $u100 = SUM ($u200 C : 5) (W)



- $u100 = SUM ($u200 C : 5) (D)



Supplemental remarks

- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.5 Mathematics/trigonometric

EXP F0 = EXP(F1) (F)

All models	○

Function: Calculation of the exponent

This macro command is used to store the exponent of [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- $u100 = EXP ($u200) (F)

$$2.71828 = e^{1.0}$$

When $u200 = “1.0”, on command execution “2.71828” is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real number, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

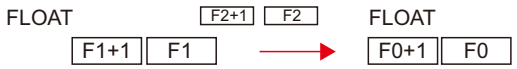
EXPT

F0 = EXPT(F1,F2) (F)

All models	○

Function: Calculation of powers

This macro command is used to store [F1] to the power of [F2] in [F0].
Specify [F0], [F1], and [F2] as floating decimal point (FLOAT) type values.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	

Example

- $u100 = EXPT ($u200,$u300) (F)

$$8 = 2^{3}$$

When $u200 = “2” and $u300 = “3”, on command execution “8” is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

LN

F0 = LN(F1) (F)

All models	○

Function: Calculation of natural logarithms

This macro command is used to store the value of the natural logarithm of [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- $u100 = LN ($u200) (F)

$$2.302585 = \log_e (10.0)$$

When $u200 = “10.0”, on command execution “2.302585” is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real number, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

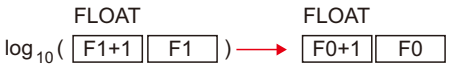
LOG

F0 = LOG(F1) (F)

All models	○

Function: Calculation of common logarithms

This macro command is used to store the value of the common logarithm of [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- $u100 = LOG ($u200) (F)

$$1.0 = \log_{10}(10.0)$$

When $u200 = “10.0”, on command execution “1.0” is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real number, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

SQRT

F0 = SQRT(F1) (F)

All models	○

Function: Calculation of square roots

This macro command is used to store the value of the square root of [F1] in [F0]. Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- $u100 = SQRT ($u200) (F)

$$1.41421 = \sqrt{\ } (2.0)$$

When $u200 = “2.0”, on command execution “1.41421” is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real number, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

ABS

All models	○

F0 = ABS (F1) (W) . . . . . WORD
F0 = ABS (F1) (D) . . . . . DWORD
F0 = ABS (F1) (F) . . . . . FLOAT

Function: Absolute value

This macro command is used to store an absolute value of [F1] in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

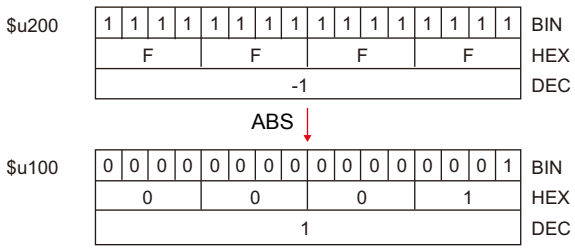
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD	FLOAT
F0	–32767 to +32767 (Decimal system with signs)	–2147483647 to +2147483647 (Decimal system with signs)	IEEE 32-bit single precision real number
F1			

Example

- $u100 = ABS ($u200) (W)
  When $u200 = "–1", on command execution "1" is stored in $u100.



Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

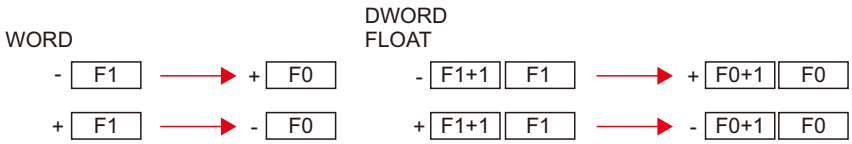
NEG

F0 = NEG (F1) (W) . . . . . WORD
F0 = NEG (F1) (D) . . . . . DWORD
F0 = NEG (F1) (F) . . . . . FLOAT

All models	○

Function: Sign inversion

This macro command is used to store a value with its sign inverted from [F1] in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD	FLOAT
F0	–32767 to +32767 (Decimal system with signs)	–2147483647to +2147483647 (Decimal system with signs)	IEEE 32-bit single precision real number
F1			

Example

- $u100 = NEG ($u200) (W)
  When $u200 = “–1”, on command execution “1” is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

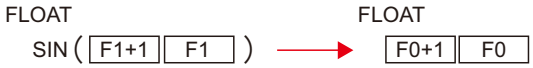
SIN

F0 = SIN (F1) (F) . . . . . FLOAT

All models	○

Function: Sine

This macro command is used to store a sine of the angle (in radians) specified for [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

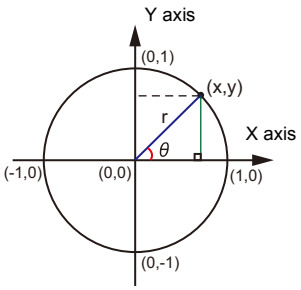
	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain the value for sin 90° in radians;
  $u200 = RAD (90) (F)
  $u100 = SIN ($u200) (F)
  The operation result of “1” is stored in $u100.

* The sine, cosine and tangent of the trigonometric functions can be obtained based on the formulae below.
- Radian (circular measure)
  1 rad = 360/2 π
  = approx. 57.29578 deg.

$$\sin\theta = \frac{y}{r}$$
$$\cos\theta = \frac{x}{r}$$
$$\tan\theta = \frac{y}{x}$$



Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- To convert the unit of an angle, use the macro command of DEG (page 4-32) or RAD (page 4-33).

COS

F0 = COS (F1) (F) . . . . . .FLOAT

All models	○

Function: Cosine

This macro command is used to store a cosine of the angle (in radians) specified for [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain the value for cos 0° in radians;
  $u200 = RAD (0) (F)
  $u100 = COS ($u200) (F)
  The operation result of “1” is stored in $u100.

  * For more information on cosθ of the trigonometric functions, refer to “Example” of “Function: Sine” on page 4-26.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- To convert the unit of an angle, use the macro command of DEG (page 4-32) or RAD (page 4-33).

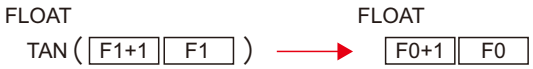
TAN

F0 = TAN (F1) (F) . . . . . FLOAT

All models	○

Function: Tangent

This macro command is used to store a tangent of the angle (in radians) specified for [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain the value for tan 45° in radians;
  $u200 = RAD (45) (F)
  $u100 = TAN ($u200) (F)
  The operation result of “1” is stored in $u100.

* For more information on tanθ of the trigonometric functions, refer to “Example” of “Function: Sine” on page 4-26.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2
3	Calculation operation execution error*3

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].
*3 When the value specified for [F1] is $\pi \times (0.5 + n)$, “−1” is stored in [F0]. (n: integer)

- To convert the unit of an angle, use the macro command of DEG (page 4-32) or RAD (page 4-33).

ASIN

F0 = ASIN (F1) (F) . . . . . .FLOAT

All models	○

Function: Arcsine

This macro command is used to store an arcsine of the angle (in radians) specified for [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

FLOAT FLOAT
SIN^{-1} (F1+1 F1) → F0+1 F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

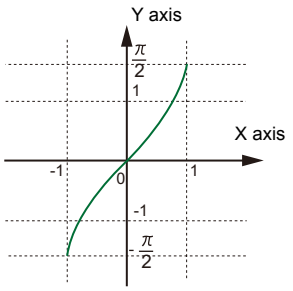
Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain the value for sin^{-1} 1;
  $u100 = ASIN (1) (F)
  The operation result of "1.570796" (= π/2) is stored in $u100.

* The sin^{-1} of the trigonometric functions is expressed in the graph shown on the right.



Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2
3	Calculation operation execution error*3

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].
*3 When the value specified for [F1] is outside the range from "-1" to "1", "–1" is stored in [F0].

- To convert the unit of an angle, use the macro command of DEG (page 4-32) or RAD (page 4-33).

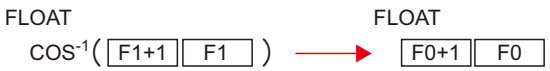
ACOS

F0 = ACOS (F1) (F). . . . . FLOAT

All models	○

Function: Arccosine

This macro command is used to store an arccosine of the angle (in radians) specified for [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

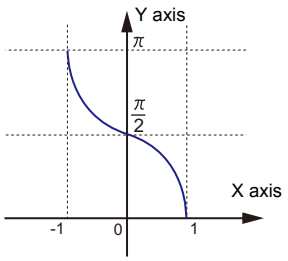
Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain the value for $\cos^{-1} 0$;
  $u100 = ACOS (0) (F)
  The operation result of “1.570796” (= $\pi/2$) is stored in $u100.

* The $\cos^{-1}$ of the trigonometric functions is expressed in the graph shown on the right.



Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2
3	Calculation operation execution error*3

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].
*3 When the value specified for [F1] is outside the range from “-1” to “1”, “–1” is stored in [F0].

- To convert the unit of an angle, use the macro command of DEG (page 4-32) or RAD (page 4-33).

ATAN

F0 = ATAN (F1) (F) . . . . . .FLOAT

All models	○

Function: Arctangent

This macro command is used to store an arctangent of the angle (in radians) specified for [F1] in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

FLOAT FLOAT
TAN^{-1} (F1+1 F1) → F0+1 F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

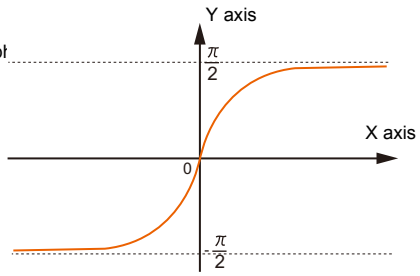
Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain the value for $\tan^{-1}$ 0;
  $u100 = ATAN (0) (F)
  The operation result of “0” is stored in $u100.

* The $\tan^{-1}$ of the trigonometric functions is expressed in the graph shown on the right.



Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

- To convert the unit of an angle, use the macro command of DEG (page 4-32) or RAD (page 4-33).

DEG

F0 = DEG (F1) (F) . . . . . FLOAT

All models	○

Function: Convert radians to degrees

This macro command is used to convert the unit of an angle specified for [F1] from radians to degrees and store the converted value in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain a value in degrees;
  $u100 = ASIN (1) (F)
  $u200 = DEG ($u100) (F)
  The operation result of “90” is stored in $u200.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

RAD

F0 = RAD (F1) (F) . . . . . .FLOAT

All models	○

Function: Convert degrees to radians

This macro command is used to convert the unit of an angle specified for [F1] from degrees to radians and store the converted value in [F0].
Specify [F0] and [F1] as floating decimal point (FLOAT) type values.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	

Example

- To obtain 180° in radians;
  $u100 = RAD (180) (F)
  The operation result of “3.141592” (= π) is stored in $u100.

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*1	Normal
1	Overflow*2
2	Underflow*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)
*2 An indefinite value is stored in [F0].

4.6 Bit Operation

BSET

All models	○

F0 (ON)

Function: Bit set

This macro command is used to set (ON) the memory bit specified in [F0].

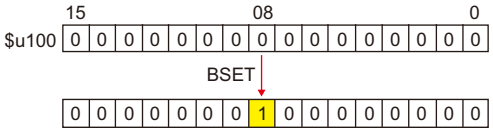
Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

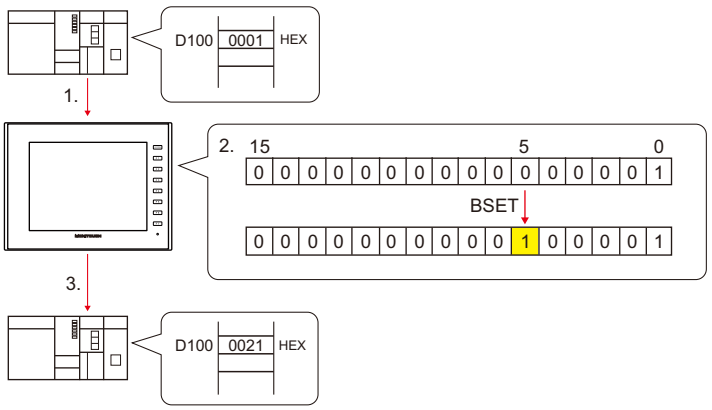
Example

- $u100 - 08 (ON)



Supplemental remarks

- If you use PLC memory or temperature controller memory that is disabled for bit-by-bit read and write, the macro operation as the following takes place.
  Ex.) Mitsubishi PLC D100-05 (ON)
  1. One word that specifies the bit is read.
  2. The bit specified by the above one word is set (ON).
  3. The data is written to the PLC.



* If the bit is changed in a sequence program during processing of step 2, step 3 for data writing is performed.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

BCLR

F0 (OFF)

All models	○

Function: Bit reset

This macro command is used to reset (OFF) the memory bit specified in [F0].

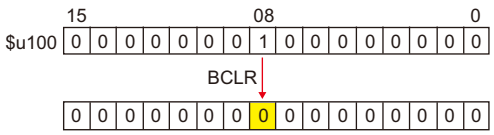
Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

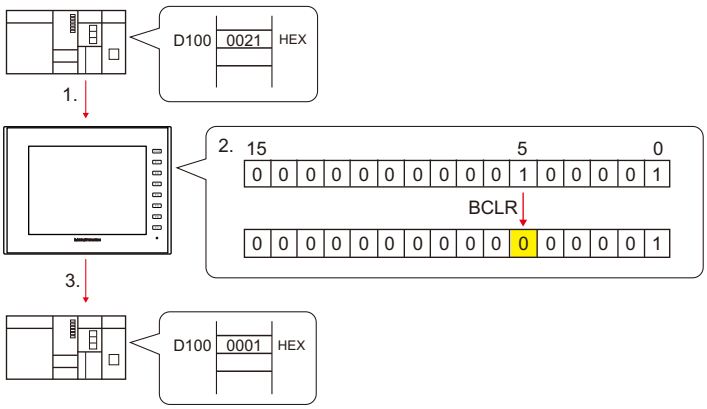
Example

- $u100 - 08 (OFF)



Supplemental remarks

- If you use PLC memory or temperature controller memory that is disabled for bit-by-bit read and write, the macro operation as the following takes place.
  Ex.) Mitsubishi PLC D100-05 (OFF)
  1. One word that specifies the bit is read.
  2. The bit specified by the above one word is reset (OFF).
  3. The data is written to the PLC.



* If the bit is changed in a sequence program during processing of step 2, step 3 for data writing is performed.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

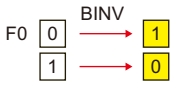
BINV

F0 (INV)

All models	○

Function: Bit inversion

This macro command is used to invert the memory bit specified in [F0].



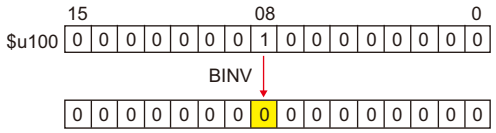
Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

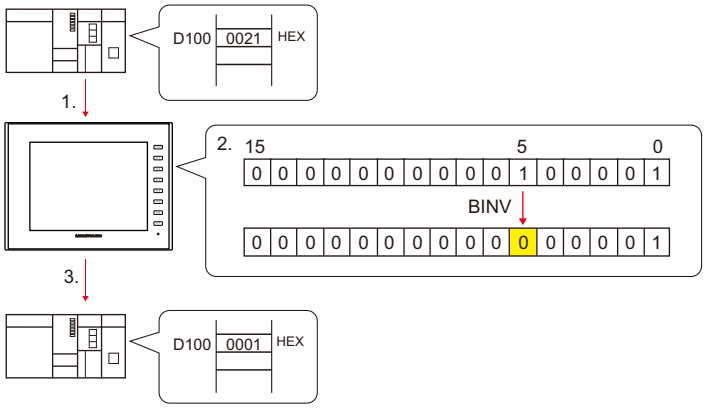
Example

- $u100 - 08 (INV)



Supplemental remarks

- If you use PLC memory or temperature controller memory that is disabled for bit-by-bit read and write, the macro operation as the following takes place.
  Ex.) Mitsubishi PLC D100-05 (INV)
  1. One word that specifies the bit is read.
  2. The bit specified by the above one word is inverted.
  3. The data is written to the PLC.



* If the bit is changed in a sequence program during processing of step 2, step 3 for data writing is performed.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.7 Conversion

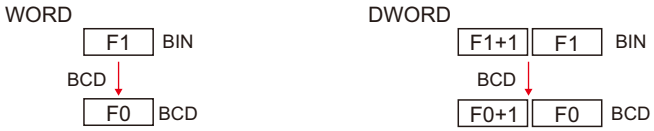
BCD

All models	○

F0 = F1 (W) BCD . . . . . WORD
F0 = F1 (D) BCD . . . . . DWORD

Function: Conversion to BCD

This macro command is used to convert the binary data specified in [F1] to BCD and write the result to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			

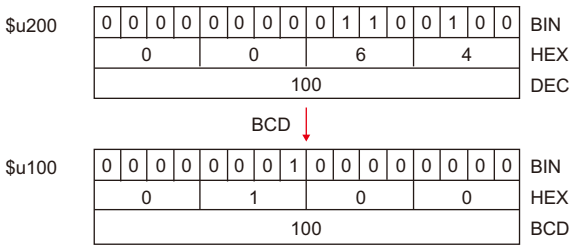
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	0 - 9999 (BCD)	0 - 99999999 (BCD)
F1	0 - 9999 (Decimal system without signs)	0 - 99999999 (Decimal system without signs)

Example

- $u100 = $u200 (W) BCD



Supplemental remarks

- If the value in [F1] is outside the permissible range, [F0] becomes “0”.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

BIN

All models	○

F0 = F1 (W) BIN . . . . . WORD

F0 = F1 (D) BIN. . . . . DWORD

Function: Conversion to BIN

This macro command is used to convert the BCD data specified in [F1] to binary data and write the result to [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			

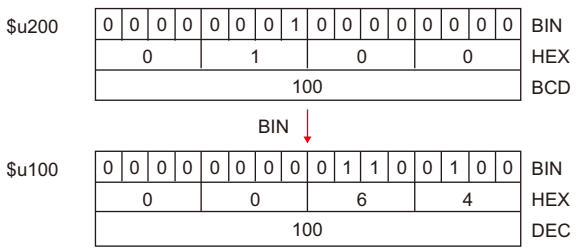
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	0 - 9999 (Decimal system without signs)	0 - 99999999 (Decimal system without signs)
F1	0 - 9999 (BCD)	0 - 99999999 (BCD)

Example

- $u100 = $u200 (W)BIN



Supplemental remarks

- If the value in [F1] is outside the permissible range, [F0] becomes “0”.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

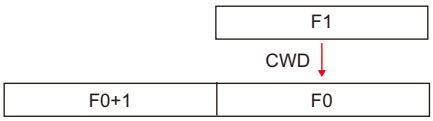
CWD

F0 = F1 D <-W

All models	○

Function: Convert one-word → double-word

This macro command is used to convert the one-word data with sign specified in [F1] to double-word data with sign and write the result to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			

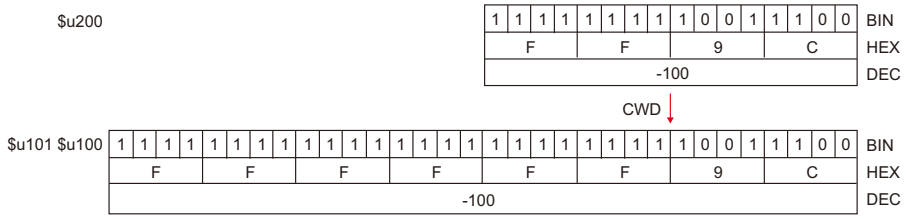
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	−32768 - +32767 (Decimal system with signs)
F0+1	
F1	

Example

- $u100 = $u200 D <- W



Supplemental remarks

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

CVP

All models	○

F0 = F1 (W) PLC <-. . . . . WORD

F0 = F1 (D) PLC <- . . . . . DWORD

Function: Convert binary data to PLC1-format data

This macro command is used to convert the binary data specified in [F1] to the PLC1-format data and write the result to [F0].
The following PLCs manipulate PLC-format data.

- Fuji Electric: MICREX-F all types
- Yaskawa: Memobus [Transmission Mode: Type 1]
- OMRON: All [Transmission Mode: Transmission Mode 2]

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			

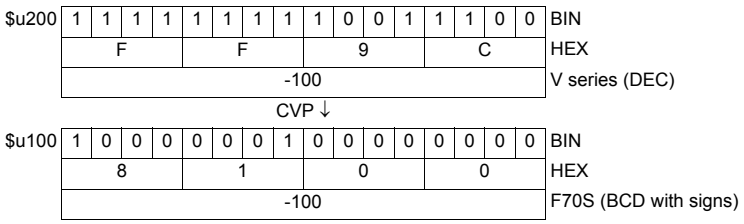
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

- The available memory address range and the type of data vary, depending on the PLCs. Refer to the PLC manual for details.

Example

- Fuji MICREX-F F70S BCD with signs (−7999 to +7999)
  The most significant bit
  OFF: Positive
  ON: Negative
  $u100 = $u200 (W) PLC<-



Supplemental remarks

- The macro command is used in combination with MOV or BMOV.
- To convert to characteristic data other than for PLC1, use “ CVPFMT” (page 4-41).
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

CVPFMT

All models	○

F0 = F1 (W) PLC F2 <- . . . . . WORD

F0 = F1 (D) PLC F2 <- . . . . . DWORD

Function: Convert binary data to PLC-format data specified at [F2]

This macro command is used to convert the binary data specified in [F1] to the PLC-format data specified at [F2] and write the result to [F0].
The following PLCs manipulate PLC-format data.

- Fuji Electric: MICREX-F all types
- Yaskawa: Memobus [Transmission Mode: Type 1]
- OMRON: All [Transmission Mode: Transmission Mode 2]

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

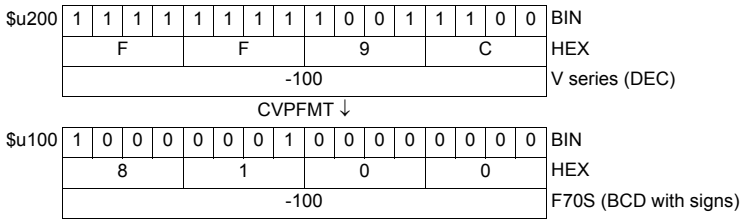
Setting range

	Value
F0	The available memory address range and the type of data vary, depending on the PLCs. Refer to the PLC manual for details.
F1	
F2	1 - 8

Example

- Fuji's MICREX-F series is connected as PLC2.
- Fuji MICREX-F F70S BCD with signs (-7999 to +7999)
  The most significant bit
  OFF: Positive
  ON: Negative

$u100 = $u200 (W) PLC2 <-



Supplemental remarks

- The macro command is used in combination with MOV or BMOV.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

CVB

All models	○

F0 = F1 (W) <- PLC. . . . . WORD

F0 = F1 (D) <- PLC . . . . . DWORD

Function: Convert PLC1-format data to binary data

This macro command is used to convert the PLC1-format data specified in [F1] to binary data and write the result to [F0].
The following PLCs manipulate PLC-format data.

- Fuji Electric: MICREX-F all types
- Yaskawa: Memobus [Transmission Mode: Type 1]
- OMRON: All [Transmission Mode: Transmission Mode 2]

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○

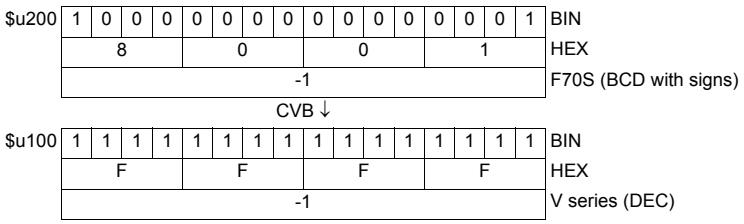
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

- The available memory address range and the type of data vary, depending on the PLCs. Refer to the PLC manual for details.

Example

- Fuji MICREX-F F70S BCD with signs (−7999 to +7999)
  The most significant bit
  OFF: Positive
  ON: Negative
  $u100 = $u200 (W) <-PLC



Supplemental remarks

- The macro command is used in combination with MOV or BMOV.
- To convert to characteristic data other than for PLC1, use “ CVBFMT” (page 4-43).
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

CVBFMT

All models	○

F0 = F1 (W) <- PLC F2 . . . . . WORD

F0 = F1 (D) <- PLC F2 . . . . . DWORD

Function: Convert PLC-format data specified at [F2] to binary data

This macro command is used to convert the PLC-format data specified at [F2] in [F1] to the binary data and write the result to [F0].
The following PLCs manipulate PLC-format data.

- Fuji Electric: MICREX-F all types
- Yaskawa: Memobus [Transmission Mode: Type 1]
- OMRON: All [Transmission Mode: Transmission Mode 2]

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

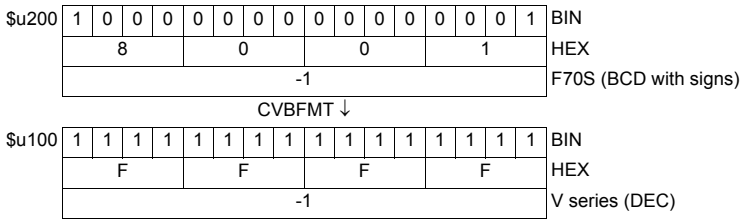
Setting range

	Value
F0	The available memory address range and the type of data vary, depending on the PLCs. Refer to the PLC manual for details.
F1	
F2	1 - 8

Example

- Fuji's MICREX-F series is connected as PLC2.
- Fuji MICREX-F F70S BCD with signs (-7999 to +7999)
  The most significant bit
  OFF: Positive
  ON: Negative

$u100 = $u200 (W) <- PLC2



Supplemental remarks

- The macro command is used in combination with MOV or BMOV.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

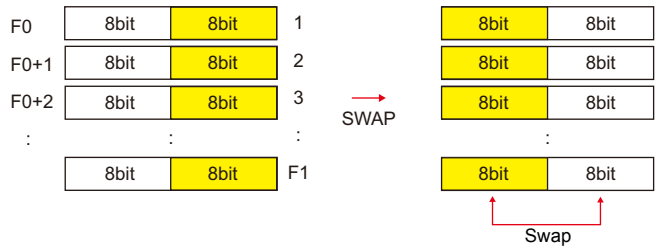
SWAP

F0 = C : F1 (SWAP)

All models	○

Function: Swap MSB with LSB

This macro command is used to perform a swap between the higher-order byte and the lower-order byte of the data at the location starting from the address specified in [F0]. The data count is specified in [F1].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	○			○

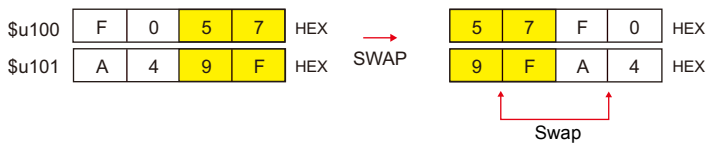
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0000 - FFFF (HEX)
F1	0 - 1024

Example

- $u100 C : 2 (SWAP)



Supplemental remarks

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

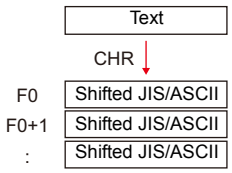
CHR

F0 = ' '

All models	○

Function: Convert text → code

This macro command is used to convert the text placed in quotation marks ' ' to the shifted JIS/ASCII codes and write the result to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	Shifted JIS/ASCII	82 bytes maximum Variable depending on the bytes of the text
F0+1		
:		
' '	Text	80 bytes maximum

Example

- When [MSB → LSB] is selected for [Text Process] on the [Communication Setting] tab window.
  $u100 = string

Text	string					
	CHR↓					
$u100	7	3	7	4	HEX	ts
$u101	7	2	6	9	HEX	ir
$u102	6	E	6	7	HEX	gn
$u103	0	0	0	0	HEX	Null code

Supplemental remarks

- Swap between the higher-order byte and the lower-order byte can be set by selecting an option for [Text Process] under [Communication Setting].
- Regardless of the setting above, use a " STRING" command (page 4-46) for [LSB → MSB] conversions.
- A null code is added to the end. Even-number-byte text thereby uses one extra word.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

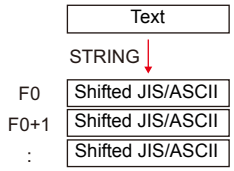
STRING

F0 = ’ ’(STRING)

All models	○

Function: Convert text → code

This macro command is used to convert the text placed in quotation marks ’ ’ to the shifted JIS/ASCII codes and write the result to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	Shifted JIS/ASCII	128 bytes maximum Variable depending on the bytes of the text
F0+1		
:		
‘ ’	Text	128 bytes maximum

Example

$u100 = string

Text	string					
	CHR↓					
$u100	7	4	7	3	HEX	ts
$u101	6	9	7	2	HEX	ir
$u102	6	7	6	E	HEX	gn
$u103	0	0	0	0	HEX	Null code

Supplemental remarks

- Regardless of the [Text Process] setting under [Communication Setting] for PLC1, the data is stored in memory in the [LSB → MSB] sequence.
- A null code is added to the end. Even-number-byte text thereby uses one extra word.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

CVFD

F0(D) <- F1 (F) F2 (D)

All models	○

Function: Convert floating decimal point → 32-bit binary

This macro command is used to convert the 32-bit single precision real number specified in [F1] to 32-bit binary data and store the result in [F0].
[F2] specifies the exponent of "10" at the time of conversion.
If [F2] = 0, rounding to the nearest whole number* is performed. If [F2] = 1, rounding to the nearest tenth* is performed. The result is stored in [F0].

