YASKAWA

YRC1000/YRC1000micro SUPPLEMENTAL INSTRUCTIONS FOR Smart Pendant

(JZRCR-APP30-1)
(INSTRUCTIONS FOR INFORM LANGUAGE)

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

MOTOMAN INSTRUCTIONS

YRC1000 INSTRUCTIONS
YRC1000 OPERATOR'S MANUAL (GENERAL) (SUBJECT SPECIFIC)
YRC1000 MAINTENANCE MANUAL
YRC1000 ALARM CODES (MAJOR ALARMS) (MINOR ALARMS)

YRC1000micro INSTRUCTIONS
YRC1000micro OPERATOR'S MANUAL
YRC1000micro MAINTENANCE MANUAL
YRC1000micro ALARM CODES (MAJOR ALARMS) (MINOR ALARMS)

YRC1000/YRC1000micro INSTRUCTIONS FOR Smart Pendant (JZRCR-APP30-1)

Have the following information available when contacting the YASKAWA Representative:
- System
- Primary Application
- Software Version *(Located on Programming Pendant by selecting: {Main Menu} - {System Info} - {Version})*
- Warranty ID *(Located on Robot Controller)*
- Robot Serial Number *(Located on Manipulator data plate)*
- Robot Sales Order Number *(Located on Robot controller data plate)*

The Americas YASKAWA Representative

- 24-hour Telephone Number: **(937) 847-3200**
  Use for urgent or emergency needs for technical support, service and/or replacement parts
- Routine Technical Inquiries: *techsupport@motoman.com*
  Allow up to 36 hours for response

Part Number: 184774-1CD
Revision: 7

MANUAL NO.
HW1485511 ⑦

DANGER

- This manual explains the INFORM language of the YRC Controller. Read this manual carefully and be sure to understand its contents before handling the YRC Controller. Any matter, including operation, usage, measures, and an item to use, not described in this manual must be regarded as “prohibited” or “improper”.
- General information related to safety are described in “Chapter 1. Safety” of “YRC CONTROLLER INSTRUCTIONS”. To ensure correct and safe operation, carefully read “Chapter 1. Safety” of “YRC CONTROLLER INSTRUCTIONS”.

CAUTION

- In some drawings in this manual, protective covers or shields are removed to show details. Make sure that all the covers or shields are installed in place before operating this product.
- YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids the product warranty.

NOTICE

- The drawings and photos in this manual are representative examples and differences may exist between them and the delivered product.
- YASKAWA may modify this model without notice when necessary due to product improvements, modifications, or changes in specifications. If such modification is made, the manual number will also be revised.
- If your copy of the manual is damaged or lost, contact your local YASKAWA representative to order a new copy. Be sure to tell the representative the manual number listed on the front cover.

NOTE

- This instruction manual is applicable to both the YRC1000 and YRC1000micro controlller.
- The description of “YRC Controller” refers to “YRC1000” and “YRC1000micro” in this manual unless otherwise specified.

Notes for Safe Operation

Read this manual carefully before installation, operation, maintenance, or inspection of the YRC Controller

In this manual, the Notes for Safe Operation are classified as “DANGER”, “WARNING”, “CAUTION”, or “NOTICE”.

Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

Safety Signs identified by the signal word DANGER should be used sparingly and only for those situations presenting the most serious hazards.

Indicates a potentially hazardous situation which, if not avoided, will result in death or serious injury.

Hazards identified by the signal word WARNING present a lesser degree of risk of injury or death than those identified by the signal word DANGER.

Indicates a hazardous situation, which if not avoided, could result in minor or moderate injury. It may also be used without the safety alert symbol as an alternative to “NOTICE”.

NOTICE is the preferred signal word to address practices not related to personal injury. The safety alert symbol should not be used with this signal word. As an alternative to “NOTICE”, the word “CAUTION” without the safety alert symbol may be used to indicate a message not related to personal injury.

Even items described as “CAUTION” may result in a serious accident in some situations.

At any rate, be sure to follow these important items.

To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as “DANGER”, “WARNING” and “CAUTION”.

<YRC1000>

DANGER

- Before operating the manipulator, make sure the servo power is turned OFF by performing the following operations. When the servo power is turned OFF, the SERVO ON LED on the Smart Pendant is turned OFF.
  - Press the emergency stop buttons on the front door of the YRC1000, on the top of the Smart Pendant, on the external control device, etc.
  - Disconnect the safety plug of the safety fence. (when in PLAY (AUTOMATIC) mode or in REMOTE mode)

If operation of the manipulator cannot be stopped in an emergency, personal injury and/or equipment damage may result.

Fig. : Emergency Stop Button

- Before releasing the emergency stop, make sure to remove the obstacle or error caused the emergency stop, if any, and then turn the servo power ON.

Failure to observe this instruction may cause unintended movement of the manipulator, which may result in personal injury.

Fig. : Release of Emergency Stop

- Observe the following precautions when performing a teaching operation within the manipulator's operating range:
  - Be sure to perform lockout by putting a lockout device on the safety fence when going into the area enclosed by the safety fence. In addition, the operator of the teaching operation must display the sign that the operation is being performed so that no other person closes the safety fence.
  - View the manipulator from the front whenever possible.
  - Always follow the predetermined operating procedure.
  - Always keep in mind emergency response measures against the manipulator’s unexpected movement toward a person.
  - Ensure a safe place to retreat in case of emergency.

Failure to observe this instruction may cause improper or unintended movement of the manipulator, which may result in personal injury.

- Confirm that no person is present in the manipulator's operating range and that the operator is in a safe location before:
  - Turning ON the YRC1000 power
  - Moving the manipulator by using the Smart Pendant
  - Running the system in the check mode
  - Performing automatic operations

Personal injury may result if a person enters the manipulator's operating range during operation. Immediately press an emergency stop button whenever there is a problem. The emergency stop buttons are located on the front panel of the YRC1000 and on the upper right of the Smart Pendant.

- Read and understand the Explanation of the Warning Labels before operating the manipulator

<YRC1000micro>

DANGER

- Before operating the manipulator, make sure the servo power is turned OFF by performing the following operations. When the servo power is turned OFF, the SERVO ON LED on the Smart Pendant is turned OFF.
  - Press the emergency stop button on the top of the Smart Pendant or on the external control device, etc.
  - Disconnect the safety plug of the safety fence. (when in play mode or in the remote mode)

If operation of the manipulator cannot be stopped in an emergency, personal injury and/or equipment damage may result.

Fig. : Emergency Stop Button

- Before releasing the emergency stop, make sure to remove the obstacle or error caused the emergency stop, if any, and then turn the servo power ON.

Failure to observe this instruction may cause unintended movement of the manipulator, which may result in personal injury.

Fig. : Release of Emergency Stop

- Observe the following precautions when performing a teaching operation within the manipulator's operating range:
  - Be sure to perform lockout by putting a lockout device on the safety fence when going into the area enclosed by the safety fence. In addition, the operator of the teaching operation must display the sign that the operation is being performed so that no other person closes the safety fence.
  - View the manipulator from the front whenever possible.
  - Always follow the predetermined operating procedure.
  - Always keep in mind emergency response measures against the manipulator’s unexpected movement toward a person.
  - Ensure a safe place to retreat in case of emergency.

Failure to observe this instruction may cause improper or unintended movement of the manipulator, which may result in personal injury.

- Confirm that no person is present in the manipulator's operating range and that the operator is in a safe location before:
  - Turning ON the YRC1000micro power
  - Moving the manipulator by using the Smart Pendant
  - Running the system in the check mode
  - Performing automatic operations

Personal injury may result if a person enters the manipulator's operating range during operation. Immediately press an emergency stop button whenever there is a problem. The emergency stop button is located on the top right of the Smart Pendant.

- Read and understand the Explanation of the Warning Labels before operating the manipulator.

<YRC1000micro only>

DANGER

- In the case of not using the Smart Pendant, be sure to supply the emergency stop button on the equipment. Then before operating the manipulator, check to be sure that the servo power is turned OFF by pressing the emergency stop button.
  Connect the external emergency stop button to the 4-14 pin and 5-15 pin of the Safety connector (Safety).
- Upon shipment of the YRC1000micro, this signal is connected by a jumper cable in the safety signal short circuit connector. To use the signal, make sure to supply a new connector, and then input it.

If the signal is input with the jumper cable connected, it does not function, which may result in personal injury or equipment damage.

<YRC1000/YRC1000micro>

WARNING

- Perform the following inspection procedures prior to conducting manipulator teaching. If there is any problem, immediately take necessary steps to solve it, such as maintenance and repair.
  - Check for a problem in manipulator movement.
  - Check for damage to insulation and sheathing of external wires.
- Always return the Smart Pendant to a safe place after use.

If the Smart Pendant is left unattended on the manipulator, on a fixture, or on the floor, etc., the Enable Switch may be activated due to surface irregularities of where it is left, and the servo power may be turned ON. In addition, in case the operation of the manipulator starts, the manipulator or the tool may hit the Smart Pendant left unattended, which may result in personal injury and/or equipment damage.

General Safety on the Smart Pendant

CAUTION

- Be careful not to drop the Smart Pendant on the floor.
- Pay attention to the handling of the cable so that it will not stumble on the Smart Pendant cable.
- Handle carefully so as not to damage the cable.
- Do not place the Smart Pendant with the touch screen facing down.
- Do not place the Smart Pendant close to a heat source or in direct sunlight.
- Do not place the Smart Pendant in an environment with excessive dust, humidity or strong magnetic fields.
- Pay attention not to adhere chemicals, cutting oil (including coolant), rust preventive oil, organic solvent etc. to the Smart Pendant.
- Do not clean the Smart Pendant with scrubbing sponges. Use a soft cloth and a little water or diluted neutral detergent (mild cleaning liquids).
- Operate the touch screen with your fingers, or use a touch-pen designed for use with capacitive screens. Never use sharp objects (ex. screwdriver) for operating the touch screen. This could damage the touch screen.
- Be careful not to get foreign objects or liquid from the USB connector.
- Do not connect the Smart Pendant to anything except for the YRC Controller.

Definition of Terms Used Often in This Manual

<YRC1000>

The MOTOMAN is a YASKAWA industrial robot product.

The MOTOMAN usually consists of the manipulator, the YRC1000 Controller, the Smart Pendant, and supply cables.

In this manual, the equipment is designated as follows:

Equipment	Manual Designation
YRC1000 Controller	YRC1000
YRC Controller Smart Pendant	Smart Pendant
Cable between the manipulator and the YRC Controller	Manipulator cable

<YRC1000micro>

The MOTOMAN is the YASKAWA industrial robot product.

The MOTOMAN usually consists of the manipulator, the YRC1000micro controller, manipulator cables, the YRC1000micro Smart Pendant, and the YRC1000micro Smart Pendant safety signal short circuit connector (optional).

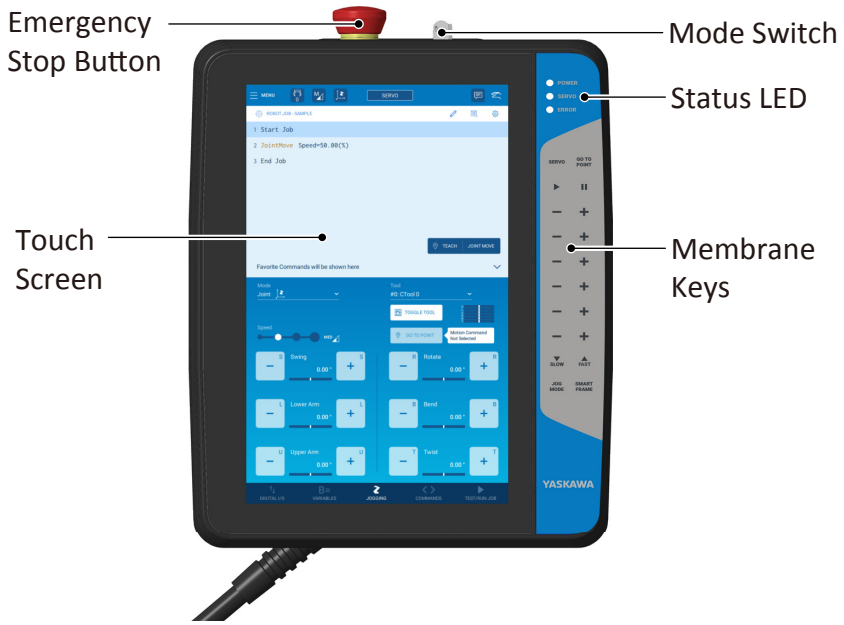
In this manual, the equipment is designated as follows:

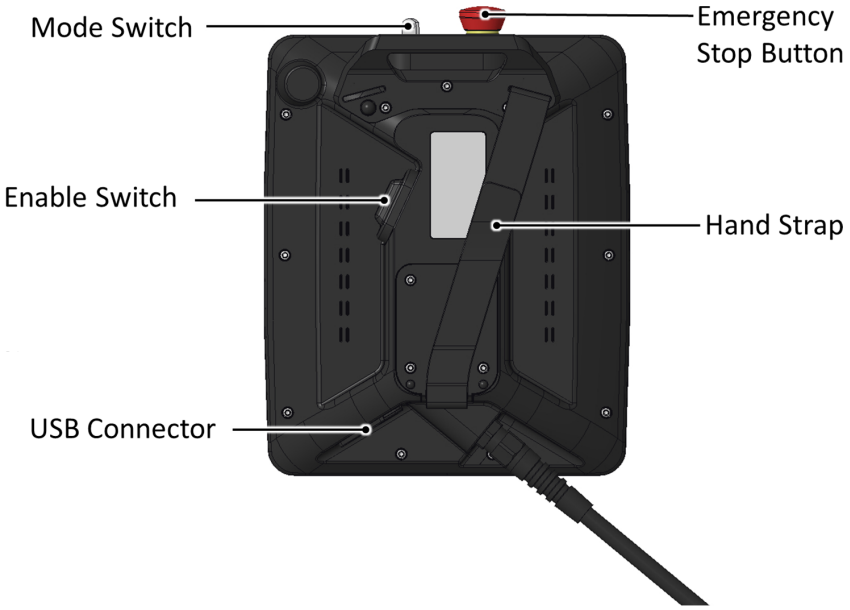
Equipment	Manual Designation
YRC1000micro Controller	YRC1000micro
YRC1000micro Smart pendant	Smart Pendant
Cable between the manipulator and the YRC Controller	Manipulator cable
YRC1000micro Smart pendant safety signal short circuit connector	YRC1000micro Smart Pendant safety signal short circuit connector (optional)

<YRC1000/YRC1000micro>

Descriptions of the Smart Pendant, buttons, and displays are shown as follows:

Equipment		Manual Designation
Smart Pendant	Emergency Stop button	This button on the Smart Pendant will be denoted as Emergency Stop button
	Mode Switch	Three kinds of modes that can be selected by the mode switch are denoted as follows: REMOTE, PLAY (AUTOMATIC), or TEACH (MANUAL)
	Displays	The buttons and items displayed in the Smart Pendant is denoted with { }. ex. {Save}
	Status LED	These LED indicators will be denoted as: POWER LED, SERVO LED, or ERROR LED
	Membrane Key	The membrane keys are denoted with [ ]. ex. [JOG MODE]
	Jog Keys	“Jog Keys” is a generic names for the jog operation keys.
	Keys pressed simultaneously (for membrane key only)	When two keys are to be pressed simultaneously, the keys are shown with a “+” sign between them, ex. [S+] + [L+].





Description of the Operation Procedure

<YRC1000/YRC1000micro>

In the explanation of the operation procedure, the expression “Select” means that the item is directly selected by touching the screen.

Registered Trademark

<YRC1000/YRC1000micro>

In this manual, names of companies, corporations, or products are trademarks, registered trademarks, or brand names for each company or corporation. The indications of (R) and ^{TM} are omitted.

Contents

1 Outline of INFORM 1-1
1.1 About INFORM 1-1
1.1.1 INFORM 1-1
1.1.2 Type of Instruction 1-1
1.1.3 Variable 1-2
1.1.3.1 Set Value of by Numeric Data 1-2
1.1.3.2 Array Index 1-2
1.1.3.3 Local Variable 1-3
1.1.4 INFORM Display Settings 1-4
1.1.5 Detail INFORM Substitution 1-8
1.2 INFORM Syntax 1-9
1.2.1 Example of Syntax 1-9
1.2.2 INFORM Syntax Elements 1-10
1.2.3 Meaning of INFORM Syntax 1-10
1.2.4 Explanation Table 1-11
1.3 Expressions 1-12
1.3.1 Set Expression Editor 1-12
1.3.2 Parameter Editor 1-13
1.3.3 Syntax Checking 1-15

2 INFORM Explanation 2-1
2.1 General Instruction 2-1
// (Comment) 2-1
SetShockDetection 2-2
ResetShockDetection 2-5
AdvInit 2-6
SkillSend 2-7
2.2 Move Instruction 2-8
JointMove 2-8
LinearMove 2-13
CircleMove 2-17
SplineMove 2-22
Incremental 2-26
SetSpeed 2-29
ShiftOn 2-31
ShiftOff 2-33
ShiftOn3D 2-34
ShiftOff3D 2-34

MakeShift 2-35
Smooth 2-38

2.3 I/O Instructions 2-40

DigitalOut 2-40
DigitalIn 2-43
PulseOut 2-46
Wait 2-49
AnalogOut 2-53

2.4 Math Instruction 2-54

Set 2-54
Increment 2-60
Decrement 2-61
Add 2-62
Subtract 2-67
Multiply 2-72
Divide 2-77
Clear 2-82
Convert 2-84
And 2-86
Or 2-87
Not 2-88
Xor 2-89
SetElement 2-90
GetElement 2-92
GetSystemVar 2-94
SquareRoot 2-100
Sine 2-101
Cosine 2-102
ArcTangent 2-103
MultiplyMatrix 2-104
InvertMatrix 2-106
GetPosition 2-107
MakeFrame 2-108
StoreStrPos 2-110

2.5 Control Instruction 2-112

Label *(Label) 2-112
Jump 2-113
IfThen 2-114
ElseIf 2-115
Else 2-116
While 2-118
For 2-119
Call 2-120
GetArgument 2-125
Return 2-127

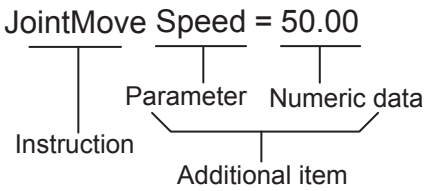
Timer 2-129
Pause 2-130
Abort 2-131
SetUserAlarm 2-132
Switch 2-134
Case 2-135
Default 2-136
StartParallelJob 2-138
WaitForParallelJob 2-139
ThreadSync 2-140
2.6 Instruction Which Adheres to an Instruction 2-141
Until 2-141

1 Outline of INFORM

1.1 About INFORM

1.1.1 INFORM

The robot programming language used with Smart Pendant is called INFORM. INFORM is composed of the instruction and the additional item (parameter and numeric data).



- **Instruction:** It is used to execute the operation and processing. In the case of a move instruction, when a position is taught, the move instruction is automatically displayed according to the interpolation method.
- **Additional item:** The speed, time, etc. are set according to the type of instruction. Numeric data and character data are added to the parameter that specifies the condition as necessary.

1.1.2 Type of Instruction

The instruction is divided into several types in terms of each process and operation.

Type	Content	Instruction Example
General Instruction	It is an instruction used for specific features	Comment
Move Instruction	It is an instruction concerning the movement and the speed.	JointMove, LinearMove
I/O Instruction	It is the instruction used to control the I/O.	DigitalOut, PulseOut
Math Instruction	It is the instruction by which the variables, etc. are used and operated.	Add, Set
Control Instruction	It is the instruction used to control the processing and operation.	Jump, For

1.1.3 Variable

Smart Pendant can use the following variables as the parameters of the command.

Variable	Symbol
Byte Variable	B, B[]
Integer Variable	I, I[]
Double Integer Variable	D,D[]
Real Number Variable	R,R[]
String Variable	S,S[]
Position Variable	P,P[]

1.1.3.1 Set Value of by Numeric Data

The unit of the numeric data for the additional item of the instruction decides the set value of variable and the value of the additional item at execution.

< Example >
Timer Time=I001
(I001=1000)

When a variable is used for the numeric data of the Time parameter, the unit of numeric data is 0.01 seconds for the YRC1000 and 0.001 seconds for the YRC1000micro. If I001 is 1000, the pause time will be 10 seconds for the YRC1000 and 1 second for the YRC1000micro.

1.1.3.2 Array Index

The variable index can also be set as an array.

<Example>
Timer Time=I[B002]
(B002=3, I003=2000)

In this case, I[B002]=I003, and the pause time will be 20 seconds for the YRC1000 and 2 seconds for the YRC1000micro.

1.1.3.3 Local Variable

Local Variable has the same data format as User Variable and are prefixed with an “L”. For example, “LB” denotes a local version of a Byte (B) variable, and “LD” denotes a local version of a Double (D) variable. The table below shows all types of Local Variables. Local Variable is available for an instruction parameter whenever the corresponding User Variable is available. For example, the PositionLevel parameter of JointMove can be inserted using User Variable of types “B/B[]/I/I[]/D/D[]”, so Local Variables of types “LB/LB[]/LI/LI[]/LD/LD[]” can also be used.

Local Variable	Symbol
Byte Variable	LB, LB[]
Integer Variable	LI, LI[]
Double Integer Variable	LD,LD[]
Real Number Variable	LR, LR[]
String Variable	LS, LS[]
Position Variable	LP, LP[]

NOTE The number of Local Variables to be used must be set in the Job List. For setting the number of Local Variables, refer to “Additional Settings” in YRC1000/YRC1000micro INSTRUCTIONS FOR Smart Pendant (HW1485509).

1.1.4 INFORM Display Settings

Smart Pendant introduces a new representation of INFORM, referred to as Detail INFORM. Detail INFORM uses descriptive words for commands and parameters as compared to INFORM that uses abbreviated terms. The representation can easily be switched by selecting Classic View in Display Settings.

<Example>
INFORM:

```
Start Job
MOVJ P001 VJ=50.00 PL=1 ACC=50
MOVL P002 VR=30.00 CR=20.00
SETE P003 (1) 0
DOUT OT#(1) ON
End Job
```

Detail INFORM:

```
Start Job
Timer Time=0.50(seconds)
JointMove P001 Speed=50.00(%) PositionLevel=1 Acceleration=50(%)
LinearMove P002 RotationSpeed=30.00(deg/sec) CornerRadius=20.00
SetElement P003 (1) 0
DigitalOut Output#(1) ON
End Job
```

Following table provides a comparison between the two INFORM representations:

General Commands	
Detail INFORM	**INFORM**
Comment	COMMENT
SetShockDetection	SHCKSET
ResetShockDetection	SHCKRST
AdvInit	ADVINIT
SkillSend	SKILLSND

1 Outline of INFORM
1.1 About INFORM

Motion Commands	
Detail INFORM	**INFORM**
JointMove	MOVJ
LinearMove	MOVL
CircleMove	MOVC
SplineMove	MOVS
Incremental	IMOV
SetSpeed	SPEED
ShiftOn	SFTON
ShiftOff	SFTOF
ShiftOn3D	SFTON3D
ShiftOff3D	SFTOF3D
MakeShift	MSHIFT
Smooth	SMOOTH

I/O Commands	
Detail INFORM	**INFORM**
DigitalOut	DOUT
DigitalIn	DIN
PulseOut	PULSE
Wait	WAIT
AnalogOut	AOUT

Math Commands	
Detail INFORM	INFORM
Set	SET
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Clear	CLEAR
Convert	CNVRT
And	AND
Or	OR
Not	NOT
Xor	XOR
SetElement	SETE
GetElement	GETE
GetSystemVar	GETS
SquareRoot	SQRT
Sine	SIN
Cosine	COS
ArcTangent	ATAN
MultiplyMatrix	MULMAT
InvertMatrix	INVMAT
GetPosition	GETPOS
MakeFrame	MFRAME
StoreStrPos	STRSTR

Control Commands	
Detail INFORM	**INFORM**
Label	LABEL
Jump	JUMP
IfThen	IFTHEN
ElseIf	ELSEIF
Else	ELSE
While	WHILE
For	FOR
Call	CALL
GetArgument	GETARG
Return	RET
Timer	TIMER
Pause	PAUSE
Abort	ABORT
SetUserAlarm	SETUALM
Switch	SWITCH
Case	CASE
Default	DEFAULT
StartParalleJob	PSTART
WaitForParallalJob	PWAIT
ThreadSync	TSYNC

Parameter	
Detail INFORM	**INFORM**
Time=x,I,I[]	T=x,I,I[]
UserFrame=x,B,D,I	UF#(x,B,D,I)
Level=x	LEVEL=x
Speed=x,B,I,D,B[],I[],D[]	VJ/V=x,B,I,D,B[],I[],D[]
RotationSpeed=x,D,D[]	VR=x,D,D[]
PositionLevel=x,B,I,D,B[],I[],D[]	PL=x,B,I,D,B[],I[],D[]
CornerRadius=x,D,D[]	CR=x,D,D[]
Acceleration=x,B,I,D,B[],I[],D[]	ACC=x,B,I,D,B[],I[],D[]
Deceleration=x,B,I,D,B[],I[],D[]	DEC=x,B,I,D,B[],I[],D[]
Output#(x,B,I,D,B[],I[],D[])	OT#(x,B,I,D,B[],I[],D[])
OutputGroup#(x,B,I,D,B[],I[],D[])	OG#(x,B,I,D,B[],I[],D[])
OutputGroupHalf#(x,B,I,D,B[],I[],D[])	OGH#(x,B,I,D,B[],I[],D[])
Input#(x,B,I,D,B[],I[],D[])	IN#(x,B,I,D,B[],I[],D[])
InputGroup#(x,B,I,D,B[],I[],D[])	IG#(x,B,I,D,B[],I[],D[])
SpecificIn#(x,B,I,D,B[],I[],D[])	SIN#(x,B,I,D,B[],I[],D[])
SpecificOut#(x,B,I,D,B[],I[],D[])	SOUT#(x,B,I,D,B[],I[],D[])
AnalogOutput#(x,B,I,D)	AO#(x,B,I,D)

1.1.5 Detail INFORM Substitution

Smart Pendant uses a subset of the INFORM language. Some instructions are removed as they are legacy and alternatives are provided and some instructions are not supported in the current version. The Table below will highlights missing instructions and presents alternatives:

INFORM	Detail INFORM

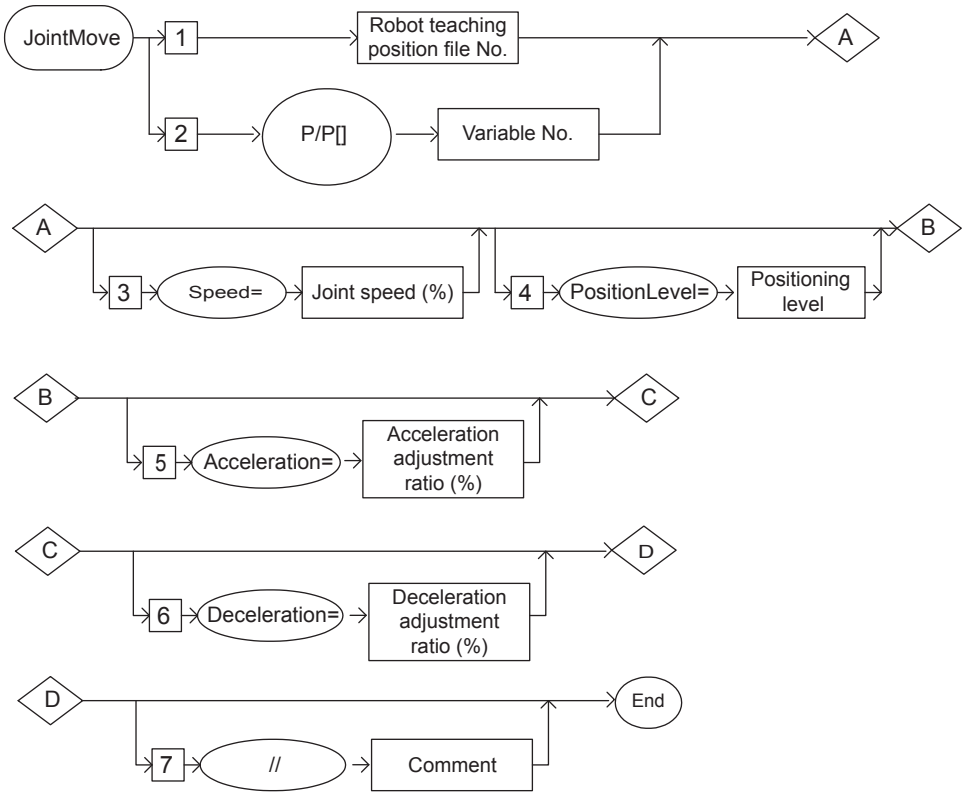
IF statements:
 	
JUMP *LABEL IF B000>5
	Place entire statement inside an IFTHEN/ENDIF block:
	
	If (B000>5) Then
	Jump *LABEL
	EndIf

1.2 INFORM Syntax

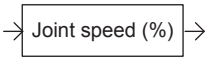
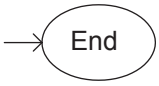
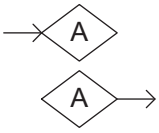
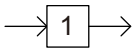
An example of the INFORM syntax is shown in the following syntax flowchart.

The INFORM syntax chart is composed of the syntax elements (instruction, parameter, and data). The order of the rows are shown with the numbers and arrows.

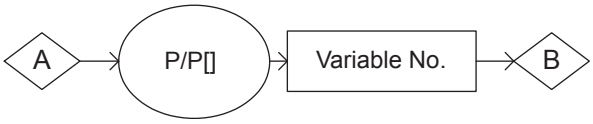
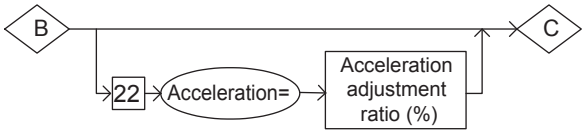
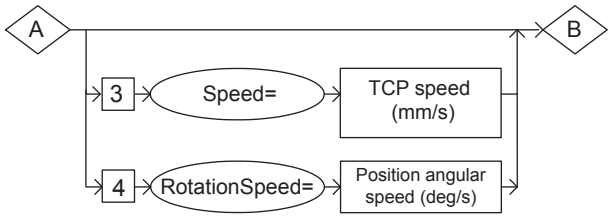
1.2.1 Example of Syntax



1.2.2 INFORM Syntax Elements

INFORM Syntax Element	Explanation	Note
	Indicates the instruction.	In this example, the “JointMove” instruction is indicated.
	Indicates the parameter.	In this example, the “Speed=” parameter is indicated.
	Indicates the numeric data.	In this example, “Joint speed” is set with the unit %.
	Indicates the end of the instruction.	
	Indicates the connection.	
	Indicates the parameter order.	

1.2.3 Meaning of INFORM Syntax

INFORM Syntax	Meaning
	This is an indispensable parameter. In this example, it is necessary to add a parameter from P Variable /P [Array]
	This is a parameter that can be omitted.And if omitted, a default value will be used. In this example, the Acceleration parameter can be omitted. And the default value will be 100%.
	This is a parameter that can be selected. In this example, either Speed=TCP speed or RotationSpeed=Position angular speed can be selected, or can be omitted.

1.2.4 Explanation Table

The explanation table in this manual can be described as follows.

No	Parameter	Explanation	Note
1	Output# (Output number)	Specify the output number signal.	No:1 to 4096 Variable B/I/D can be used.

- No
  Indicates the parameter number. Corresponds to the number in the INFORM syntax.

- Parameter
  Indicates the description of the parameter.

- Explanation
  Provides an explanation of the parameter.

1.3 Expressions

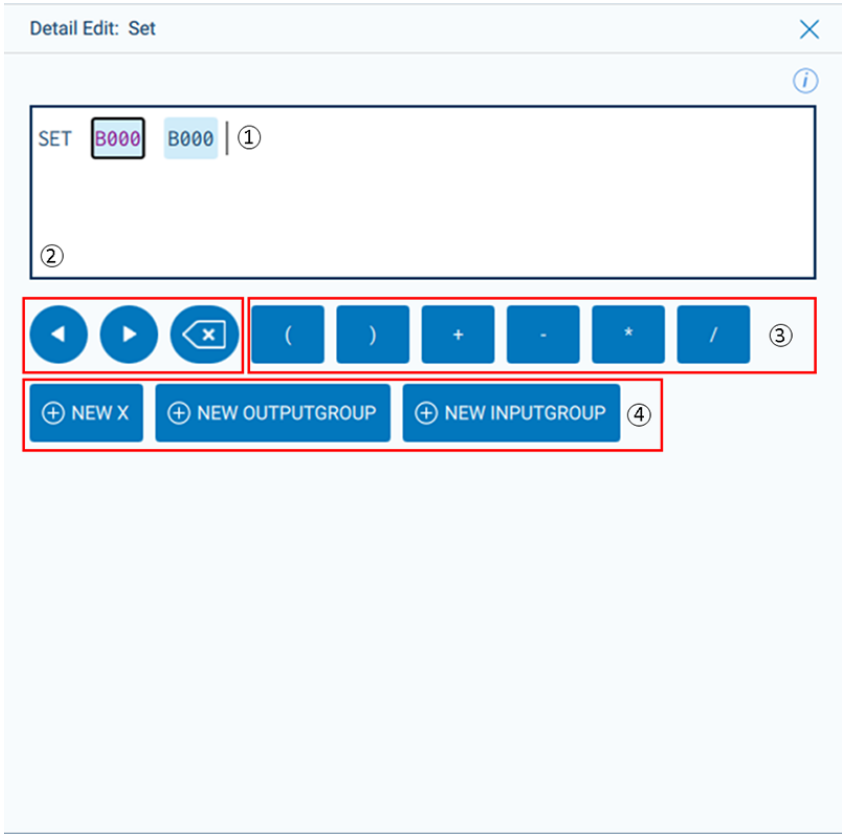
The Set Instruction can be used to assign variable parameters from Constants, other Variables, or Expressions. Using Expressions allows for complex operations to be programmed in a single line. For example, the line below is equivalent to B000 = (B001+B002)*5.

```
2     Set B000 (B001 + B002) * 5
```

1.3.1 Set Expression Editor

The Set Instruction has a custom editor, shown in *fig. 1-1*. The main areas of operation are described below.

Fig. 1-1: Set Expression Editor



① Display of the current Set instruction. If a large expression is entered, it will wrap to multiple lines.

② These buttons can be used to move the cursor position to the left or right as well as delete items from the instruction.

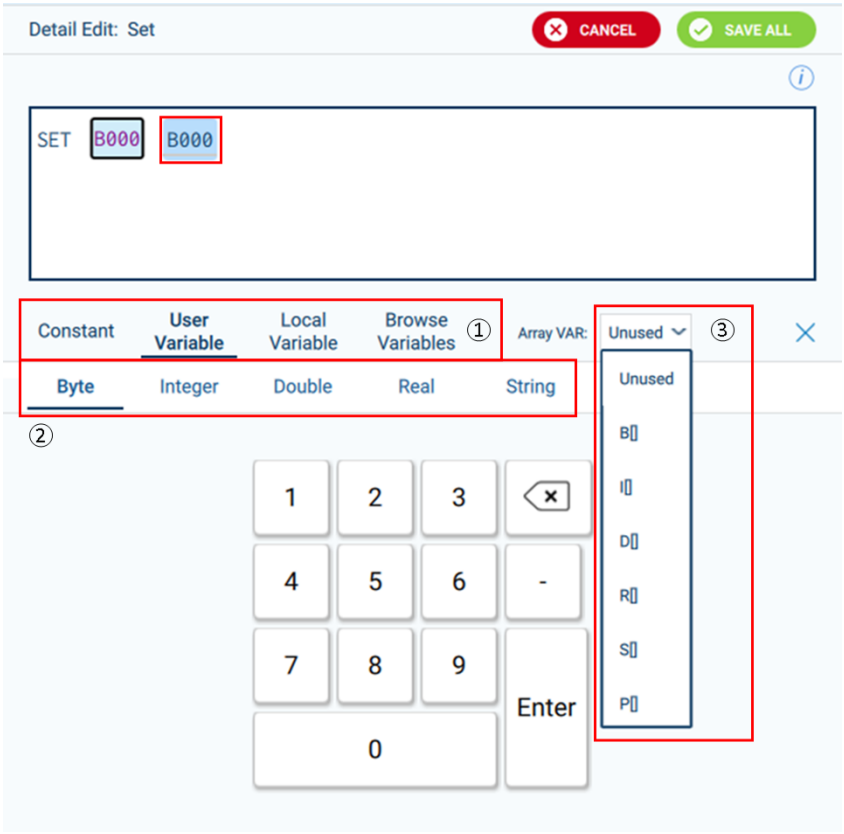
③ These buttons can be used to add new mathematic operators to an expression (open/close parentheses, add, subtract, multiply, divide).

④ These buttons can be used to add new Constant/Variable {+ NEW X} or add an OutputGroup/InputGroup to the instruction.

1.3.2 Parameter Editor

If a parameter inside the Set instruction is clicked (e.g. “B000” in *fig. 1-2*), the Parameter Editor opens:

Fig. 1-2: Set Parameter Editor



① Use the top bar to select the type of parameter to enter. Options are:

- Constant - a constant value such as “5”, “-2.6”, “1.25e7”, etc.
- User Variable - a global variable such as B004, I123, R055, etc.
- Local Variable - a local variable such as LB001, LD004, etc.

NOTE Only Local Variable types that are allocated for the current job displays here

- Browse Variable - browse user and local variables by name to select. See *fig. 1-3*.

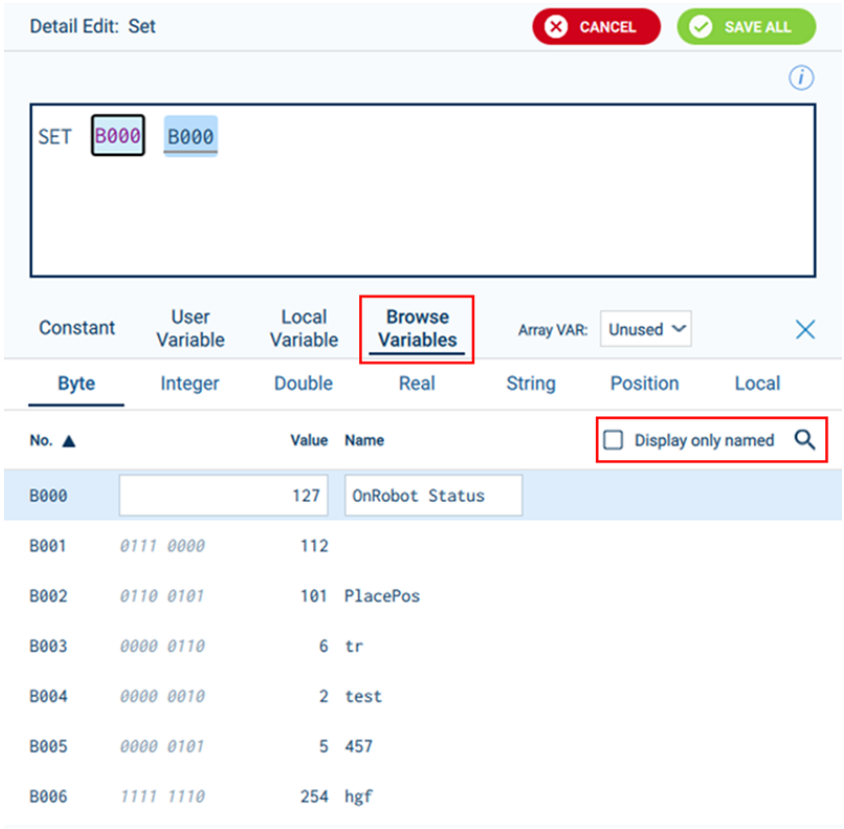
② These buttons can be used to select the type of Variable after selecting “User Variables” or “Local Variables” above.

NOTE If selecting Constant from the top bar, then this bar will not show.

③ Use this drop-down to select an Array Variable such as B[], I[], etc.