* Rounding down and rounding up are also possible. Refer to page 4-48.

F1

31	30	29	-	24	23	22	21	-	5	4	3	2	1	0
Sign	Exponent					Mantissa								

Real number

0 < Exponent < 255	:	$(-1)^{Sign} \times (1 + Mantissa \times 2^{-23}) \times 2^{(Exponent - 127)}$
Exponent = 0, Mantissa ≠ 0	:	$(-1)^{Sign} \times (Mantissa \times 2^{-23}) \times 2^{-126}$
Exponent = 0, Mantissa = 0	:	0
Sign = 0, Exponent = 255, Mantissa = 0	:	∞
Sign = 1, Exponent = 255, Mantissa = 0	:	-∞
Exponent = 255, Mantissa ≠ 0	:	NaN

CVFD ↓

F0

31	30	29	-	5	4	3	2	1	0
2^{31}	2^{30}	2^{29}	-	2^5	2^4	2^3	2^2	2^1	2^0

BIN

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	–2147483648 - 2147483647 (BIN)
F1	IEEE 32-bit single precision real number
F2	–32 - +32

Example

- $u100 (D) <- $u200 (F) 0 (D)

$u201,$u200

31	30	29	-	24	23	22	21	-	2	1	0
0	127					4194304					
Sign	Exponent					Mantissa					

$(-1)^0 \times (1 + 4194304 \times 2^{-23}) \times 2^{(127-127)} = 1.5$

CVFD ↓

$u101,$u100

31	30	29	-	2	1	0
0	0	0	-	0	1	0

2_{DEC}

- $u100 (D) <- $u200 (F) 1 (D)

$u201,$u200	31	30	29	-	24	23	22	21	-		2	1	0
	0	127							4194304				
	Sign	Exponent							Mantissa				
	$(-1)^0 \times (1 + 4194304 \times 2^{-23}) \times 2^{(127 - 127)} = 1.5$												

CVFD ↓

$u101,$u100	31	30	29	-	2	1	0
	0	0	0	-	1	1	1
	15_{DEC}						

Supplemental remarks

- You can select whether to round to the nearest whole number, round down or round up by specifying the appropriate value for $s99.*

Setting	Operation	
Other than 1 or 2	Round to the nearest whole number	0 - 4 : Round down 5 - 9 : Round up
1	Round down	
2	Round up	0: Round down Other than 0: Round up

* If [Retain compatibility with negative value handling of CVFD macro command] is checked in the [General Settings] tab window ([System Setting] → [Unit Setting] → [General Settings]), the action to round down is performed, irrespective of the value in memory at $s99.

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

CVDF

F0(F) <- F1 (D) F2 (D)

All models	○

Function: Convert 32-bit binary → floating decimal point

This macro command is used to convert the 32-bit binary data specified in [F1] to 32-bit single precision real number and store the result in [F0].
[F2] specifies the exponent of “10” at the time of conversion.

F1

31	30	29	-	5	4	3	2	1	0
2^{31}	2^{30}	2^{29}	-	2^5	2^4	2^3	2^2	2^1	2^0

BIN

CVDF ↓

F0

31	30	29	-	24	23	22	21	-	5	4	3	2	1	0
Sign	Exponent							Mantissa						

Real number

0 < Exponent < 255	:	$(-1)^{Sign} \times (1 + Mantissa \times 2^{-23}) \times 2^{(Exponent - 127)}$
Exponent = 0, Mantissa ≠ 0	:	$(-1)^{Sign} \times (Mantissa \times 2^{-23}) \times 2^{-126}$
Exponent = 0, Mantissa = 0	:	0
Sign = 0, Exponent = 255, Mantissa = 0	:	∞
Sign = 1, Exponent = 255, Mantissa = 0	:	-∞
Exponent = 255, Mantissa ≠ 0	:	NaN

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	–2147483648 - 2147483647 (BIN)
F2	–32 - +32

Example

- $u100 (F) <- $u200 (D) 0 (D)

$u201,$u200

31	30	29	-	2	1	0
00000001_{BIN}						

BIN

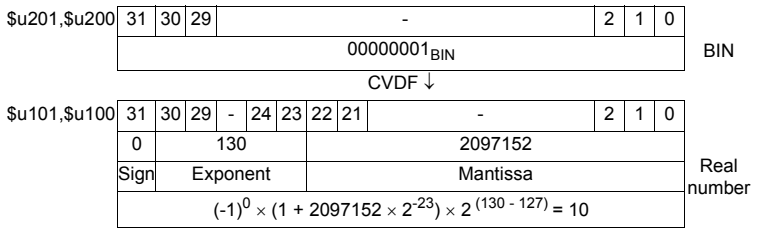
CVDF ↓

$u101,$u100

31	30	29	-	24	23	22	21	-	2	1	0
0	127							0			
Sign	Exponent							Mantissa			
$(-1)^0 \times (1 + 0 \times 2^{-23}) \times 2^{(127 - 127)} = 1$											

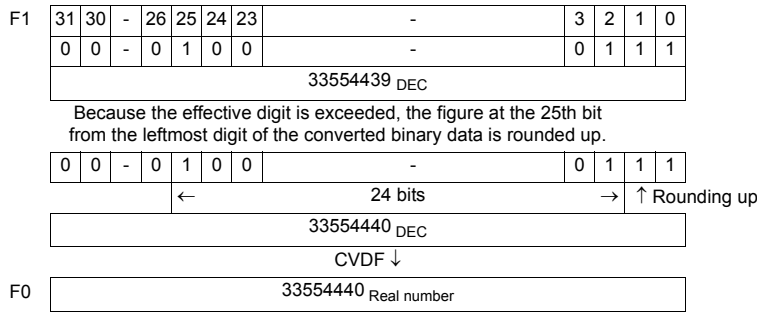
Real number

- $u100 (F) <- $u200 (D) 1 (D)



Supplemental remarks

The V series manipulates 32-bit single precision real numbers. Therefore, in the case of 24-bit binary data that exceeds the significant digit (−16777216 to 16777215 in the decimal system), the figure at the 25th bit from the leftmost digit of the converted binary data is rounded up and the figures at the 26th bit and after are truncated. Since the value obtained in the above manner is used for conversion to real number, an error is introduced.



- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

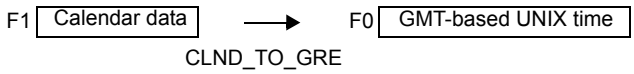
CLND_TO_GRE

CLND_TO_GRE F0 F1 F2

All models	○

Function: Convert calendar data → GMT-based UNIX time

This macro is used to convert the calendar data [F1] in format [F2] to the UNIX time based on GMT, and to store the converted result in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	Time data 0	DEC only
F0+1	Time data 1	Time data 1 | Time data 0 GMT-based UNIX time from January 1, 1970
F1	4 or 2 digits: Year	
F1+1	1 - 12: Month	
F1+2	1 - 31: Day	
F1+3	0 - 23: Hour	
F1+4	0 - 59: Minute	
F1+5	0 - 59: Second	
F2	Data format for [F1] 0: DEC 1: BCD	

: ← V series (return data)

Example

The calendar data in $u200 - $u205 in DEC format, 17 (hour):25 (minutes):10 (seconds) on June 10 in 2010, is converted to the GMT-based UNIX time, and the converted result is stored in $u100 and $u101.

```
$u200 = 2010 (W)
$u201 = 6 (W)
$u202 = 10 (W)
$u203 = 17 (W)
$u204 = 25 (W)
$u205 = 10 (W)
$u300 = 0 (W)
CLND_TO_GRE $u100 $u200 $u300
```

The GMT-based UNIX time “1276190710 seconds” is obtained.

Time data 0 → $u100 = 8182 DEC
Time data 1 → $u101 = 19473 DEC

Supplemental remarks

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Restrictions

- When setting a numerical data display to show the converted result of calendar data, 3 (hour):14 (minutes):7 (seconds) on January 19, 2038 or after, enable the display to show 2-word long data without sign.
- This macro handles any year divisible by 4 as a leap year. For example, the year 2100 is recognized as a leap year though it is not so. Therefore, an error of one day will result.
- The calendar data displayable on the V9 unit ranges from January 1, 2012 to January 19, 2038. Any data outside this range cannot be converted with this macro correctly.

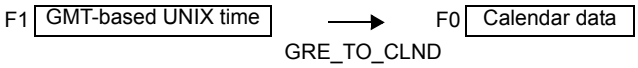
GRE_TO_CLND

GRE_TO_CLND F0 F1 F2

All models	○

Function: Convert GMT-based UNIX time → calendar data

This macro is used to convert the UNIX time based on GMT in [F1] to the calendar data in format [F2], and to store the converted result in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	4 digits: Year	
F0+1	1 - 12: Month	
F0+2	1 - 31: Day	
F0+3	0 - 23: Hour	
F0+4	0 - 59: Minute	
F0+5	0 - 59: Second	
F0+6	0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday	
F1	Time data 0	DEC only
F1+1	Time data 1	Time data 1 | Time data 0 GMT-based UNIX time from January 1, 1970
F2	Data format for [F0] 0: DEC 1: BCD	

: ← V series (return data)

Example

The GMT-based UNIX time, 1278663500 seconds, in $u200 is converted to the calendar data in DEC format, and the converted result is stored in $u100 and after.

GRE_TO_CLND $u100 $u200 0

The calendar data, “8 (hour):18 (minutes):20 (seconds) on Friday on July 9, 2010,” is obtained.

Year → $u100 = 2010 DEC
Month → $u101 = 7 DEC
Day → $u102 = 9 DEC
Hour → $u103 = 8 DEC
Minutes → $u104 = 18 DEC
Seconds → $u105 = 20 DEC
Day of the week → $u106 = 5 DEC

Supplemental remarks

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Restrictions

- This macro handles any year divisible by 4 as a leap year. For example, the year 2100 is recognized as a leap year though it is not so. Therefore, an error of one day will result.
- The calendar data displayable on the V9 unit ranges from January 1, 2012 to January 19, 2038. Any data outside this range cannot be converted with this macro correctly.

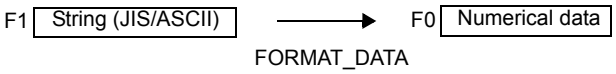
FORMAT_DATA

FORMAT_DATA F0 F1 F2

All models	○

Function: Convert string → numerical data

This macro is used to convert the string [F1] according to the attributes [F2], and to store the converted result in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	◎			
F2	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

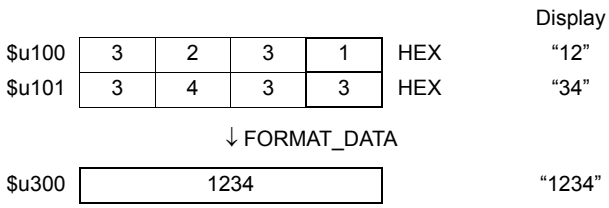
	Value	Remarks
F0	Target memory: BIN data	The number of words depends on [F2+1] (data length).
F1	Source memory: String (ASCII code)	The number of bytes depends on [F2+3] (character count). 32 bytes maximum (16 words) Character processing LSB → MSB fixed
F2	0: DEC without sign (decimal) 1: DEC with a negative sign (decimal) 2: DEC with a positive/negative sign (decimal) 3: HEX (hexadecimal) 4: OCT (octal) 5: BIN (binary) 6: FLOAT (real number)	Format for [F1] If “DEC with a negative sign” or “FLOAT” is selected for [F2] for the conversion of a positive value, add a space code (20H) to the leftmost position of the positive value. Otherwise, an error will result. A space code is not included in the number of digits. Example: For a string “123” to be converted, add a space to make it as “␣123”.
F2+1	0: 1 word 1: 2 words	Data length for [F0] If “FLOAT” is selected for [F2], specify “0”.
F2+2	0: DEC 1: BCD	Data format for [F0] If “HEX,” “OCT,” “BIN,” or “FLOAT” is selected for [F2], specify “0”.
F2+3	1 - 32: [F2] = 0, 1, 2, 5, or 6 1 - 8: [F2] = 3 1 - 11: [F2] = 4	Number of digits for [F1] A positive/negative sign and a decimal point are not included in the number of digits. Example: For a string “−12.3” to be converted, the number of digits is three.
F2+4	0 - 10: [F2] = 0, 1, or 2 0 - 31: [F2] = 6	Decimal place for [F1] Example: For a string “12.34” to be converted, specify two decimal places.
F2+5	0: With zero suppress 1: Without zero suppress	Format for [F1]

	Value	Remarks
F2+6	Valid only when F2+5 = 0 0: Leading spaces removed 1: Trailing spaces removed	Format for [F1] When a value in [F1] includes leading spaces, specify “0”. When a value in [F1] includes trailing spaces, specify “1”. Example: 0: ␣␣␣12 → 12 1: 12 ␣␣␣ → 12
F2+7	0 fixed	

Example

The string in $u100 is converted to the numerical data, and the converted result is stored in $u300.

- String “1234”: DEC without sign



$u00100 = ‘1234’ (STRING)
$u00200 = 0 (W) [DEC without sign]
$u00201 = 0 (W) [1 word]
$u00202 = 0 (W) [DEC]
$u00203 = 4 (W) [4 digits]
$u00204 = 0 (W) [Without decimal point]
$u00205 = 0 (W) [With zero suppress]
$u00206 = 0 (W) [Leading spaces removed]
$u00207 = 0 (W) [0 fixed]
FORMAT_DATA $u00300 $u00100 $u00200

The result “1234” is stored in $u300.

- String “12.34”: A positive value in DEC with a negative sign format and with two decimal places

$u00100 = ‘ ␣12.34’ (STRING)
; (For a positive value, add a space code 20H to the leftmost position.)
$u00200 = 1 (W) [DEC with a negative sign]
$u00201 = 0 (W) [1 word]
$u00202 = 0 (W) [DEC]
$u00203 = 4 (W) [4 digits]
$u00204 = 2 (W) [Two decimal places]
$u00205 = 0 (W) [With zero suppress]
$u00206 = 0 (W) [Leading spaces removed]
$u00207 = 0 (W) [0 fixed]
FORMAT_DATA $u00300 $u00100 $u00200

The result “1234” is stored in $u300.

- String “−12.34”: A negative value in DEC with a negative sign format and with two decimal places

```
$u00100 = ‘−12.34’ (STRING)
$u00200 = 1 (W)  [DEC with a negative sign]
$u00201 = 0 (W)  [1 word]
$u00202 = 0 (W)  [DEC]
$u00203 = 4 (W)  [4 digits]
$u00204 = 2 (W)  [Two decimal places]
$u00205 = 0 (W)  [With zero suppress]
$u00206 = 0 (W)  [Leading spaces removed]
$u00207 = 0 (W)  [0 fixed]
FORMAT_DATA $u00300 $u00100 $u00200
```

The result “−1234” is stored in $u300.

- String “1234”: FLOAT

```
$u00100 = ‘ ␣1234’ (STRING)
;(For a positive value, add a space code 20H to the leftmost position.)
$u00200 = 6 (W)  [FLOAT]
$u00201 = 0 (W)  [0 fixed]
$u00202 = 0 (W)  [0 fixed]
$u00203 = 4 (W)  [4 digits]
$u00204 = 0 (W)  [Without decimal point]
$u00205 = 0 (W)  [With zero suppress]
$u00206 = 0 (W)  [Leading spaces removed]
$u00207 = 0 (W)  [0 fixed]
FORMAT_DATA $u00300 $u00100 $u00200
```

The result “1234” is stored in $u300 and $u301.

- String “001234”: DEC without sign format and without zero suppress

```
$u00100 = ‘001234’ (STRING)
$u00200 = 0 (W)  [DEC without sign]
$u00201 = 0 (W)  [1 word]
$u00202 = 0 (W)  [DEC]
$u00203 = 6 (W)  [6 digits]
$u00204 = 0 (W)  [Without decimal point]
$u00205 = 1 (W)  [Without zero suppress]
$u00206 = 0 (W)  [Leading spaces removed]
$u00207 = 0 (W)  [0 fixed]
FORMAT_DATA $u00300 $u00100 $u00200
```

The result “1234” is stored in $u300.

- String “␣ ␣1234”: DEC without sign format and with two leading spaces

```
$u00100 = ‘ ␣ ␣1234’ (STRING)
$u00200 = 0 (W)  [DEC without sign]
$u00201 = 0 (W)  [1 word]
$u00202 = 0 (W)  [DEC]
$u00203 = 6 (W)  [6 digits]
$u00204 = 0 (W)  [Without decimal point]
$u00205 = 0 (W)  [With zero suppress]
$u00206 = 0 (W)  [Leading spaces removed]
$u00207 = 0 (W)  [0 fixed]
FORMAT_DATA $u00300 $u00100 $u00200
```

The result “1234” is stored in $u300.

- String “1234 ␣ ␣”: DEC without sign format and with two trailing spaces
  $u00100 = ‘1234 ␣ ␣’ (STRING)
  $u00200 = 0 (W) [DEC without sign]
  $u00201 = 0 (W) [1 word]
  $u00202 = 0 (W) [DEC]
  $u00203 = 6 (W) [6 digits]
  $u00204 = 0 (W) [Without decimal point]
  $u00205 = 0 (W) [With zero suppress]
  $u00206 = 1 (W) [Trailing spaces removed]
  $u00207 = 0 (W) [0 fixed]
  FORMAT_DATA $u00300 $u00100 $u00200

  The result “1234” is stored in $u300.

Supplemental remarks

- If “HEX” is specified as an attribute for conversion, characters “A” - “F” of the source data is not case-sensitive.
- If this macro, with “FLOAT” specified as an attribute, results in underflow, “0” is obtained as the converted result.
- Conversion with this macro is in the order of LSB → MSB.
- The following PLCs provided with PLC-specific data format are capable of handling negative values in BCD with a sign format. When you run this macro using such a value with any of these PLCs, the internal memory is not valid for [F0]. Therefore, be sure to assign the PLC memory (specific to the PLC model) to [F0].
  - Fuji Electric: All of the MICREX-F series
  - Yaskawa: Memobus [Trans. Mode: Type 1]
  - Omron: All [Transmission Mode 2]
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

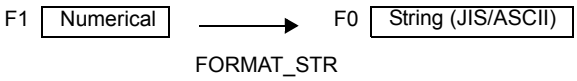
FORMAT_STR

FORMAT_STR F0 F1 F2

All models	○

Function: Convert numerical data → string

This macro is used to convert the numerical data [F1] according to the attributes [F2], and to store the converted result in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎	◎		
F2	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	Target device memory: String (ASCII code)	The number of bytes depends on [F2+3] (character count). 32 bytes maximum (16 words) Character processing LSB → MSB fixed
F1	Source device memory: BIN data	The number of words depends on [F2+1] (data length).
F2	0: DEC without sign (decimal) 1: DEC with a negative sign (decimal) 2: DEC with a positive/negative sign (decimal) 3: HEX (hexadecimal) 4: OCT (octal) 5: BIN (binary) 6: FLOAT (real number)	Format for [F1] If “DEC with a negative sign” or “FLOAT” is selected for [F2] and the converted result is a positive value, a space code (20H) is added to the leftmost position of the positive value. Example: For numerical data “123” to be converted, a space is added to provide a converted result as “␣123”.
F2+1	0: 1 word 1: 2 words	Data length for [F1] If “FLOAT” is selected for [F2], specify “0”.
F2+2	0: DEC 1: BCD	Data format for [F1] If “HEX,” “OCT,” “BIN,” or “FLOAT” is selected for [F2], specify “0”.
F2+3	1 - 32: [F2] = 0, 1, 2, 5, or 6 1 - 8: [F2] = 3 1 - 11: [F2] = 4	Number of digits for [F0] A positive/negative sign and a decimal point are not included in the number of digits. If the number of digits specified for [F2+3] is smaller than that of the converted string, the result is given as a hyphen “-”. Example: For a string “-12.3” as the converted result, the number of digits is three.
F2+4	0 - 10: [F2] = 0, 1, or 2 0 - 31: [F2] = 6	Decimal place for [F0] Example: For a string “12.34” as the converted result, the number of digits is four and two decimal places are given.

	Value	Remarks
F2+5	0: With zero suppress 1: Without zero suppress	Format for [F0] Select whether to execute zero suppress. Example: For a string “00012” as the converted result, specify “1”.
F2+6	Valid only when F2+5 = 0 0: Leading spaces added 1: Trailing spaces added	Format for [F0] When a value in [F0] includes leading spaces, specify “0”. When a value in [F1] includes trailing spaces, specify “1”. Example: 0: 12 → ␣␣12 1: 12 → 12␣␣
F2+7	0 fixed	

Example

The numerical data in $u100 is converted to a string according to the specified attributes, and the converted result is stored in $u300.

- Numerical data “1234”: DEC without sign

						Display
$u100	1234					“1234”
	↓ FORMAT_STR					
$u300	3	2	3	1	HEX	“12”
$u301	3	4	3	3	HEX	“34”

$u00100 = 1234 (W)
$u00200 = 0 (W) [DEC without sign]
$u00201 = 0 (W) [1 word]
$u00202 = 0 (W) [DEC]
$u00203 = 4 (W) [4 digits]
$u00204 = 0 (W) [Without decimal point]
$u00205 = 0 (W) [With zero suppress]
$u00206 = 0 (W) [Leading spaces added]
$u00207 = 0 (W) [0 fixed]
FORMAT_STR $u00300 $u00100 $u00200

The result “1234” is stored in $u300 and $u301.

- Numerical data “1234”: DEC without sign format and with zero suppress and leading spaces

$u00100 = 1234 (W)
$u00200 = 0 (W) [DEC without sign]
$u00201 = 0 (W) [1 word]
$u00202 = 0 (W) [DEC]
$u00203 = 6 (W) [6 digits]
$u00204 = 0 (W) [Without decimal point]
$u00205 = 0 (W) [With zero suppress]
$u00206 = 0 (W) [Leading spaces added]
$u00207 = 0 (W) [0 fixed]
FORMAT_STR $u00300 $u00100 $u00200

The result “␣␣1234” is stored in $u300 to $u302.

- Numerical data “1234”: DEC without sign format and with zero suppress and trailing spaces
  $u00100 = 1234 (W)
  $u00200 = 0 (W) [DEC without sign]
  $u00201 = 0 (W) [1 word]
  $u00202 = 0 (W) [DEC]
  $u00203 = 6 (W) [6 digits]
  $u00204 = 0 (W) [Without decimal point]
  $u00205 = 0 (W) [With zero suppress]
  $u00206 = 1 (W) [Trailing spaces added]
  $u00207 = 0 (W) [0 fixed]
  FORMAT_STR $u00300 $u00100 $u00200

  The result “1234␣␣” is stored in $u300 to $u302.

- Numerical data “1234”: DEC without sign format and without zero suppress
  $u00100 = 1234 (W)
  $u00200 = 0 (W) [DEC without sign]
  $u00201 = 0 (W) [1 word]
  $u00202 = 0 (W) [DEC]
  $u00203 = 6 (W) [6 digits]
  $u00204 = 0 (W) [Without decimal point]
  $u00205 = 1 (W) [Without zero suppress]
  $u00206 = 0 (W) [Leading spaces added]
  $u00207 = 0 (W) [0 fixed]
  FORMAT_STR $u00300 $u00100 $u00200

  The result “001234” is stored in $u300 to $u302.

- Numerical data “12.34”: DEC with a negative sign format and with two decimal places
  $u00100 = 1234 (W)
  $u00200 = 1 (W) [DEC with a negative sign]
  $u00201 = 0 (W) [1 word]
  $u00202 = 0 (W) [DEC]
  $u00203 = 4 (W) [4 digits]
  $u00204 = 2 (W) [Two decimal places]
  $u00205 = 0 (W) [With zero suppress]
  $u00206 = 0 (W) [Leading spaces added]
  $u00207 = 0 (W) [0 fixed]
  FORMAT_STR $u00300 $u00100 $u00200

  The result “␣12.34” is stored in $u300 to $u302.
  (For a positive value, a space code 20H is added to the leftmost position.)

- Numerical data “1234.00”: FLOAT
  $u00100 = 1234 (D)
  $u00100(F) <- $u00100(D) 0 (D)
  $u00200 = 6 (W) [FLOAT]
  $u00201 = 0 (W) [0 fixed]
  $u00202 = 0 (W) [0 fixed]
  $u00203 = 6 (W) [6 digits]
  $u00204 = 2 (W) [Two decimal places]
  $u00205 = 0 (W) [With zero suppress]
  $u00206 = 0 (W) [Leading spaces added]
  $u00207 = 0 (W) [0 fixed]
  FORMAT_STR $u00300 $u00100 $u00200

  The result “␣1234.00” is stored in $u300 to $u303.
  (For a positive value, a space code 20H is added to the leftmost position.)

Supplemental remarks

- Conversion with this macro is in the order of LSB → MSB.
- A NULL code is added to the end of the string as a result of conversion. Even-number-byte string thereby uses one extra word.
- The following PLCs provided with PLC-specific data format are capable of handling negative values in BCD with a sign format. When you run this macro using such a value with any of these PLCs, the internal memory is not valid for [F1]. Therefore, be sure to assign the PLC memory (specific to the PLC model) to [F1].
  - Fuji Electric: All of the MICREX-F series
  - Yaskawa: Memobus [Trans. Mode: Type 1]
  - Omron: All [Transmission Mode 2]
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.8 Transfer

MOV

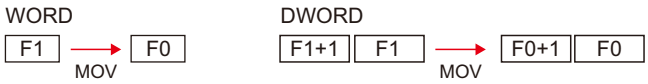
All models	○

F0 = F1 (W) . . . . . WORD

F0 = F1 (D) . . . . . DWORD

Function: Transfer

This macro command is used to transfer the data at the address specified in [F1] to the address in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	○

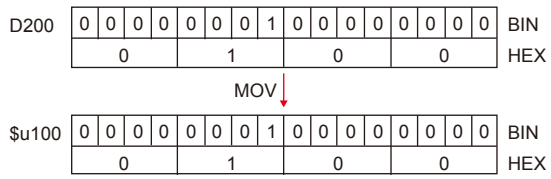
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

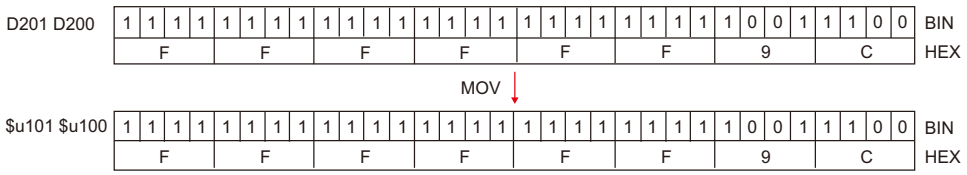
	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		

Example

- $u100 = PLC1 [D200] (W)



- $u100 = PLC1 [D200] (D)



Supplemental remarks

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

BMOV

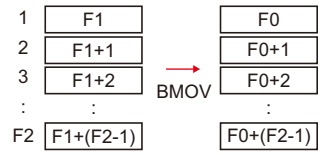
All models	○

F0 = F1 C : F2 (BMOV)(W) . . . . . WORD
F0 = F1 C : F2 (BMOV)(D) . . . . . DWORD

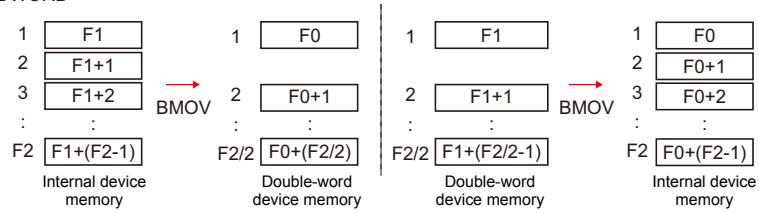
Function: Block transfer

This macro command is used to transfer the data at the location starting from the address specified in [F1] in a block to the top address in [F0]. The data count is specified in [F2].