When {Browse Variables} is selected, the panel below is shown. This allows a user to look through a list to pick which variable they want to use. The “Display Only Named” and search icon (highlighted below) can make finding the correct variable easier.

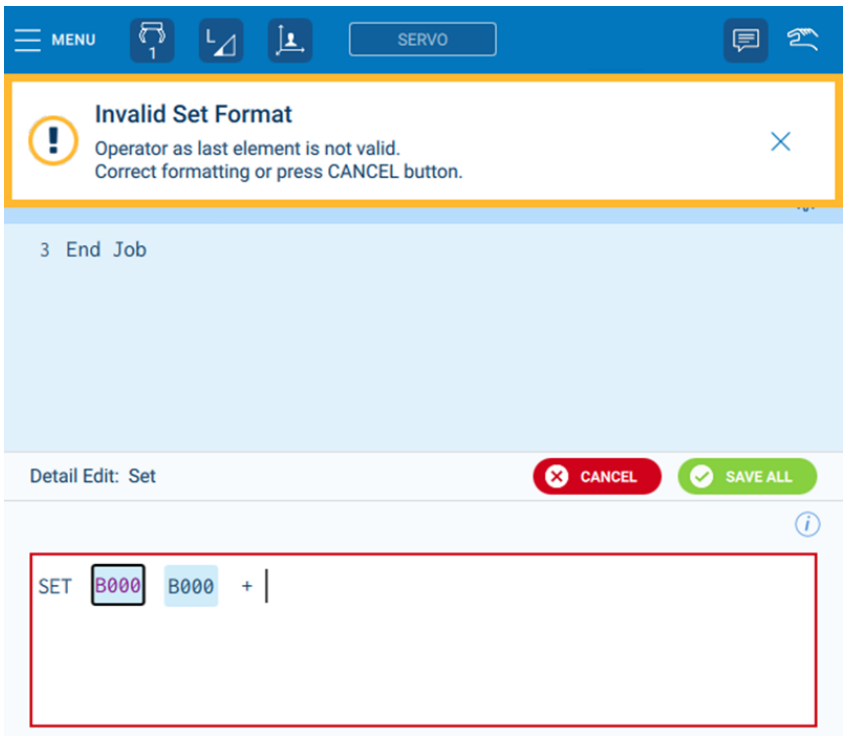
Fig. 1-3: Browse Variables



1.3.3 Syntax Checking

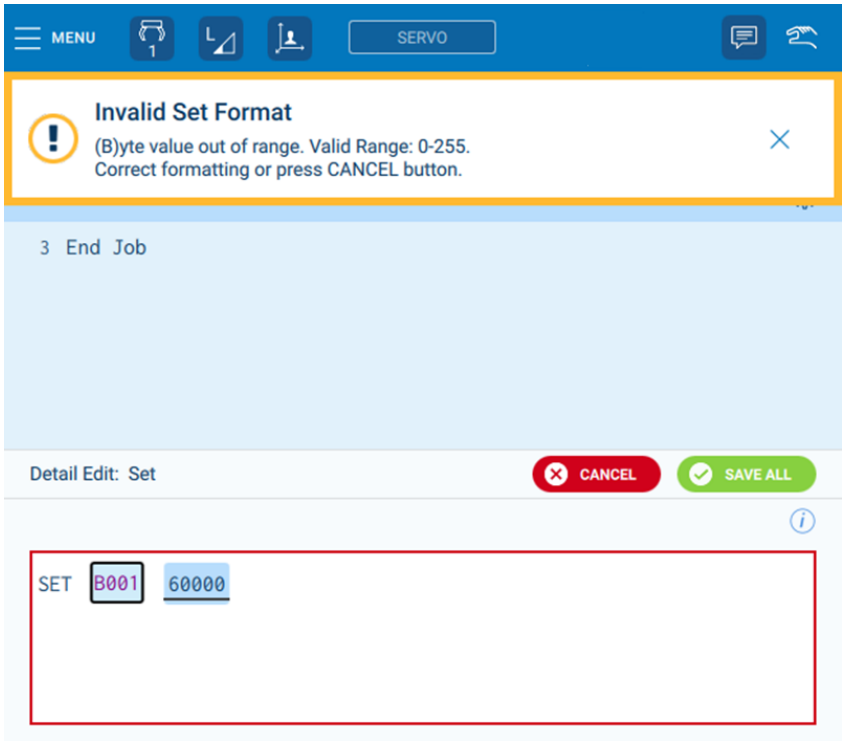
If the expression is edited and a syntax error is detected, the expression rectangle becomes red. A message describing the error displays when {Save All} is pressed, as shown in *fig. 1-4*.

Fig. 1-4: Syntax Error Message



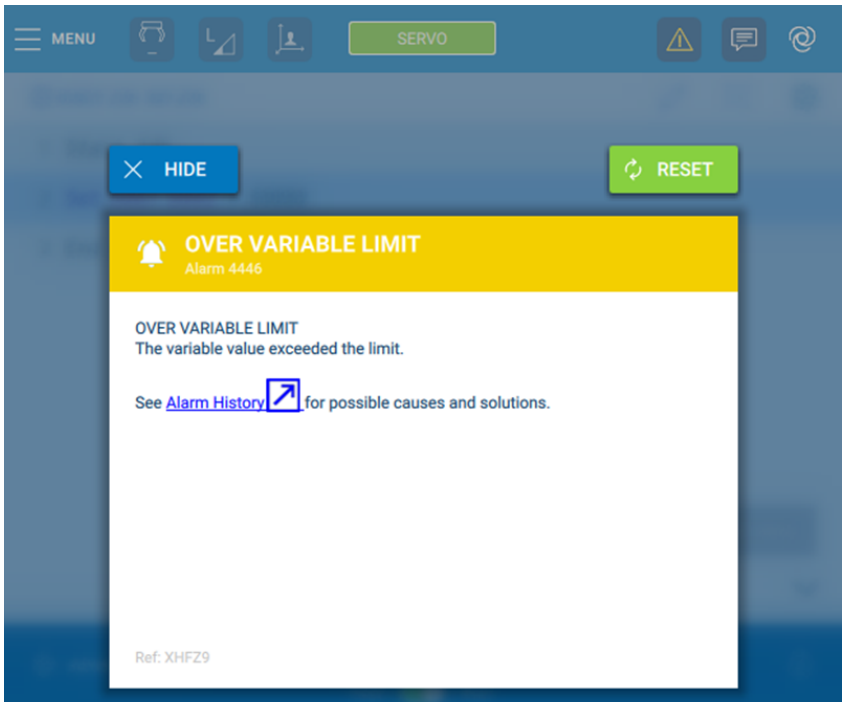
When there is one variable parameter and one constant, the constant value is validated against the range for the variable type. If it is not valid, the expression rectangle becomes red and an error message is displayed when {Save All} is pressed. For example, *fig. 1-5* shows the error message when setting a (B)yte value outside of its allowed range of 0-255

Fig. 1-5: Constant Value Range Check



If there are more parameters, the constant is not range checked. This can cause an alarm when the job executes if the expression results in a value outside the allowed range for the destination variable type, as shown in *fig. 1-6*.

Fig. 1-6: Variable Limit Run Time Error



2 INFORM Explanation

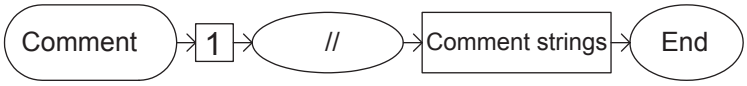
2.1 General Instruction

// (Comment)

Function

Specify the comment.

Syntax



Explanation

1. // Comment strings

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
1	// Comment strings	Specify the comment.	String: 32 characters

Example

(1) Start Job

```
// Waiting Position
JointMove Speed=100.00
JointMove Speed=25.00
// Waiting Position
JointMove Speed=100.00
End Job
```

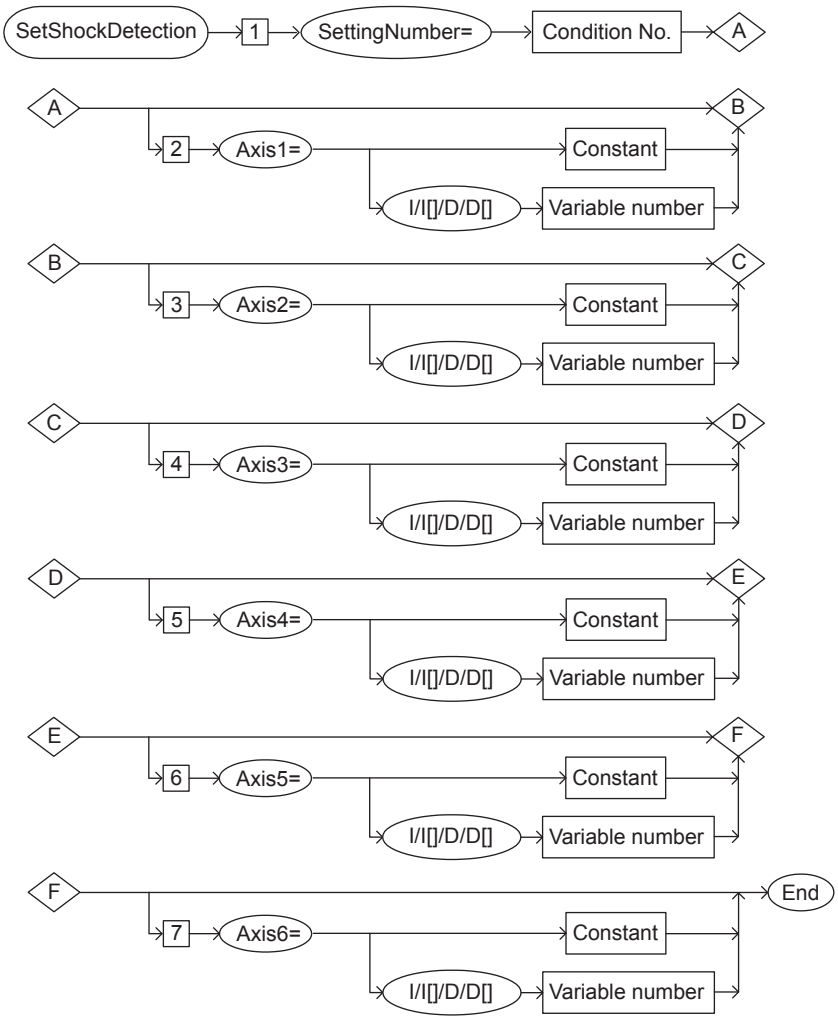
The comment clarifies the job content.

SetShockDetection

Function

Changes the shock detection level to the value set in the specified condition file.

Syntax



Explanation

1. SettingNumber

The following parameter must be added.

No	Parameter	Explanation	Note
1	Condition number	Specify the shock detection level condition number in which the detection level in playback mode is set.	Number: 1 to 7

2. Axis1

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
2	[Constant] I Variable number/ I [Array Number]/ D Variable number/ D [Array Number]	Specify the change of the shock detection level for axis 1 (S-axis for the manipulator with six axes). If the shock detection level is not specified, the level will be the detection level specified in the shock detection level condition number.	Setting level range: 1 to 500

3. Axis2

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
3	[Constant] I Variable number/ I [Array Number]/ D Variable number/ D [Array Number]	Specify the change of the shock detection level for axis 2 (L-axis for the manipulator with six axes). If the shock detection level is not specified, the level will be the detection level specified in the shock detection level condition number.	Setting level range: 1 to 500

4. Axis3

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
4	[Constant] I Variable number/ I [Array Number]/ D Variable number/ D [Array Number]	Specify the change of the shock detection level for axis 3 (U-axis for the manipulator with six axes). If the shock detection level is not specified, the level will be the detection level specified in the shock detection level condition number.	Setting level range: 1 to 500

5. Axis4

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
5	[Constant] I Variable number/ I [Array Number]/ D Variable number/ D [Array Number]	Specify the change of the shock detection level for axis 4 (R-axis for the manipulator with six axes). If the shock detection level is not specified, the level will be the detection level specified in the shock detection level condition number.	Setting level range: 1 to 500

6. Axis5

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
6	[Constant] I Variable number/ I [Array Number]/ D Variable number/ D [Array Number]	Specify the change of the shock detection level for axis 5 (B-axis for the manipulator with six axes). If the shock detection level is not specified, the level will be the detection level specified in the shock detection level condition number.	Setting level range: 1 to 500

7. Axis6

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
7	[Constant] I Variable number/ I [Array Number]/ D Variable number/ D [Array Number]	Specify the change of the shock detection level for axis 6 (T-axis for the manipulator with six axes). If the shock detection level is not specified, the level will be the detection level specified in the shock detection level condition number.	Setting level range: 1 to 500

Example

SetShockDetection SettingNumber=1
The shock detection level condition number 1 is used for detecting the shock.

SetShockDetection SettingNumber=1 Axis4=120 Axis5=120 Axis6=120
The shock detection level condition number 1 is used. Since the shock detection level is not specified for Axis1 (S-axis), Axis2 (L-axis) and Axis3 (U-axis), the level will be the shock detection level specified in the shock detection level condition number. The shock detection level for Axis4 (R-axis), Axis5 (B-axis) and Axis6 (T-axis) are overridden as 120.

ResetShockDetection

Function

Reset shock detection level to the standard (default) value.

Syntax

Example

```
SetShockDetection   SettingNumber=1
ResetShockDetection
```

The shock detection level changed by the SetShockDetection instruction is reset and returned to the detection level of the standard PLAY (AUTOMATIC): Default) by the ResetShockDetection instruction.

AdvInit

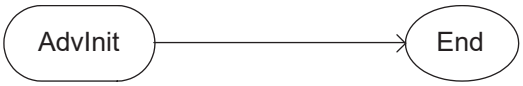
SUBSET	STANDARD	EXPANDED
Not available	Not available	Available

Function

When the same variable is used for multiple systems in the optional independent control function, AdvInit controls the timing to change the variable data among the systems.

AdvInit is an instruction used to control the robot controller internal processing, therefore, executing this instruction does not affect the job.

Syntax

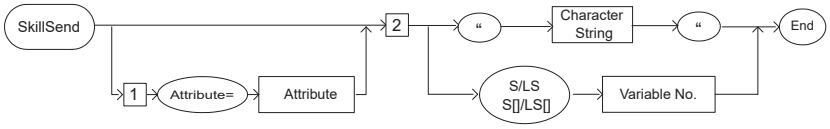


SkillSend

Function

Notify MotoPlus application with specified command.

Syntax



Explanation

1. Attribute

The following tag can be added or omitted.

No	Parameter	Explanation	Note
1	D Variable number / D [Array number] / [Constant]	Specify the attribute	

2. Command String

Add the following tag.

No	Parameter	Explanation	Note
2	S Variable number / S [Array number] / [Constant]	Specify the command string.	String: 32 characters S variable: 16 characters

2.2 Move Instruction

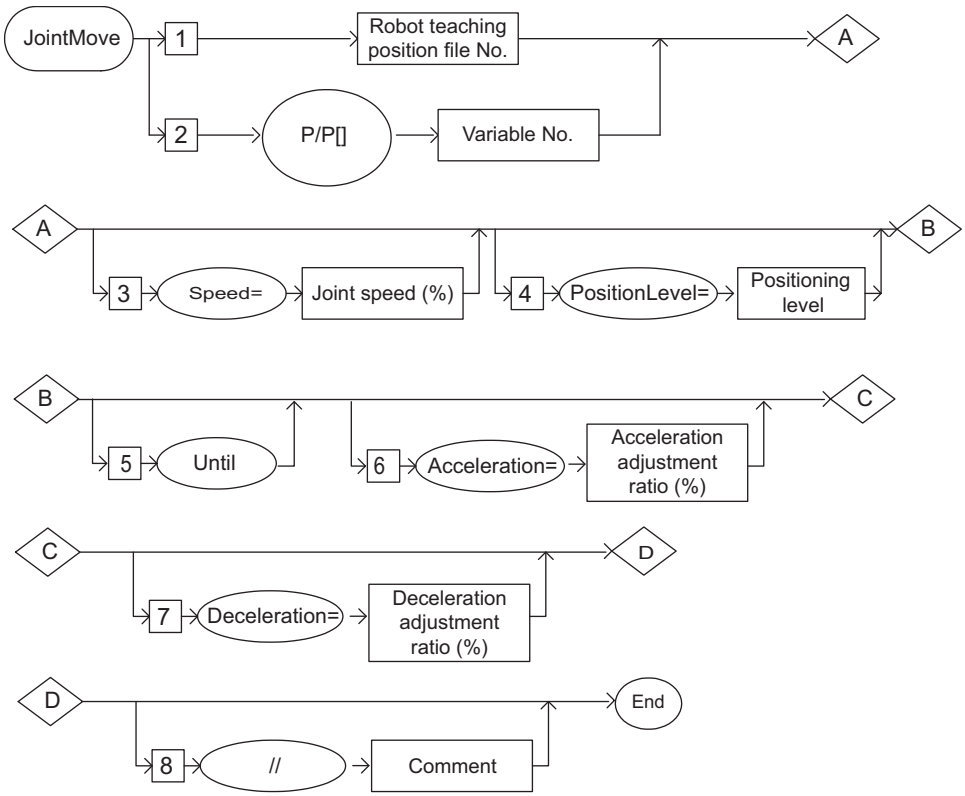
JointMove

Function

Moves to the teaching position by joint interpolation.

Syntax

The parameter which can be used is limited by the type of the job.



Explanation

1. Robot teaching position file number /P Variable number /P [Array number]

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Robot teaching position file number	The robot axis teaching position in each step is written in this file unconditionally when the step is taught. This teaching position cannot be edited.	On the job display, this parameter is not displayed.
2	P Variable number/ P [Array number]	Specify the position variable number of the robot axis. Moves to the position data set in the variable of the specified number.	Variable number: 000 to 127

SUPPLE -MENT

Robot teaching position is taught by pressing the button “Teach” at the Job Contents View. P Variable is used for the Motion Command under Command Bar.

2. Speed=Joint speed

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
3	Speed=Joint speed	Specify the joint speed. The joint speed is shown in the ratio to the highest speed. When the joint speed is omitted, the operation is performed at the speed decided beforehand.	Speed: 0.01% to 100.00% Variable B/B[]/I/I[]/D/D[] can be used. (Units: 0.01%)

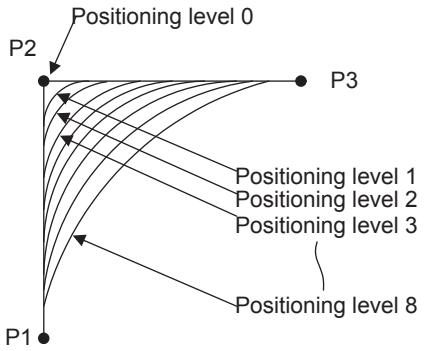
3. Position level

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
4	PositionLevel= Position level	Specify the position level. The approach level when the manipulator passes the position where the teaching procedure was performed is called a positioning level.	Level: 0 to 8 Variable B/B[]/I/I[]/D/D[] can be used.

Position level

The approach level when the manipulator passes the position where the teaching procedure was performed is called a position level.



4. Until

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
5	Until	Specify the Until instruction. The Until instruction is a tag instruction by which the condition of the input signal is evaluated during operation. When the condition of the input signal is full, the robot executes the next instruction.	Refer to Until in *chapter 2.6 "Instruction Which Adheres to an Instruction"*

5. Acceleration=Acceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
6	Acceleration= Acceleration adjustment ratio	Specify the acceleration adjustment ratio. The acceleration instruction reduces the amount of acceleration in the specified ratio.	Acceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

6. Deceleration=Deceleration adjustment ratio

The following parameter can be added or omitted.

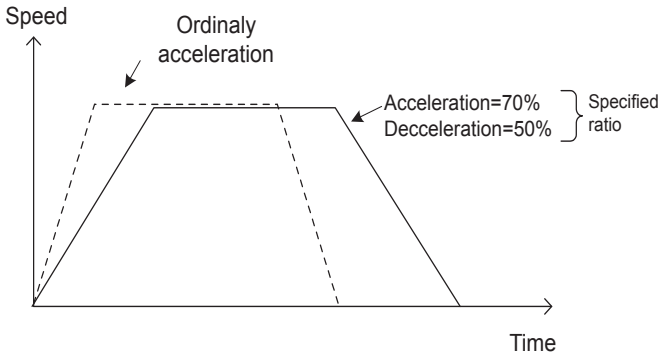
No	Parameter	Explanation	Note
7	Deceleration= Deceleration adjustment ratio	Specify the deceleration adjustment ratio. The deceleration instruction reduces the amount of deceleration in the specified ratio.	Deceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

SUPPLE -MENT

Acceleration adjustment ratio

The acceleration adjustment ratio (acceleration) reduces the amount of acceleration in the specified ratio.