WORD



DWORD



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

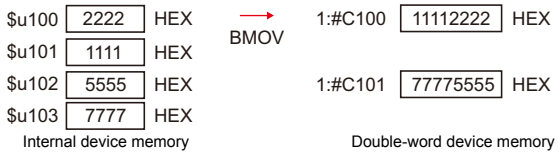
Setting range

	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2	0 - 4096	0 - 4096

Example

- $u100 = PLC1 [D200] C : 3 (BMOV) (W)

- PLC2 [1:#C100] = $u100 C : 4 (BMOV) (D) or
  PLC2 [1:#C100] = $u100 C : 3 (BMOV) (D)



- $u100 = PLC2 [1:#C100] C : 4 (BMOV) (D) or
  $u100 = PLC2 [1:#C100] C : 3 (BMOV) (D)

4

Supplemental remarks

- If [☐ Permit Double-Word Transfer by BMOV] is not checked on the [General Settings] tab window ([System Setting] → [Unit Setting] → [General Settings]), DWORD cannot be selected.
  If BMOV in double-word device memory is executed though the option is not checked, the following results:

  PLC2 [1:#C100] = $u100 C : 4 (BMOV)

  $u100 = PLC2 [1:#C100] C : 4 (BMOV)

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

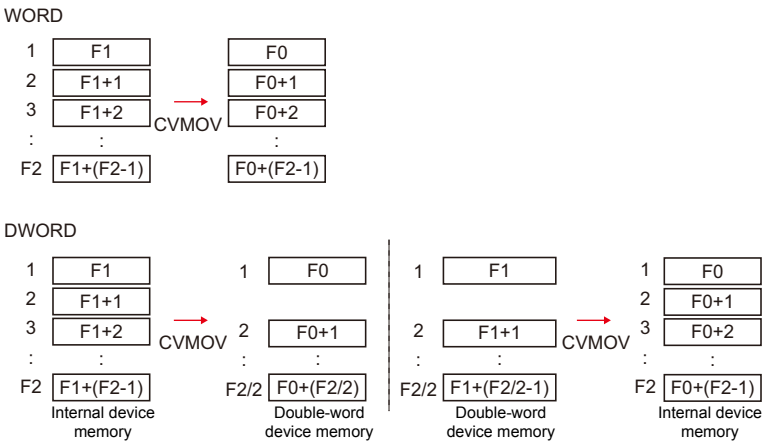
CVMOV

All models	○

F0 = F1 C : F2 (CVMOV)(W). . . . . WORD
F0 = F1 C : F2 (CVMOV)(D) . . . . . DWORD

Function: Block transfer

This macro command is used to transfer the data at the location starting from the address specified in [F1] in a block to the top address in [F0]. The data count is specified in [F2]. Depending on the PLC models, data conversion takes place at the same time.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2	0 - 4096	0 - 4096

Example

Refer to the operation example applicable to your PLC model. If any PLC other than listed blow is in use, the operation identical to the BMOV command takes place.

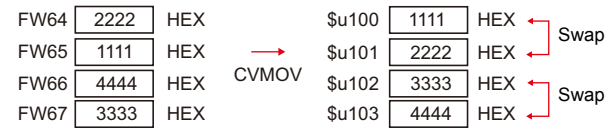
Device selection		Remarks	Operation
Fuji Electric	MICREX-F Series		2
Hitachi	HIDIC-S10/2α, S10mini		1
	HIDIC-S10/2α, S10mini (Ethernet)		
	HIDIC-S10/4α		
	HIDIC-S10V		
	HIDIC-S10V (Ethernet)		
OMRON	All models	[Transmission Mode: Transmission Mode 2] in the [Communication Setting] tab window	2
Siemens	S5 PG Port*		1
	S7		
	S7-200PPI		
	S7-300/400MPI		
	TI500/505		
Yaskawa	Memobus	[Transmission Mode: Type 1] in the [Communication Setting] tab window	2

Device selection (temperature controller / servo / inverter)		Remarks	Operation
IAI	PCON/ACON/SCON(MODBUS RTU)		1

- Operation 1: With Hitachi's PLC selected as PLC1
  - $u100 = PLC1 [FW0064] C : 3 (CVMOV) (W)

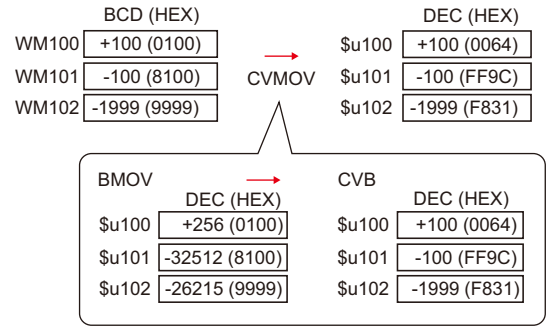
In the case of WORD, the operation identical to BMOV takes place.

  - $u100 = PLC1 [FW0064] C : 3 (CVMOV) (D) or
    $u100 = PLC1 [FW0064] C : 4 (CVMOV) (D)



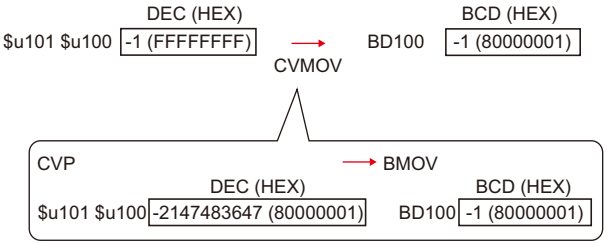
In the case of DWORD, a swap between the higher-order word and the lower-order word takes place.

- Operation 2: With Fuji's PLC selected as PLC2
  - $u100 =PLC2 [WM100] C : 3 (CVMOV) (W)



PLC-format data (BCD with signs) converted to binary data is stored.

  - PLC2 [BD100] = $u100 C : 2 (CVMOV) (D)



Binary data converted to PLC-format data (BCD with signs) is stored.

Supplemental remarks

- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

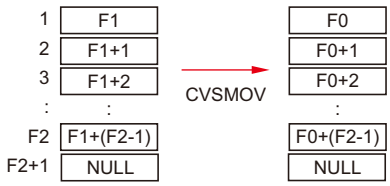
CVSMOV

All models	○

F0 = F1 C : F2 (CVSMOV) (W) . . . . . WORD
F0 = F1 C : F2 (CVSMOV) (D) . . . . . DWORD

Function: Block transfer with text process conversion

This macro command is used to transmit the data at the location starting from the address specified in [F1] in a block to the top address in [F0]. The data count is specified in [F2]. In transfer from the internal device memory to the PLCn device memory, from the PLCn device memory to the internal device memory, or from PLCm device memory to the PLCn device memory, text conversion is executed at the same time.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	Text	102 bytes maximum (Varies depending on the bytes of the text)
F0+1		
:		
F1	Text	102 bytes maximum (Varies depending on the bytes of the text)
F1+1		
:		
F2	0 - 100	100 bytes maximum

Example

- When the [Communication Setting] → [Text Process] setting for the PLC that is the transfer destination (PLC3) is [MSB → LSB]:
  - PLC3 [D100] = $u100 C : 8 (CVSMOV) (W)

$u100	7	4	7	3	HEX	ts
$u101	6	9	7	2	HEX	ir
$u102	6	7	6	E	HEX	gn
$u103	0	0	0	0	HEX	Null code
	↓ CVSMOV					
D100	7	3	7	4	HEX	ts
D101	7	2	6	9	HEX	ir
D102	6	E	6	7	HEX	gn
D103	0	0	0	0	HEX	Null code

Supplemental remarks

- A null code is added to the end. Even-number-byte text thereby uses one extra word.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

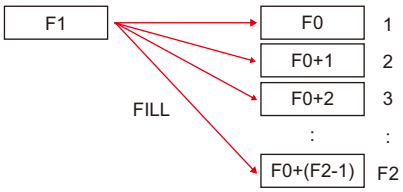
FILL

F0 = F1 C : F2 (FILL)

All models	○

Function: Transfer all

This macro command is used to write the data specified in [F1] to the words starting from the address in [F0]. The number of the words is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎		
F1	○			○
F2	○			○

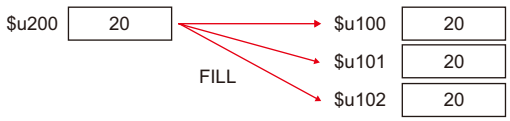
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD
F0	0000 - FFFF (HEX)
F1	
F2	0 - 4096

Example

- $u100 = $u200 C : 3 (FILL)



Supplemental remarks

- When a PLC device memory address is specified for [F0], code conversion is not performed.
- The result of macro execution is stored in $s1057.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.9 Comparison

CMP

IF (F0 condition F1) LB F2 (W) . . . . . WORD
IF (F0 condition F1) LB F2 (D) . . . . . DWORD

All models	○

Function: Comparison

This macro command is used to compare the data with signs specified in [F0] and [F1] and to execute a jump to the label in [F2] if the comparison satisfies the condition.

Conditions

Symbol	Contents
==	Equal
!=	Different
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			○
F1	◎			○
F2				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2	0 - 127	0 - 127

Example

- IF ($u100 == 500) LB 0 (W)
  RET
  LB0
  :

  If $u100 = 500, a jump to LB0 (label 0) takes place and then macro execution proceeds to the next line.
  If $u100 ≠ 500, macro execution proceeds to the next line. In this example, RET terminates the macro.

Supplemental remarks

- A label (LB) must be specified as the jump target. If no label exists, “Error: 83” (no destination label for the jump) occurs as a result of an error check on MONITOUCH.
- The result of macro execution is stored in $s1058.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

TST

IF condition (F0 & F1) LB F2 (W) . . . . . WORD
IF condition (F0 & F1) LB F2 (D). . . . . DWORD

All models	○

Function: Comparison with 0

This macro command is used to compare the result of [F0] ANDed with [F1] with “0”, and to execute a jump to the label specified in [F2] if the comparison satisfies the condition.

Conditions

Conditions	Contents
ZERO	0
NON ZERO	Other than 0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			○
F1	◎			○
F2				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD
F0	0000 - FFFF (HEX)	00000000 - FFFFFFFF (HEX)
F1		
F2	0 - 127	0 - 127

Example

- IFNZ ($u100 & 8000H) LB0 (W)
  RET
  LB0
  :

  If bit 15 at $u100 is set (ON), a jump to LB0 (label 0) takes place and then macro execution proceeds to the next line.
  If bit 15 at $u100 is reset (OFF), macro execution proceeds to the next line. In this example, RET terminates the macro.

Supplemental remarks

- A label (LB) must be specified as the jump target. If no label exists, “Error: 83” (no destination label for the jump) occurs as a result of an error check on MONITOUCH.
- The result of macro execution is stored in $s1058.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

IF
ELSE
ENDIF

All models	○

IF (F0 (condition 1) F1) (W) . . . . . .WORD
IF (F0 (condition 1) F1) (D). . . . . .DWORD
IF (condition 2) (F0) (B) . . . . . .BIT
(1)
ELSE
(2)
ENDIF

Function: Conditional branch

For WORD or DWORD, this macro command is used to compare [F0] and [F1], and to execute processing (1) if true, or (2) if false.
For BIT, [F0] and condition 2 is compared, and processing (1) is executed if true, or (2) if false.
Processing of “ELSE” and (2) can be omitted.

Condition 1

Symbol	Contents
==	Equal
!=	Different
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Condition 2

Symbol	Contents
ZERO	0
NON ZERO	Other than 0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	○
F1	◎	◎	◎	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	WORD	DWORD	BIT
F0	–32768 - +32767 (Decimal system with signs)	–2147483648 - +2147483647 (Decimal system with signs)	0, 1
F1			-

Example

- For WORD comparison
  IF ($u100 < 10) (W)
  $u100 = $u100 + 1 (W)
  ELSE
  $u100 = 0 (W)
  ENDIF
  “$u100 = $u100 + 1” is executed when $u100 is smaller than 10. When $u100 is 10 or more, “$u100 = 0” is executed.

- For BIT comparison
  IFNZ ($u100-00) (B)
  $u100 = $u100 + 1 (W)
  ELSE
  $u100 = 0 (W)
  ENDIF
  “$u100 = $u100 + 1” is executed when $u100-00 is set (ON). When $u100-00 is reset (OFF), “$u100 = 0” is executed.

Restrictions

- IF-ELSE-ENDIF commands can be nested up to 8 levels.

Supplemental remarks

- An error occurs to the macro editor when any of the following conditions is met.
  1. When IF-ELSE-ENDIF commands are nested beyond 8 levels;

Ex.:
```
IF ($u100 > 0)
IF ($u100 < 10)
      :
IF ($u200 == 1)
ENDIF
```
✖ There are 9 or more IF commands between IF-ENDIF commands.

  2. When the number of IF commands is not the same as the one of ENDIF commands;

Ex.:
```
IF ($u100 == 0)
IF ($u100 == 0)
ENDIF
```
✖ There are two IF commands while there is one ENDIF command.

  3. When the number of IF commands is not the same as the one of ELSE commands;

Ex.:
```
IF ($u100 == 0)
ELSE
ELSE
ENDIF
```
✖ There is one IF command while there are two ELSE commands.

  4. When FOR and NEXT commands are specified in a series of IF-ELSE-ENDIF commands.

Ex.:
```
IF ($u100 == 0)
FOR 10
ELSE
ENDIF
NEXT
```
✖ Only ELSE and ENDIF commands are specified between FOR and NEXT commands.

- The result of macro execution is stored in $s1059.

Code (DEC)	Contents
0*1	Normal
−1	Execution error*2

*1 Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

*2 When reading from [F0] and [F1] ends in failure, an error occurs and “−1” is stored in $s1059.
When an execution error occurs, it is regarded as a fault.

4.10 Macro Operation Control

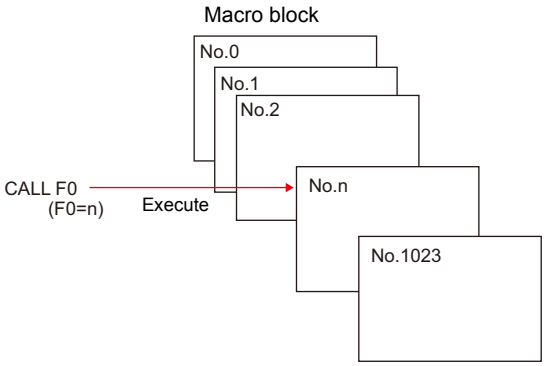
CALL

All models	○

CALL F0

Function: Macro block number designation

This macro command is used to execute the macro block specified in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

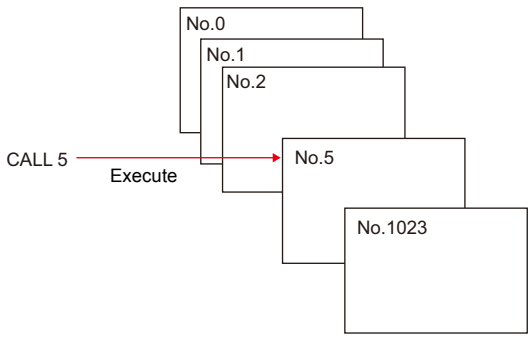
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 1023

Example

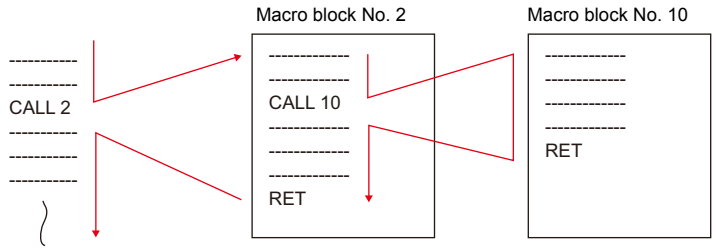
- CALL 5



Supplemental remarks

- If the macro block number called by CALL is not registered, an error check triggers a warning.

- The macro command can be nested up to 8 levels.
  Ex.) 2 levels



- The result of macro execution is stored in $s1059.

Code (DEC)	Contents
0*	Normal
−1	End in error (9 or more levels of macro commands are nested, an attempt is made to execute macro commands of 160001 lines or more, etc.)

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

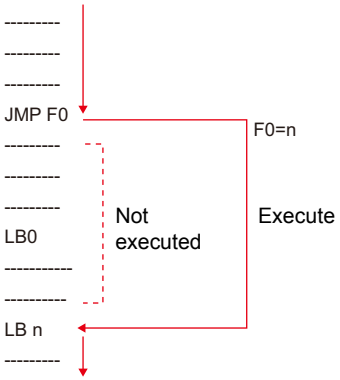
JMP

All models	○

JMP LB F0

Function: Unconditional jump

This macro command is used to execute a jump to the label specified in [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0				○

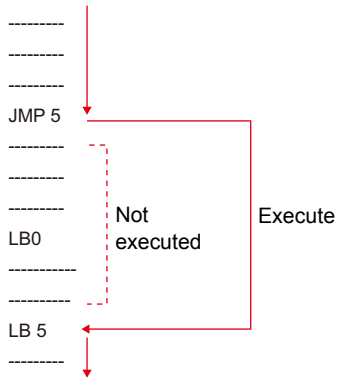
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 127

Example

- JMP LB5



Supplemental remarks

- A label (LB) must be specified as the jump target. If no label exists, error 83 (there is no destination label for the jump) will be detected by error check on MONITOUCH.
- The result of macro execution is stored in $s1059.

Code (DEC)	Contents
0*	Normal
–1	End in error (number of executed macro lines of 160001 or greater, etc.)

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

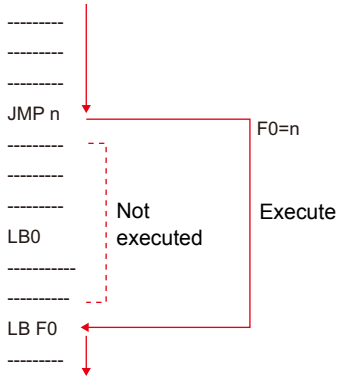
LABEL

All models	○

LB F0:

Function: Label number

This macro command is used to create jump target labels for CMP, TST, and JMP.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 127

Supplemental remarks

- A label (LB) must be specified as the jump target. If no label exists, error 83 (there is no destination label for the jump) will be detected by error check on MONITOUCH.
- The result of macro execution is stored in $s1059.

Code (DEC)	Contents
0*	Normal
−1	End in error (number of executed macro lines of 160001 or greater, etc.)

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

FOR/NEXT

All models	○

FOR F0
NEXT

Function: FOR - NEXT

This macro command is used to execute a loop between FOR and NEXT the number of times specified in [F0].

```
FOR F0
    $u300 = $u300+5   ← The loop executes the number
NEXT                    of times specified in F0.
```

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 65535

Example

```
$u300 = 0 (W)
$u301 = 0 (W)
FOR 3
    $u300 = $u300 + 1 (W)

    FOR $u400
        $u301 = $u301 + 5 (W)
    NEXT

NEXT
```

If $u400 = 5, the loop is executed 5 times.

The loop is executed 3 times.

- Result
  $u300 = 3
  $u301 = 75

Supplemental remarks

- Loop between FOR and NEXT can be nested* up to 8 levels. Nesting beyond 8 levels triggers error 81 (macro: FOR-NEXT command number is wrong) as a result of error check on MONITOUCH.
  * Nesting means incorporating a FOR-NEXT loop into a loop of the same kind.

```
:
FOR 2
:
 FOR 3
  :
 NEXT
 :
NEXT
:
```

- The result of macro execution is stored in $s1059.

Code (DEC)	Contents
0*	Normal
−1	End in error (nesting of 9 or more levels / number of executed macro lines of 160001 or greater, etc.)

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

RET

All models	○

RET

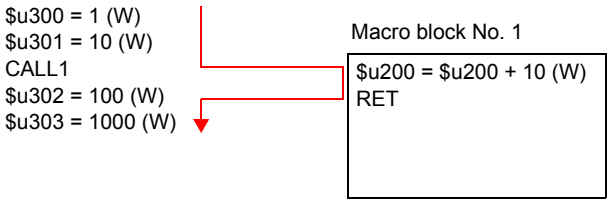
Function: Macro finish

This macro command is used to finish a macro. Any lines after RET are not executed.

```
$u300 = 1 (W)       ← Execute
$u301 = 10 (W)      ← Execute
RET                 ← Finish
$u302 = 100 (W)     ← Not executed
$u303 = 1000 (W)    ← Not executed
```

Supplemental remarks

- In the case of a macro block called by CALL, RET executes a return to the original sequence.



SWRET

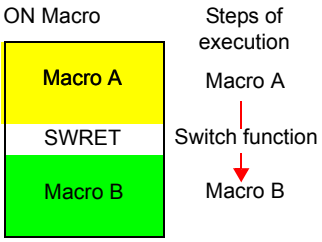
SWRET

All models	○

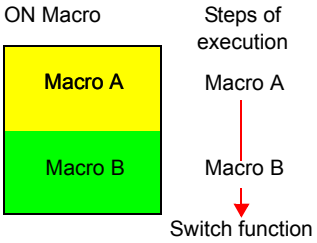
Function: Execute switch function

This macro command is used in a switch ON macro.

- With SWRET:
  Processing takes place in the order of the interruption of the macro, the execution of the switch function, and the execution of the remaining program of the macro.

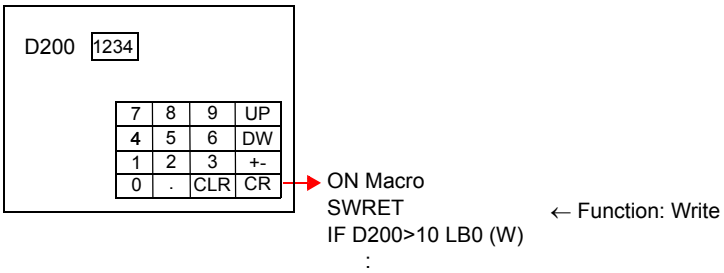


- Without SWRET:
  Processing takes place in the order of the execution of the switch ON macro and the execution of the switch function.



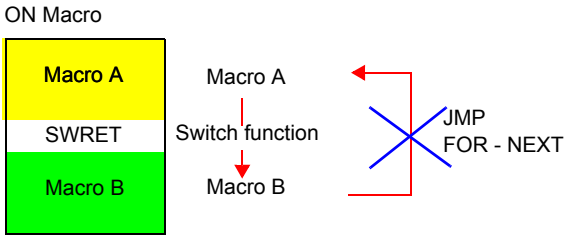
Example

- In a case where a macro runs based on the result written by the ENT key (in the entry mode) to the entry target D200, executing the switch function (for writing) by SWRET is required.



Supplemental remarks

- The macro command is valid in switch ON macros.
  The command, however, is not executed normally in the following cases:
  - SWRET exists in a macro block called by CALL.
  - JMP or FOR-NEXT triggers a movement to a label before the execution of SWRET.



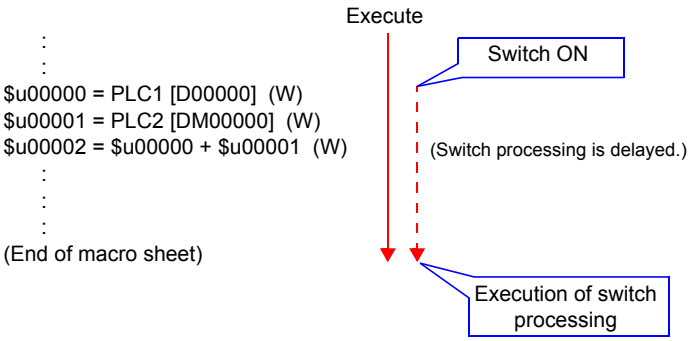
EN_INT

All models	○

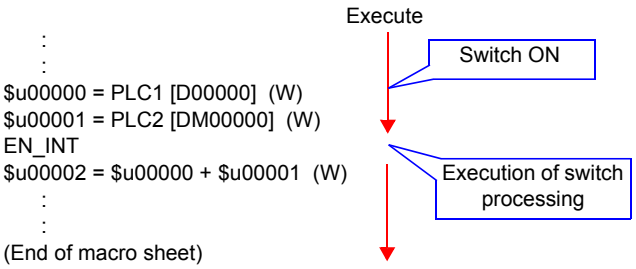
EN_INT

Function: Enabling interruption of switch processing

If a switch on the V series unit is pressed during the execution of macro processing, the switch processing is not executed immediately but is delayed until completion of the macro sheet.



When this command is executed while switch processing is pending, macro processing is suspended while the switch processing is executed. On completion of the switch processing macro execution is continued from the point of suspension.



Supplemental remarks

- If there is no switch processing pending, nothing happens in response to this command.

4.11 FROM Backup

In the FP-ROM (flash memory) for the V series screen program, its empty area can be used to back up the PLC device memory, internal device memory, and memory card. A maximum of 16k words can be allocated to the backup area.

FROM_WR

FROM_WR F0 F1

All models	○

Function: Write to FROM

This macro command is used to write the data of words starting from the address specified in [F0] to the FP-ROM. The number of the words is specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Address in each device memory
F1	1 - 16384 (= 16k words)

Supplemental remarks

- When using the macro command, go to the [General Settings] tab window in the [Unit Setting] dialog ([System Setting] → [Unit Setting] → [General Settings]). On the window, check [☐ Use Internal Flash ROM as Back-up Area].
  * Checking this option reduces the available screen program capacity by 66 kbytes.
- Each FP-ROM allows 100,000 write operations.
  (Each execution of FROM_WR is counted as one time, regardless of the number of words.) It is thereby recommended that backup data be read after power-on and be written before power-off.
- Do not execute FROM_WR in every cycle using a CYCLE macro, etc.
- Writing to FP-ROM takes three to five seconds.
- The result of macro execution is stored in $s728.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

FROM_RD

All models	○

FROM_RD F0 F1

Function: Read from FROM

This macro command is used to read the data of words from the FP-ROM into the address specified in [F0]. The number of the words is specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Address in each device memory
F1	1 - 16384 (= 16k words)

Supplemental remarks

- When using the macro command, go to the [General Settings] tab window in the [Unit Setting] dialog ([System Setting] → [Unit Setting] → [General Settings]). On the window, check [☐ Use Internal Flash ROM as Back-up Area].
  * Checking this option reduces the available screen program capacity by 66 kbytes.
- Do not execute FROM_RD in every cycle using a CYCLE macro, etc.
- The result of macro execution is stored in $s728.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.12 PLC

PLC_CLND

PLC_CLND F0 PLC F1 F2 F3

All models	○

Function: Calendar control function for PLC [F1]

This macro command is used to control the calendar for the PLC specified in [F1]. Depending on the value specified in [F0] it specifies reading or writing of the calendar data.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○
F1	○			○
F2	○			
F3	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	0: Calendar reading *1 1: Calendar writing (specified by user) *2 2: Calendar writing (by the system) *3	
F1	2 - 8: PLC number	
F2	0 - 31: PLC station number	Invalid with 1:1 connections
F2+1	0 - 255: PLC sub station number	Invalid with 1:1 connections Only valid for PLCs with sub station number designations
F3	0 - : Year (4-digit/2-digit)	
F3+1	1 - 12: Month	
F3+2	1 - 31: Day	
F3+3	0 - 23: Hour	
F3+4	0 - 59: Minute	
F3+5	0 - 59: Second	
F3+6	0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday	Only valid with a read ([F0] = 0) setting Invalid with a write ([F0] = 1 or 2) setting because the calculation is done internally in the unit

* Details of calendar function specification

*1 When [F0] = 0: Calendar reading
When the connection method specified in [F1] is “1:1”, the calendar is read for the connected device and the information is saved in the [F3] device memory. (The contents in the [F2] device memory are ignored.)
When the connection method specified in [F1] is “1:n”, the calendar for the connected device with the station number specified in [F2] or the sub station number specified in [F2+1] is read and saved in the [F3] device memory.
The V series system calendar is not changed by any command. To change the system calendar, use “ SYS (SET_SYS_CLND) F1” (page 4-213).

*2 When [F0] = 1: Calendar reading (specified by user)
When the connection method specified in [F1] is “1:1”, the calendar data in the [F3] device memory is written to the connected device.
(The contents in the [F2] device memory are ignored.)
When the connection method specified in [F1] is “1:n”, the calendar data specified in [F3] is written to the connected device with the station number specified in [F2] or the sub station number specified in [F2+1].

*3 When [F0] = 2: Calendar reading (by the system)
When the connection method specified in [F1] is “1:1”, the V series unit’s system calendar data is written to the connected device.
(The contents in the [F2] device memory and the [F3] device memory are ignored.)
When the connection method specified in [F1] is “1:n”, the system’s calendar data specified in [F3] is written to the connected device with the station number specified in [F2] or the sub station number specified in [F2+1].
(The contents in the [F3] device memory are ignored.)

Example

- Setting the calendar for PLC2, station No. 1 to 20:00:00 on October 15, 2007

```
$u100 = 1 (W)        — [PLC station number: 1]
$u200 = 2007 (W)   ┐
$u201 = 10 (W)     │
$u202 = 15 (W)     │
$u203 = 20 (W)     │  [October 15, 2007, Monday, 20:00:00]
$u204 = 0 (W)      │
$u205 = 0 (W)      ┘
PLC_CLND  1  PLC2  $u100  $u200
SYS (SET_SYS_CLND) $u200 (V series calendar setting)
```

Supplemental remarks

- If the relevant equipment doesn’t incorporate a calendar, nothing happens in response to the command. (The V series automatically judges whether or not the equipment incorporates a calendar.)
- Nothing happens to the equipment whose link has been dead in response to the command.
- The result of macro execution is stored in $s729.

Code (HEX)	Contents
0*	Normal
2004	A PLC [F1] communication error has occurred during processing.
FFFF	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

PLC_CTL

PLC_CTL PLC F0 F1 F2

All models	○

Function: PLC [F1] control function

This macro command is used to control the operation specified in the words starting from the address in [F1] in relation to the PLC specified in [F0]. The number of words is specified in [F2].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○
F1	○			
F2				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	1 - 8: PLC number
F1	0 - 31: PLC station number
F1+1	Command and others The items to be set differ depending on the equipment. For more information, refer to the V9 Series Connection Manual.
:	
F2	The number of words to be transferred

Example

- Bringing Omron's E5ZN (station No. 1) connected to the PLC2 to a state of RUN:
  $u100 = 1 (W) [PLC station number]
  $u101 = 30H (W) [Command]
  $u102 = 100H (W) [Operation command (RUN)]
  PLC_CTL PLC2 $u100 3

Contents	F0	F1 (=$u n) =$u100		F2
Operation command	1 - 8 (PLC1 - 8)	n =$u100	Station number*	3
		n+1 =$u101	Command: 0030H	
		n+2 =$u102	0000H: Communication writing OFF (disabled) 0001H: Communication writing ON (enabled)	
			0100H: RUN 0101H: STOP	
			0200H: Multi-SP (Set point 0) 0201H: Multi-SP (Set point 1) 0202H: Multi-SP (Set point 2) 0203H: Multi-SP (Set point 3)	
			0300H: AT cancel 0301H: AT execution	
			0400H: Write mode (Backup) 0401H: Write mode (RAM)	
			0500H: Save RAM data	
			0600H: Software reset	
			0700H: Move to set area 1	
			0800H: Move to protect level	

* 8000 (HEX): broadcasting

- The result of macro execution is stored in $s729.