Using this function can reduce the load inertia on the tool and the workpiece.



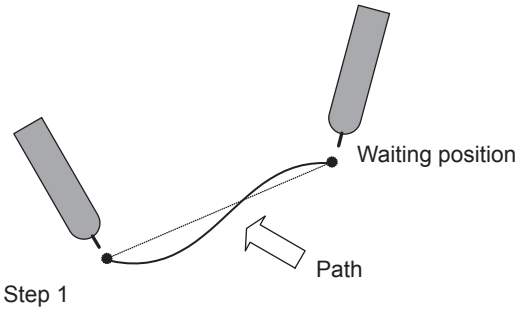
7. Comment Strings

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
8	// Comment strings	Specify the comment.	String: 32 characters

Example

(1) JointMove P000 Speed=50.00
Move from the manipulator's waiting position to step 1. Move by joint interpolation at a speed of 50%.
The position in Step 1 is registered to the P variable no. 0.
The path during movement is not specified. Be careful of interference.



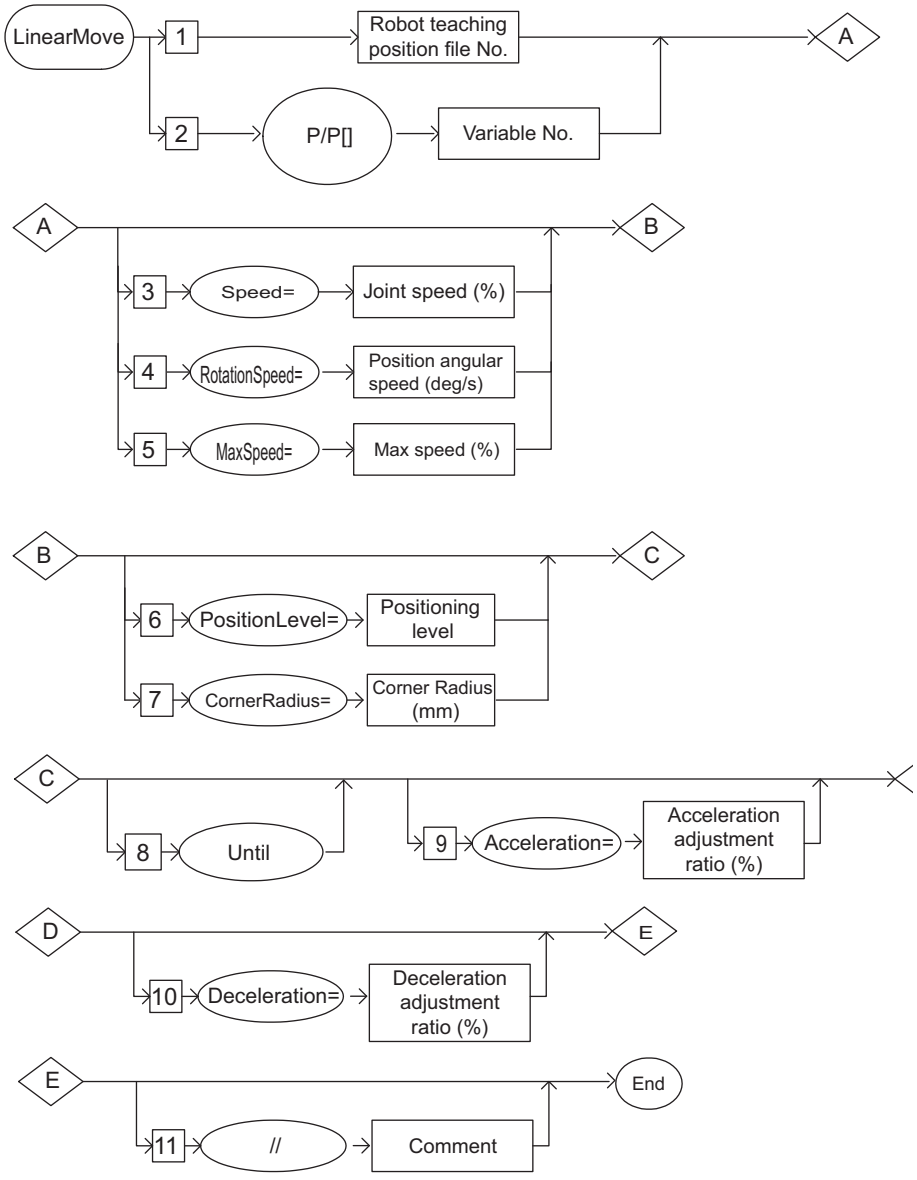
LinearMove

Function

Moves to the teaching position by linear interpolation.

Syntax

The parameter which can be used is limited by the type of the job.



Explanation

1. Robot teaching position file number /P Variable number /P [Array number]

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Robot teaching position file number	The robot axis teaching position in each step is written in this file unconditionally when the step is taught. This teaching position cannot be edited.	On the job display, this parameter is not displayed.
2	P Variable number/ P [Array number]	Specify the position variable number of the robot axis. Moves to the position data set in the variable of the specified number.	Variable number: 000 to 127

SUPPLE -MENT

Robot teaching position is taught by pressing the button “Teach” at the Job Contents View. P Variable is used for the Motion Command under Command Bar.

2. Speed=Tool center point speed/RotationSpeed= Position angular speed/ MaxSpeed=Max speed

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
3	Speed=Tool center point speed	Specify the tool center point speed. Specify the unit of rate using the operation condition setting screen.	Variable B/B[]/I/I[]/D/D[] can be used. (Units: 0.1 mm/s)
4	RotationSpeed= Position angular speed	Specify the position angular speed.	Variable B/B[]/I/I[]/D/D[] can be used. (Units: 0.1 deg/s)
5	MaxSpeed=Max speed	Specify max speed. Max speed is expressed as the rate with respect to the maximum speeds of each axes.	Speed: 50% to 100% Variable B/B[]/LB/LB[]/I/ I[]/LI/LI[]/D/D[]/ LD/LD[] can be used. (Units: %)

3. Position Level/Corner Radius

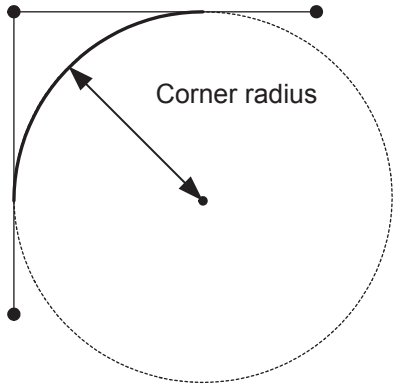
The following parameter can be added or omitted.

No	Parameter	Explanation	Note
6	PositionLevel= Position level	Specify the position level. The position level is a level of the approach when the manipulator passes the position where the teaching procedure was performed.	Level: 0 to 8 Variable B/B[]/I/I[]/D/D[] can be used.
7	CornerRadius= Corner radius	Specify the corner radius. The manipulator operates by circular interpolation in which the corner radius is specified.	Radius: 0.1mm to 6553.5 mm Variable D/D[] can be used. (Units: 0.1 mm)

SUPPLE-MENT

Corner radius

The manipulator operates by circular interpolation in which the corner radius is specified.



4. Until

No	Parameter	Explanation	Note
8	Until	Specify the Until instruction. The Until instruction is a tag instruction by which the condition of the input signal is evaluated during operation. When the condition of the input signal is full, the robot executes the next instruction.	Refer to Until in *chapter 2.6 "Instruction Which Adheres to an Instruction"*

5. Acceleration=Acceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
9	Acceleration= Acceleration adjustment ratio	Specify the acceleration adjustment ratio. The acceleration instruction reduces the amount of acceleration in the specified ratio.	Acceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

6. Decceleration=Deceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
10	Deceleration= Deceleration adjustment ratio	Specify the deceleration adjustment ratio. The deceleration instruction reduces the amount of deceleration in the specified ratio.	Deceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

7. Comment Strings

The following parameter can be added or omitted

No	Parameter	Explanation	Note
11	// Comment strings	Specify the comment.	String: 32 characters

Example

(1) JointMove Speed=50.00
JointMove Speed=25.00
JointMove Speed=12.50 ♦♦♦ Step 3
LinearMove Speed=138 ♦♦♦ Step 4
Moves from Step 3 to Step 4 by the linear interpolation at a rate of 138mm/s.

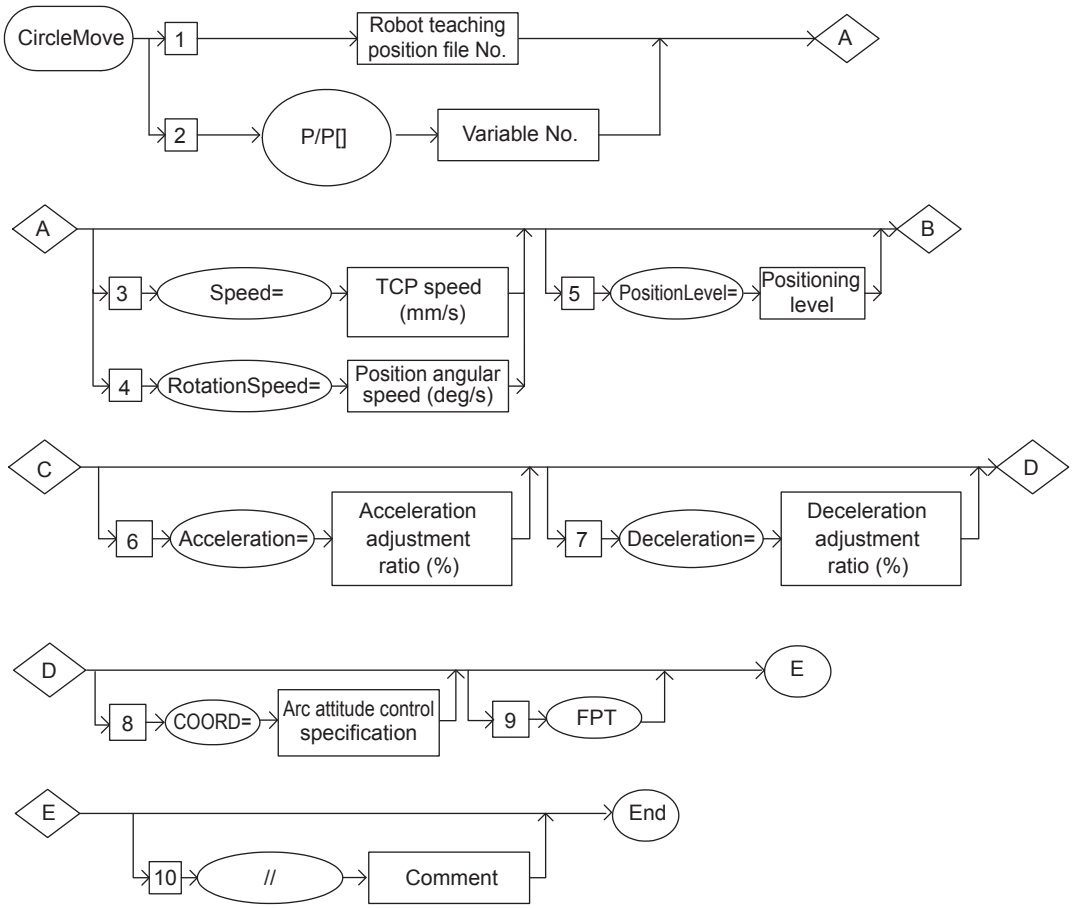
CircleMove

Function

Moves to the teaching position by circular interpolation.

Syntax

The parameter which can be used is limited by the type of the job.



NOTE

The three CircleMove instructions for the three points which define a circular arc must be registered in the same job.
If all of the three CircleMove instructions are not registered in the same job due to the Jump/Call instruction, the manipulator cannot perform the circular interpolation motion which passes the three points.

Example

1. Robot teaching position file number /P Variable number /P [Array number]

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Robot teaching position file number	The robot axis teaching position in each step is written in this file unconditionally when the step is taught. This teaching position cannot be edited.	On the job display, this parameter is not displayed.
2	P Variable number/ P [Array number]/	Specify the position variable number of the robot axis. Moves to the position data set in the variable of the specified number.	Variable number: 000 to 127

SUPPLE -MENT

Robot teaching position is taught by pressing the button “Teach” at the Job Contents View. P Variable is used for the Motion Command under Command Bar.

2. Speed=Tool center point speed/Rotation-Speed=Position angular speed

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
3	Speed=Tool center point speed	Specify the tool center point speed. Specify the unit of rate using the operation condition setting screen.	Variable B/B[]/I/I[]/D/D[]/ can be used. (Units: 0.1 mm/s)
4	RotationSpeed= Position angular speed	Specify the position angular speed.	Variable B/B[]/I/I[]/D/D[] can be used. (Units: 0.1 deg/s)

3. PositionLevel=Position level

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
5	PositionLevel= Position level	Specify the position level. The position level is a level of the approach when the manipulator passes the position where the teaching procedure was performed.	Level: 0 to 8 Variable B/B[]/LB/I/I[]/D/ D[] can be used.

4. Acceleration=Acceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
6	Acceleration= Acceleration adjustment ratio	Specify the acceleration adjustment ratio. The acceleration instruction reduces the amount of acceleration in the specified ratio.	Acceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/ D[]can be used.

5. Deceleration=Deceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
7	Deceleration= Deceleration adjustment ratio	Specify the deceleration adjustment ratio. The deceleration instruction reduces the amount of deceleration in the specified ratio.	Deceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

6. Circle Pose Control=Arc attitude control specification

The following parameter can be added or omitted.

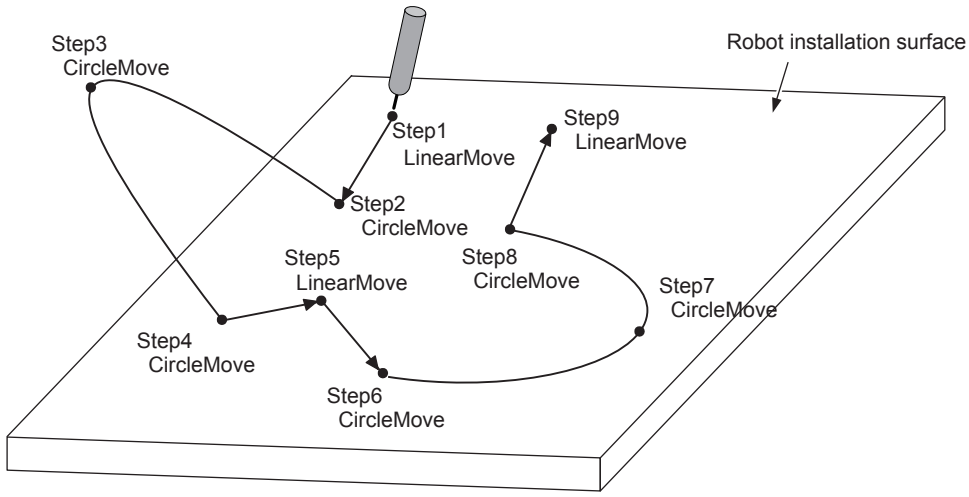
No	Parameter	Explanation	Note
8	COORD=Arc attitude control specification	Specify the attitude control of arc. This is not usually needed, but depending on teaching, the intended circular motion may not be performed. In this case, specify as follows. When the circular surface is parallel with the robot installation surface, specify as COORD = 0. When the circular surface is not parallel with the robot installation surface, specify as COORD = 1.	Variable B/B[]/I/I[]/D/D[] can be used to specify the attitude control

Example

```
LinearMove Speed=138
CircleMove Speed=138  COORD=1        ♦♦♦Step 2
CircleMove Speed=138  COORD=1        ♦♦♦Step 3
CircleMove Speed=138  COORD=1        ♦♦♦Step 4
LinearMove Speed=138
CircleMove Speed=138  COORD=0        ♦♦♦Step 6
CircleMove Speed=138  COORD=0        ♦♦♦Step 7
CircleMove Speed=138  COORD=0        ♦♦♦Step 8
LinearMove Speed=138
End Job
```

Step 2 to 4 performs the attitude control based on the circular surface.

Step 6 to 7 performs the attitude control based on the robot installation surface.



7. FPT Circle End Point setting

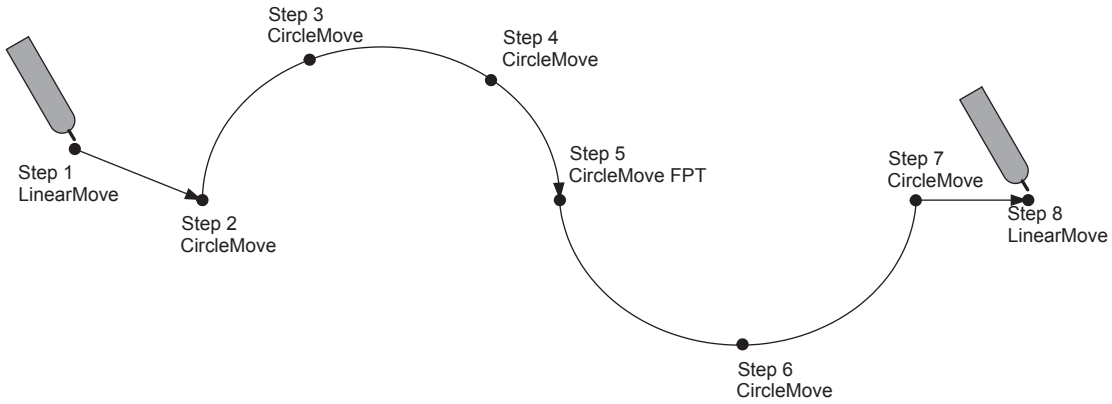
The following parameter can be added or omitted.

No	Parameter	Explanation	Note
9	FPT	Specify the end-point of the arc (the point at which the curvature of the arc is to be changed).	

Example

(1) LinearMove Speed=138
CircleMove Speed=138 ♦♦♦Step 2
CircleMove Speed=138 ♦♦♦Step 3
CircleMove Speed=138 ♦♦♦Step 4
CircleMove FPT Speed=138 ♦♦♦Step 5
CircleMove Speed=138 ♦♦♦Step 6
CircleMove Speed=138 ♦♦♦Step 7
LinearMove Speed=138
End Job

Moves from Step 2 to Step 7 by circular interpolation at a rate of 138 mm/s.
Moves to Step 3 in a circular arc formed with the teaching points in Steps 2, 3, and 4.
Moves to Step 4 in a circular arc formed with the teaching points in Steps 3, 4, and 5.
Moves to Step 5 in a circular arc formed with the teaching points in Steps 3, 4, and 5.
Moves to Step 6 in a circular arc formed with the teaching points in Steps 5, 6, and 7.
Moves to Step 7 in a circular arc formed with the teaching points in Steps 5, 6, and 7.



8. Comment Strings

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
10	// Comment strings	Specify the comment.	String: 32 characters

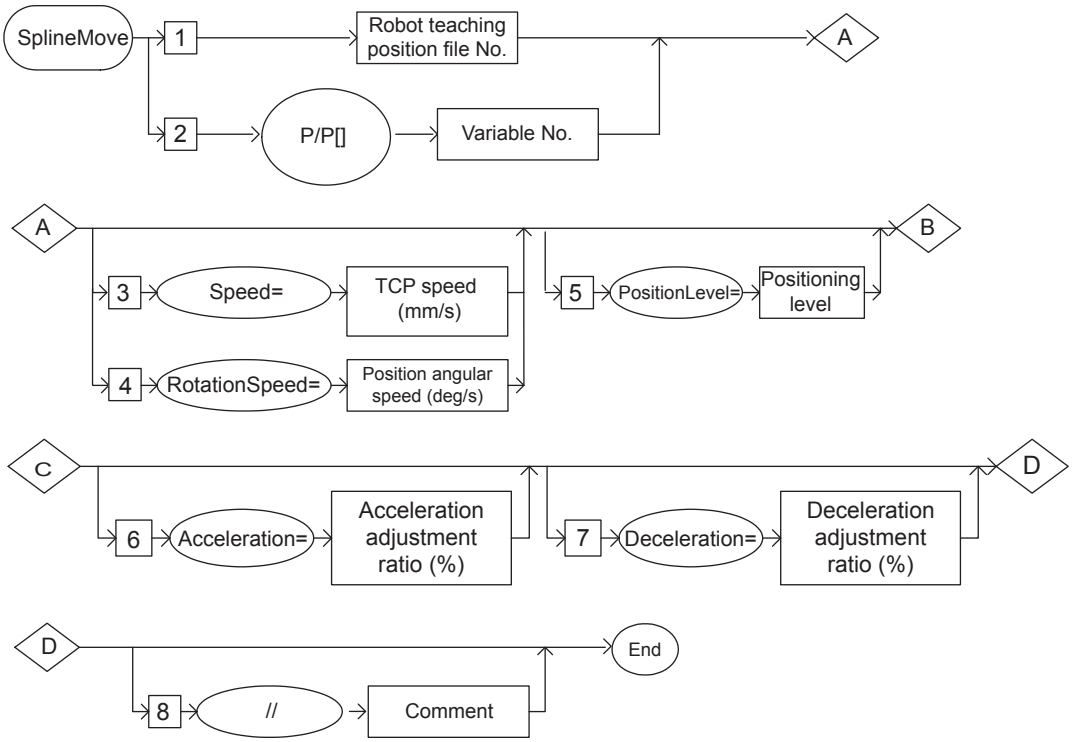
SplineMove

Function

Moves to the teaching position by spline interpolation.

Syntax

The parameter which can be used is limited by the type of the job.



Explanation

1. Robot teaching position file number /P Variable number /P [Array number]

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Robot teaching position file number	The robot axis teaching position in each step is written in this file unconditionally when the step is taught. This teaching position cannot be edited.	On the job display, this parameter is not displayed.
2	P Variable number / P [Array number]	Specify the position variable number of the robot axis. Moves to the position data set in the variable of the specified number.	Variable number: 000 to 127

SUPPLE -MENT

Robot teaching position is taught by pressing the button “Teach” at the Job Contents View. P Variable is used for the Motion Command under Command Bar.

2. Speed=Tool center point speed/Rotation-Speed=Position angular speed

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
3	Speed=Tool center point speed	Specify the tool center point speed. Specify the unit of rate using the operation condition setting screen.	Variable B/B[]/I/I[]/D/D[]/ can be used. (Units: 0.1 mm/s)
4	RotationSpeed= Position angular speed	Specify the position angular speed.	Variable B/B[]/I/I[]/D/D[] can be used. (Units: 0.1 deg/s)

3. PositionLevel=Position level

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
5	PositionLevel= Position level	Specify the position level. The position level is a level of the approach when the manipulator passes the position where the teaching procedure was done.	Level: 0 to 8 Variable B/B[]/I/I[]/D/D[] can be used.

4. Acceleration=Acceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
6	Acceleration= Acceleration adjustment ratio	Specify the acceleration adjustment ratio. The acceleration instruction reduces the amount of acceleration in the specified ratio.	Acceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

5. Deceleration=Deceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
7	Deceleration= Deceleration adjustment ratio	Specify the deceleration adjustment ratio. The deceleration instruction reduces the amount of deceleration in the specified ratio.	Deceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

6. Comment Strings

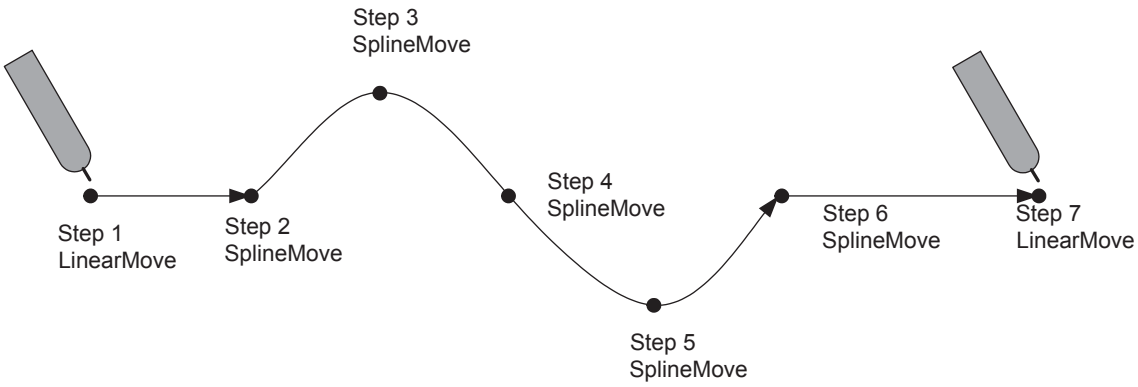
The following parameter can be added or omitted.

No	Parameter	Explanation	Note
8	// Comment strings	Specify the comment.	String: 32 characters

Example

```
(1)     LinearMove Speed=138
        SplineMove Speed=138    ♦♦♦ Step 2
        SplineMove Speed=138    ♦♦♦ Step 3
        SplineMove Speed=138    ♦♦♦ Step 4
        SplineMove Speed=138    ♦♦♦ Step 5
        SplineMove Speed=138    ♦♦♦ Step 6
        LinearMove Speed=138
        End Job
```

Moves from Step 2 to Step 6 by spline interpolation at a rate of 138mm/s.
Moves to Step 3 by spline interpolation defined by the teaching points in Steps 2, 3, and 4.
Moves to Step 4 by synchronized spline interpolation defined by the teaching points in Steps 2, 3, 4 and by the synchronized spline interpolation defined by the teaching points in Steps 3, 4, and 5.
Moves to Step 5 by synchronized spline interpolation defined by the teaching points in Steps 3, 4,5 and by synchronized spline interpolation defined by the teaching points in Steps 4, 5, and 6.
Moves to Step 6 by spline interpolation defined by the teaching points in Steps 4, 5, and 6.



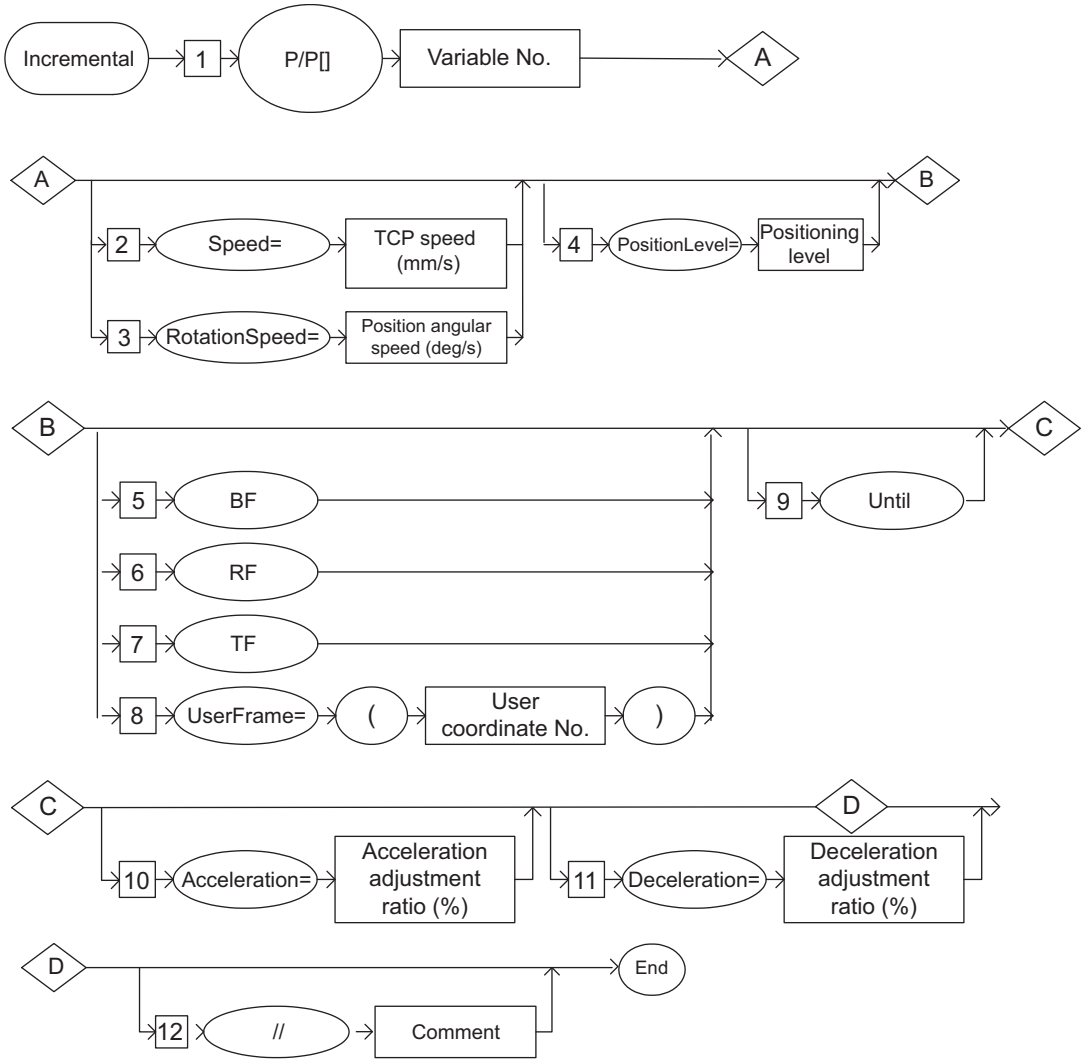
Incremental

Function

Moves by linear interpolation from the current position for the specified incremental value.

Syntax

The parameter which can be used is limited by the type of the job.



Explanation

1. P Variable number /P [Array number]

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	P Variable number / P [Array number]	Specify the position variable number of the manipulator axis. Moves the axis according to the position data set in the specified variable number.	Variable number: 000 to 127

2. Speed=Tool center point speed/Rotation-Speed=Position angular speed

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
2	Speed=Tool center point speed	Specify the tool center point speed. Specify the unit of rate using the operation condition setting screen.	Variable B/B[]/I/I[]/D/D[]/ can be used. (Units: 0.1 mm/s)
3	RotationSpeed= Position angular speed	Specify the position angular speed.	Variable B/B[]/I/I[]/D/D[] can be used. (Units: 0.1 deg/s)

3. PositionLevel=Position level

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
4	PositionLevel= Position level	Specify the position level. The positioning level is a level of the approach when the manipulator passes the position where the teaching procedure was done.	Level: 0 to 8 Variable B/B[]/I/I[]/D/D[] can be used.

4. BF/RF/TF/UserFrame= (User coordinate number)

The following parameters can be added or omitted.

No	Parameter	Explanation	Note
5	BF	Specify the increment value in the base coordinate system.	
6	RF	Specify the increment value in the robot coordinate system.	
7	TF	Specify the increment value in the tool coordinate system.	
8	UserFrame= (User coordinate number)	Specify the increment value in the user coordinate system.	No: 1 to 24 Variable B/I/D can be used.

5. Until

No	Parameter	Explanation	Note
9	Until	Specify the Until instruction. The Until instruction is a tag instruction by which the condition of the input signal is evaluated during operation. When the condition of the input signal is full, the robot executes the next instruction.	Refer to Until in *chapter 2.6 "Instruction Which Adheres to an Instruction"*

6. Acceleration=Acceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
10	Acceleration= Acceleration adjustment ratio	Specify the acceleration adjustment ratio. The Acceleration instruction reduces the amount of acceleration in the specified ratio.	Acceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

7. Deceleration=Deceleration adjustment ratio

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
11	Deceleration= Deceleration adjustment ratio	Specify the deceleration adjustment ratio. The deceleration instruction reduces the amount of deceleration in the specified ratio.	Deceleration adjustment ratio: 20% to 100% Variable B/B[]/I/I[]/D/D[] can be used.

8. Comment Strings

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
12	// Comment strings	Specify the comment.	String: 32 characters

Example

(1) Incremental P000 Speed=138 RF
Moves from the current position at a rate of 138 mm/s for the incremental value specified in P000 in the robot coordinate system.

SetSpeed

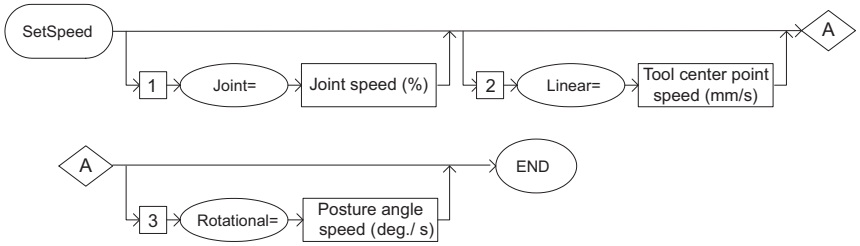
SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Sets the playback speed. The manipulator operates at the speed specified in the SetSpeed instruction when the speed is not specified in the move instruction.

Construction

The tag which can be used is limited by the type of job.



Explanation

1. Joint=Joint speed

The following tag can be added or omitted.

No	Parameter	Explanation	Note
1	Joint=Joint speed	Specifies the joint speed. The joint speed is shown in the ratio to the highest speed. Operates at the speed decided beforehand when the joint speed is omitted.	Speed: 0.01% to 100.00% Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used. (Units: 0.01%)

2. Linear=Tool center point speed

The following tag can be added or omitted.

No	Parameter	Explanation	Note
2	Linear=Tool center point speed	Specifies the tool center point speed. Specifies the unit of rate using the operation condition setting screen.	Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used. (Units: 0.1 mm/s)

3. Rotational=Position angular speed

The following tag can be added or omitted.

No	Parameter	Explanation	Note
3	Rotational=Position angular speed	Specifies the position angular speed.	Variable B/B[]/LB/LB[]/I/I[]/LI/LI[]/D/D[]/LD/LD[] can be used. (Units: 0.1 deg/s)

Example

```
Start Job
JointMove Speed=100.00
LinearMove Speed=138.0

SetSpeed Joint=50.00 Linear=276.0 Rotational=30.0
JointMove
LinearMove

LinearMove Rotation Speed=60.0
End Job
```

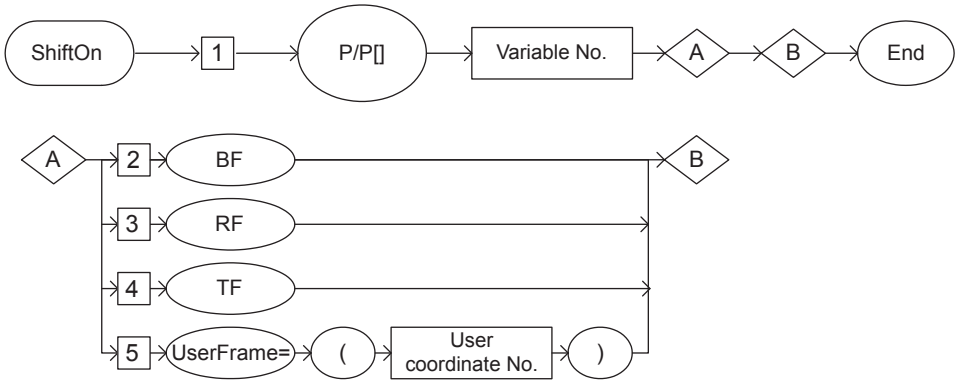
ShiftOn

Function

Begins the parallel shift operation. The amount of the parallel shift is set in a positional variable by the increment value of X, Y, Z, and Rx, Ry, Rz in each coordinate system.

Syntax

The parameter which can be used is limited by the type of the job.



Explanation

1. P Variable number /P [Array number]

The following parameter must be added.

No	Parameter	Explanation	Note
1	P Variable number/ P [Array number]	Specify the position variable number of the robot axis by which the shift value is set. The shift is performed by the only increment value which is set in the variable of the specified number.	Variable number: 000 to 127

2. BF/RF/TF/UserFrame= (User coordinate number)

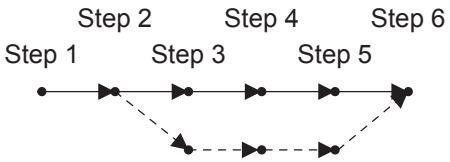
When a P Variable number or P [Array number] is selected from the table in part 2 of this explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
2	BF	Specify the shift operation in the base coordinate system.	
3	RF	Specify the shift operation in the robot coordinate system.	
4	TF	Specify the shift operation in the tool coordinate system.	
5	UserFrame= (User coordinate number)	Specify the shift operation in the user coordinate system.	No.1 to 63 Variable B/I/D can be used.

Example

(1) JointMove Speed=50.0
LinearMove Speed=138
ShiftOn P000 RF
LinearMove Speed=138
LinearMove Speed=138
LinearMove Speed138
ShiftOff
LinearMove Speed=138
:

Shifted section



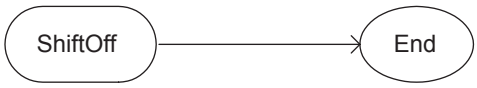
The shift is performed between Step 3 and Step 5 in the robot coordinate system by the amount of the shift which is set in P000.

ShiftOff

Function

Ends the parallel shift operation.

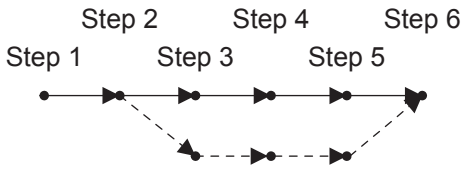
Syntax



Example

(1)
```
JointMove Speed=50.0
LinearMove Speed=138
ShiftOn P000 RF
LinearMove Speed=138
LinearMove Speed=138
LinearMove Speed138
ShiftOff
LinearMove Speed=138
```

Shifted section



The shift is performed between Step 3 and Step 5 in the robot coordinate system by the amount of the shift which is set in P000.

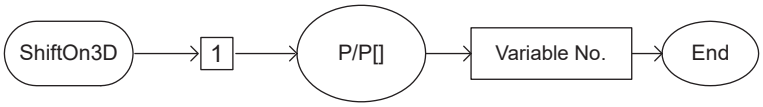
ShiftOn3D

Function

Begins the 3D shift operation. This will perform a matrix transformation on each position based on a Position Variable defined in World (Base) Frame.

Syntax

The parameter which can be used is limited by the type of the job.



Explanation

1. P Variable number /P [Array number]

The following parameter must be added

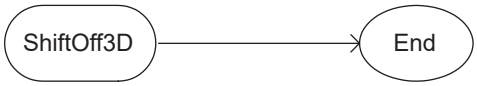
No	Parameter	Explanation	Note
1	P Variable number/ P [Array number]	Specify the position variable number by which the 3D shift value is set. The shift is performed by the matrix transformation which is set in the variable of the specified number.	Variable number: 000 to 127

ShiftOff3D

Function

Ends the 3D shift operation.

Syntax



MakeShift

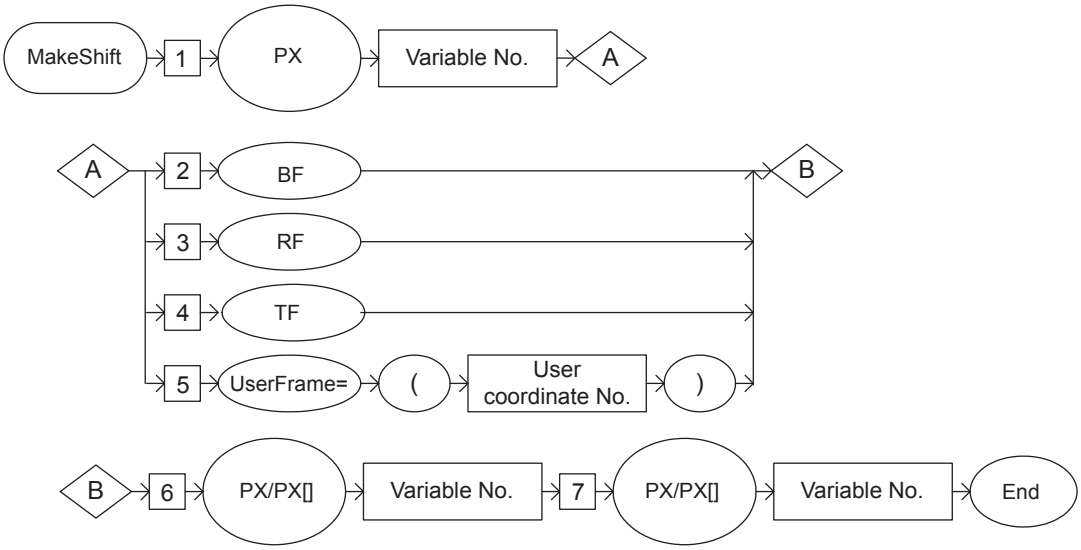
Function

Calculates the amount of the shift in the specified coordinate system according to Data 2 and Data 3 and stores the result in Data 1.