Code (HEX)	Contents
0*	Normal
2002	Memory cannot be allocated.
2004	A PLC [F0] communication error has occurred during processing.

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

TBL_READ

TBL_READ F0 <- TABLE:PLC F1 : F2

All models	○

Function: Read from device memory map

This macro command is used to transfer the data at the addresses registered in the device memory map specified in [F2] of the PLC specified in [F1] to the addresses starting with the one specified in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	○			○
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Top address of the target
F1	1 - 8: PLC number
F2	0 - 31: Device memory map No.

Example

- Transferring the data of the addresses registered in device memory map No. 5 defined at PLC3 to $u500 onward
  TBL_READ $u500 <- TABLE : PLC3 : 5

Supplemental remarks

- As many addresses as the data count set in the device memory map must be allocated to the target memory, to which data will be transferred.
- The result of macro execution is stored in $s729.

Code (HEX)	Contents
0*	Normal
2001	The address set in the device memory map does not exist.
2002	The device memory cannot be allocated.
2004	A PLC [F1] communication error has occurred during processing.

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

TBL_WRITE

TBL_WRITE TABLE:PLC F1 : F0 <- F2

All models	○

Function: Write to device memory map

This macro command is used to transfer the data at the location starting from the address specified in [F2] to the address registered in the device memory map [F0] for the PLC [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○
F1	○			○
F2	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 31: Device memory map No.
F1	1 - 8: PLC number
F2	Top memory address of the source

Example

- Transferring the data of $u500 onward to the addresses registered in device memory map No. 5 defined at PLC3
  TBL_WRITE TABLE : PLC3 : 5 <- $u00500

Supplemental remarks

- As many addresses as the data count set in the device memory map must be allocated to the target memory, to which data will be transferred.
- The result of macro execution is stored in $s729.

Code (HEX)	Contents
0*	Normal
2001	The address set in the device memory map does not exist.
2002	The device memory cannot be allocated.
2004	A PLC [F1] communication error has occurred during processing.

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.13 Ethernet

SEND

SEND F0 C:F1 TO F2

All models	○

Function: Transfer to server

This macro command is used to transfer the data of words starting from the address specified in [F0] to the server of the network table number in [F2]. The number of the words is specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎			○
F2	◎			○

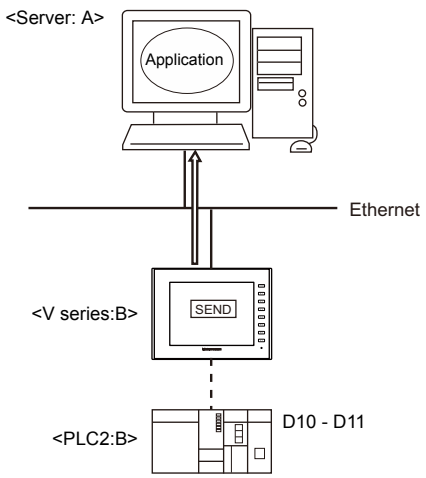
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Top address of the source
F1	0 - 2000: The number of words to be transferred
F2	0 - 255: Transfer target (network table number)

Example

- SEND PLC2 [D10] C:2 TO:3
  The above program transfers two words of data starting from D10 of PLC2:B to network table No. 3 (server A).



Supplemental remarks

The following system devices are related to this command. For more information, refer to the V9 Series Connection Manual.

Address	Contents	Remarks
$s514	The macro execution format (wait request) is set.	→V
$s515	The result of macro execution is stored.	←V

EREAD

EREAD F0 = F1 C:F2 F3

All models	○

Function: Read on the network

This macro command is used to read the data of words starting from the address specified in [F1] set in the [F3]-specified network table into the address in [F0]. The number of the words is specified in [F2].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	◎			○
F3	◎			○

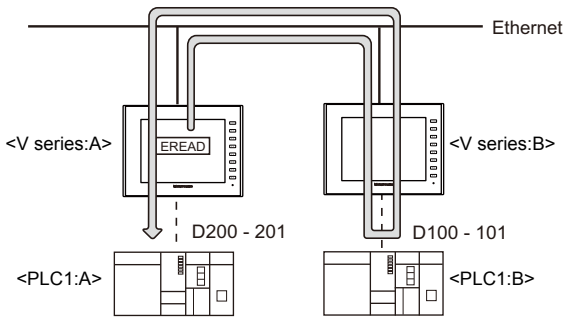
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Top address of the target
F1	Top address of the source
F2	0 - 2000: The number of words to be transferred
F3	0 - 255: Transfer source (network table number)

Example

- EREAD PLC1 [D200] = PLC1 [D100] C:2 5
  The above program reads two words of data starting from D100 of PLC2:B, which is connected to network table No. 5 (V series:B), into D200 onward of PLC1:A.



Supplemental remarks

The following system devices are related to this command. For more information, refer to the V9 Series Connection Manual.

Address	Contents	Remarks
$s514	The macro execution format (wait request) is set.	→V
$s515	The result of macro execution is stored.	←V

EWRITE

EWRITE F0 F1 = F2 C:F3

All models	○

Function: Write on the network

This macro command is used to write data starting from the address specified in [F2] to the address specified in [F0] of the equipment connected to the network table number specified in [F1]. The number of words is specified in [F3].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎			○
F2	◎	◎	◎	
F3	◎			○

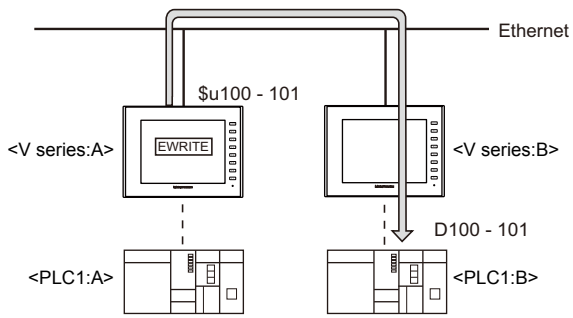
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Top address of the target
F1	0 - 255: Transfer target (network table number)
F2	Top address of the source
F3	0 - 2000: The number of words to be transferred

Example

- EWRITE PLC1 [D100] 5 = $u100 C:2
  The above program writes two words of data starting from $u100 of the V series:A to D100 onward of PLC2:B which is connected to network table No. 5 (V series:B).



Supplemental remarks

The following system devices are related to this command. For more information, refer to the V9 Series Connection Manual.

Address	Contents	Remarks
$s514	The macro execution format (wait request) is set.	→V
$s515	The result of macro execution is stored.	←V

4.14 Storage (Recipe)

LD_RECIPE

All models	○

LD_RECIPE F0 F1

Function: Read CSV file

This macro command is used to transfer the CSV file specified in [F1] to the location starting from the address in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	○	○	○	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Transfer target address
F1	0000 - 9999: CSV file number

CSV file

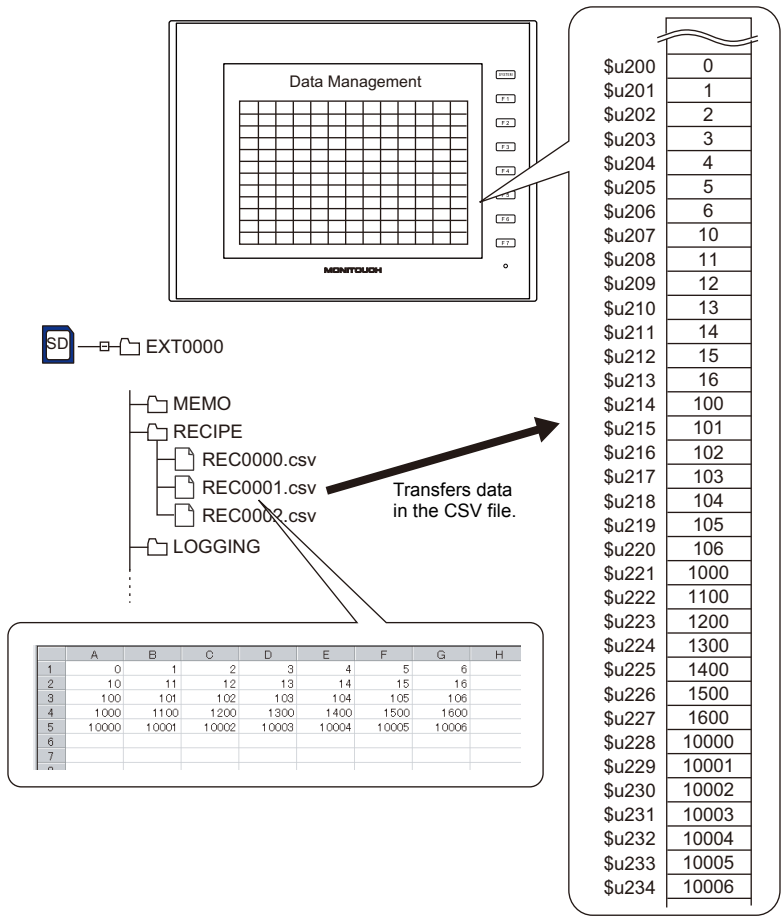
Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File No.

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in the CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

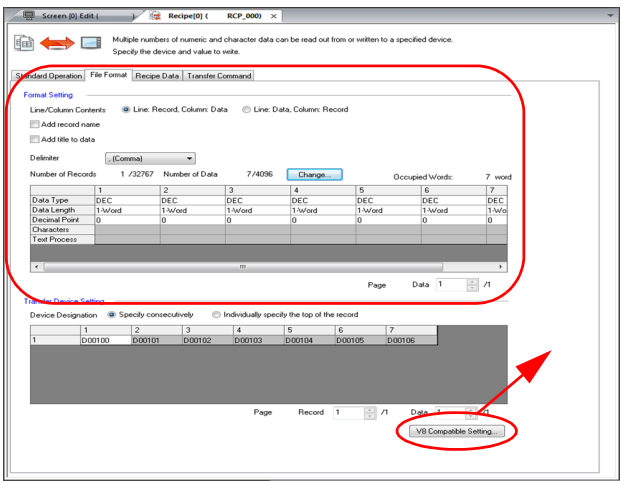
Example

- LD_RECIPE $u200 1
  The data in the REC0001.csv file is transferred to the location starting from $u200.

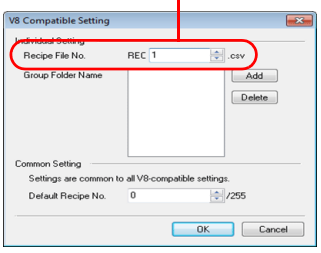


Supplemental remarks

- Recipe settings are required for each CSV file.



The file “REC0001.csv” is used.



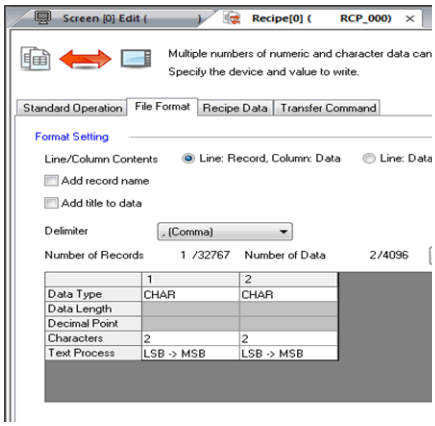
- For reading text, whether to convert a null to 20H (space) or read it as “00” can be selected.
  Go to the [General Setting] tab window in the [Unit Setting] dialog ([System Setting] → [Unit Setting] → [General Settings]). On the tab window, check or uncheck [☐ Convert NULL to Space with the LD/RD Macro].

  Example:

  CSV file

  ```
  A,B,
  C,,
  ```

  Format setting



  Execution result

Storage target	Checked	Unchecked
n	2041H	0041H
n+1	2042H	0042H
n+2	2043H	0043H
n+3	2020H	0000H
	A null is converted to 20H.	A null remains “00”.

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

LD_RECIPE2

All models	○

LD_RECIPE2 F0 F1 F2

Function: Read CSV file (recipe number designation)

This macro command is used to transfer the CSV file number [F1] in the format of the recipe number [F2] to the location starting from the address [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	○	○	○	○
F2	○	○	○	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Transfer target address
F1	0000 - 9999: CSV file number
F2	0 - 255: Recipe number

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv

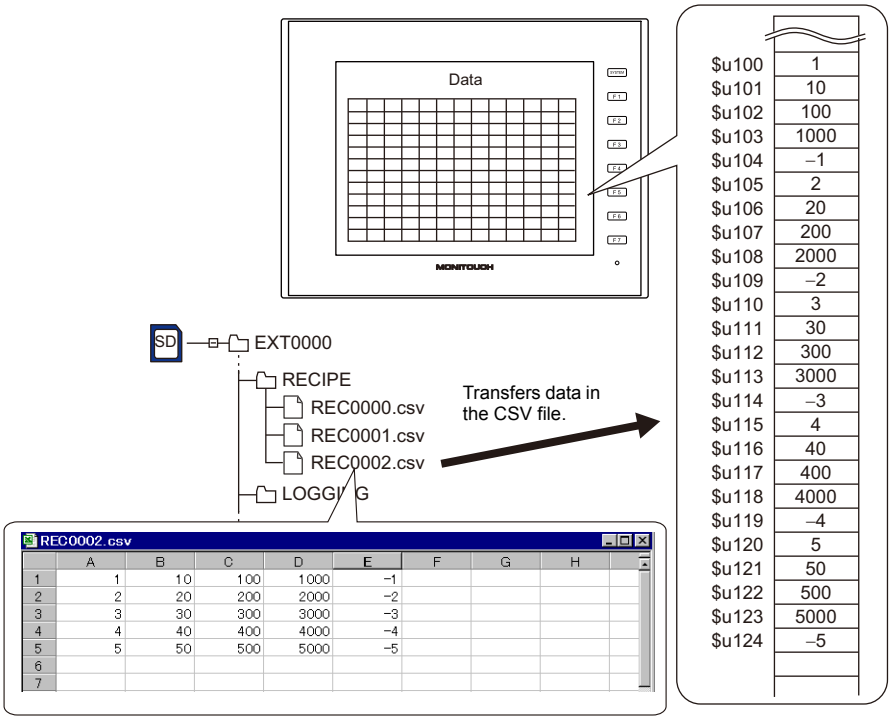
0000 - 9999: File No.

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

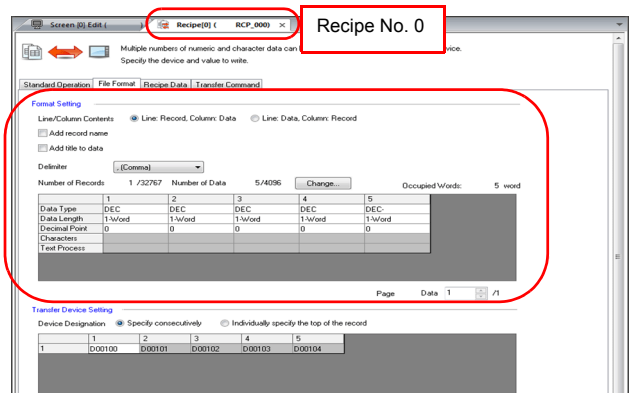
Example

- LD_RECIPE2 $u100 2 0
  The above program transfers the data in the file “REC0002.csv” in the format of recipe No. 0 to the location starting from $u100.



Supplemental remarks

- Recipe settings must be made in the same format as the CSV file.



- For reading text, whether to convert a null to 20H (space) or read it as “00” can be selected. For more information, refer to page 4-99.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

LD_RECIPESEL

All models	○

LD_RECIPESEL F0 F1

Function: Read CSV file (in units of a cell)

This macro command is used to transfer part of the CSV file specified in [F1] to the location starting from the address in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer source address	
F1	0000 - 9999: CSV file number	
F1+1	1 - 32767: Top line number	1 - 4096: Top line number
F1+2	0* - 4096: Top column number	0* - 4096: Top column number
F1+3	1 - 32767: Number of lines	1 - 4096: Number of lines
F1+4	1 - 4096: Number of columns	1 - 4096: Number of columns

* Specify “0” if you wish to transfer the record name as well. In that case, select [Record Name + Data] for [Transfer Target] under [Transfer Device Setting] ([Recipe] → [File Format]) The number of columns specified in F1+4 includes the cell of the record name.

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File No.

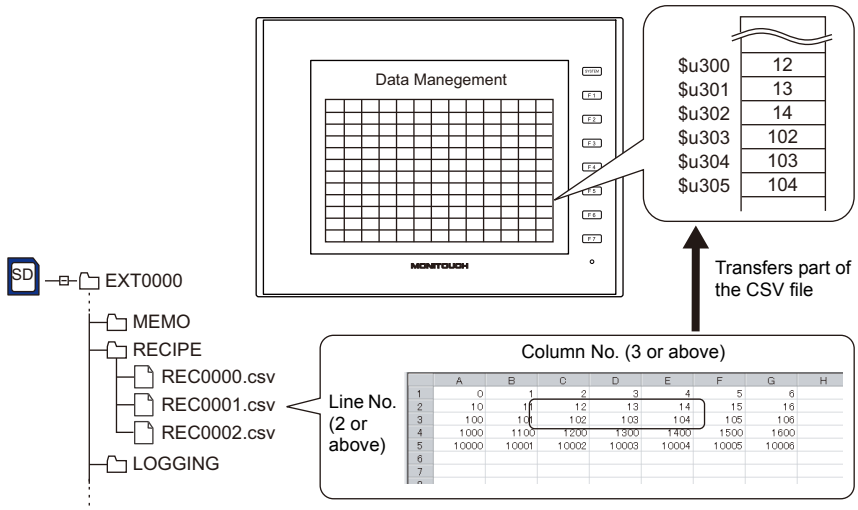
The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

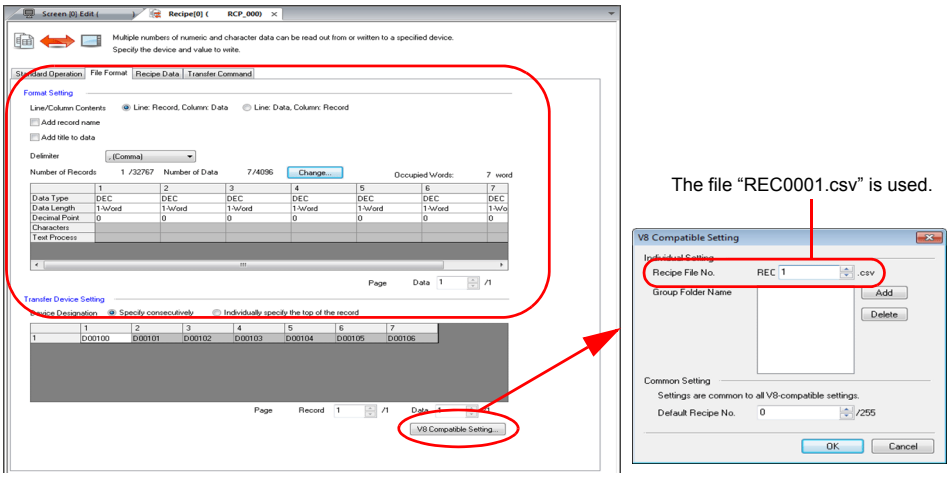
- $u100 = 1 (W) [File number 1]
  $u101 = 2 (W) [Top line number]
  $u102 = 3 (W) [Top column number]
  $u103 = 2 (W) [Number of lines]
  $u104 = 3 (W) [Number of columns]
  LD_RECIPESEL $u300 $u100

  The above program transfers part of the data in the REC0001.csv file to the location starting from $u300.



Supplemental remarks

- Attribute setting is required for each CSV file.



- For reading text, whether to convert a null to 20H (space) or read it as “00” can be selected. For more information, refer to page 4-99.

- Difference between reading one line and reading multiple lines

<table>
<tr><th></th><th>Line: Record, Column: Data</th><th>Line: Data, Column: Record</th></tr>
<tr><td>CSV</td><td>CSV file
<table><tr><th>DEC</th><th>CHAR</th><th>DEC</th></tr><tr><td>1</td><td>A</td><td>100</td></tr><tr><td>2</td><td>B</td><td>200</td></tr><tr><td>3</td><td>C</td><td>300</td></tr><tr><td>4</td><td>D</td><td>400</td></tr></table></td><td>CSV file
<table><tr><td>DEC</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>CHAR</td><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>DEC</td><td>100</td><td>200</td><td>300</td><td>400</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>One line</td><td>Reading one line and two columns from top line No. 2 and top column No.2
<table><tr><th>DEC</th><th>CHAR</th><th>DEC</th></tr><tr><td>1</td><td>A</td><td>100</td></tr><tr><td>2</td><td>B</td><td>200</td></tr><tr><td>3</td><td>C</td><td>300</td></tr><tr><td>4</td><td>D</td><td>400</td></tr></table></td><td>Reading one line and two columns from top line No. 2 and top column No.2
<table><tr><td>DEC</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>CHAR</td><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>DEC</td><td>100</td><td>200</td><td>300</td><td>400</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>Two lines</td><td>Reading two lines and two columns from top line No. 2 and top column No. 2
<table><tr><th>DEC</th><th>CHAR</th><th>DEC</th></tr><tr><td>1</td><td>A</td><td>100</td></tr><tr><td>2</td><td>B</td><td>200</td></tr><tr><td>3</td><td>C</td><td>300</td></tr><tr><td>4</td><td>D</td><td>400</td></tr></table></td><td>Reading two lines and two columns from top line No. 2 and top column No. 2
<table><tr><td>DEC</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>CHAR</td><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>DEC</td><td>100</td><td>200</td><td>300</td><td>400</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>* Execute reading line by line (specifying multiple lines at one time is not allowed).</td></tr>
</table>

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

LD_RECIPESEL2 LD_RECIPESEL2 F0 F1 F2

All models	○

Function: Read CSV file (in units of a cell/recipe No. designation)

This macro command is used to transfer a part of data in the CSV file number [F1] in the format of the recipe number [F2] to the location starting from the address [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	○	○	○	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer target address	
F1	0000 - 9999: CSV file number	
F1+1	1 - 32767: Top line number	1 - 4096: Top line number
F1+2	0* - 4096: Top column number	0* - 4096: Top column number
F1+3	1 - 32767: Number of lines	1 - 4096: Number of lines
F1+4	1 - 4096: Number of columns	1 - 4096: Number of columns
F2	0 - 255: Recipe number	

* Specify “0” if you wish to transfer the record name as well. In that case, select [Record Name + Data] for [Transfer Target] under [Transfer Device Setting] ([Recipe] → [File Format]) The number of columns specified in F1+4 includes the cell of the record name.

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File number

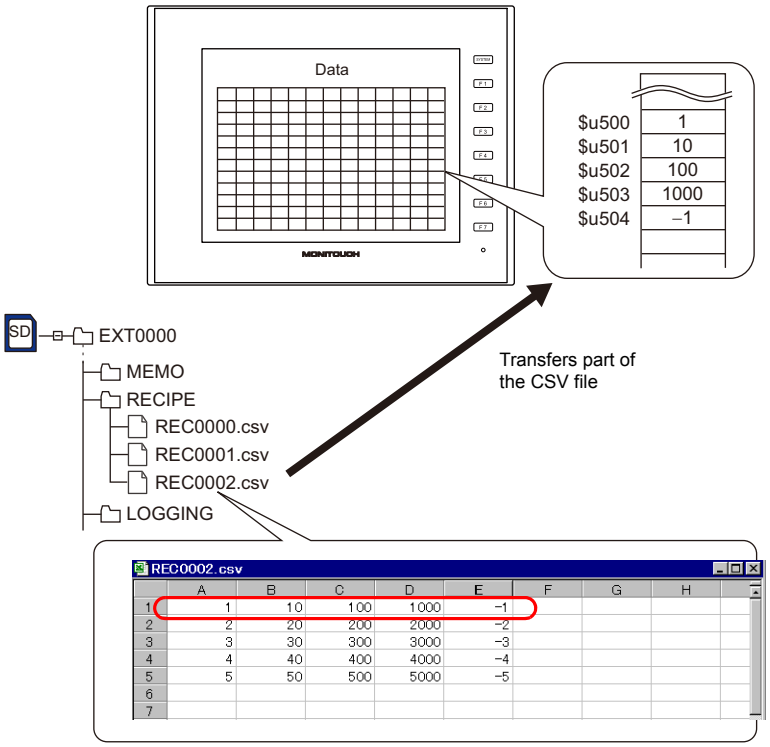
The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Title Record ♦

Example

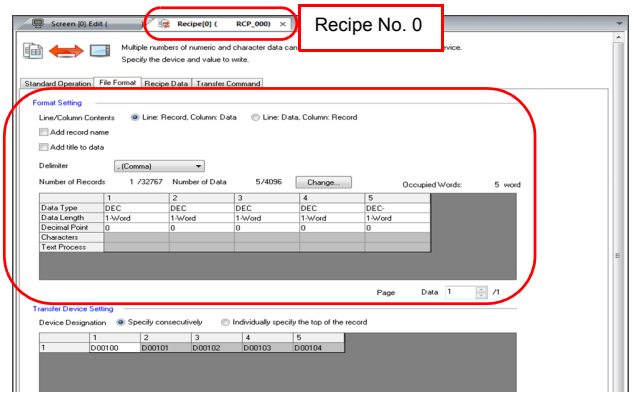
- $u100 = 2 (W) [File number]
  $u101 = 1 (W) [Top line number]
  $u102 = 1 (W) [Top column number]
  $u103 = 1 (W) [Number of lines]
  $u104 = 5 (W) [Number of columns]
  LD_RECIPESEL2 $u500 $u100 0

  The above program transfers a part of data in the file “REC0002.csv” in the format of recipe No. 0 to the location starting from $u500.



Supplemental remarks

- Recipe settings must be made in the same format as the CSV file.



- For reading text, whether to convert a null to 20H (space) or read it as “00” can be selected. For more information, refer to page 4-99.
- Difference between reading one line and reading multiple lines

	Line: Record, Column: Data	Line: Data, Column: Record
CSV	CSV file DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400	CSV file DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400
One line	Reading one line and two columns from top line No. 2 and top column No. 2 DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400	Reading one line and two columns from top line No. 2 and top column No. 2 DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400
Two lines	Reading two lines and two columns from top line No. 2 and top column No. 2 DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400	Reading two lines and two columns from top line No. 2 and top column No. 2 DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400 * Execute reading line by line (specifying multiple lines at one time is not allowed).

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SV_RECIPE

All models	○

SV_RECIPE F0 F1 F2

Function: Save to CSV file

This macro command is used to save the data of words starting from the address specified in [F0] to the CSV file in [F2]. The number of the words is specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	○	○	○	○
F2	○	○	○	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Transfer source address
F1	1 - 4096: Word count
F2	0000 - 9999: CSV file number

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File number

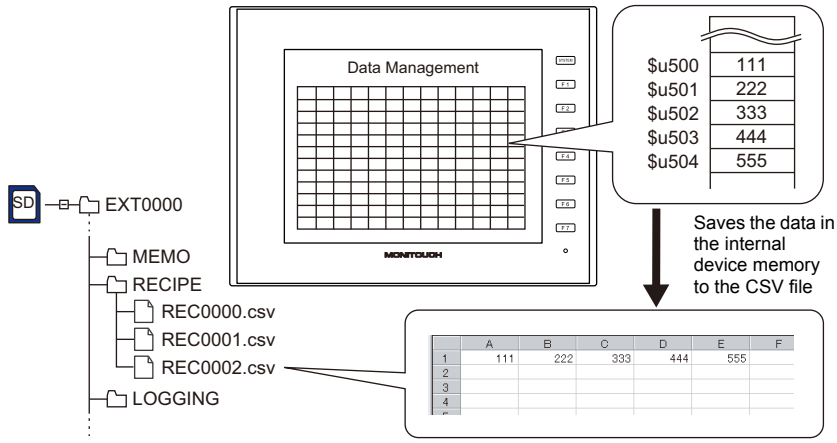
The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

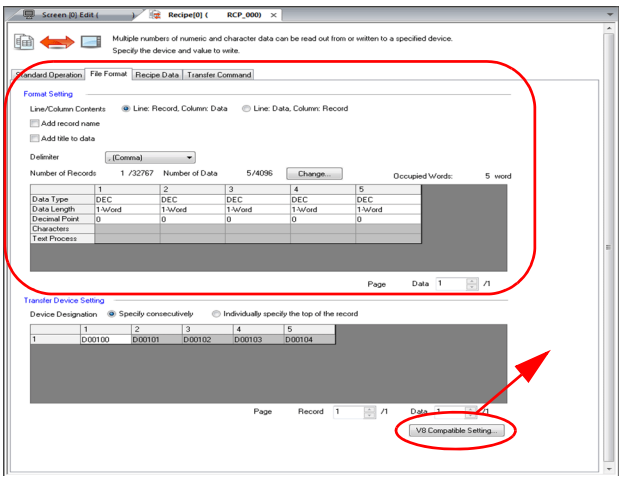
- SV_RECIPE $u500 5 2

  The above program saves the five-word data at $u500 - 504 to the REC0002.csv file.