Data 2 indicates the reference position to carry out the parallel shift, and Data 3 is the target position (shifted position).

Syntax

MakeShift <Data 1> Coordinate system designation <Data 2> <Data 3>



Explanation

1. PX Variable number

The following parameter must be added.

No.	Parameter	Explanation	Note
1	PX Variable number	Specify the number of the expanded position variable to store the calculated shift.	<Data 1>

2. BF/RF/TF/UserFrame= (User coordinate number)

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	BF	Specify the calculation of the shift amount in the base coordinate system.	
3	RF	Specify the calculation of the shift amount in the robot coordinate system.	
4	TF	Specify the calculation of the shift amount in the tool coordinate system.	
5	UserFrame= (User coordinate number)	Specify the calculation of the shift amount in the user coordinate system.	No.: 1 to 63 Variable B/I/D can be used.

3. PX Variable number/PX [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
6	PX Variable number/ PX [Array number]	Specify the expanded position type variable number of the reference position to calculate the amount of the shift.	<Data 2>

4. PX Variable number/PX [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
7	PX Variable number/ PX [Array number]	Specify the expanded position type variable number of the target position to calculate the amount of the shift.	<Data 3>

Example

(1) Start Job

```
JointMove Speed=20.00(%)          :Moves to the reference position.
GetPosition PX000 Step#(1)        :Sets the current position (the reference
                                   position) in the position variable P000.
JointMove Speed=20.00(%)          :Moves to the target position
GetPosition PX001 Step#(2)        :Sets the current position (the target
                                   position) in the position variable P001.
MakeShift PX010 RF PX000 PX001    :Calculates the shift amount and stores it in
End Job                            the position variable P010.
```

Smooth

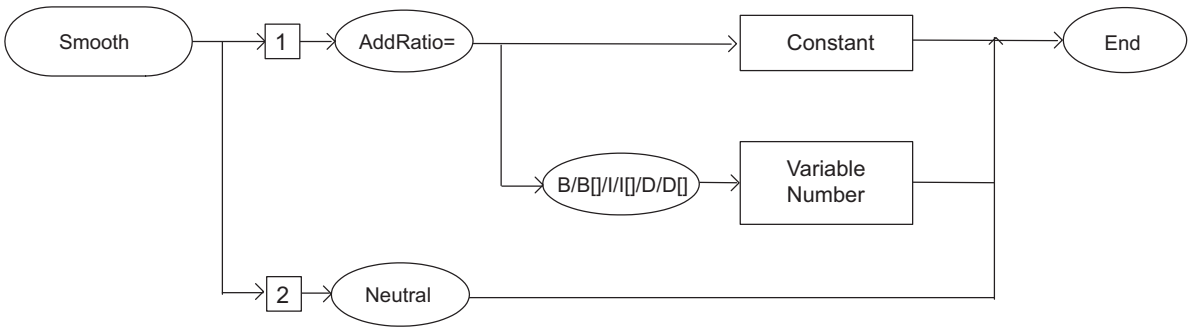
Function

Activates a motion filter to smooth the robot's motion and reduce vibration.

NOTE

- The Smooth command is available for the
  - YRC1000 (YAS4.13.00+)
  - YRC1000micro (YBS2.49.00+)
- S2C1596 must be set to 1 to see this command in the INFORM list.

Syntax



Explanation

1. AddRatio

Add the following tag.

No.	Parameter	Explanation	Note
1	AddRatio=	This tag activates the robot motion filter. The motion filter ratio applies to subsequent move instructions *(see Example on page "2-39")*	Range: 0 to 300 (%)

2. Neutral

Add the following tag.

No.	Parameter	Explanation	Note
2	Neutral	The tag disables the previously active robot motion filter *(see Example on page "2-39")*.	

Example

Line	Instruction	
0000	NOP	
0001	MOVJ VJ=10.00	
0002	MOVL V=200.0	
0003	SMOOTH ADDRATIO=100	•••Set manipulator motion filter (100%).
0004	MOVL V=200.0	These instructions operate with the manipulator motion filter set by the SMOOTH instruction (100%).
0005	MOVL V=200.0	
0006	MOVL V=200.0	
0018	MOVL V=200.0	
0019	MOVL V=200.0	
0020	MOVL V=200.0	
0021	MOVL V=200.0	
0022	SMOOTH NEUTRAL	•••Cancel manipulator motion filter.
0023	MOVJ VJ=10.00	
0024	END	

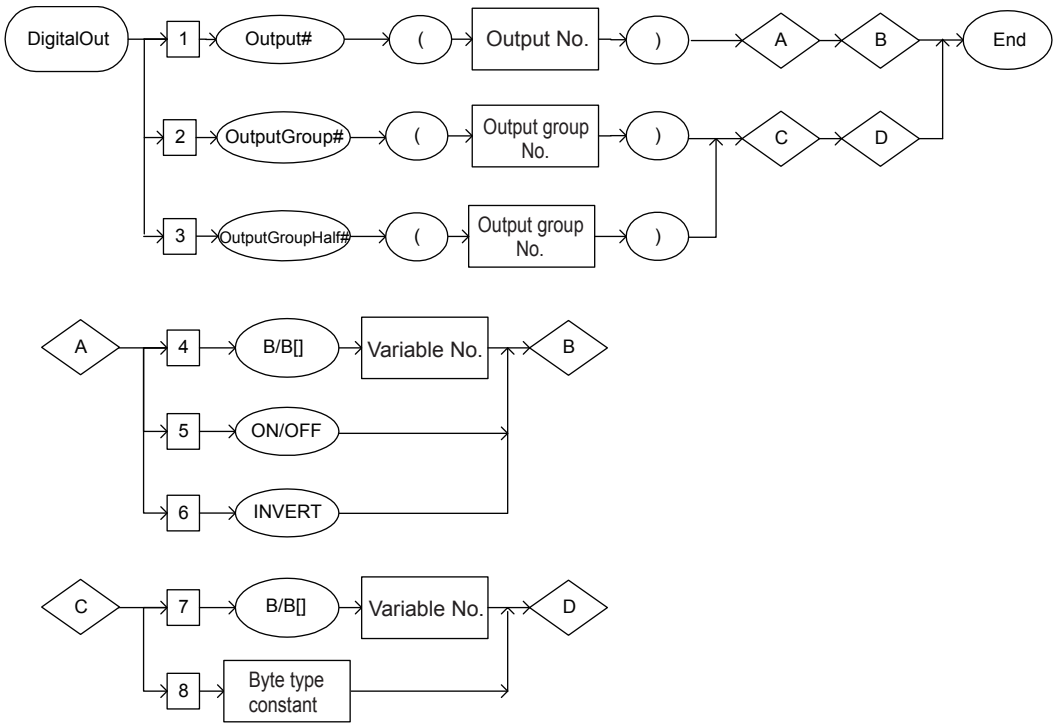
2.3 I/O Instructions

DigitalOut

Function

Turns the output signal ON and OFF.

Syntax



Explanation

1. Output# (Output number) /OutputGroup# (Output group number) /OutputGroupHalf# (Output group number)

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Output# (Output number)	Specify the number of output signal.	No:1 to 4096 Variable B/I/D can be used.
2	OutputGroup# (Output group number)	Specify the group number of output signal (1 group 8 points).	No:1 to 512 Variable B/I/D can be used.
3	OutputGroupHalf# (Output group number)	Specify the group number of output signal (1 group 4 points).	No:1 to 1024 Variable B/I/D can be used.

SUPPLE -MENT

Output signal

Output signal Output#(xx) is 1 point, OutputGroupHalf#(xx) is 1 group 4 points, and OutputGroup#(xx) is 1 group 8 points.

Output#(8)	Output#(7)	Output#(6)	Output#(5)	Output#(4)	Output#(3)	Output#(2)	Output#(1)
OutputGroupHalt#(2)				OutputGroupHalt#(1)			
OutputGroup#(1)							

2. Output# (Output number) /OutputGroup# (Output group number) /OutputGroupHalf# (Output group number)

When Output# (output number) is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
4	B Variable number/ B [Array number]/	The least significant bit of the specified byte type variable specifies on/off of the output signal.	Least significant bit: 0: OFF 1: ON
5	ON/OFF	Specify on/off of the output signal.	
6	INVERT	Refers the current signal status to output OFF when the status is ON, and output ON when the status is OFF.	

3. B Variable number / B [Array number] / Byte type constant

When OutputGroup# (Output group number) or OutputGroupHalf# (Output group number) is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
7	B Variable number/ B [Array number]/	Specify ON/OFF of the corresponding output signal when the content of the specified byte type variable is expressed in bits.	bit: 0: OFF 1: ON
8	Byte type constant	When the specified byte type constant is expressed in bit form, the corresponding on/off output signal is specified.	

Example

(1) DigitalOut Output#(12) ON
GP output signal no. 12 is turned ON.

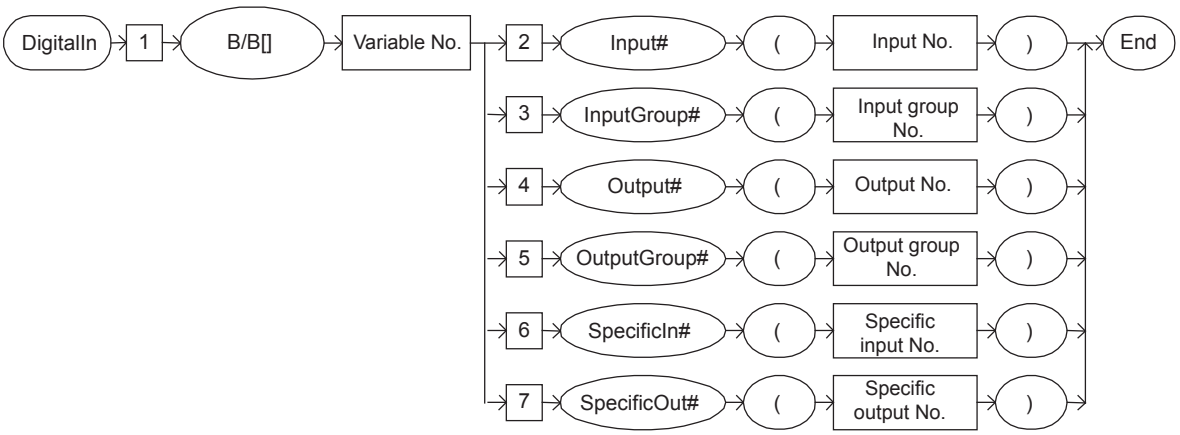
(2) Set B000 24
DigitalOut OutputGroup#(3) B000
B000=24(Decimal)= 00011000(Binary)

DigitalIn

Function

The signal status is read in the byte type variable.

Syntax



Explanation

1. B Variable Number / B [Array Number]

The following parameters must be selected.

No	Parameter	Explanation	Note
1	B Variable Number/ B [Array Number]	Specify the number of byte type variable which reads the signal status.	

2. Input#(Input number) / InputGroup#(Input group number) / Output#(Output number) / OutputGroup#(Output group number) / SpecificIn#(Specific input number) / SpecificOut#(Specific output number)

When a B Variable number, B [Array number] is selected from the table in part 1 of this Explanation, one of the parameters is selected from the following table

No	Parameter	Explanation	Note
2	Input# (Input number)	Specify the number of input signal to read the signal status	No:1 to 4096 Variable B/I/D can be used.
3	InputGroup# (Input group number)	Specify the number of the input group (1 group 8 points) signal to read the signal status.	No:1 to 512 Variable B/I/D can be used.
4	Output# (Output number)	Specify the number of the output signal to read the signal status.	No:1 to 4096 Variable B/I/D can be used.
5	OutputGroup# (Output group number)	Specify the number of the output group signal (1 group 8 points) to read the signal status.	No:1 to 512 Variable B/I/D can be used.
6	SpecificIn# (Specific input number)	Specify the number of the specific input signal to read the signal status.	No:1 to 1280 Variable B/I/D can be used.
7	SpecificOut# (Specific output number)	Specify the number of the specific output signal to read the signal status.	No:1 to 2400 Variable B/I/D can be used.

SUPPLE -MENT

Input signal

Input signal Input#(xx) is 1 point, InputGroupHalf#(xx) is 1 group 4 points, and Input-Group#(xx) is 1group 8 points.

Input#(8)	Input#(7)	Input#(6)	Input#(5)	Input#(4)	Input#(3)	Input#(2)	Input#(1)
InputGroupHalf#(2)				InputGroupHalf#(1)			
InputGroup#(1)							

Example

(1) DigitalIn B016 Input#(12)
The ON/OFF status of input signal No.12 is read in byte type variable No.16. When the input signal No.12 is on, the status of the input signal is B016=1 (decimal)=00000001 (binary).

(2) DigitalIn B002 OutputGroup#(8)
The ON/OFF status of GP output signal number from no. 57 to no.64 is read in byte type variable No.2.
In the following cases, the status of the output signal is B002=150 (decimal)= 10010110 (binary).

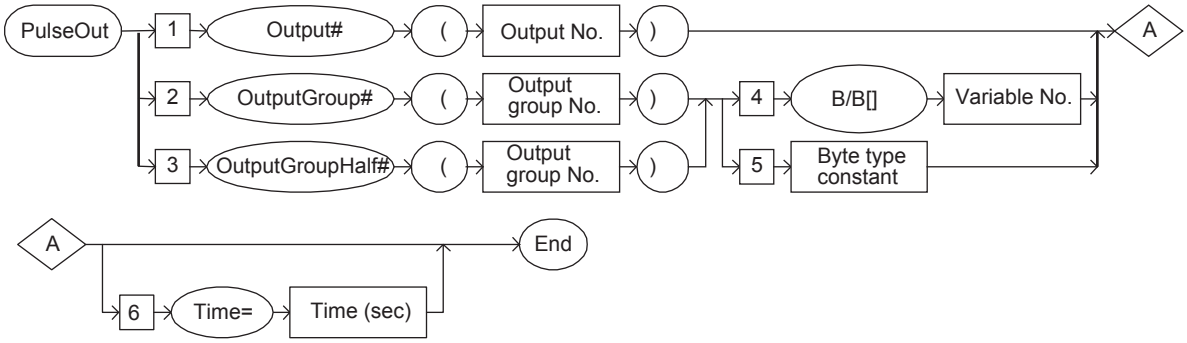
PulseOut

Function

The pulse signal is output to the output signal only for the specified time.

The PulseOut instruction executes the next instruction without waiting for the completion of pulse output (the specified output time).
However, in case the pulse output of the previous executed PulseOut instruction is not completed, this time PulseOut instruction is executed after the completion when the instruction is executed continuously for the same signal.

Syntax



Explanation

1. Output# (output number) / OutputGroup# (output group number) / OutputGroupHalf# (output group number)

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Output# (Output number)	Specify the number of the signal to which the pulse signal is output.	No.: 1 to 4096 Variable B/I/D can be used.
2	OutputGroup# (Output group number)	Specify the group number of the signal (1 group 8 points) to which the pulse signal is output.	No.: 1 to 512 Variable B/I/D can be used.
3	OutputGroupHalf# (Output group number)	Specify the group number of the signal (1 group 4 points) to which the pulse signal is output.	No.: 1 to 1024 Variable B/I/D can be used.

2. B Variable number / B [Array number] / Byte type constant

When OutputGroup# (output group number) or OutputGroupHalf# (output group number) in the above table is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
4	B Variable number/ B [Array Number]	Specify ON/OFF of the corresponding pulse output signal when the content of the specified byte type variable is expressed in bits.	Bit: 0: OFF 1: ON
5	Byte type constant	Specify ON/OFF of the corresponding pulse output signal when the specified byte type constant is expressed in bits.	

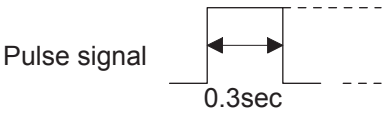
3. Time=Time

The following parameter can be added or omitted.

No.	Parameter	Explanation	Note
6	Time=Time	Specify the time to output. The pulse signal is output during the specified time Time.	For the YRC1000 Timer: 0.01 to 655.35 seconds Variable I/I[] can be used. Units: 0.01 seconds For the YRC1000micro Timer: 0.001 to 65.535 seconds Variable I/I[] can be used. Units: 0.001 seconds When the time is not specified, the pulse signal is output for 0.30 seconds. (YRC1000/ YRC1000micro)

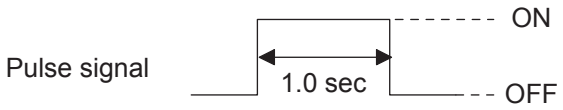
Example

(1) PulseOut Output#(128)
The pulse signal is output for 0.30 seconds to output signal No.128.



(2) Set B000 5
PulseOut Output#(B000) Time=1.0

The pulse signal is output for 1.0 seconds to output signal No.5.

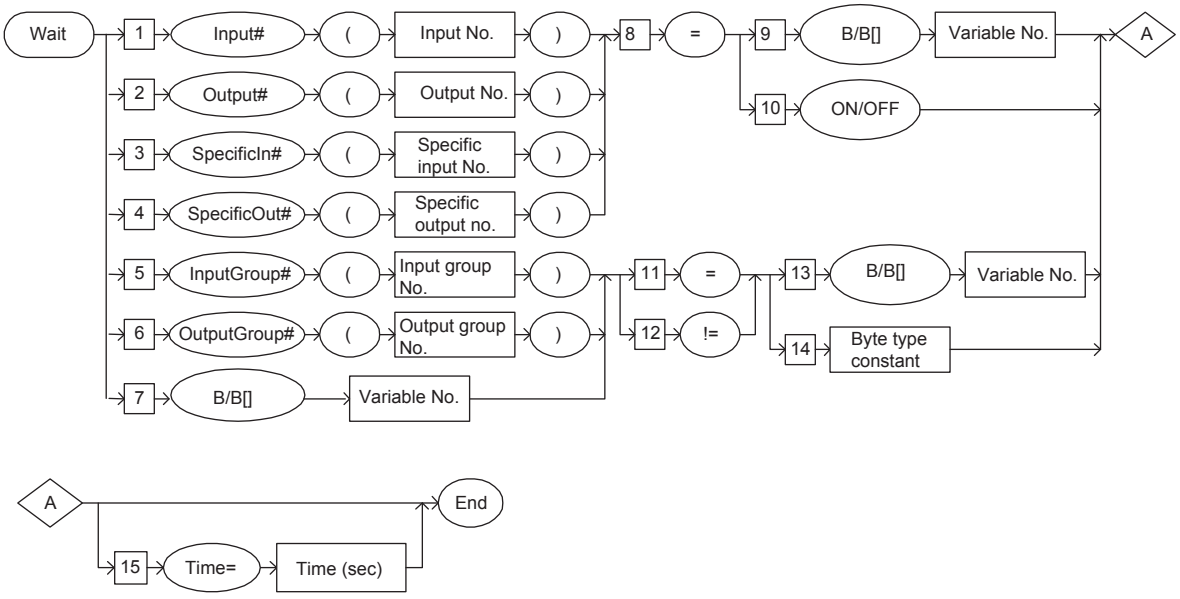


Wait

Function

Wait until the various statuses coincide with the specified status.