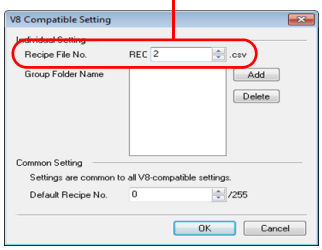


Supplemental remarks

- Recipe settings are required for each CSV file.



The file “REC0002.csv” is used.



- If the specified CSV file does not exist in the storage, a new file will be created. Creating the CSV file in advance is not necessary.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SV_RECIPE2

All models	○

SV_RECIPE2 F0 F1 F2 F3

Function: Save to CSV file (recipe No. designation)

This macro command is used to save the data of words specified in [F1] starting from the address [F0] to the CSV file number [F2] in the format of the recipe number [F3].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	○	○	○	○
F2	○	○	○	○
F3	○	○	○	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	Transfer source address	
F1	1 - 4096:	Word count
F2	0000 - 9999:	CSV file number
F3	0 - 255:	Recipe number

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File No.

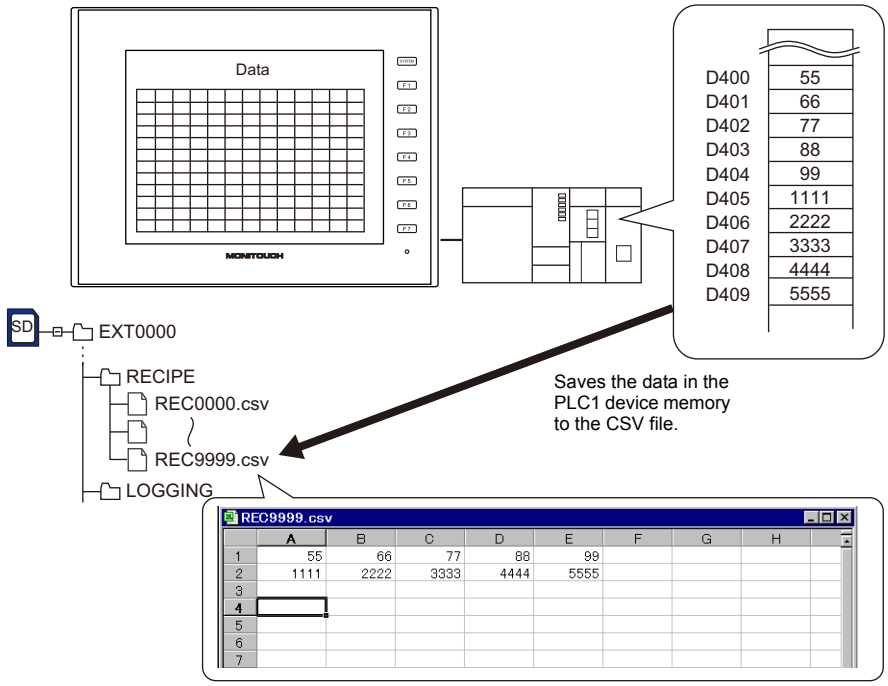
The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Title Record ♦

Example

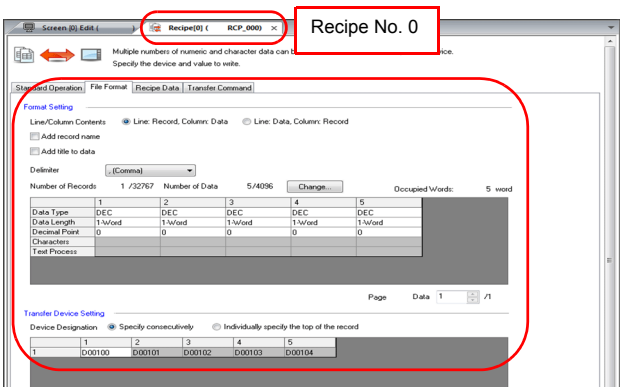
- SV_RECIPE2 PLC1 [D400] 10 9999 0

  The above program saves the ten-word data at D400 - 409 in PLC1 to the file “REC9999.csv” in the format of recipe No. 0.



Supplemental remarks

- Recipe settings must be made in the same format as the CSV file.



- If the specified CSV file does not exist in the storage, a new file will be created. Creating the CSV file in advance is not necessary.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SV_RECIPESEL

SV_RECIPESEL F0 F1

All models	○

Function: Save to CSV file

This macro command is used to save the data at the location starting from the address specified in [F0] to the specified line/column in the CSV file in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer source address	
F1	0000 - 9999: CSV file number	
F1+1	1 - 32767: Top line number	1 - 4096: Top line number
F1+2	0* - 4096: Top column number	0* - 4096: Top column number
F1+3	1 - 4096: Number of lines	1 - 4096: Number of lines
F1+4	1 - 4096: Number of columns	1 - 4096: Number of columns

* Specify “0” if you wish to transfer the record name as well. In that case, select [Record Name + Data] for [Transfer Target] under [Transfer Device Setting] ([Recipe] → [File Format]) The number of columns specified in F1+4 includes the cell of the record name.

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File No.

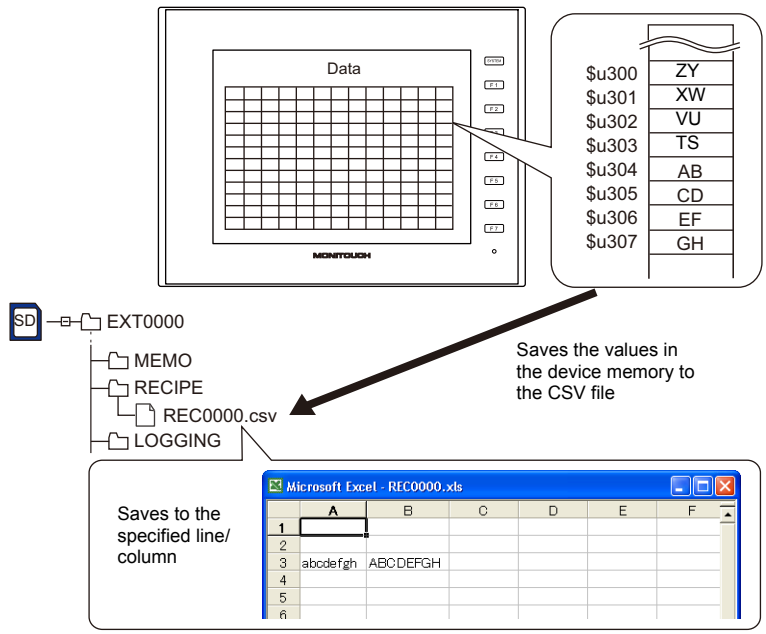
The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Title Record ♦

Example

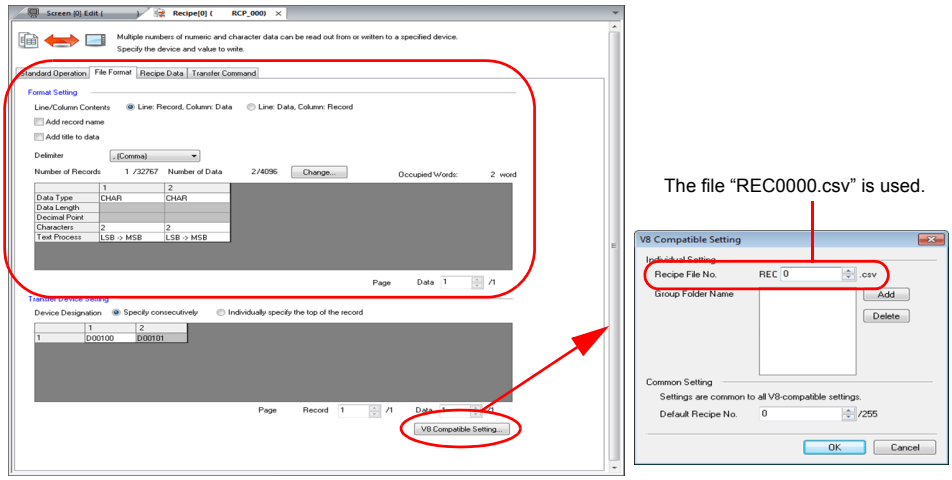
- $u100 = 0 (W) [File number]
  $u101 = 3 (W) [Top line number]
  $u102 = 1 (W) [Top column number]
  $u103 = 1 (W) [Number of lines]
  $u104 = 2 (W) [Number of columns]
  SV_RECIPESEL $u300 $u100

  The above program saves the data at the location starting from $u300 to line No. 3 in the REC0000.csv file.



Supplemental remarks

- Recipe settings are required for each CSV file.



- If the specified CSV file does not exist in the storage, a new file will be created. Creating the CSV file in advance is not necessary.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SV_RECIPESEL2 SV_RECIPESEL2 F0 F1 F2

All models	○

Function: Save to CSV file (recipe No. designation)

This macro command is used to save the data at the location starting from the address specified in [F0] in the format of the recipe number in [F2] to the specified line/column in the CSV file in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	○	○	○	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer source address	
F1	0000 - 9999: CSV file number	
F1+1	1 - 32767: Top line number	1 - 4096: Top line number
F1+2	0* - 4096: Top column number	0* - 4096: Top column number
F1+3	1 - 32767: Number of lines	1 - 4096: Number of lines
F1+4	1 - 4096: Number of columns	1 - 4096: Number of columns
F2	0 - 255: Recipe number	

* Specify “0” if you wish to transfer the record name as well. In that case, select [Record Name + Data] for [Transfer Target] under [Transfer Device Setting] ([Recipe] → [File Format]) The number of columns specified in F1+4 includes the cell of the record name.

CSV file

Storage target: \(access folder)\RECIPE
File name: \RECxxxx.csv
0000 - 9999: File No.

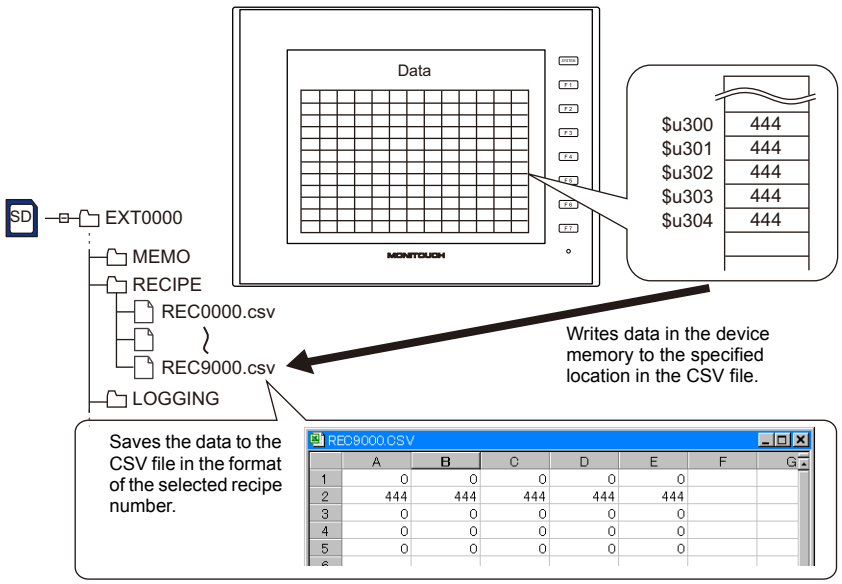
The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Title Record ♦

Example

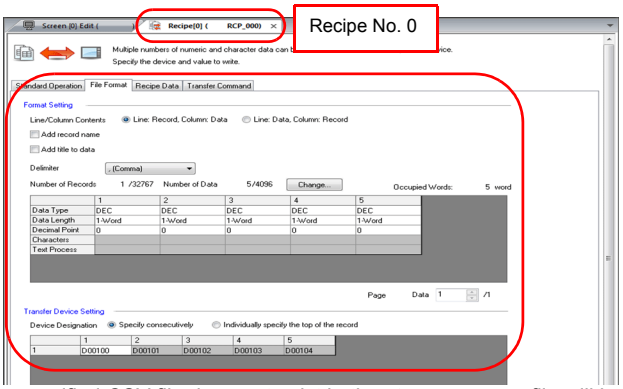
- $u100 = 9000 (W) [File number]
  $u101 = 2 (W) [Top line number]
  $u102 = 1 (W) [Top column number]
  $u103 = 1 (W) [Number of lines]
  $u104 = 5 (W) [Number of columns]
  SV_RECIPESEL2 $u300 $u100 0

  The above program saves the data at the location starting from $u300 in the format of recipe No. 3 to line No. 2 in the REC9000.csv file.



Supplemental remarks

- Recipe settings must be made in the same format as the CSV file.



- If the specified CSV file does not exist in the storage, a new file will be created. Creating the CSV file in advance is not necessary.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SET_ RECIPEFOLDER

All models	○

SET_RECIPEFOLDER F0

Function: Folder designation

This macro command is used to designate the folder storing CSV files in [F0].

Available device memory

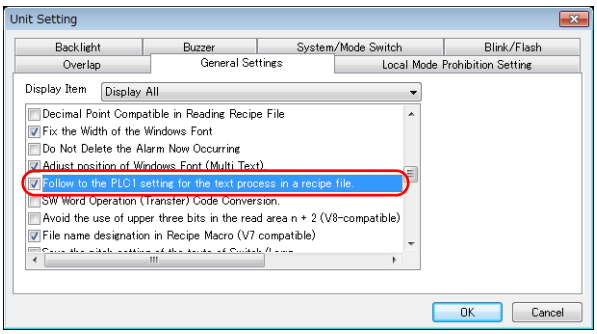
	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	ASCII code (8 one-byte upper-case alphanumeric characters): Access target folder name*
F0+1	
F0+2	
F0+3	

* Text processing (LSB → MSB or MSB → LSB) for the folder name is determined whether [Follow to the PLC1 setting for the text process in a recipe file.] on the [General Settings] tab window that is displayed by [System Setting] → [Unit Setting] is checked or not.



☑ Follow to the PLC1 setting for the text process in a recipe file.	☐ Follow to the PLC1 setting for the text process in a recipe file.
Text processing specified for the PLC1	Fixed to “LSB → MSB”

Example

- $u100 = 4154H (W)
  $u101 = 4752H (W)
  $u102 = 5445H (W)
  $u103 = 0000H (W)
  SET_RECIPEFOLDER $u100

  54 41 52 47 45 54 = TARGET (ASCII)

  The above program specifies the folder at \(access folder)\RECIPE\TARGET.

- The CHR or STRING macro command will simplify the designation of a folder if it is a fixed name.
  - (When text processing is performed according to the setting on the PLC1: use a “CHR” command.)
    $u100 = ’TARGET’
    SET_RECIPEFOLDER $u100
  - (When “LSB → MSB” is selected: use a “STRING” command.)
    $u100 = ‘TARGET’ (STRING)
    SET_RECIPEFOLDER $u100

Supplemental remarks

- Four consecutive words starting from the address in [F0] are used. Be sure that these words are not already used elsewhere.
- Once the macro command is executed, the effect is maintained until any of the following takes place.
  - Turning off the power
  - Switching the V series from a state of RUN to STOP (Local mode)
  - Removing the storage device

  Execute the macro command again after any of the above or if you access a CSV file in a different folder.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

RD_RECIPE_FILE RD_RECIPE_FILE F0 F1

All models	○

Function: Read CSV file

This macro command is used to transfer all data in the CSV file specified in [F1] to the address in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Transfer target
F1	ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name*
F1+1	
F1+2	
F1+3	

* For details on text processing of the file name, refer to “Supplemental remarks” on Page 4-121.

CSV file

Storage target: \(access folder)\RECIPE\(arbitrary folder)
File name: \xxxxxxxx.csv

8 one-byte upper-case alphanumeric characters or less

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

- $u100 = 'TARGET'
  SET_RECIPEFOLDER $u100
  $u110 = 5250H (W)
  $u111 = 444FH (W)
  $u112 = 4355H (W)
  $u113 = 3154H (W)
  RD_RECIPE_FILE PLC1 [D200] $u110

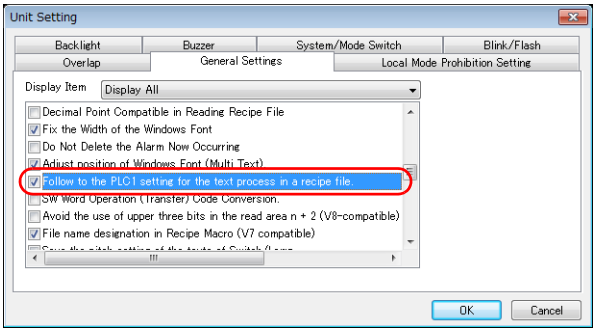
  Not required if SET_FOLDER has already been executed

  50 52 4F 44 55 43 54 31 = PRODUCT1 (ASCII)

  The above program transfers all data in the PRODUCT1.csv file stored in the TARGET folder to PLC1: D200.

Supplemental remarks

- Four consecutive words starting from the address in [F1] are used. Be sure that these words are not already used elsewhere.
- If the CSV file specified in [F1] does not exist, a storage read error occurs ($s497 = 16).
- For reading text, whether to convert a null to 20H (space) or read it as "00" can be selected. For more information, refer to page 4-99.
- Text processing (LSB → MSB or MSB → LSB) for the file name is determined whether [Follow to the PLC1 setting for the text process in a recipe file.] on the [General Settings] tab window that is displayed by [System Setting] → [Unit Setting] is checked or not.



Device memory	☑ Follow to the PLC1 setting for the text process in a recipe file.	☐ Follow to the PLC1 setting for the text process in a recipe file.
Internal device memory	Text processing specified for the PLC1	Fixed to "LSB → MSB"
PLC 1 - 8 device memory	Text processing specified for the PLC1	Text processing specified for each PLC

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

RD_RECIPE_LINE RD_RECIPE_LINE F0 F1 F2 F3

All models	○

Function: Read CSV file (line designation)

This macro command is used to transfer the data of specified lines in the [F1]-specified CSV file to the address in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	◎	◎	◎	○
F3	◎	◎	◎	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer target	
F1 F1+1 F1+2 F1+3	ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name*	
F2	1 - 32767: Top line	1 - 4096: Top line
F3	1 - 32767: Final line	1 - 4096: Final line

* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

CSV file

Storage target: \(access folder)\RECIPE\(arbitrary folder)
File name: \xxxxxxxx.csv

8 one-byte upper-case alphanumeric characters or less

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

- $u100 = 'TARGET'
  SET_RECIPEFOLDER $u100
  $u110 = 5250H (W)
  $u111 = 444FH (W)
  $u112 = 4355H (W)
  $u113 = 3154H (W)
  RD_RECIPE_LINE PLC1 [D200] $u110 3 3

  Not required if SET_FOLDER has already been executed

  50 52 4F 44 55 43 54 31 = PRODUCT1 (ASCII)

  The above program transfers line No. 3 (record No. 3) data in the PRODUCT1.csv file stored in the TARGET folder to PLC1: D200.

Supplemental remarks

- Four consecutive words starting from the address in [F1] are used. Be sure that these words are not already used elsewhere.
- If the CSV file specified in [F1] does not exist, a storage read error occurs ($s497 = 16).
- For reading text, whether to convert a null to 20H (space) or read it as "00" can be selected. For more information, refer to page 4-99.
- Difference between reading one line and reading multiple lines

	Line: Record, Column: Data	Line: Data, Column: Record
CSV	CSV file DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400	CSV file DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400
One line	Reading based on top line No. 2 and final line No. 2 DEC CHAR DEC 1 A 100 ~~2 B 200~~ 3 C 300 4 D 400	Reading based on top line No. 2 and final line No. 2 DEC 1 2 3 4 CHAR ~~A B C D~~ DEC 100 200 300 400
Two lines	Reading based on top line No. 2 and final line No. 3 DEC CHAR DEC 1 A 100 ~~2 B 200~~ ~~3 C 300~~ 4 D 400	Reading based on top line No. 2 and final line No. 3 DEC 1 2 3 4 CHAR ~~A B C D~~ DEC ~~100 200 300 400~~ * Execute reading line by line (specifying multiple lines at one time is not allowed).

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

RD_RECIPE_ COLUMN

All models	○

RD_RECIPE_COLUMN F0 F1 F2 F3

Function: Read CSV file (column designation)

This macro command is used to transfer the data of specified columns in the [F1]-specified CSV file to the address in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	◎	◎	◎	○
F3	◎	◎	◎	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer target	
F1	ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name*	
F1+1		
F2+2		
F1+3		
F2	0: Column of record name 1 - 4096: Top column of data	
F3	0: Column of record name 1 - 4096: Final column of data	

* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

CSV file

Storage target: \(access folder)\RECIPE\(arbitrary folder)
File name: \xxxxxxxx.csv

8 one-byte upper-case alphanumeric characters or less

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

- $u100 = 'TARGET'
  SET_RECIPEFOLDER $u100
  $u110 = 5250H (W)
  $u111 = 444FH (W)
  $u112 = 4355H (W)
  $u113 = 3154H (W)
  RD_RECIPE_COLUMN PLC1 [D300] $u110 5 5

  Not required if SET_FOLDER has already been executed

  50 52 4F 44 55 43 54 31 = PRODUCT1 (ASCII)

  The above program transfers column No. 5 data in the PRODUCT1.csv file stored in the TARGET folder to PLC1: D300.

Supplemental remarks

- Four consecutive words starting from the address in [F1] are used. Be sure that these words are not already used elsewhere.
- If the CSV file specified in [F1] does not exist, a storage read error occurs ($s497 = 16).
- For reading text, whether to convert a null to 20H (space) or read it as "00" can be selected. For more information, refer to page 4-99.
- Difference between reading one column and reading multiple columns

	Line: Record, Column: Data	Line: Data, Column: Record
CSV	CSV file DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400	CSV file DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400
One line	Reading based on top column No. 2 and final column No. 2 DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400	Reading based on top column No. 2 and final column No. 2 DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400
Two lines	Reading based on top column No. 2 and final column No. 3 DEC CHAR DEC 1 A 100 2 B 200 3 C 300 4 D 400 * Execute reading column by column (specifying multiple columns at one time is not allowed).	Reading based on top column No. 2 and final column No. 3 DEC 1 2 3 4 CHAR A B C D DEC 100 200 300 400

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

WR_RECIPE_FILE WR_RECIPE_FILE F0 F1

All models	○

Function: Save to CSV file

This macro command is used to save the data at the location starting from the address specified in [F0] to the CSV file in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Transfer source
F1	ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name*
F1+1	
F1+2	
F1+3	

* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

CSV file

Storage target: \(access folder)\RECIPE\(arbitrary folder)
File name: \xxxxxxxx.csv

8 one-byte upper-case alphanumeric characters or less

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Title Record ♦

Example

- $u100 = 'TARGET'
  SET_RECIPEFOLDER $u100
  $u110 = 5250H (W)
  $u111 = 444FH (W)
  $u112 = 4355H (W)
  $u113 = 3754H (W)
  WR_RECIPE_FILE PLC1 [D200] $u110

  Not required if SET_FOLDER has already been executed

  50 52 4F 44 55 43 54 37 = PRODUCT7 (ASCII)

  The above program overwrites the PRODUCT7.csv file stored in the TARGET folder with the data at the location starting from PLC1: D200.

Supplemental remarks

- Four consecutive words starting from the address in [F1] are used. Be sure that these words are not already used elsewhere.
- If the CSV file specified in [F1] does not exist, a storage read error occurs ($s497 = 16).
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

WR_RECIPE_LINE WR_RECIPE_LINE F0 F1 F2 F3

All models	○

Function: Save to CSV file (line designation)

This macro command is used to save the data at addresses starting from the one specified in [F0] in a specified line, or an additional final line, of the CSV file specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	◎	◎	◎	○
F3	◎	◎	◎	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer source	
F1	ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name *1	
F1+1		
F1+2		
F1+3		
F2	1 - 32767: Top line -1: Additional final line*2	1 - 4096: Top line
F3	1 - 32767: Final line -1: Additional final line*2	1 - 4096: Final line

*1 For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.
*2 An additional final line is only saved if “-1” is set for both F2 and F3.

CSV file

Storage target: \(access folder)\RECIPE\(arbitrary folder)
File name: \xxxxxxxx.csv
8 one-byte upper-case alphanumeric characters or less

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

- $u100 = 'TARGET'
  SET_RECIPEFOLDER $u100 ┐ Not required if SET_FOLDER has already been executed
  $u110 = 5250H (W)
  $u111 = 444FH (W)
  $u112 = 4355H (W)
  $u113 = 3754H (W) ┐ 50 52 4F 44 55 43 54 37 = PRODUCT7 (ASCII)
  WD_RECIPE_LINE PLC1 [D200] $u110 3 3

  The above program overwrites line No. 3 in the PRODUCT7.csv file stored in the TARGET folder with the data at the location starting from PLC1: D200.

Supplemental remarks

- Four consecutive words starting from the address in [F1] are used. Be sure that these words are not already used elsewhere.
- If the specified CSV file does not exist, specifying "1" or "−1" for [F2] creates a new file. If [F2] ≠ 1, a storage read error ($s497 = 16) occurs.
  However, when [Line: Data, Column: Record] is selected, use "WR_RECIPE_COLUM" to create a new file.
- When setting "−1" for [F2] and [F3] and adding an additional final line, make sure that the number of lines does not exceed 32767. The macro will not operate correctly on files with more than 32767 lines.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

WR_RECIPE_ COLUMN

All models	○

WR_RECIPE_COLUMN F0 F1 F2 F3

Function: Save to CSV file (column designation)

This macro command is used to save the data at the location starting from the address in [F0] to the specified column in the F1-specified CSV file.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	
F1	◎	◎	◎	
F2	◎	◎	◎	○
F3	◎	◎	◎	○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Line: Record, Column: Data	Line: Data, Column: Record
F0	Transfer source	
F1	ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name*	
F1+1		
F2+2		
F1+3		
F2	0: Column of record name 1 - 4096: Top column of data	
F3	0: Column of record name 1 - 4096: Final column of data	

* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

CSV file

Storage target: \(access folder)\RECIPE\(arbitrary folder)
File name: \xxxxxxxx.csv

8 one-byte upper-case alphanumeric characters or less

The designation of the line and column numbers in a CSV file differs, depending on the options selected for [Format Setting] ([Recipe] → [File Format]). The ♦ mark indicates the position of line No. 1 and column No. 1 in a CSV file.

	☐ Add title to data	☑ Add title to data
☐ Add record name	♦	Title ♦
☑ Add record name	Record ♦	- Record Title ♦

Example

- $u100 = 'TARGET'
  SET_RECIPEFOLDER $u100
  $u110 = 5250H (W)
  $u111 = 444FH (W)
  $u112 = 4355H (W)
  $u113 = 3754H (W)
  WR_RECIPE_COLUMN PLC1 [D300] $u110 5 5

  Not required if SET_FOLDER has already been executed

  50 52 4F 44 55 43 54 37 = PRODUCT7 (ASCII)

  The above program overwrites column No. 5 in the PRODUCT7.csv file stored in the TARGET folder with the data at the location starting from PLC1: D300.

Supplemental remarks

- Four consecutive words starting from the address in [F1] are used. Be sure that these words are not already used elsewhere.
- If the CSV file specified in [F1] does not exist, a storage read error occurs ($s497 = 16).
- When [Line: Data, Column: Record] is selected, a new CSV file is created by specifying [F2] = 1.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

GET_RECIPE_FILEINFO

All models	○

GET_RECIPE_FILEINFO F0 F1 F2

Function: CSV file information

This macro command is used to store the number of lines/columns of the F1-specified CSV file in memory at the address in [F2].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎	◎	◎	○
F1	◎	◎	◎	
F2	◎	◎	◎	

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0: Number of lines 1: Number of columns
F1	0000 - 9999: CSV file number designation (RECxxxx.csv) −1 (FFFFH): CSV file name designation (xxxxxxxx.csv)
F1+1	Valid if F1 = −1 ASCII code (8 one-byte upper-case alphanumeric characters): CSV file name*
F1+2	
F1+3	
F1+4	
F2	Information storage device memory

* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

Example

- CSV file number designation

```
$u100 = 0 (W)        [Line]
$u200 = 1 (W)        [File number]
GET_RECIPE_FILEINFO $u100 $u200 $u300
```

The above program stores the number of lines of the REC0001.CSV file located in the RECIPE folder in $u300.

- CSV file name designation

```
$u400 = 'TEST'                          ┐ Not required if SET_FOLDER has
SET_RECIPEFOLDER $u400                  ┘ already been executed
$u100 = 1 (W)          [Column]
$u200 = -1 (W)         [File name]
$u201 = 'SUBDATA'      [File name]
GET_RECIPE_FILEINFO $u100 $u200 $u300
```

The above program reads the number of columns in the file “SUBDATA.CSV” under the TEST folder from the recipe setting and stores it in $u300.

Supplemental remarks

- When a CSV file name is specified, the next four consecutive words starting from the address in [F1+1] are used. Be sure that these words are not already used elsewhere.

- If [Add title to data] is checked under [Format Setting] ([Recipe] → [File Format]), the number of lines to be stored does not include the title line.
- If [Add record name] is checked under [Format Setting] ([Recipe] → [File Format]), the number of columns to be stored does not include the column of the record name.
- In the event of storing the number of columns with [Line: Record, Column: Data] checked or storing the number of lines with [Line: Data, Column: Record] checked under [Format Setting] ([Recipe] → [File Format]), the data is stored based on the readout from the settings made under [Format Setting].
- The result of macro execution is stored in $s990.

Code (DEC)	Contents
0	Normal
1	F0 parameter invalid
2	F1 parameter invalid
3	F2 parameter invalid
4	F3 parameter invalid
5	Error found during accessing the specified file
6	Unable to process the specified file

- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.15 Storage (Sampling)

SMPL_BAK

SMPL_BAK F0

All models	○

Function: Save backup (bin file)

This macro command is used to make a backup file of logging or alarm data in block No. [F0] and to save the file to the year/month/day folder in the storage.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 11: Block number

File

- Logging server
  Storage target: \(access folder)\LOGGING\(year/month folder)\(year/month/day folder)
  File name: LOGGINGxx_YYYYMMDDHHMMSS.bin

  00 - 11: Block number
  Output time in year, month, day, hour, minute, and second

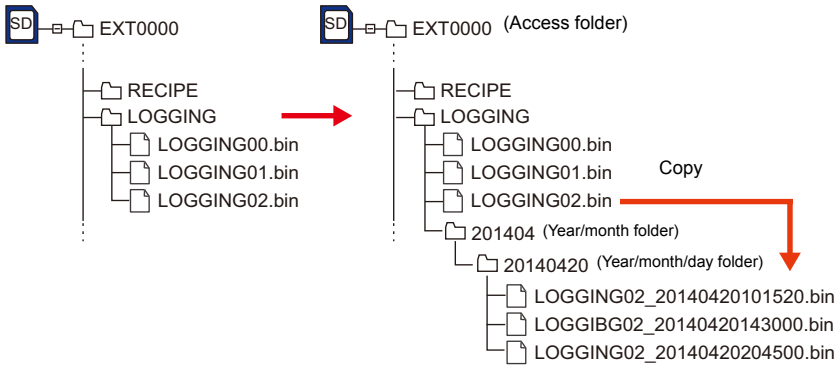
- Alarm server
  Storage target: \(access folder)\ALARM\(year/month folder)\(year/month/day folder)
  File name: ALARMxx_YYYYMMDDHHMMSS.bin (alarm)
  EVENTxx_YYYYMMDDHHMMSS.bin (event)

  00 - 11: Block number
  Output time in year, month, day, hour, minute, and second

Example

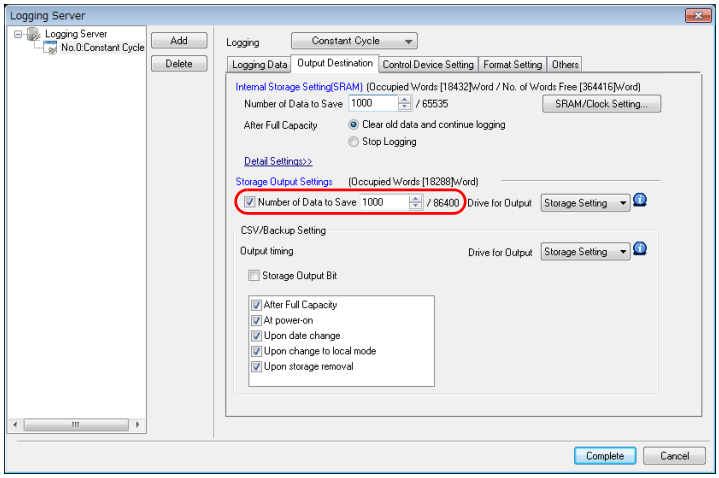
- SMPL_BAK 2

  The above program creates a backup file for logging block 2 (LOGGING02.bin) on April 20, 2014.



Supplemental remarks

- This command can be used only in the V8-compatible mode.
- This macro command is valid when [Number of Data to Save] under [Storage Output Settings] is checked in the [Output Destination] window of the logging or alarm block.



- Data stored in SRAM is output to the storage and saved in a backup file.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SMPL_CSV

All models	○

SMPL_CSV F0

Function: Create CSV file

This macro command is used to convert the logging or alarm data in block No. [F0] to the CSV file, and to save the file to the LOGGING or ALARM folder in the storage.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 11: Block number

File

- Logging server
  Storage target: \(access folder)\LOGGING
  File name: \xxxxxxxx.csv
  File name
- Alarm server
  Storage target: \(access folder)\ALARM
  File name: \xxxxxxxx.csv
  File name

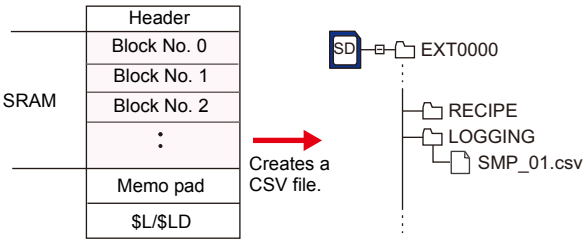
* Specify a file name as desired in [Form Setting] in the [Logging Block] or [Alarm Block] window.

Example

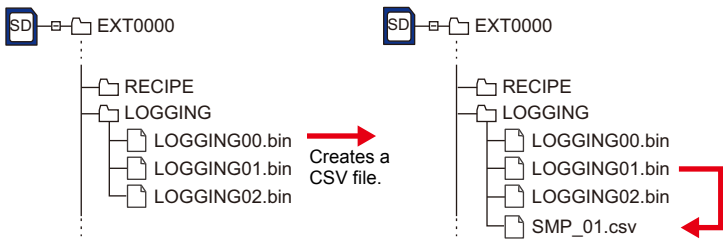
- SMPL_CSV 1

  The above program converts the data of logging block 1 to CSV format (SMP_01.CSV) and saves the file.

With [Number of Data to Save] under [Storage Output Settings] unchecked:

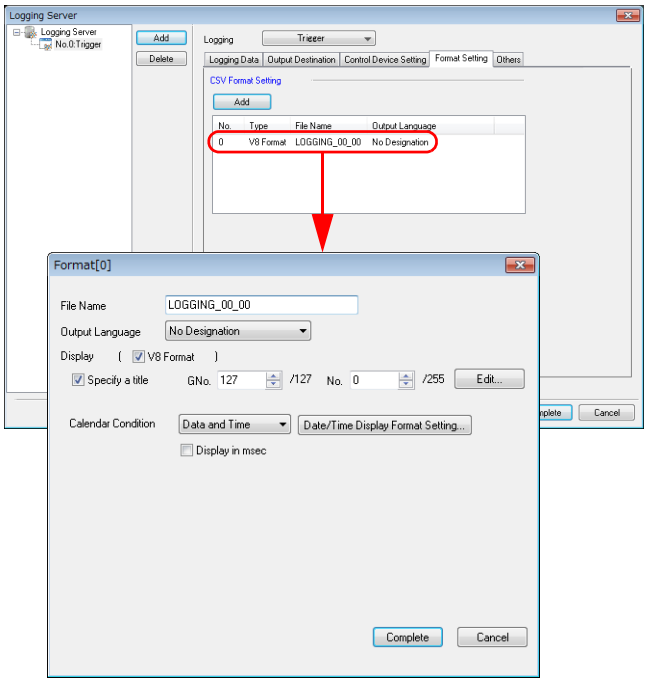


With [Number of Data to Save] under [Storage Output Settings] checked:



Supplemental remarks

- This command can be used only in the V8-compatible mode.
- When [Number of Data to Save] under [Storage Output Settings] is checked in the [Output Destination] window, data stored in SRAM is output to the storage device and saved in a CSV file.
- [Format Setting] is required for each block number.



- If the specified file already exists, it will be overwritten.
- If the block is empty, no CSV file will be created.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SMPL_CSV2

All models	○

SMPL_CSV2 F0 F1

Function: Create CSV file (file name designation)

This macro command is used to convert the logging or alarm data in block No. [F0] to the CSV file under a name specified for [F1], and to save the file to the LOGGING or ALARM folder in the storage. If the specified file does not exist, a new file will be created.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			○
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

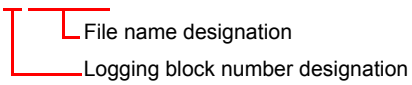
	Value
F0	0 - 11: Block number
F1	ASCII code (64 one-byte uppercase alphanumerics at the maximum): CSV file name

File

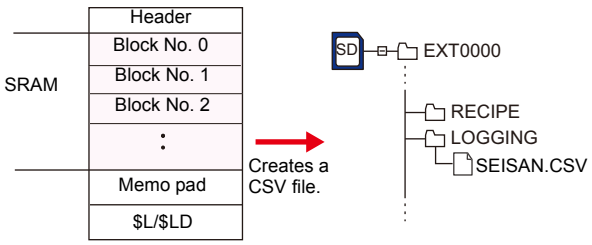
- Logging server
  Storage target: \(access folder)\LOGGING
  File name: xxxxxxxx.csv
- Alarm server
  Storage target: \(access folder)\ALARM
  File name: xxxxxxxx.csv

Example

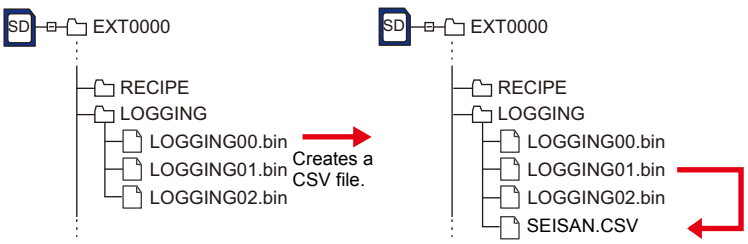
- The file named “SEISAN.CSV” is created from the data in logging block No. 1.
  $u00100 = ‘SEISAN’ (STRING)
  SMPL_CSV2 1 $u00100



With [Number of Data to Save] under [Storage Output Settings] unchecked:



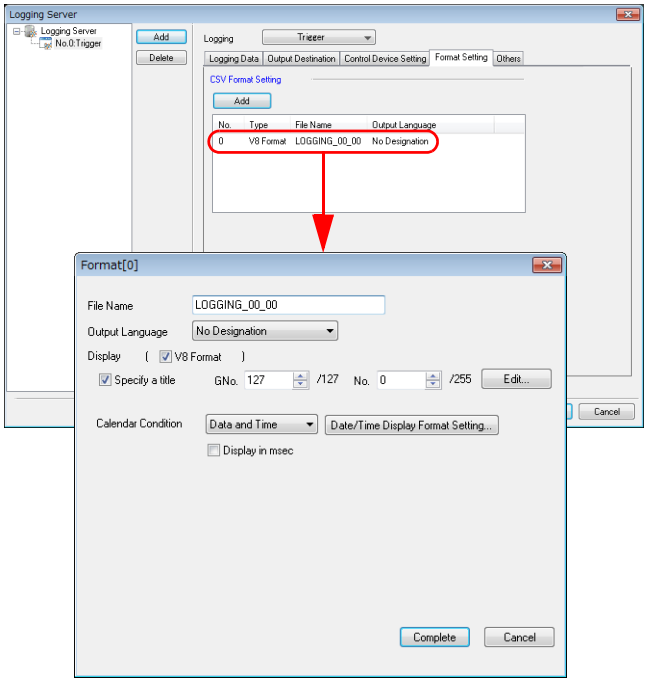
With [Number of Data to Save] under [Storage Output Settings] checked:



* If [☐ Insert/Overwrite together with STRING Command] is checked in the [Device Setting] or [Macro Editing Support] dialog, the macro command STRING can also be registered.
For more information on STRING, refer to page 4-46.

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- When [Number of Data to Save] under [Storage Output Settings] is checked in the [Output Destination] window, data stored in SRAM is output to the storage device and saved in a CSV file.
- [Format Setting] is required for each block number.



- If the specified file already exists, it will be overwritten.
- If the block is empty, no CSV file will be created.
- A full pathname can be specified for [F1].
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Restrictions

- Symbols, [\], [/], [:], [*], [?], ["], [<], [>] and [|], cannot be used for a file name.

SMPL_SAVE

All models	○

SMPL_SAVE

Function: Save logging/alarm data stored in SRAM

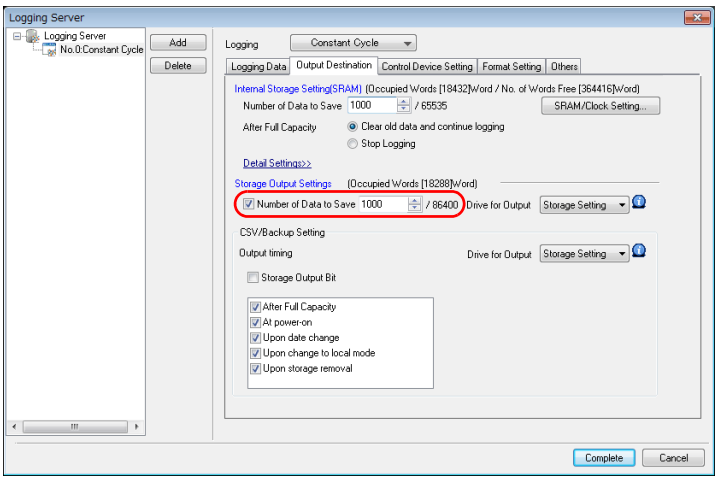
This macro command is used to save the logging or alarm data stored in SRAM to the storage at the desired set timing.

File

- Logging server
  Storage target: \(access folder)\LOGGING
  File name: \LOGGINGxx.bin
  00 - 11: Block number
- Alarm server
  Storage target: \(access folder)\ALARM
  File name: \ALARMxx.bin
  00 - 11: Block number

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- This macro command is valid when [Number of Data to Save] under [Storage Output Settings] is checked in the [Output Destination] window of the logging or alarm block.



- Data stored in SRAM is output to the storage and saved in a backup file.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SMPLCSV_BAK

All models	○

SMPLCSV_BAK F0

Function: Save backup (CSV file)

This macro command is used to convert the logging or alarm data in block No. [F0] to the CSV file, and to save the file to the year/month/day folder in the storage.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

Device	Value
F0	0 - 11: Block number

File

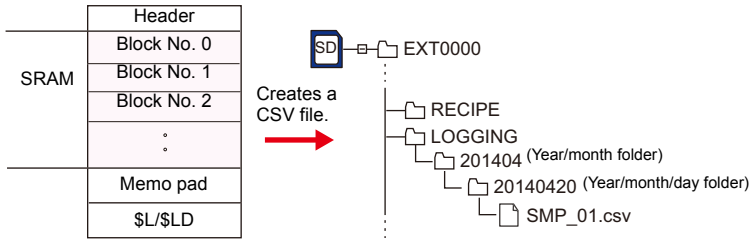
- Logging server
  Storage target: \(access folder)\LOGGING\(year/month folder)\(year/month/day folder)
  File name: \xxxxxxxx_YYYYMMDDHHMMSS. csv
  File name — Output time in year, month, day, hour, minute, and second
- Alarm server
  Storage target: \(access folder)\ALARM\(year/month folder)\(year/month/day folder)
  File name: \xxxxxxxx_YYYYMMDDHHMMSS. csv
  File name — Output time in year, month, day, hour, minute, and second

* Specify a file name as desired in [Form Setting] in the [Logging Block] or [Alarm Block] window.

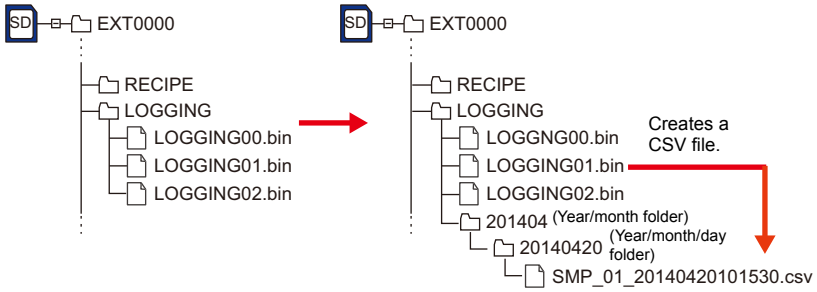
Example

- SMPLCSV_BAK 1
  The above program creates a CSV file for logging block 1 (LOGGING01.bin) on April 20, 2014.

With [Number of Data to Save] under [Storage Output Settings] unchecked:

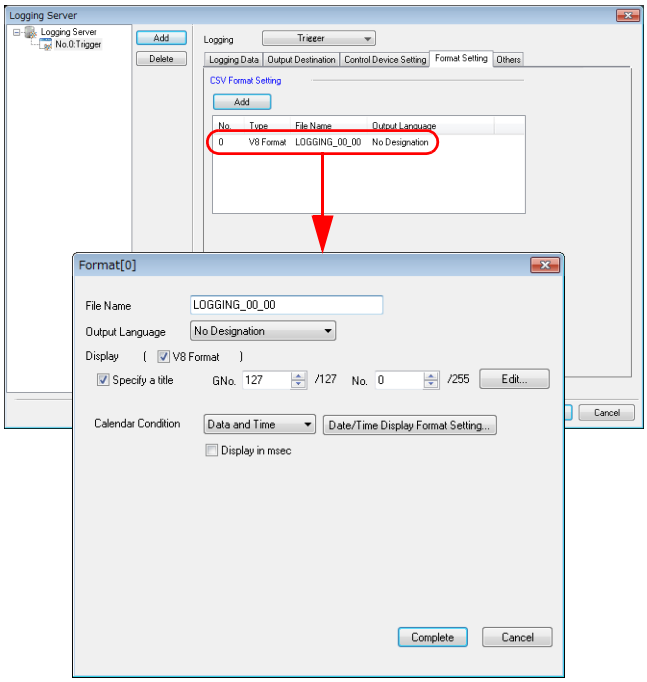


With [Number of Data to Save] under [Storage Output Settings] checked:



Supplemental remarks

- This command can be used only in the V8-compatible mode.
- When [Number of Data to Save] under [Storage Output Settings] is checked in the [Output Destination] window, data stored in SRAM is output to the storage device and saved in a CSV file.
- [Format Setting] is required for each block number.



- If the block is empty, no CSV file will be created.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SMPLCSV_BAK2 SMPLCSV_BAK2

All models	○

Function: Create CSV backup file (file name designation)

This macro command is used to convert the logging or alarm data in block No. [F0] to the CSV file under a name specified for [F1], and to save the file to the year/month/day folder in the storage.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			○
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 11: Block number
F1	ASCII code (64 one-byte uppercase alphanumerics at the maximum): CSV file name

File

- Logging server
  Storage target: \(access folder)\LOGGING\(year/month folder)\(year/month/day folder)
  File name: \xxxxxxxx_YYYYMMDDHHMMSS. csv
  File name
  Output time in year, month, day, hour, minute, and second
- Logging server
  Storage target: \(access folder)\LOGGING\(year/month folder)\(year/month/day folder)
  File name: \xxxxxxxx_YYYYMMDDHHMMSS. csv
  File name
  Output time in year, month, day, hour, minute, and second

Example

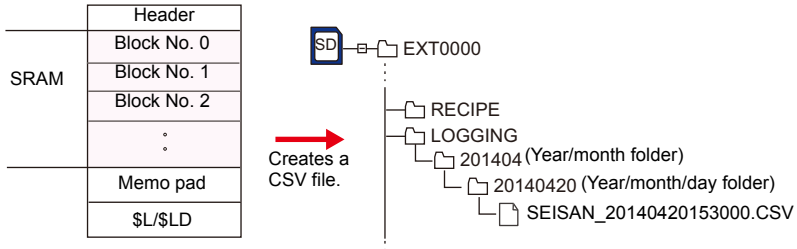
- A CSV file is created for block No. 1 backup.
  April 20, 2014, file name “SEISAN.CSV”

```
$u00100 = ‘SEISAN’ (STRING)
SMPLCSV_BAK2 1 $u00100
```

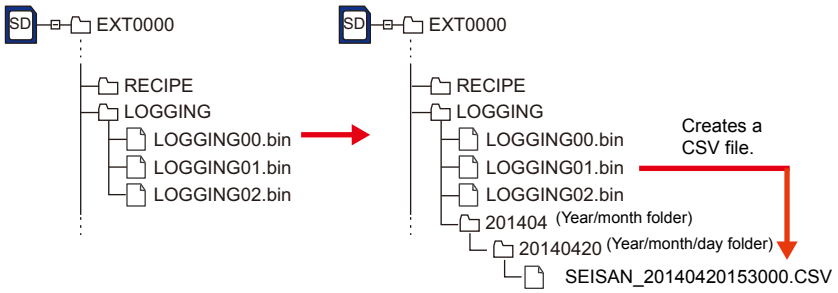
File name designation
Block number designation

* If [☐ Insert/Overwrite together with STRING Command] is checked in the [Device Setting] or [Macro Editing Support] dialog, the macro command STRING can also be registered.
For more information on STRING, refer to page 4-46.

With [Number of Data to Save] under [Storage Output Settings] unchecked:



With [Number of Data to Save] under [Storage Output Settings] checked:



Supplemental remarks

- This command can be used only in the V8-compatible mode.
- When [Number of Data to Save] under [Storage Output Settings] is checked in the [Output Destination] window, data stored in SRAM is output to the storage device and saved in a CSV file.
- The format setting must be made for each block number. (Refer to page 4-139.)
- If the block is empty, no CSV file will be created.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Restrictions

- Symbols, [\], [/], [:], [*], [?], ["], [<], [>] and [|], cannot be used for a file name.

4.16 Storage (Others)

HDCOPY

All models	○

HDCOPY

Function: Hardcopy

This macro command is used to save the image of the screen displayed at the time of the macro execution to the storage.

Storage target

Storage target: \(access folder)\HDCOPY
File name: \HDxxxx.PNG

0000 - 1023: Screen number

Supplemental remarks

- One file saves one screen. If a screen file you wish to save already exists in the storage, the file will be overwritten.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

HDCOPY2

All models	○

HDCOPY2 F0

Function: Hardcopy

This macro command is used to save the image of the screen displayed at the time of macro execution with the backup number specified in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0 - 99: Backup number

Storage target

Storage target: \(access folder)\HDCOPY
File name: \HDxxx~yy.PNG

00 - 99: Backup number

000 - 999: Screen number
(Screen Nos. 1000 - 1023 invalid)

Supplemental remarks

- With the use of backup numbers, a maximum of 100 hardcopy images can be saved per screen. You can, therefore, view time-series variations in these images.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

HDCOPY3

All models	○

HDCOPY3

Function: Hardcopy (file name designation)

This macro command is used to save the screen image (PNG) displayed at the time of the macro execution, under a file name specified in [F0], to the storage.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	ASCII code (64 one-byte uppercase alphanumerics at the maximum): CSV file name

File

Storage target: \(access folder)\HDCOPY
File name: \xxxxxxxx.PNG

File name

Example

- The file named “SCREEN10.PNG” is created.
  $u00100 = ‘SCREEN10’ (STRING)
  HDCOPY3 $u00100

File name designation

* If [☐ Insert/Overwrite together with STRING Command] is checked in the [Device Setting] or [Macro Editing Support] dialog, the macro command STRING can also be registered. For more information on STRING, refer to page 4-46.

Supplemental remarks

- One file saves one screen. If a screen file you wish to save already exists in the storage, the file will be overwritten.
- A full pathname can be specified for [F0].
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Restrictions

- Symbols, [\], [/], [:], [*], [?], ["], [<], [>] and [|], cannot be used for a file name.

SET_DRIVE

All models	○

SET_DRIVE F0

Function: Select drive

This macro command is used to select a storage drive to be accessed by a macro command.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Drive name designation* C: Built-in SD card drive D: Storage device connected to USB port

* The drive name must be followed by a colon.
For details on text processing of the drive name, refer to “Supplemental remarks” on page 4-121.

Example

- $u0010 =‘D:’
  SET_DRIVE $u0010

  The above program switches access to the D drive (storage device connected to the USB port).

Supplemental remarks

- If the drive name is not correctly specified, no operation takes place.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

- A drive change due to this macro command occurs only when any recipe macro command is executed.
  No drive change will be made for sampling data storage and macro commands other than that which is recipe-related.
- After the drive has been changed with this command, files under the folder specified for [Access Folder Name] in [System Setting] → [Storage Setting] are accessed. To change the folder to access using a recipe-related macro command, use “SET_ RECIPEFOLDER” (page 4-118).

COPY_FILE

All models	○

COPY_FILE F0 F1

Function: Copy file

This macro command is used to copy the file specified in [F0] to the file specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			
F1	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

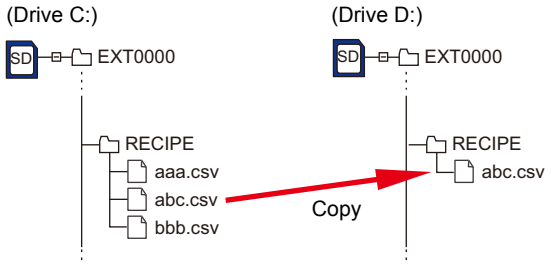
Setting range

	Value	Remarks
F0	Full pathname of the copy source*	Drive name designation C: Built-in SD card drive D: Memory device connected to USB port
F1	Full pathname of the copy destination*	

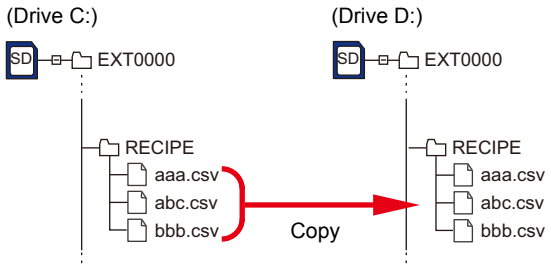
* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

Example

- Operation 1
  The program below copies “C:\EXT0000\RECIPE\abc.csv” to “D:\EXT0000\RECIPE\abc.csv”.
  - $u00100 = ‘C:\EXT0000\RECIPE\abc.csv’
    $u00200 = ‘D:\EXT0000\RECIPE\’
    COPY_FILE $u00100 $u00200



- Operation 2
  The program below copies all files stored in “C:\EXT0000\RECIPE\” to “D:\EXT0000\RECIPE\”.
  - $u00100 = ‘C:\EXT0000\RECIPE*.*’
    $u00200 = ‘D:\EXT0000\RECIPE\’
    COPY_FILE $u00100 $u00200



Supplemental remarks

- When an asterisk "*" is specified for the copy source file name (F0) or extension name, all of the files or files with all extensions are copied. The contents of subfolders are also copied.
- If the file name of the copy destination (F1) is omitted, the data is copied to the file under the same name.
- If the full pathname is not correctly specified, no operation takes place.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

MOVE_FILE

MOVE_FILE F0 F1 F2

All models	○

Function: Move file

This macro command is used to move the file or folder specified in [F0] to the path specified in [F1].
File renaming is also possible.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			
F1	○			
F2	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

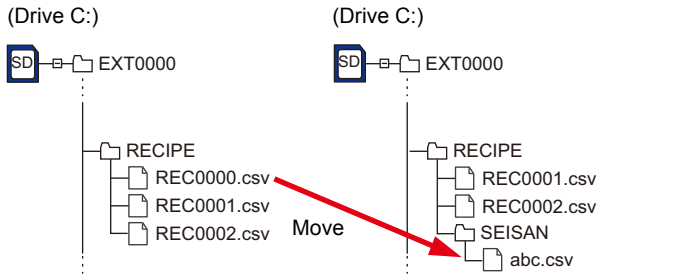
Setting range

	Value	Remarks
F0	Source full pathname (within 255 alphanumerics)*	Drive name designation C: Built-in SD card drive D: Storage device connected to USB port
F1	Target full pathname (within 255 alphanumerics)*	
F2	0 fixed	

* For details on text processing of the file name, refer to “Supplemental remarks” on page 4-121.

Example

- The program below moves “C:\EXT0000\RECIPE\REC0000.csv” to “C:\EXT0000\RECIPE\SEISAN\abc.csv”.
  $u00100 = ‘C:\EXT0000\RECIPE\REC0000.csv’
  $u00200 = ‘C:\EXT0000\RECIPE\SEISAN\abc.csv’
  $u00300 = 0 (W)
  MOVE_FILE $u00100 $u00200 $u00300



* The file “REC0000.csv” is deleted.

Supplemental remarks

- If the full pathname is not correctly specified, no operation takes place. An error will result.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

- In the case of a read-only file movement between drives, the file is copied to the target location, and the file at the original location is not deleted.
- A folder to be moved is allowed to contain a maximum of 5 hierarchical levels under the folder. If files or folders at further lower levels exist under the folder, they can be copied to the target location, but those at the original location are not deleted.

Restrictions

- Use alphanumerics to specify full pathnames as the source and the target. If any characters other than alphanumerics are used, the function of this macro command is not assured.
- Wildcard characters (such as “*” and “?”) cannot be used for full pathnames as the source and the target.
- If a file of the same name already exists in the target location, it will not be overwritten.
  In this case, “−1” is set in $s1062 (execution error). Change the file name and execute the macro again.