Syntax



Explanation

1. Input#(Input number) / Output#(Output number) / SpecificIn#(Specific input number) / SpecificOut#(Specific output number) / InputGroup#(Input group number) / OutputGroup#(Output group number) / B Variable Number / B[Array Number]

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	Input# (Input number)	Specify the number of the GP input signal for the waiting condition.	No:1 to 4096 Variable B/I/D can be used.
2	Output# (Output number)	Specify the number of the GP output signal for the waiting condition.	No:1 to 4096 Variable B/I/D can be used.
3	SpecificIn# (Specific input number)	Specify the number of the specific input signal for the waiting condition.	No:1 to 2048 Variable B/I/D can be used.
4	SpecificOut# (Specific output number)	Specify the number of the specific output signal for the waiting condition.	No:1 to 4096 Variable B/I/D can be used.
5	InputGroup# (Input group number)	Specify the number of GP input group (1 group 8 points) signal for the waiting condition.	No:1 to 512 Variable B/I/D can be used.
6	OutputGroup# (Output group number)	Specify the number of GP output group (1 group 8 points) signal for the waiting condition.	No:1 to 512 Variable B/I/D can be used.
7	B Variable number/ B [Array number]	Specify the byte type variable for the waiting condition.	

2. =

When an Input# (input number), Output# (output number), SpecificIn# (specific input number), or SpecificOut# (specific output number) is selected from the table in part 1 of this Explanation, add the following parameter.

No	Parameter	Explanation	Note
8	=	It is equal.	

3. B Variable number / B [Array number] / ON / OFF

When an Input# (input number), Output# (output number), SpecificIn# (specific input number), or SpecificOut# (specific output number) is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table after the equal sign (=) is added from the table in part 2 of this Explanation.

No	Parameter	Explanation	Note
9	B Variable number/ B [Array number]	Specify byte type variable which becomes a waiting condition.	Least significant bit: 0:OFF 1:ON
10	ON/OFF	Specify on/off of the waiting condition.	

4. =/!=

When an InputGroup# (input group number), OutputGroup# (output group number), B variable number, B [array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
11	=	It is equal.	
12	!=	It is not equal.	

5. B Variable number / B [Array number] / Byte type constant

When an InputGroup# (input group number), OutputGroup# (output group number), B variable number, B [array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table after “=” or “!=” are added from the table in part 4 of this Explanation.

No	Parameter	Explanation	Note
13	B Variable number/ B [Array number]	Specify byte type variable which becomes a waiting condition.	
14	Byte type constant	The waiting condition is specified by byte type constant.	

6. Time=time

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
15	Time=time	Specify the waiting time. When the condition is specified and the specified waiting time passes, the next instruction is executed even if the various statuses do not coincide with the condition.	For the YRC1000 Timer : 0.01 to 655.35 seconds Variable I/I[] can be used. Units : 0.01 seconds For the YRC1000micro Timer : 0.001 to 65.535 seconds Variable I/I[] can be used. Units : 0.001 seconds

Example

(1) Wait Input#(12)=ON
Waits until input signal no.12 is turned on.

(2) Set B000 5
Set B002 16
Wait SpecificIn#(B000)=B002 Time=3.0

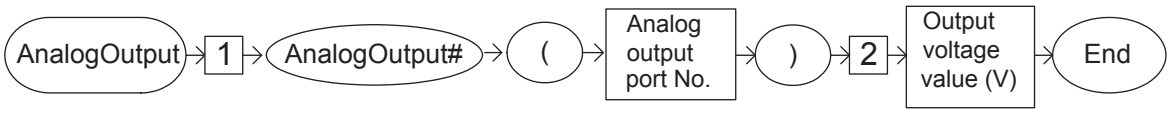
B002=16 (Decimal)=00010000 (Binary)
Waits until specific input signal no.5 is turned off.
However, after three seconds, even if the signal is not turned off, the next instruction is executed.

AnalogOut

Function

Outputs the set voltage value to the general-purpose analog output port.

Syntax



Explanation

1. AnalogOutput# (Analog output port number)

The following parameter must be added.

No	Parameter	Explanation	Note
1	AnalogOutput# (Analog output port number)	Specify the number of the analog output port to which the set voltage value is output.	No.: 1 to 40 Variable B/I/D can be used.

2. Output Analog Data parameter value

The following parameter must be added.

No.	Parameter	Explanation	Note
2	Output voltage value	Specify the output voltage value.	Voltage value: -14.00 to +14.00 Variable I/I[] can be used. (Units: 0.01 V)

Example

(1) Set I000 1270
AnalogOut AnalogOutput#(1) I000
The voltage of 12.7 V is output to the analog output port No. 1.

2.4 Math Instruction

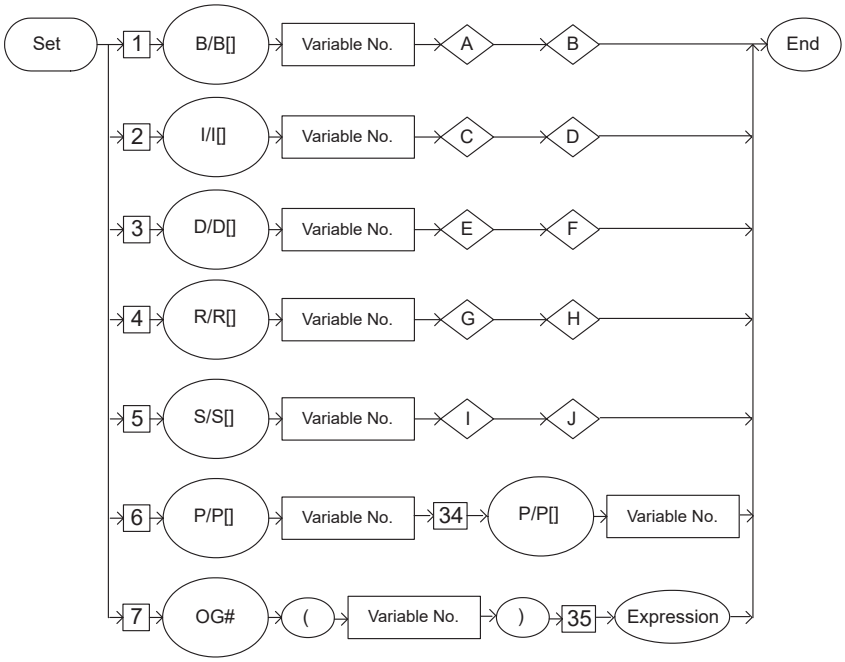
Set

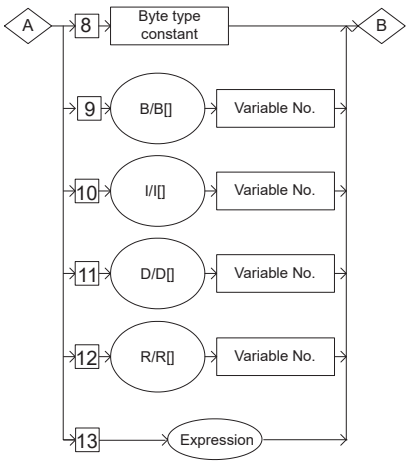
Function

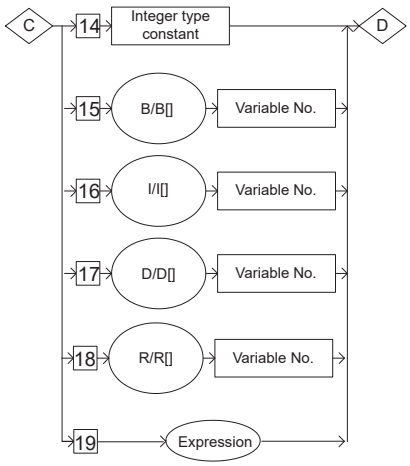
Sets Data 2 to Data 1.

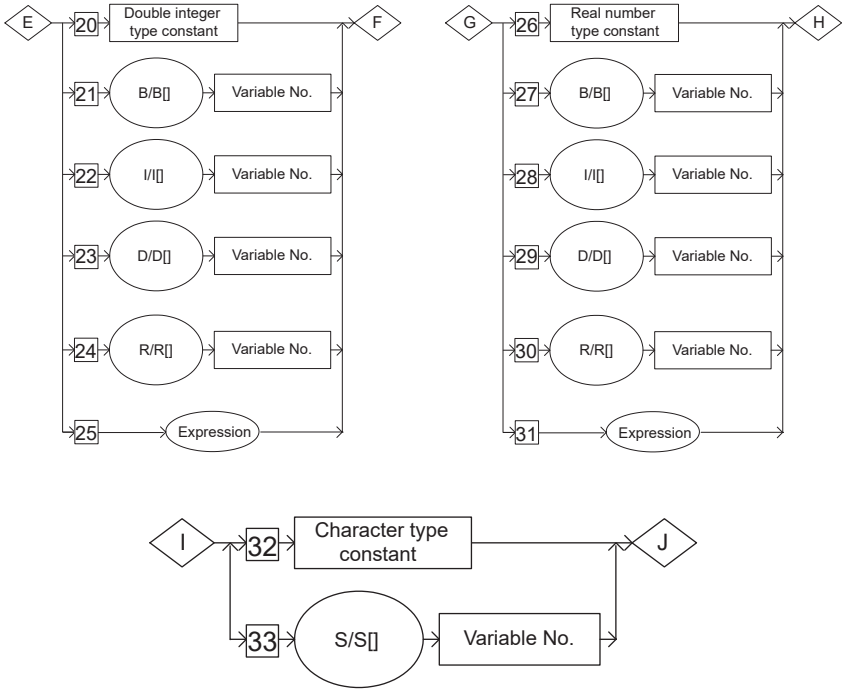
Syntax

SET <Data 1> <Data 2>








E
20
Double integer type constant
F
21
B/B[]
Variable No.
22
I/I[]
Variable No.
23
D/D[]
Variable No.
24
R/R[]
Variable No.
25
Expression
G
26
Real number type constant
H
27
B/B[]
Variable No.
28
I/I[]
Variable No.
29
D/D[]
Variable No.
30
R/R[]
Variable No.
31
Expression

I
32
Character type constant
J
33
S/S[]
Variable No.

Explanation

1. B Variable number /B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number] /R Variable number /R [Array number] /S Variable number /S [Array number] /P Variable number /P [Array number] /OG# (Output group number)

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to which the data is set.	< Data1>
2	I Variable number/ I [Array number]	Specify the number of the integer type variable to which the data is set.	< Data 1>
3	D Variable number/ D [Array number]	Specify the number of the double precision type variable which the data is set.	< Data 1>
4	R Variable number/ R [Array number]	Specify the number of the real type variable to which the data is set.	< Data 1>
5	S Variable number/ S [Array number]	Specifies the number of the character type variable to which data is set.	< Data 1>
6	P Variable number/ P [Array number]	Specify the number of the robot axis positional variable to which the data is set	< Data 1>
7	OG# (Output group number)	Specify the number of GP output group signal (1 group 8 points) to which the data is set.	< Data 1> No:1 to 512 Variable B/I/D can be used.

2. Byte type constant /B Variable number /B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number] /R Variable number /R [Array number] /Expression

When a B Variable number, B [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
8	Byte type constant	Specify the byte type constant.	< Data 2>
9	B Variable number/ B [Array number]	Specify the number of the byte type variable.	< Data 2>
10	I Variable number/ I [Array number]	Specify the number of the integer type variable.	< Data 2>
11	D Variable number/ D [Array number]	Specify the number of the double integer type variable.	< Data 2>
12	R Variable number/ R [Array number]	Specify the number of the real type variable.	< Data 2>
13	Expression	Specify the expression.	< Data 2> For details of setting the expression, refer to *chapter 1.3 "Expressions"*

3. Integer type constant /B Variable number /B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number] /R Variable number /R [Array number] /Expression

When an I Variable number, or I [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table..

No	Parameter	Explanation	Note
14	Integer type constant	Specify the integer type constant.	< Data 2>
15	B Variable number/ B [Array number]	Specify the number of the byte type variable.	< Data 2>
16	I Variable number/ I [Array number]	Specify the number of the integer type variable.	< Data 2>
17	D Variable number/ D [Array number	Specify the number of the double integer type variable.	< Data 2>
18	R Variable number/ R [Array number]	Specify the number of the real type variable.	< Data 2>
19	Expression.	Specify the expression	< Data 2> For details of setting the expression, refer to *chapter 1.3 "Expressions"*

4. Double precision type constant /B Variable number / B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number] /R Variable number /R [Array number] / Expression

When a D Variable number, or D [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
20	Double integer type constant	Specify the double integer type constant.	< Data 2>
21	B Variable number/ B [Array number]	Specify the number of the byte type variable.	< Data 2>
22	I Variable number/ I [Array number]	Specify the number of the integer type variable.	< Data 2>
23	D Variable number/ D [Array number]	Specify the number of the double precision type variable.	< Data 2>
24	R Variable number/ R [Array number]	Specify the number of the real type variable.	< Data 2>
25	Expression.	Specify the expression	< Data 2> For details of setting the expression, refer to *chapter 1.3 "Expressions"*

5. Real type constant /B Variable number /B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number] /R Variable number /R [Array number] /Expression

When an R Variable number, or R [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
26	Real type constant	Specify the real type constant	< Data 2>
27	B Variable number/ B [Array number]	Specify the number of the byte type variable.	< Data 2>
28	I Variable number/ I [Array number]	Specify the number of the integer type variable.	< Data 2>
29	D Variable number/ D [Array number]	Specify the number of the double precision type variable.	< Data 2>
30	R Variable number/ R [Array number]	Specify the number of the real type variable.	< Data 2>
31	Expression.	Specify the expression	< Data 2> For details of setting the expression, refer to *chapter 1.3 "Expressions"*

6. Character type constant /S Variable number /S [Array number]

When an S Variable number, or S [Array number] is selected from the table in part 1 of this Explanation, one of the parameter must be selected from the following table.

No	Parameter	Explanation	Note
32	Character type constant	Specify the character type data	< Data 2>
33	S Variable number/ S [Array number]	Specify the number of the byte type variable.	< Data 2>

7. P Variable number /P [Array number]

When a P Variable number, or P [Array number] is selected from the table in part 1 of this Explanation, the following parameter must be added.

No	Parameter	Explanation	Note
34	P Variable number/ P [Array number]	Specify the number of the robot axis position variable.	< Data 2>

8. Expression

When OG# (output group number) is selected from the table in part 1 of this Explanation, the following parameter must be added.

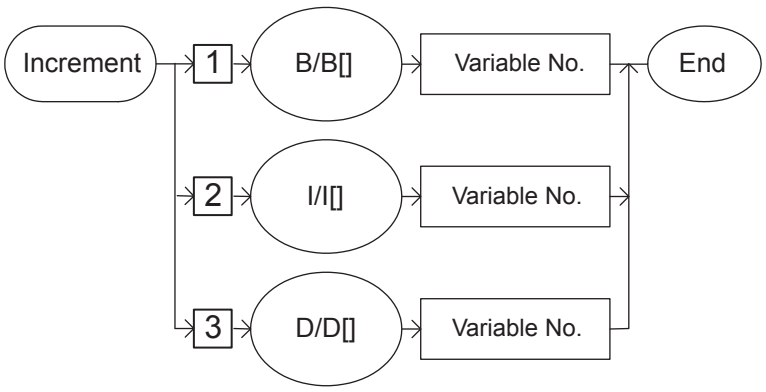
No	Parameter	Explanation	Note
35	Expression.	Specify the expression	< Data 2>
For details of setting the expression, refer to *chapter 1.3 “Expressions”*

Increment

Function

Adds one to the content of the specified variable.

Syntax



Explanation

1. B Variable number /B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number]

One of the parameters must be selected from the following table.

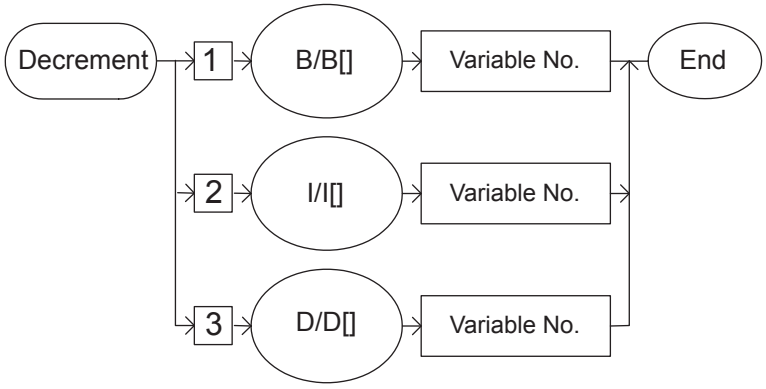
No	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable.	
2	I Variable number I [Array number]	Specify the number of the integer type variable.	
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable.	

Decrement

Function

Subtracts 1 from a specified variable.

Syntax



Explanation

1. B Variable number/B [Array number]/I Variable number/I [Array number]/ D Variable number/D [Array number]

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable from which 1 is subtracted.	
2	I Variable number/ I [Array number]	Specify the number of the integer type variable from which 1 is subtracted.	
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable from which 1 is subtracted.	

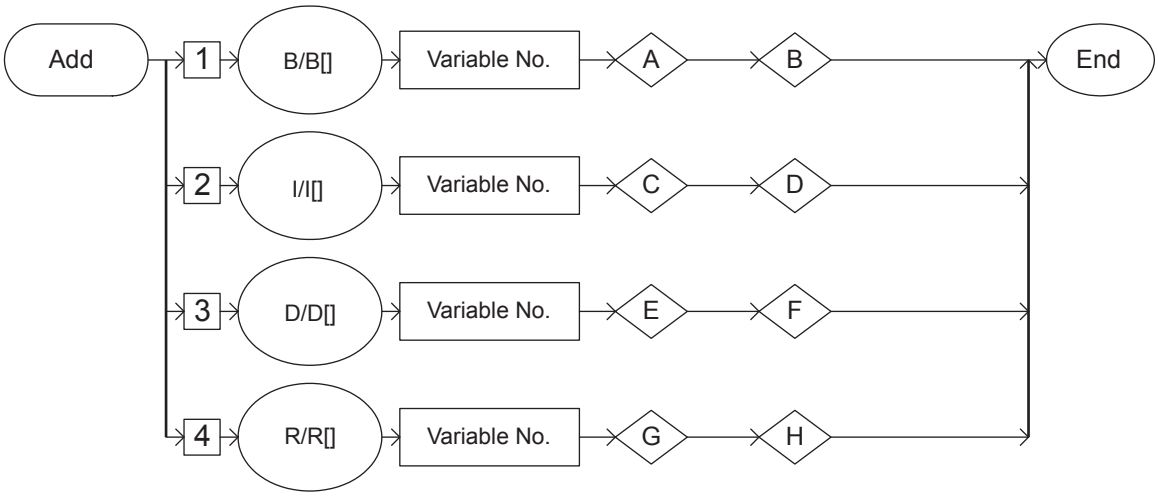
Add

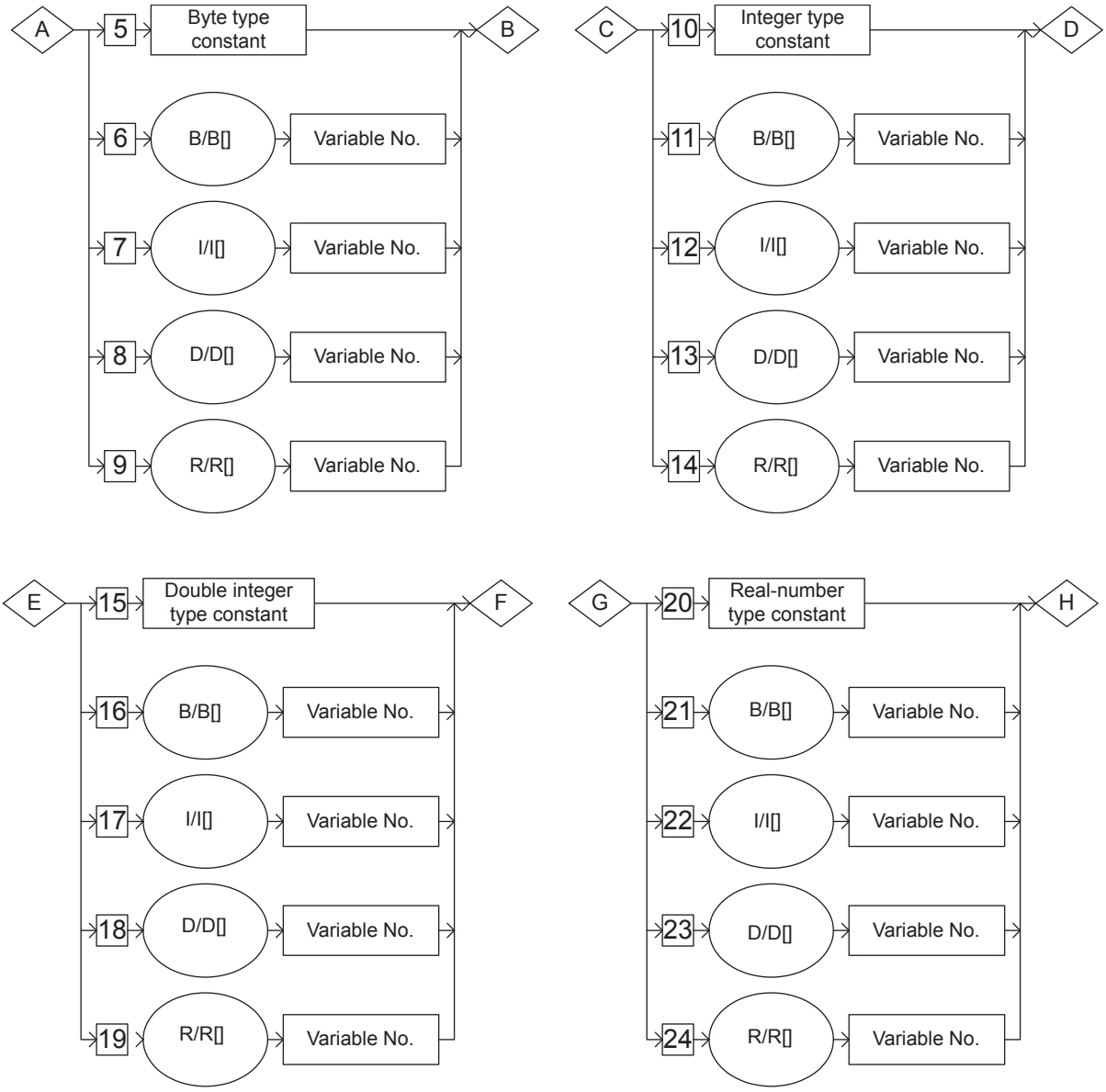
Function

Adds Data 1 and Data 2, and stores the result in Data 1.