READ_FILE

READ_FILE F0 F1 F2 F3

All models	○

Function: Read universal file

This macro command is used to read the file [F0] in binary format and to store the obtained data in memory [F1] and after.
It is also possible to acquire the size of the file [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			
F1	◎			
F2	○			
F3	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value		Remarks
	File read	File size acquisition	
F0	Source full pathname (within 255 alphanumerics)		Drive designation C: Built-in SD card drive D: Memory connected to USB port
F1	Storage memory	0 fixed	
F2	0 - 10485760 bytes: Size	0 fixed	DEC
F2+1			
F2+2	0 - 10485760 bytes: Offset from the top of the file	0 fixed	DEC
F2+3			
F2+4	0 fixed		
F3	Read data size storage memory (Data size successfully read)	File size storage memory	
F3+1			

: ← V series (return data)

Example

- File read
  The file “ABC.DAT” is read from its 11th byte by 512 bytes into $u1000 - $u1255.

```
$u00100 = ‘C:\EXT0000\ABC\ABC.DAT’    [Source full pathname]
$u00200 = 512 (D)                     [Size]
$u00202 = 10 (D)                      [Offset]
$u00204 = 0 (W)                       [0 fixed]
READ_FILE $u00100 $u01000 $u00200 $u00300
```

- File size acquisition
  The size of the file “ABC.DAT” is read into $u300.

```
$u00100 = ‘C:\EXT0000\ABC\ABC.DAT’    [Source full pathname]
$u00200 = 0 (D)                       [0 fixed]
$u00202 = 0 (D)                       [0 fixed]
$u00204 = 0 (W)                       [0 fixed]
READ_FILE $u00100 $u01000 $u00200 $u00300
```

Supplemental remarks

- If any characters other than alphanumerics are used to specify a source full pathname, this macro command may not work normally. Be sure to use alphanumerics.
- Wildcard characters (such as “*” and “?”) cannot be used for a full pathname as the source.
- If the file specified as the source does not exist, an error will result.
- If an illegal full pathname is specified, this macro command does not work. An error will result.
- In the event of an error during file reading, the data having been read is stored in memory. However, the size of the data does not affect the successfully read data size in [F3] and [F3+1].
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

WRITE_FILE

WRITE_FILE F0 F1 F2

All models	○

Function: Write to universal file

This macro command is used to write the data from memory [F1] and after in binary format to the file [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			
F1	◎			
F2	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value			Remarks
	New creation	Overwriting	Addition	
F0	Target full pathname			Drive designation C: Built-in SD card drive D: Memory connected to USB port
F1	Source memory			
F2	0 fixed	1 fixed	2 fixed	
F2+1 F2+2	0 - 10485760 bytes: Size			DEC
F2+3 F2+4	0 fixed	0 - 10485760 bytes: Offset from the top of the file	0 fixed	
F2+5	0 fixed			

Example

- New creation
  The 512 bytes of data in $u1000 - $u1255 is written to the new file “ABC.DAT” created in the folder “ABC”.

```
$u00100 = ‘C:\EXT0000\ABC\ABC.DAT’   [Target full pathname]
$u00200 = 0 (W)                      [0: New creation]
$u00201 = 512 (D)                    [Size]
$u00203 = 0 (D)                      [0 fixed]
$u00205 = 0 (W)                      [0 fixed]
WRITE_FILE $u00100 $u01000 $u00200
```

- Overwriting
  The 33rd byte and after in the existing file “ABC.DAT” is overwritten with the 16 bytes of data in $u1000 - $u1007.

```
$u00100 = ‘C:\EXT0000\ABC\ABC.DAT’   [Target full pathname]
$u00200 = 1 (W)                      [1: Overwriting]
$u00201 = 16 (D)                     [Size]
$u00203 = 32 (D)                     [Offset]
$u00205 = 0 (W)                      [0 fixed]
WRITE_FILE $u00100 $u01000 $u00200
```

- Addition
  The 512 bytes of data in $u1000 - $u1255 is added to the existing file “ABC.DAT”.

```
$u00100 = ‘C:\EXT0000\ABC\ABC.DAT’   [Target full pathname]
$u00200 = 2 (W)                      [2: Addition]
$u00201 = 512 (D)                    [Size]
$u00203 = 0 (D)                      [0 fixed]
$u00205 = 0 (W)                      [0 fixed]
WRITE_FILE $u00100 $u01000 $u00200
```


Supplemental remarks

- If the name of a new file you intend to create is already used, delete the existing file first and create a new file.
- If the size specified with [F2+1] and [F2+2] is zero for a new file, an empty file will be created.
- If the file you specified for overwriting or data addition does not exist, an error will result.
- Wildcard characters (such as “*” and “?”) cannot be used for a full pathname as the target, to which data is written.
- If an illegal full pathname is specified, this macro command does not work. An error will result.
- In the event of an error during writing to a file, the data having been written remains in the file.
- The result of macro execution is stored in $s1062.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.17 Real No. Arithmetical Operation

F_ADD(+)

F0 = F1 + F2 (F)

All models	○

Function: Real number addition

This macro command is used to write the result of [F1] real number data plus [F2] real number data to [F0].

DWORD

	F1+1	F1
+	F2+1	F2
	F0+1	F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

F_SUB(-)

F0 = F1 - F2 (F)

All models	○

Function: Real number subtraction

This macro command is used to write the result of [F1] real number data minus [F2] real number data to [F0].

DWORD

	F1+1	F1
−	F2+1	F2
	F0+1	F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

F_MUL(X)

All models	○

F0 × F2 (F)

Function: Real number multiplication

This macro command is used to write the result of [F1] real number data multiplied by [F2] real number data to [F0].

DWORD

	F1+1	F1
×	F2+1	F2
	F0+1	F0

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

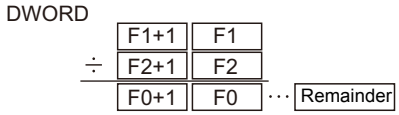
F_DIV(/)

All models	○

F0 = F1 / F2 (F)

Function: Real number division

This macro command is used to write the result of [F1] real number data divided by [F2] real number data to [F0].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			○
F2	◎			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
3	Calculation operation execution error
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.18 Real No. Statistics

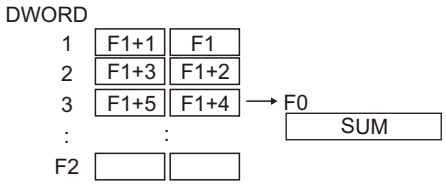
F_SUM

F0 = F_SUM (F1 C:F2) (F)

All models	○

Function: Sum of real number data

This macro command is used to sum the real number data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	0 - 512

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

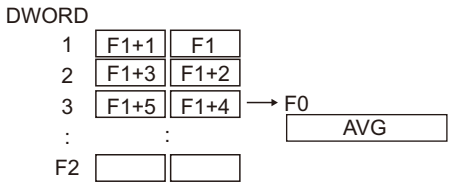
F_AVG

All models	○

F0 = F_AVG (F1 C:F2) (F)

Function: Average of real number data

This macro command is used to average the real number data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	0 - 512

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
–1	Execution error

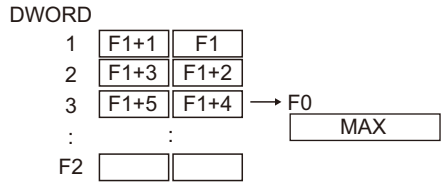
* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

F_MAX F0 = F_MAX (F1 C:F2) (F)

All models	○

Function: Maximum of real number data

This macro command is used to find the maximum of the real number data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	0 - 512

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

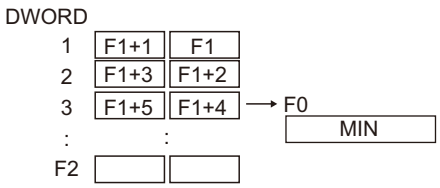
F_MIN

F0 = F_MIN (F1 C:F2) (F)

All models	○

Function: Minimum of real number data

This macro command is used to find the minimum of the real number data at the location starting from the address specified in [F1] and write the result to [F0]. The data count is specified in [F2].



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			
F2	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	IEEE 32-bit single precision real number
F1	
F2	0 - 512

Supplemental remarks

- For more information on the IEEE 32-bit single precision real numbers, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s1056.

Code (DEC)	Contents
0*	Normal
1	Overflow
2	Underflow
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

4.19 Others

;(Comment)

; (Comment)

All models	○

Function: Comment

This is treated as a comment line. No command processing is required.

BRIGHT

All models	○

BRIGHT F0

Function: Brightness adjustment

This command is used to change the brightness of the TFT display to the level specified in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0: Bright : 127: Dark

Supplemental remarks

- The current brightness is output to $s956.
- When the macro command is executed, communication will pause for several hundred milliseconds to allow for saving the setting value to the FROM. Avoid the frequent use of the macro command.
- If MONITOUCH set to a low brightness is turned off, the backlight may not light up at the next power-on.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

GET_MSGBLK

GET_MSGBLK F0 F1

All models	○

Function: Message acquisition

This macro command is used to store the [F1]-specified message (text) in [F0] memory using ASCII/shifted JIS codes.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			
F1	○			○

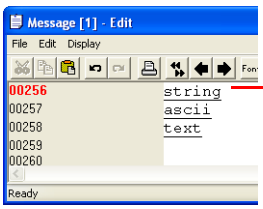
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Storage memory
F1	0 - 32767: Message No.

Example

- $u00050 = 256 (W)
  GET_MSGBLK $u00100 $u00050



GET_MSGBLK

$u100	7	4	7	3	HEX	ts
$u101	6	9	7	2	HEX	ir
$u102	6	7	6	E	HEX	gn
$u103	0	0	0	0	HEX	Null code

The above program stores message No. 256 (= GNo. 1 and line No. 0) in memory at $u100 and after using shifted JIS codes.

Supplemental remarks

- Regardless of the [Text Process] setting under [Communication Setting] for PLC1, the data is stored in memory in the [LSB → MSB] sequence.
- A null code is added to the end. Even-number-byte text thereby uses one extra word.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

PLC_ULR

All models	○

PLC_ULR F0 F1

Function: Read user log

This macro command is used to read the user log of the PLC with the station number / CPU number specified in [F0] of the PLC1 into the address specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			
F1	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

		Value		Remarks
Device memory information definition	F0	Higher-order	01 - 1F: Station number	Setting required only for 1:n connection
		Lower-order	00 : CPU No.1 01 : CPU No.2 02 : CPU No.3 03 : CPU No.4	
	F0+1	−1: 0: 1 - 63:	Reading the number of user log registrations Reading the most recent user log Reading user log No. n	
Reading the number of registrations	F1	Number of registrations (decimal)		Stored also in the special register Z105
	F1+1			
Log read	F1	0: Normal −1: Error		"−1" to be stored if no data exists in the user log specified in F0 or a communication error occurs
	F1+1	Year (ASCII)		
	F1+2	Month (ASCII)		
	F1+3	Day (ASCII)		
	F1+4	Hour (ASCII)		
	F1+5	Minute (ASCII)		
	F1+6	Second (ASCII)		
	F1+7	Main code (decimal)		
	F1+8	Sub-code (decimal)		

:← V series (Return data)

Example

If a user log reading results in “05/10/19 11 : 20 : 34 +1 +23”, its format for storage is as the following:

	Storage format
m+0	0
m+1	3530HEX (= 05DEC)
m+2	3031HEX (= 10DEC)
m+3	3931HEX (= 19DEC)
m+4	3131HEX (= 11DEC)
m+5	3032HEX (= 20DEC)
m+6	3433HEX (= 34DEC)
m+7	1DEC
m+8	23DEC

Supplemental remarks

- The macro command is valid only when Yokogawa’s FA-M3xxx is selected as the PLC1.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

RECONNECT

All models	○

RECONNECT F0

Function: Multi-drop reconnection (PLC1)

This macro command is used to establish a connection again to the stations specified in [F0] or the sub stations specified in [F0+1] when a multi-drop connection is set at the PLC1.
When "–1" is specified for [F0], reconnection with all ports is established, and when "–1" is specified for [F0+1], reconnection with all sub ports is established.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	0 - 255: PLC station number	–1: All station numbers designation
F0+1	0 - 255: PLC sub-station number	–1: All sub-station numbers designation

Supplemental remarks

- This command is only valid when a multi-drop connection (1:n) is set at PLC1. To re-establish a connection other than with PLC1, use a " RECONNECT_EX" command (page 4-172).
- The macro command is used in the event of a communication fault.
- Reconnection with the specified station is performed only once.
- When reconnection is successful, the "interrupted" information in system device memory ($s114 to 159) and 8-way communication device memory ($p[1] : 10 to 25) in the PLC1 are cleared.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

RECONNECT_EX RECONNECT_EX PLC F0 F1

All models	○

Function: Reconnection

This macro command is used to establish a connection again with the station number [F1] or the sub-station number [F1+1] specified in [F0] of the PLC.
When "−1" is specified for [F1], reconnection with all stations is established, and when "−1" is specified for [F1+1], reconnection with all sub stations is established.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○
F1	○			○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	1 - 8: PLC number	
F1	0 - 255: PLC station number	−1: All station numbers designation
F1+1	0 - 255: PLC sub-station number	−1: All sub-station number designation

Supplemental remarks

- The macro command is used in the event of a communication fault.
- Reconnection with the specified station and the specified sub-station is performed only once.
- When reconnection is successful, the "interrupted" information in 8-way communication device memory ($p[F0] : 10 to 25) in the PLC is cleared. For the PLC1, the "interrupted" information in system device memory ($s114 to 129) is also cleared at the same time.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SAMPLE

SAMPLE F0 F1 F2

All models	○

Function: Acquire logging/alarm data

This macro command is used to store sampling data specified in [F2] of the block number specified in [F1] at the device memory address specified in [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	○			
F2	○			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value		
F0	Storage target		
F1	0: Cursor specification 1: Block specification		
F1+1	(F1 = 0)	(F1 = 1)	
	0: Base 1 - 10: Overlap ID 0 - 9	0 - 11: Block number	
F1+2	0 - 255: ID No. of the item displayed	Not used	
F2	0: Acquisition of sampling data 1: Acquisition of average / maximum / minimum / total data 2: Acquisition of alarm data		
F2+1	([F2] = 0)	([F2] = 1)	([F2] = 2)
	0: With no time data 1: With time data	0 - : Word No.	Not used

1. Acquiring sampling data (with no time data)
   - When [F1] = 0
     When the specified logging viewer is selected (the cursor is displayed), the data at the cursor position is stored.
     When the specified logging viewer is not selected (the cursor is not displayed), the most recent sampling data is stored.
   - When [F1] = 1
     The most recent sampling data is stored.
   - Specify the sampling data to be acquired in [F1+1] and [F1+2].
   - Set “0” for [F2] and [F2+1].
   - The following data is stored in the [F0] memory.

Device memory	Contents	Word count
F0	Sampling data (1)	1
F0+1	Sampling data (2)	1
F0+2	Sampling data (3)	1
:	:	:
F0 + (sampling word count – 1)	Sampling data (sampling word count)	1

* When [Real Time Display] is selected for [Display Mode] in the logging viewer, no operation takes place.

2. Acquiring sampling data (with time data)
   - When [F1] = 0
     When the specified logging viewer is selected (the cursor is displayed), the data at the cursor position is stored.
     When the specified logging viewer is not selected (the cursor is not displayed), the most recent sampling data is stored.
   - When [F1] = 1
     The most recent sampling data is stored.
   - Specify the sampling data to be acquired in [F1+1] and [F1+2].
   - Set “0” for [F2] and “1” for [F2+1].
   - The following data is stored in the [F0] device memory.

Device memory	Contents	Word count
F0	Sampling time (Greenwich data)	2
F0+2	Sampling time in msec (0 - 999)	1
F0+3	Sampling data (1)	1
F0+4	Sampling data (2)	1
:	:	:
F0 + (3 + sampling word count – 1)	Sampling data (sampling word count)	1

* When [Real Time Display] is selected for [Display Mode] in the logging viewer, no operation takes place.

3. Acquiring average / maximum / minimum / total data
   - Specify the sampling data to be acquired in [F1+1] and [F1+2].
   - Set “1” for [F2].
   - Set the number of words for [F2+1].
   - The following data is stored in the [F0] device memory.

Device memory	Contents	Word count
F0	Average	2
F0+2	Maximum	2
F0+4	Minimum	2
F0+6	Total	2
F0+8	Result of overflow 0: No overflow 1: Overflow occurred	1

* When [Real Time Display] is selected for [Display Mode] in the logging viewer, no operation takes place.

4. Acquiring alarm information
   - Specify the sampling data to be acquired in [F1+1] and [F1+2].
   - Set “2” for [F2].
   - The following data is stored in the [F0] device memory.

Device memory	Contents	Word count
F0	Automatic operation time	2
F0+2	Automatic operation stop time	2
F0+4	Program stop time	2
F0+6	Number of stops	1
F0+7	Rate of operation	1

* This command can be used only for the alarm server.
If [Alarm History] ([Alarm Block] → [Alarm Device]) is not checked, no operation takes place.

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SEARCH_FILE

All models	○

SEARCH_FILE F0 F1

Function: JPEG file search

This macro command is used to search for JPEG file numbers in the SNAP/JPEG folder stored in the storage based on the specified increments and store the result in the device memory [F0].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	◎			
F1	◎			

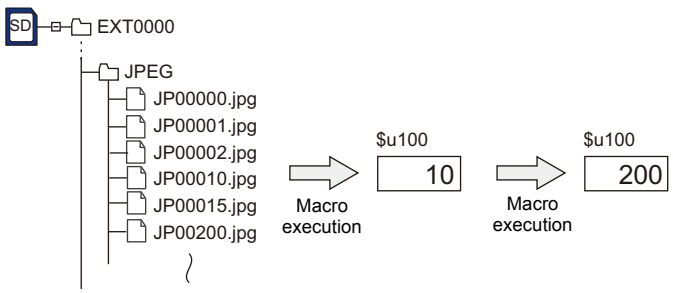
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	Search result (file number) storage target
F1	0: Searches the JPEG folder for JPxxxxx.jpg file 1: Searches the SNAP folder for VDxxxxx.jpg file
F1+1	0 - 32767: Search start file number
F1+2	−32767 - 32767: Increments

Example

- $u200 = 0 (W) [JPEG folder search]
  $u201 = 0 (W) [Search start file No. 0]
  $u202 = 10 (W) [Increments 10]
  SEARCH_FILE $u100 $u200
  $u201 = $u100 (W)



Supplemental remarks

- The macro command is valid even if no JPEG display item exists on the screen.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

ADJ_VOLUME

ADJ_VOLUME F0 F1 F2

V9 Standard	○
V9 Lite	

Function: Volume adjustment

This macro command is used to change the volume of the channel specified in [F0] to the value specified in [F1]/[F2].

	Volume adjustment value	Volume
High	7	0dB
	6	-3dB
	5	-6dB
	4	-9dB
	3	-12dB
	2	-15dB
	1	-18dB
Low	0	-21dB

*The default is “5” (−6 dB).

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0	○			○
F1	○			○
F2	○			○

○: Setting enabled (indirect designation disabled)
◎: Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0: L channel 1: R channel 2: Both channels L and R
F1	0 - 7: Volume adjustment for L channel
F2	0 - 7: Volume adjustment for R channel

Supplemental remarks

- This command is valid only for the V9 Standard model.
- To save the ADJ_VOLUME setting to the V series, use SAVE _VOLUME. When the V series is turned off without executing the SAVE_VOLUME command following the ADJ_VOLUME command, the viewing angle is reset to the one that was valid before the execution of the ADJ_VOLUME command.
- The current volume adjustment value (0 - 7) for the L channel is stored in $s1001. The current volume adjustment value (0 - 7) for the R channel is stored in $s1002.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SAVE_VOLUME

SAVE_VOLUME

V9 Standard	○
V9 Lite	

Function: Save volume adjustment value

This macro command is used to save the volume adjustment value set by the “ADJ_VOLUME” command in FROM.

Example

- ADJ_VOLUME 2 6 6
  SAVE_VOLUME

  The above program sets the volume for both L and R channels to 6.

Supplemental remarks

- This command is valid only for the V9 Standard model.
- When the SAVE_VOLUME command is executed, the communication (serial, Ethernet) is temporarily interrupted. Do not execute the SAVE_VOLUME command frequently.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

TREND REFRESH TREND REFRESH F0 F1

All models	○

Function: Refresh trend data display

The macro command is used to refresh the logging viewer display specified in [F0] and [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F0				○
F1				○

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	0: Base 1 - 10: Overlap ID 0 - 9
F1	0 - 255: ID

Supplemental remarks

- This command is valid only when [Display mode: Historical Display] and [Display method: Graph Display] are set for the logging viewer.
- If device memory addresses are specified in logging viewer settings for [Graph Min. Value] and [Graph Max. Value], and [Max. Scale Value] and [Min. Scale Value], the display must be refreshed each time data at any of these devices is changed.
- The result of macro execution is stored in $s1063.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

SYS (SET_SCRN) F1

All models	○

Function: Screen number designation

This macro command is used to display the screen specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	SET_SCRN
F1	0 - 9999: Screen number

Example

- $u100 = 55 (W) [Screen number]
  SYS (SET_SCRN) $u100

  The above program displays screen No. 55.

Supplemental remarks

- If a screen number that does not exist is specified in [F1], the macro command is disabled.
- The macro command is invalid in screen OPEN, screen CLOSE, overlap library OPEN, overlap library CLOSE, and initial macros.
- The macro command is valid only once in a macro created on the macro edit sheet. Its execution timing is set at the end of the macro on the edit sheet.
- Do not execute the macro command in every cycle using a CYCLE macro or an event timer macro.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (SET_MOVLP) F1

Function: Multi-overlap/global overlap setting

This macro command is used to display the overlap library specified in F1+1 on the overlap ID in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Dot (unit: 4 × 1)	Line/column (unit 8 × 20)
F0	SET_MOVLP	
F1	0 - 9: Overlap ID	
F1+1	0 - 1023: Overlap library number	
F1+2	0 - 1023: X coordinate	0 - 127: X coordinate
F1+3	0 - 767: Y coordinate	0 - 38: Y coordinate

Example

- $u100 = 2 (W) [Overlap ID]
  $u101 = 12 (W) [Overlap library number]
  $u102 = 50 (W) [X coordinate]
  $u103 = 5 (W) [Y coordinate]
  SYS (SET_MOVLP) $u100

  Line/Column:
  The above program displays overlap ID2 and overlap library No. 12 at the coordinates X: 400 and Y: 100.

  Dot:
  The above program displays overlap ID2 and overlap library No. 12 at the coordinates X: 48* and Y: 5.
  * Since four dots are treated as one unit on the X axis, any value in the range of “48” to “51” on the axis is regarded as “48”.

Supplemental remarks

- The macro command is valid when [Internal] is checked under [Designate] in the [Multi-Overlap] or [Global Overlap Setting] dialog.
- If [F1]>9, the macro command is disabled.
- If an overlap library number specified in [F1+1] does not exist, the macro command is disabled.
- If the specified X and Y coordinates are outside the permissible ranges, the display appears in the lower right corner of the screen.
- The macro command is invalid in a screen CLOSE macro and an initial macro.
- The macro command is valid only once in a macro created on the macro edit sheet for each overlap ID. Its execution timing is set at the end of the macro on the edit sheet.
- Use the OVLP_SHOW command to turn off the multi-overlap or global overlap.
- Do not execute the macro command in every cycle using a CYCLE macro or an event timer macro.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (OVLP_SHOW) F1

Function: Overlap ON/OFF

This macro command is used to show/hide the overlap ID specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	OVLP_SHOW
F1	0 - 9: Overlap ID
F1+1	0: OFF (non-display) 1: ON (display)

Example

- $u100 = 2 (W) [Overlap ID]
  $u101 = 0 (W) [OFF]
  SYS (OVLP_SHOW) $u100

  The above program turns off overlap ID2.

- $u100 = 2 (W) [Overlap ID]
  $u101 = 1 (W) [ON]
  SYS (OVLP_SHOW) $u100

  The above program turns on overlap ID2.

Supplemental remarks

- If [F1]>9, the macro command is disabled.
- If F1+1 = 0, the macro command is valid for normal, call-, multi-, and global ([Designate]: [Internal]) overlaps.
- The macro command is invalid in screen CLOSE, overlap library CLOSE and initial macros.
- The macro command is valid only once in a macro created on the macro edit sheet for each overlap ID. Its execution timing is set at the end of the macro on the edit sheet.
- Do not execute the macro command in every cycle using a CYCLE macro or an event timer macro.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (OVLP_POS) F1

Function: Overlap relocation

This macro command is used to move the overlap ID specified in [F1] to the coordinates X in [F1+1] and Y in [F1+2].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
	Dot (unit: 4 × 1)	Line/column (unit 8 × 20)
F0	OVLP_POS	
F1	0 - 9: Overlap ID	
F1+1	0 - 1023: X coordinate	0 - 127: X coordinate
F1+2	0 - 767: Y coordinate	0 - 38: Y coordinate

Example

- $u100 = 2 (W) [Overlap ID]
  $u101 = 50 (W) [X coordinate]
  $u102 = 5 (W) [Y coordinate]
  SYS (OVLP_POS) $u100

  Line/Column:
  The above program moves overlap ID2 to coordinates X: 400 and Y: 100.

  Dot:
  The above program moves overlap ID2 to coordinates X: 48* and Y: 5.

  * Since four dots are treated as one unit on the X axis, any value in the range of “48” to “51” on the axis is regarded as “48”.

Supplemental remarks

- If [F1]>9, the macro command is disabled.
- In the event of a normal or a call-overlap, the macro command is enabled also to display the overlap.
- The X and Y coordinates specified by the macro command take effect until the screen is switched. If OVLP_SHOW is executed after OVLP_POS, the overlap appears at the coordinates specified by OVLP_POS.
- The macro command is invalid in screen OPEN, screen CLOSE, overlap library CLOSE and initial macros.
- The macro command is valid only once in a macro created on the macro edit sheet for each overlap ID. Its execution timing is set at the end of the macro on the edit sheet.
- Do not execute the macro command in every cycle using a CYCLE macro or an event timer macro.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (GET_MSG) F1

Function: Message acquisition

This macro command is used to store the F1-specified message in memory at the $u address in F1+1 using ASCII/shifted JIS codes.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	GET_MSG
F1	0 - 32767: Message number
F1+1	0 - 16383: Storage device number
$u[F1+1] '	Shifted JIS/ASCII 50 words maximum

:← V series (Return data)

Example

- $u50 = 256 (W) [Message number]
  $u51 = 100 (W) [Storage target device memory No.]
  SYS (GET_MSG) $u50
  The above program stores message No. 256 (= GNo. 1 and line No. 0) in memory at $u100 and after using shifted JIS codes.
  The above program shows the case when [MSB → LSB] is selected for [Text Process] under [Communication Setting] for the PLC1.

Supplemental remarks

- Swap between the higher-order byte and the lower-order byte can be set by selecting an option for [Text Process] under [Communication Setting].
- Regardless of the setting above, use a " GET_MSGBLK" command (page 4-168) for storing data by [LSB → MSB].
- A null code is added to the end. Even-number-byte text thereby uses one extra word.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

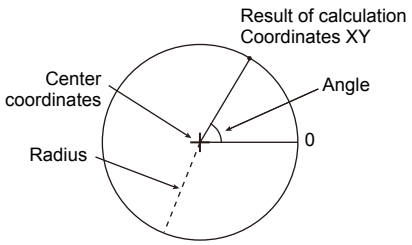
SYS

SYS (GET_XY) F1

All models	○

Function: Acquisition of X and Y coordinates on circumference

This macro command is used to calculate X and Y coordinates from a radius, an angle and, center coordinates.



Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

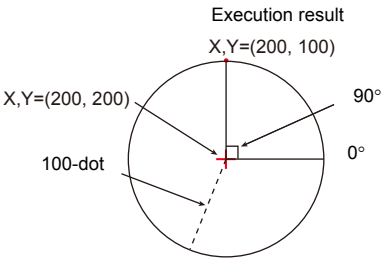
	Value
F0	GET_XY
F1	0 or above: Radius
F1+1	0 to 3600: Angle (0.1-degrees)
F1+2	0 or above: Center coordinate X
F1+3	0 or above: Center coordinate Y
F1+4	0 or above: X coordinate
F1+5	0 or above: Y coordinate

:← V series (Return data)

Example

- $u100 = 100 (W) [Radius]
  $u101 = 900 (W) [Angle]
  $u102 = 200 (W) [X coordinate of the center]
  $u103 = 200 (W) [Y coordinate of the center]
  SYS (GET_XY) $u100

  On the circumference of a circle 100 dots in radius with the center at coordinates X: 200 and Y: 200, the above program calculates the X and Y coordinates of the point at an angle of 90 degrees.
  X coordinate: $u104 = 200
  Y coordinate: $u105 = 100



Supplemental remarks

- If a value specified for the angle is 3,600 or above, the value is corrected to the remainder as the result of division by 3,600.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (SET_BZ) F1

Function: Buzzer control

This macro command is used to control the buzzer of MONITOUCH.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	SET_BZ	
F1	0: Normal 1: Error 2: Sound change	
F1+1	0: Standard 1: Short 2: None 3: Continuous	Setting required if F1 = 2

Example

- $u100 = 2 (W) [Sound change]
  $u101 = 2 (W) [None]
  SYS (SET_BZ) $u100

  The above program turns off the MONITOUCH buzzer.

Supplemental remarks

- The [Buzzer] tab window setting in the [Unit Setting] dialog ([System Setting] → [Unit Setting] → [Buzzer]) takes effect only at the time of initial connection of MONITOUCH.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (GET_TIME) F1

Function: System time acquisition

This macro command is used to acquire values from the timer that increments by one at 10-ms intervals after power-on.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	GET_TIME
F1	0 - 4294967295 (×10 msec)
F1+1	

:← V series (Return data)

Example

- SYS(GET_TIME) $u100
  The above program acquires the time that has elapsed after power-on.

  $u100 = 27900 (W)
  279000 msec = 279 sec = 4 minutes 39 seconds

Supplemental remarks

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (STA_TIME) F1
SYS (CHK_TIME) F1

Function: Timer setting

STA_TIME starts the timer. CHK_TIME confirms a time-out.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

- STA_TIME

	Value	Remarks
F0	STA_TIME	
F1	Time-out flag 0: Counting 1: Time-out	
F1+1	0: Timer type 0	F1 = 1: Stops the timer
	1: Timer type 1	F1 = 1: Updates the timer start time
F1+2	0 - 65535: Time-out time	×10 ms
F1+3	Timer start time	

:← V series (Return data)

- CHK_TIME
  For [F1], use the same device memory as for STA_TIME.

Example

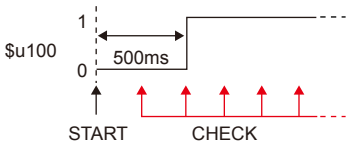
<Timer type 0>

- OPEN macro
  $u101 = 0 (W) [Timer type]
  $u102 = 50 (W) [Time-up time]
  SYS (STA_TIME) $u100

  The above program starts the timer type 0, for which a 500-ms time-out period is set.
  $u103 =current time and $u100=0 are set.

- CYCLE macro
  SYS (CHK_TIME) $u100

  $u100 = 0
  ↓ Lapse of 500 ms
  $u100 = 1 (W)
  (End)

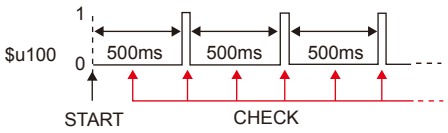


<Timer type 1>

- OPEN macro
  $u101 = 1 (W) Timer type
  $U102 = 50 (W) Time-up time
  SYS (STA_TIME) $u100

  The above program starts the timer type 1, for which a 500-ms time-out period is set.
  $u103 =current time and $u100=0 are set.

- CYCLE macro
  SYS (CHK_TIME) $u100
  IF ($u100! = 0) LB 0
  RET
  LB0
  $u200 = $u200+1 (W)
  RET



  $u100 = 0
  ↓ Lapse of 500 ms
  $u100 = 1 and $u200 = 1 are set.
  ↓
  $u103 =current time and $u100=0 are set.
  ↓ Lapse of 500 ms
  $u100 = 1 and $u200 = 2 are set.
  ↓
  (Repetition)

Supplemental remarks

- The timer base is set to 10 ms.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (GET_CLND) F1

Function: Calendar acquisition

This macro command is used to acquire the values of the system calendar.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	GET_CLND	
F1	0 or above:	Year (4-digit)
F1+1	1 - 12:	Month
F1+2	1 - 31:	Day
F1+3	0 - 23:	Hour
F1+4	0 - 59:	Minute
F1+5	0 - 59:	Second
F1+6	0: 1: 2: 3: 4: 5: 6:	Sunday Monday Tuesday Wednesday Thursday Friday Saturday

:← V series (Return data)

Example

- SYS (GET_CLND) $u100

$u100 = 2005
$u101 = 7
$u102 = 15
$u103 = 15
$u104 = 25
$u105 = 41
$u106 = 5

[July 15, 2005 Friday 15:25:41]

Supplemental remarks

- The calendar is acquired not from a PLC or other external device but from the V series unit.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (SET_CLND) F1

Function: Calendar setting

This macro command is used to set the values of eight words starting from the address specified in [F1] to the system calendar. When MONTOUCH is connected with PLC1 including the calendar function, this macro command also sets the PLC1's calendar.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	SET_CLND	
F1	0 or above: Year (4-digit/2-digit)	
F1+1	1 - 12: Month	
F1+2	1 - 31: Day	
F1+3	0 - 23: Hour	
F1+4	0 - 59: Minute	
F1+5	0 - 59: Second	
F1+6	Day of the week	Invalid; to be automatically calculated by MONITOUCH
F1+7	0 - 31: PLC station number	For 1:n connection only

Example

- $u100 = 2005 (W)
  $u101 = 7 (W)
  $u102 = 15 (W)
  $u103 = 15 (W)
  $u104 = 0 (W)
  $u105 = 0 (W)
  [July 15, 2005 Friday 15:00:00]
  SYS (SET_CLND) $u100

  The above program sets the calendars in the V series and the PLC1 to July 15, 2005 on Friday at 15:00:00.

Supplemental remarks

- When setting calendar data for PLC 2 to 8, use a macro command " PLC_CLND" (page 4-88).
- When setting calendar data only for the V series, use a macro command " SYS (SET_SYS_CLND) F1" (page 4-213).
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (SET_BUFNO) F1

Function 1: Logging information

This macro command is used to store the average, maximum, minimum, and total of logging numbers 0 to 31 located in the block number specified in [F1] in system devices $s180 to 435.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	SET_BUFNO
F1	0 - 11: Logging block No.
$s180 - 181	Logging No. 0 Average
$s182 - 183	Logging No. 0 Maximum
$s184 - 185	Logging No. 0 Minimum
$s186 - 187	Logging No. 0 Total
$s188 - 195	Logging No. 1 Average/maximum/minimum/total
$s196 - 203	Logging No. 2 Average/maximum/minimum/total
$s204 - 211	Logging No. 3 Average/maximum/minimum/total
$s212 - 219	Logging No. 4 Average/maximum/minimum/total
$s220 - 227	Logging No. 5 Average/maximum/minimum/total
$s228 - 235	Logging No. 6 Average/maximum/minimum/total
$s236 - 243	Logging No. 7 Average/maximum/minimum/total
$s244 - 251	Logging No. 8 Average/maximum/minimum/total
$s252 - 259	Logging No. 9 Average/maximum/minimum/total
$s260 - 267	Logging No. 10 Average/maximum/minimum/total
$s268 - 275	Logging No. 11 Average/maximum/minimum/total
$s276 - 283	Logging No. 12 Average/maximum/minimum/total
$s284 - 291	Logging No. 13 Average/maximum/minimum/total
$s292 - 299	Logging No. 14 Average/maximum/minimum/total
$s300 - 307	Logging No. 15 Average/maximum/minimum/total
$s308 - 315	Logging No. 16 Average/maximum/minimum/total
$s316 - 323	Logging No. 17 Average/maximum/minimum/total
$s324 - 331	Logging No. 18 Average/maximum/minimum/total
$s332 - 339	Logging No. 19 Average/maximum/minimum/total
$s340 - 347	Logging No. 20 Average/maximum/minimum/total
$s348 - 355	Logging No. 21 Average/maximum/minimum/total
$s356 - 363	Logging No. 22 Average/maximum/minimum/total
$s364 - 371	Logging No. 23 Average/maximum/minimum/total
$s372 - 379	Logging No. 24 Average/maximum/minimum/total
$s380 - 387	Logging No. 25 Average/maximum/minimum/total

	Value
$s388 - 395	Logging No. 26 Average/maximum/minimum/total
$s396 - 403	Logging No. 27 Average/maximum/minimum/total
$s404 - 411	Logging No. 28 Average/maximum/minimum/total
$s412 - 419	Logging No. 29 Average/maximum/minimum/total
$s420 - 427	Logging No. 30 Average/maximum/minimum/total
$s428 - 435	Logging No. 31 Average/maximum/minimum/total

:← V series (Return data)

Example

- $u100 = 5 (W) [Block No.]
  SYS (SET_BUFNO) $u100

  The above program stores the average, maximum, minimum and total of block No. 5 in $s180 to 435.

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- When multiple logging blocks are set, the block with the smallest number is selected as default.
- The macro command does not work if no display area exists on the screen.
- Logging numbers 32 to 255 are not available with this macro command.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Function 2: Alarm log information

This macro command is used to store alarm log information of block number specified in [F1] in $s436 - 443.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	SET_BUFNO
F1	0 - 11: Alarm block number
$s436 - 437	Automatic operation time
$s438 - 439	Automatic operation stop time
$s440 - 441	Program stop time
$s442	Number of stops
$s443	Rate of operation XX.X

:← V series (Return data)

Example

- $u100 = 4 (W) [Block No.]
  SYS(SET_BUFNO) $u100

  The above program stores the alarm log information of block No. 4 in $s436 - 443.

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- This command is valid only when [Alarm History] ([Alarm Block] → [Alarm Device]) is checked.
- For more information on each data, refer to the V9 Series Reference Manual.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (GET_SMPL) F1

Function: Acquire logging/alarm data

This macro command is used to store the data of the specified block, logging, or alarm device memory in the device memory address $u [F1+2].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

		Value
F0		GET_SMPL
F1		0 - 11: Block number
F1+1		0 or above: Logging number / Alarm device memory number
F1+2		0 - 16383: Storage target internal device memory No. “n”
Trigger / Constant cycle / Device memory map	$u n	Time data 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Month: 1 - 12 Day: 1 - 31 Hour: 0 - 23
	$u n+1	Time data 1 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Minutes & seconds (unit: seconds) [Time data 1]/[60] = [A] remainder [B] [A]: minutes [B]: seconds
	$u n+2	Logging data (maximum number of logging data: 128 words)
	$u n+3	
	:	
Alarm (Alarm logging)	$u n	Time data 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Month: 1 - 12 Day: 1 - 31 Hour: 0 - 23
	$u n+1	Time data 1 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Minutes & seconds (unit: seconds)
	$u n+2	Sampling bit number 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 0: OFF 1: ON Bit number

		Value
Alarm (Time order alarming)	$u n	Time data 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Month: 1 - 12 / Day: 1 - 31 / Hour: 0 - 23
	$u n+1	Time data 1 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Minutes & seconds (unit: seconds)
	$u n+2	Sampling relay number 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Relay number
Alarm (alarm tracking)	$u n	Time data 0
	$u n+1	Time data 1 Time data 1 | Time data 0 GMT-based UNIX time from January 1, 1970
	$u n+2	Alarm bit information 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Bit number 1: Power-off after an alarm occurrence 1: Deleted by DEL key 1: First cause 0: Reset 1: Occurrence

: ← V series (return data)

Example

- $u100 = 3 (W) [Block No.]
  $u101 = 0 (W) [Alarm device memory No.]
  $u102 = 200 (W) [Storage target device memory]
  SYS (GET_SMPL) $u100

  The above program stores the information of alarm device memory No. 0 in alarm block No. 3 in $s200.
  For an alarm (time order alarming):
  $u200 = 1E8F $_{HEX}$
  $u201 = 06B4 $_{HEX}$
  $u202 = 0002 $_{HEX}$

  0 0 0 1 1 1 1 0 1 0 0 0 1 1 1 1
  7 / 20 / 15

  July 20, 15:28:36, bit No. 2 ON

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (GET_SCUR) F1

Function: Cursor point acquisition

This macro command is used to store the sampling number and the cursor address associated with the of the logging or alarm viewer currently being displayed.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	GET_SCUR	
F1	0: Base 1 - 10: Overlap ID 0 - 9	
F1+1	0 - 255 : ID	
F1+2	0: Cursor non-display	The most recent information to be stored in F1+3 and F1+4
	1: Cursor display	The cursor information to be stored in F1+3 and F1+4
F1+3	0 or above: Sample number	Used by GET_SMPL
F1+4	0 or above: Cursor address	Sampling number comparison*

:← V series (Return data)

* Whether or not the acquired sampling number is the same as that previously acquired is checked.
Even if the sampling number remains the same, any change in the cursor address means that the data to be fetched has also changed.
Contrary, even if the sampling number has changed, no change in the cursor address means that the data to be accessed also remains the same.

Example

- $u100 = 0 (W) [Base]
  $u101 = 1 (W) [ID]
  SYS (GET_SCUR) $u100

  The above program acquires the cursor point of the logging or alarm data (ID 1) on the base screen.

  $u102 = 1 [Cursor being displayed]
  $u103 = 28 [Sample number]
  $u104 = 39 (W) [Cursor address]

Supplemental remarks

- This command can be used only in the V8-compatible mode.
- The macro command is valid in the following cases:
  - [Historical Display] is selected for [Display mode] in the logging viewer.
  - [Event History] is selected for [Display mode] in the alarm viewer.
- If the sampling number is “5” at the time of the execution of GET_SCUR, the sampling count (numerical data display) on the screen shows “6”.
  This results from the fact that the cursor point starts from “0” and the sampling count (numerical data display) starts from “1”.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (DSP_DATA) F1

Function: Show/hide numerical data display

This macro command is used to show/hide numerical data displays placed in the specified location (ID).

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	DSP_DATA
F1	0: Base 1 - 3: Overlap ID 0 - 2 4 - 7: Data block No. 0 - 3 8 - 14: Overlap ID 3 - 9
F1+1	0 - 255: ID
F1+2	0: Not display 1: Display

Example

- $u100 = 0 (W) [Base]
  $u101 = 1 (W) [ID]
  $u102 = 0 (W) [Not display]
  SYS (DSP_DATA) $u100

  The above program hides all numerical data displays of ID 1 on the base screen.

Supplemental remarks

- The macro command is valid for numerical data displays only. It cannot be used for character and message displays.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (CHG_DATA) F1

Function: Change numerical data display property

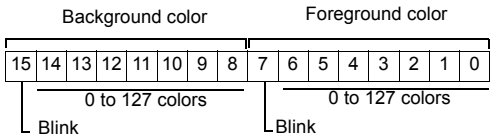
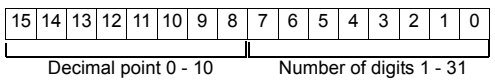
This macro command is used to change the properties of the numerical data displays placed in the specified location (ID).

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	CHG_DATA
F1	0: Base 1 - 3: Overlap ID 0 - 2 4 - 7: Data block No. 0 - 3 8 - 14: Overlap ID 3 - 9
F1+1	0 - 255: ID
F1+2	0: Without signs 1: With signs 2: With sign (+) 3: HEX 4: OCT 5: BIN
F1+3	Color  Color / Code (HEX): Black 00; Blue 01; Red 02; Magenta 03; Green 04; Cyan 05; Yellow 06; White 07
F1+4	Decimal point and number of digits 

Example

The above program changes the properties of the numerical data display of ID1 placed on the base screen.

- Type: HEX
- Background color: Black
- Foreground color: Green
- Decimal Point: None
- Number of digits: 5

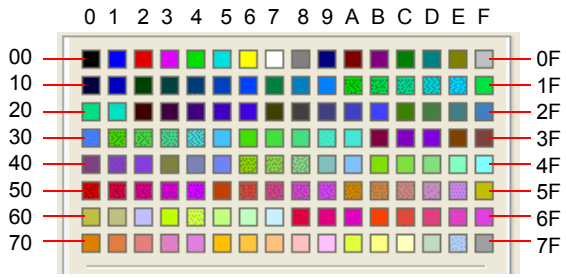
```
$u200 = 0 (W)          [Base]
$u201 = 1 (W)          [ID]
$u202 = 0 (W)          [Not display]
SYS (DSP_DATA) $u200

$u100 = 0 (W)          [Base]
$u101 = 1 (W)          [ID]
$u102 = 3 (W)          [Type]
$u103 = 0004H (W)      [Color]
$u104 = 0005H (W)      [Decimal point and number of digits]
SYS (CHG_DATA) $u100 macro execution

$u200 = 0 (W)          [Base]
$u201 = 1 (W)          [ID]
$u202 = 1 (W)          [Display]
SYS (DSP_DATA) $u200
```

Supplemental remarks

- The macro command is valid for numerical data displays only. It cannot be used for character and message displays.
- When using this macro command, be sure to execute the command DSP_DATA to redisplay the data. For more information on DSP_DATA, refer to page 4-201.
- Even on MONITOUCH with 32k- or 64k-color display, 128 colors + blink ([Custom Color] → [Palette 1]) are available with the macro command.
- 128-color codes
  The boxes on the palette are provided with their individual codes.



- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (STA_LIST) F1

Function: Data sheet print

This macro command is used to print data sheets.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

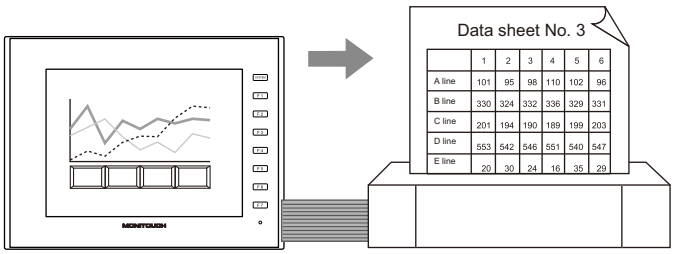
	Value
F0	STA_LIST
F1	0 - 1023: Print start number
F1+1	1 - 1023: Number of pages to be printed
F1+2	ASCII code: Output file name (64 one-byte alphanumeric characters maximum) *
:	
F1+33	

* Valid only when $s1656 = 1 (output in PDF)

Example

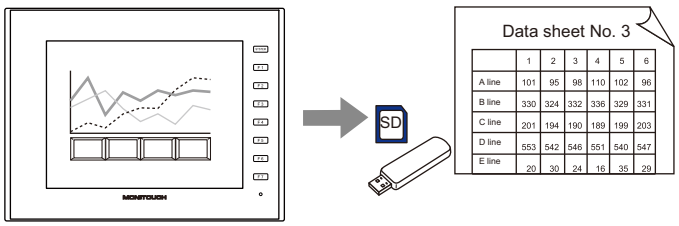
- $u100 = 3 (W) [Print start number]
  $u101 = 1 (W) [Number of pages to be printed]
  SYS (STA_LIST) $u100 macro execution

  The above program prints data sheet No. 3.



- $s1656 = 1(W) Output destination: PDF
  $u100 = 3(W) Print start number
  $u101 = 1(W) Number of pages to be printed
  $u102 = TEST (STRING)1(W) File name
  STA_LIST $u100

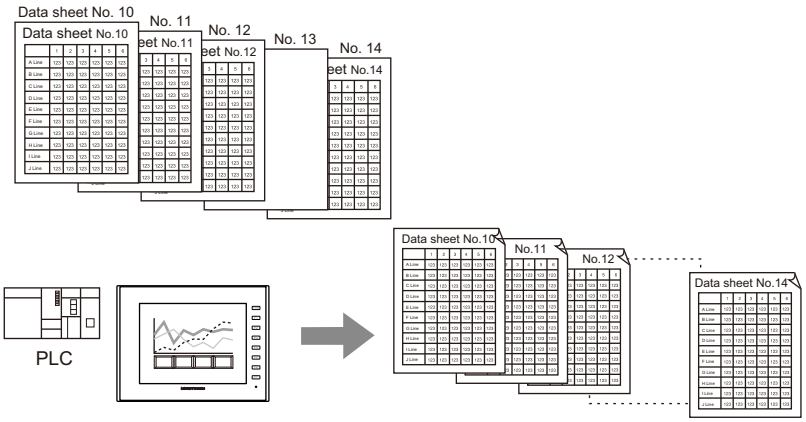
  Datasheet No. 3 is output to the storage device in PDF format.



Supplemental remarks

- If nothing is registered on a data sheet, specifying the page of this sheet does not produce a printout of it.

[Data Sheet Edit]



$u100 = 10 [Print start number]
$u101 = 5 [Number of pages to be printed]
SYS (STA_LIST) $u100

Data sheet No. 10 to 12 and 14 can be printed. The page that is not stored, No. 13, is ignored, and four pages are output.

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (SET_BKLT) F1

Function: Backlight control

This macro command is used to control the backlight.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	SET_BKLT	
F1	0: OFF 1: ON	
	2: OFF time change	Valid when [Auto 1/2/3] is selected
F1+1	0 - 65535: OFF time (sec)	Setting required if F1 = 2

Example

- $u100 = 0 (W) [OFF]
  SYS (SET_BKLT) $u100

  The above program turns off the backlight.

Supplemental remarks

- When [Always ON] is selected for [Action] on the [Backlight] tab window in the [Unit Setting] dialog ([System Setting] → [Unit Setting]), the macro command is invalid.
- The macro command is invalid with the control device memory ON.
- Do not execute the macro command in macros to be executed constantly using a CYCLE macro, an interval timer, or an event timer macro.
- The use of a switch ON macro to execute a backlight turn-on command will not be possible.
- At power-on, the backlight is restored to the status as set in [Backlight] tab window (initial status) in the [Unit Setting] dialog ([System Setting] → [Unit Setting]). The internal memory $L is available to retain the value set with this macro command. By using the initial macro at power-on, this macro command is executable according to the value you stored with $L.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (RESTART) F1

Function: Restart

This macro command is used to restart the V series when the time (in seconds) specified in [F1] has elapsed.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	RESTART
F1	0 - 60: Time (sec)

Example

- $u100 = 10 (W) [sec]
  SYS (RESTART) $u100

  The above program maintains the check screen for 10 seconds and then switches it to the RUN screen.

Supplemental remarks

- When the macro command has been executed, the data in the internal memory $u becomes “0”.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (CHG_LANG) F1

Function: Language change

This macro command is used to switch the language displayed on MONITOUCH to the language specified in [F1].

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

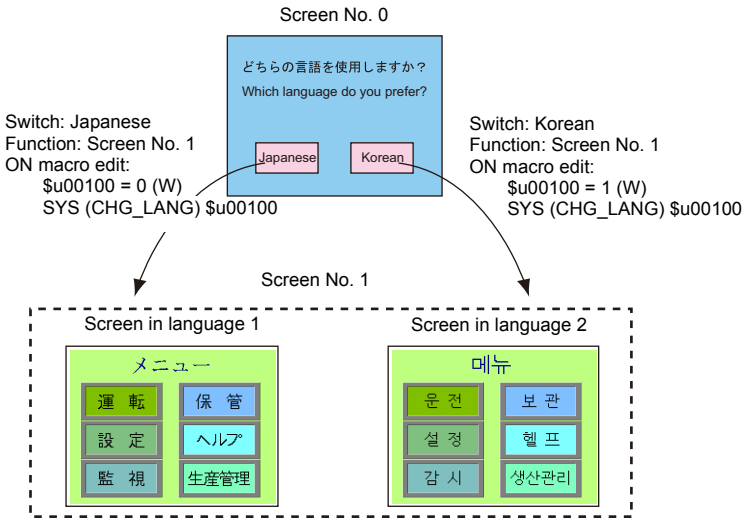
○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	CHG_LANG
F1	0: Language 1 1: Language 2 2: Language 3 : 13: Language 14 14: Language 15 15: Language 16

Example

In the example below, the ON macros for the screen change switches are used to switch between two languages.



Supplemental remarks

- When the screen is switched, the language also switches.
  To change the language on the same screen, use the “SYS(RESET_SCRN)” command (page 4-210).
- At power-on, the language as specified for [Initial Interface Language] in the [Font Setting] dialog takes effect (initial status).

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (RESET_SCRN) F1

Function: Redisplay screen

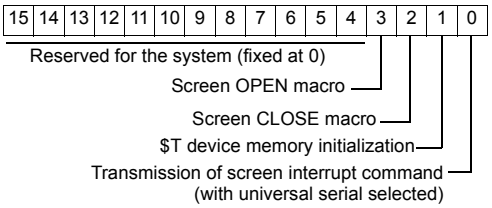
This macro command is used to reset the currently displayed screen. It is convenient for switching languages and for switching the display of screen libraries.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value
F0	RESET_SCRN
F1	Bit OFF (0) : Executed Bit ON (1) : Disabled  * Multiple bits can be specified at the same time.

Example

- $u100 = 0H (W)
  SYS (RESET_SCRN) $u100

  Close macro, open macro, and internal device memory $T are initialized and the screen currently displayed is reset.
- $u100 = CH (W)
  SYS (RESET_SCRN) $u100

  The screen currently displayed is reset without executing close macro and open macro.

Supplemental remarks

- The macro command is invalid in screen OPEN, screen CLOSE, overlap library OPEN, overlap library CLOSE, and initial macros. Executing these error results in failure.
- The macro command is valid only once in a macro created on the macro edit sheet. Its execution timing is set at the end of the macro on the edit sheet.
- When screen internal switching is disabled (the 13th bit of read area n+1 is ON), the macro is invalid.
- [Function: Return] for the switch is valid even after using this command.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (OUT_ENQ) F1

Function 1: Universal serial (interrupt)

This macro command is used to execute an interrupt.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	OUT_ENQ	
F1	10 - 2F $_{HEX}$: ENQ No.	
F1+1	Transfer format 0: Numerical 1: Characteristic	
F1+2	1 - 16384: Word count	If F1+1 = 1
	2 - 32768: Number of bytes	If F1+1 = 1
F1+3	Top address number	
F1+4	0: Non-wait	Executes the next macro
	1: Wait	Executes the next macro after a transmission is complete

Example

The following programs transmit the specified data to the host when the character display ($u200) shows “ABCD.”

- Transfer data format: Numerical
  $u100 = 10H (W) [ENQ No.]
  $u101 = 0 (W) [Numerical]
  $u102 = 2 (W) [Word count]
  $u103 = 200 (W) [Top address]
  $u104 = 0 (W) [Non-wait]
  SYS (OUT_ENQ) $u100 Macro execution

  Data received at the host: 3431343234333434H

- Transfer data format: Characteristic
  $u100 = 10H (W) [ENQ No.]
  $u101 = 1 (W) [Characteristic]
  $u102 = 4 (W) [Number of bytes]
  $u103 = 200 (W) [Top address number]
  $u104 = 0 (W) [Non-wait]
  SYS (OUT_ENQ) $u100

  Data received at the host: 41424344H

Supplemental remarks

- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

Function 2: A-link+Net10 (network designation)

This macro command is used to designate a target network, with which a connection will be established.

All models	○

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	Remarks
F0	OUT_ENQ	
F1	0: Fixed	
F1+1	2: Fixed	
F1+2	System code 1: NET/10 2: NET II (/B)	
F1+3	0: Fixed	If F1+2 = 2
	1: Network number	If F1+2 = 2

Example

- $u100 = 0 (W) [Fixed]
  $u101 = 2 (W) [Fixed]
  $u102 = 1 (W) [NET/10]
  $u103 = 3 (W) [Network number]
  SYS (OUT_ENQ) $u100

  According to the above program, the PLC connected to the V series accesses the PLC NET 10 on network No. 3.

Supplemental remarks

- The macro command is valid when [A-link + Net10] is selected for [Select PLC1 Type].
- Be sure to use the macro command in an OPEN macro for the screen. If it is used in any other way, the network will change immediately after the command is executed and a communication error will result.
- For more information, refer to the V9 Series Connection Manual.
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
−1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

SYS

All models	○

SYS (SET_SYS_CLND) F1

Function: System calendar setting

This macro command is used to set the values of seven words starting from the address specified in [F1] to the system calendar.
The PLC calendar is not changed.

Available device memory

	Internal device memory	PLC 1 - 8 device memory	Memory card	Constant
F1	◎			

○ : Setting enabled (indirect designation disabled)
◎ : Setting enabled (indirect designation enabled)

Setting range

	Value	
F0	SET_SYS_CLND	
F1	0 - : Year (4-digit/2-digit)	
F1+1	1 - 12: Month	
F1+2	1 - 31: Day	
F1+3	0 - 23: Hour	
F1+4	0 - 59: Minute	
F1+5	0 - 59: Second	
F1+6	Day of the week	Invalid; to be automatically calculated by MONITOUCH

Example

- $u100 = 2005 (W)
  $u101 = 7 (W)
  $u102 = 15 (W)
  $u103 = 15 (W)
  $u104 = 0 (W)
  $u105 = 0 (W)

  July 15, 2005 Friday 15:00:00

  SYS (SET_SYS_CLND) $u00100

  The above program sets the calendars in MONITOUCH to July 15, 2005 on Friday at 15:00:00.

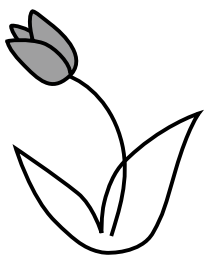
Supplemental remarks

- When setting calendar data for PLC 1 to 8, use a macro command " PLC_CLND" (page 4-88).
- The result of macro execution is stored in $s72.

Code (DEC)	Contents
0*	Normal
–1	Execution error

* Select [System Setting] → [Unit Setting] → [Environment Setting], and check [Store the result as normal upon successful completion of macro execution]. When this box is not checked, the value will not be updated even if execution of the macro is successfully completed. (Under development)

MEMO



Please use this page freely.

Hakko Electronics Co., Ltd.
www.monitouch.com

Sales
890-1, Kamikashiwano-machi, Hakusan-shi,
Ishikawa, 924-0035 Japan
TEL +81-76-274-2144 FAX +81-76-274-5136