Syntax

Add <Data 1> <Data 2>




A
5
Byte type constant
B
6
B/B[]
Variable No.
7
I/I[]
Variable No.
8
D/D[]
Variable No.
9
R/R[]
Variable No.
C
10
Integer type constant
D
11
B/B[]
Variable No.
12
I/I[]
Variable No.
13
D/D[]
Variable No.
14
R/R[]
Variable No.
E
15
Double integer type constant
F
16
B/B[]
Variable No.
17
I/I[]
Variable No.
18
D/D[]
Variable No.
19
R/R[]
Variable No.
G
20
Real-number type constant
H
21
B/B[]
Variable No.
22
I/I[]
Variable No.
23
D/D[]
Variable No.
24
R/R[]
Variable No.

Explanation

1. B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]/P Variable number/P [Array number]

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to be added.	<Data 1>
2	I Variable number/ I [Array number]	Specify the number of the integer type variable to be added.	<Data 1>
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be added.	<Data 1>
4	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be added.	<Data 1>

2. Byte type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a B Variable number, B [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
5	Byte type constant	Specify the byte type data to be added.	<Data 2>
6	B Variable number/ B [Array number]	Specify the number of the byte type variable to be added.	<Data 2>
7	I Variable number/ I [Array number]	Specify the number of the integer type variable to be added.	<Data 2>
8	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be added.	<Data 2>
9	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be added.	<Data 2>

3. Integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an I Variable number, I [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
10	Integer type constant	Specify the integer type data to be added.	<Data 2>
11	B Variable number/ B [Array number]	Specify the number of the byte type variable to be added.	<Data 2>
12	I Variable number/ I [Array number]	Specify the number of the integer type variable to be added.	<Data 2>
13	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be added.	<Data 2>
14	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be added.	<Data 2>

4. Double integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a D Variable number, D [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
15	Double integer type constant	Specify the double integer type data to be added.	<Data 2>
16	B Variable number/ B [Array number]	Specify the number of the byte type variable to be added.	<Data 2>
17	I Variable number/ I [Array number]	Specify the number of the integer type variable to be added.	<Data 2>
18	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be added.	<Data 2>
19	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be added.	<Data 2>

5. Real-number type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an R Variable number, R [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
20	Real-number type constant	Specify the real-number type data to be added.	<Data 2>
21	B Variable number/ B [Array number]	Specify the number of the byte type variable to be added.	<Data 2>
22	I Variable number/ I [Array number]	Specify the number of the integer type variable to be added.	<Data 2>
23	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be added.	<Data 2>
24	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be added.	<Data 2>

Example

(1) Add B000 10
Adds 10 to B000, and stores the result in B000.

(2) Add I000 I001
Adds I001 to I000, and stores the result in I000.

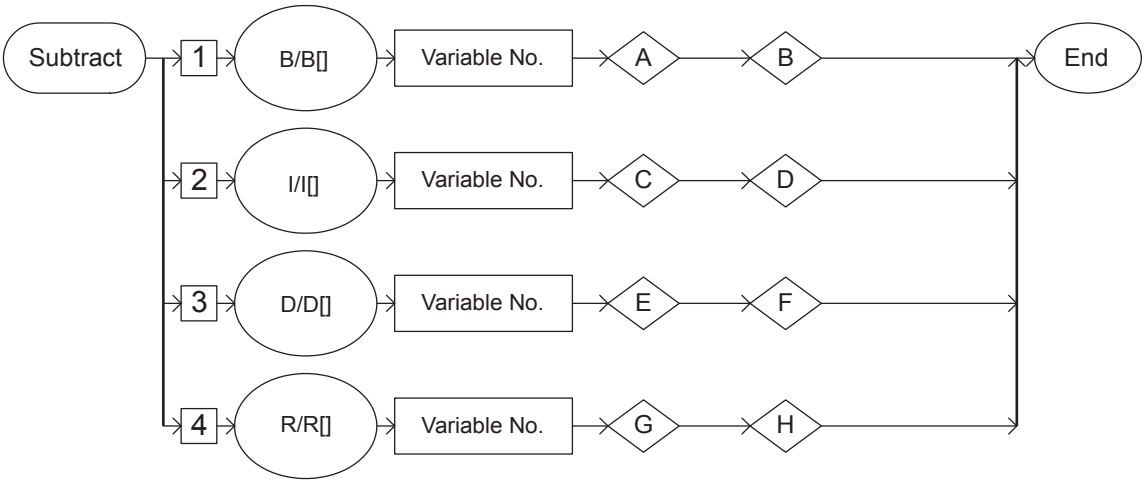
Subtract

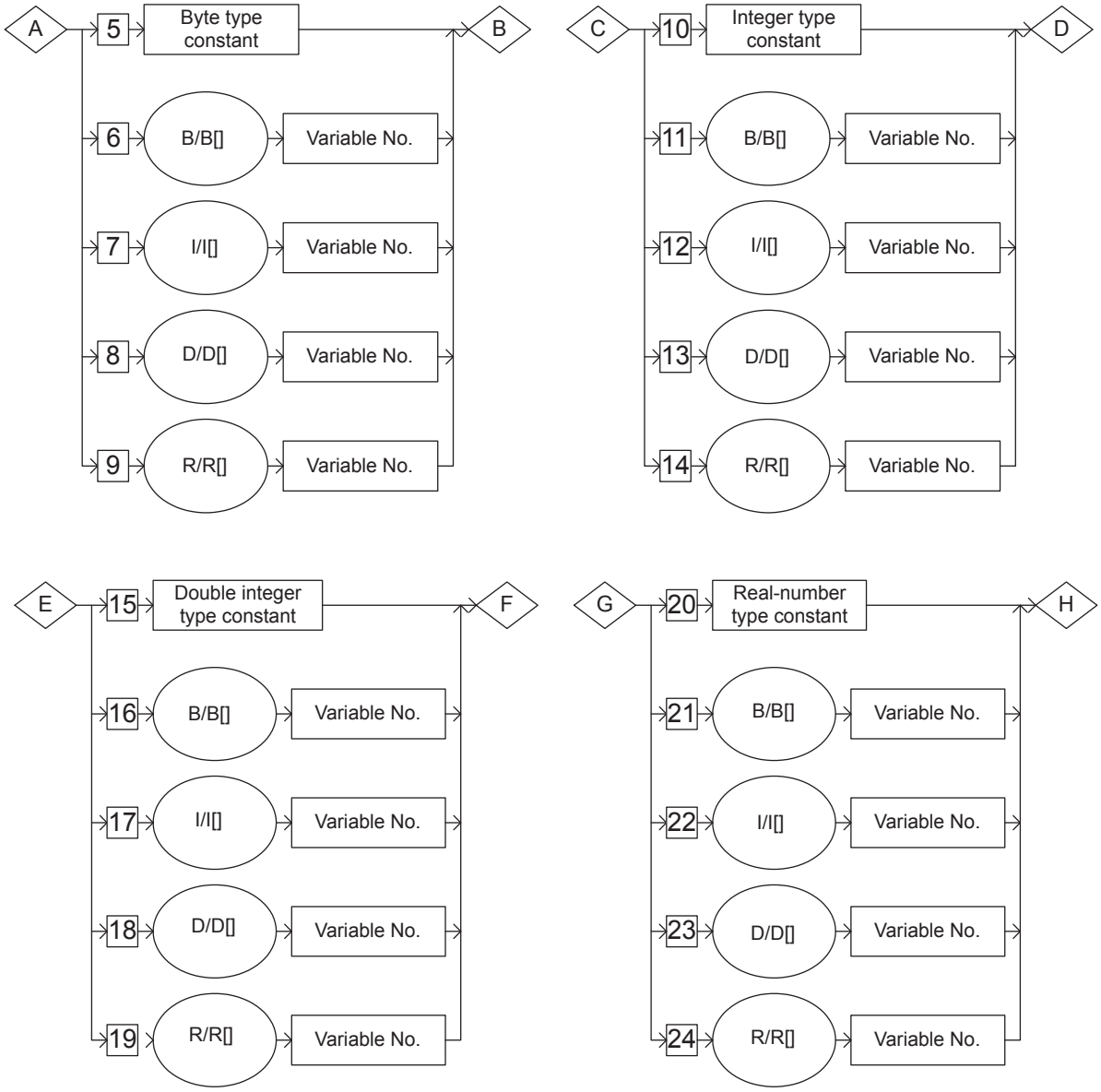
Function

Subtract Data 2 from Data 1, and stores the result in Data 1.

Syntax

Subtract <Data 1> <Data 2>




A
5
Byte type constant
B
6
B/B[]
Variable No.
7
I/I[]
Variable No.
8
D/D[]
Variable No.
9
R/R[]
Variable No.
C
10
Integer type constant
D
11
B/B[]
Variable No.
12
I/I[]
Variable No.
13
D/D[]
Variable No.
14
R/R[]
Variable No.
E
15
Double integer type constant
F
16
B/B[]
Variable No.
17
I/I[]
Variable No.
18
D/D[]
Variable No.
19
R/R[]
Variable No.
G
20
Real-number type constant
H
21
B/B[]
Variable No.
22
I/I[]
Variable No.
23
D/D[]
Variable No.
24
R/R[]
Variable No.

Explanation

1. B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to be subtracted.	<Data 1>
2	I Variable number/ I [Array number]	Specify the number of the integer type variable to be subtracted.	<Data 1>
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be subtracted.	<Data 1>
4	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be subtracted.	<Data 1>

2. Byte type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a B Variable number, B [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
5	Byte type constant	Specify the byte type data to be subtracted.	<Data 2>
6	B Variable number/ B [Array number]	Specify the number of the byte type variable to be subtracted.	<Data 2>
7	I Variable number/ I [Array number]	Specify the number of the integer type variable to be subtracted.	<Data 2>
8	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be subtracted.	<Data 2>
9	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be subtracted.	<Data 2>

3. Integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an I Variable number, I [Array number] is selected from the table part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
10	Integer type constant	Specify the integer type data to be subtracted.	<Data 2>
11	B Variable number/ B [Array number]	Specify the number of the byte type variable to be subtracted.	<Data 2>
12	I Variable number I [Array number]	Specify the number of the integer type variable to be subtracted.	<Data 2>
13	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be subtracted.	<Data 2>
14	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be subtracted.	<Data 2>

4. Double integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a D Variable number, D [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
15	Double integer type constant	Specify the double integer type data to be subtracted.	<Data 2>
16	B Variable number/ B [Array number]	Specify the number of the byte type variable to be subtracted.	<Data 2>
17	I Variable number/ I [Array number]	Specify the number of the integer type variable to be subtracted.	<Data 2>
18	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be subtracted.	<Data 2>
19	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be subtracted.	<Data 2>

5. Real-number type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an R Variable number, R [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
20	Real-number type constant	Specify the real-number type data to be subtracted.	<Data 2>
21	B Variable number/ B [Array number]	Specify the number of the byte type variable to be subtracted.	<Data 2>
22	I Variable number/ I [Array number]	Specify the number of the integer type variable to be subtracted.	<Data 2>
23	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be subtracted.	<Data 2>
24	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be subtracted.	<Data 2>

Example

(1) Subtract B000 10
Subtracts 10 from B000, and stores the result in B000.

(2) Subtract I000 I001
Subtracts I001 from I000, and stores the result in I000.

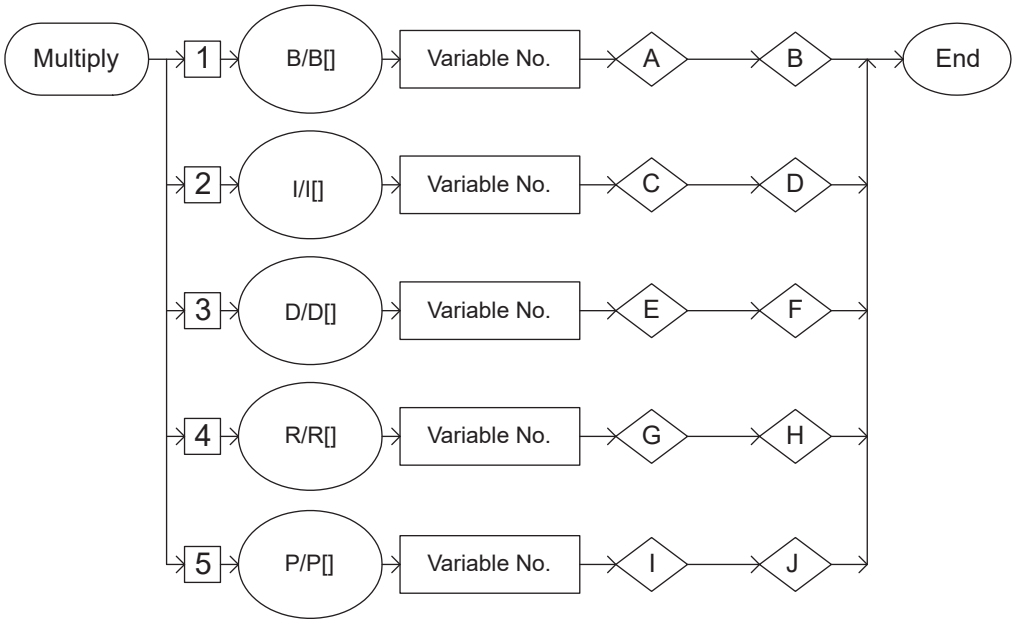
Multiply

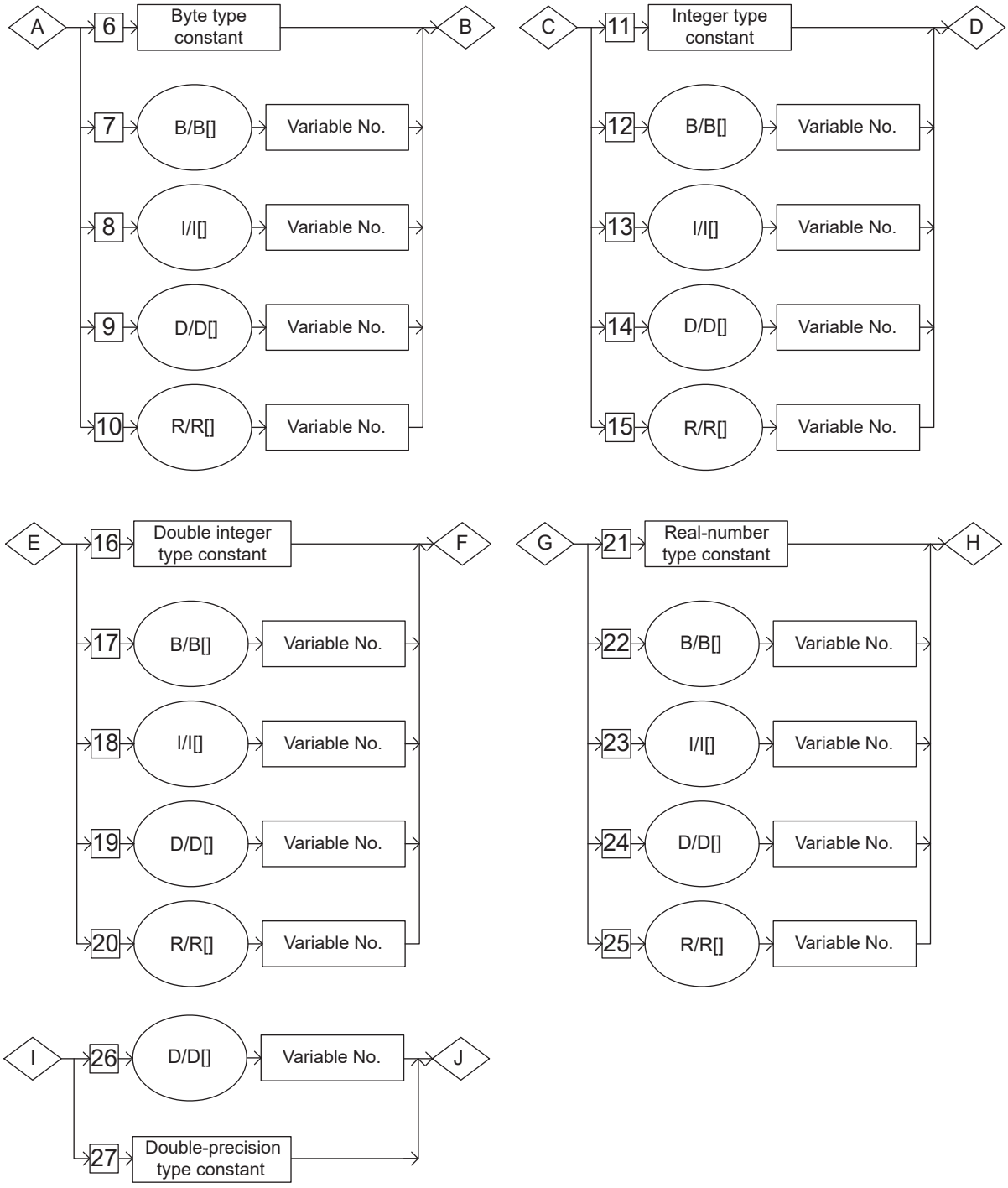
Function

Multiplies Data 1 by Data 2, and stores the result in Data 1.

Syntax

Multiply <Data 1> <Data 2>




A
6
Byte type constant
B
7
B/B[]
Variable No.
8
I/I[]
Variable No.
9
D/D[]
Variable No.
10
R/R[]
Variable No.
C
11
Integer type constant
D
12
B/B[]
Variable No.
13
I/I[]
Variable No.
14
D/D[]
Variable No.
15
R/R[]
Variable No.
E
16
Double integer type constant
F
17
B/B[]
Variable No.
18
I/I[]
Variable No.
19
D/D[]
Variable No.
20
R/R[]
Variable No.
G
21
Real-number type constant
H
22
B/B[]
Variable No.
23
I/I[]
Variable No.
24
D/D[]
Variable No.
25
R/R[]
Variable No.
I
26
D/D[]
Variable No.
J
27
Double-precision type constant

Explanation

1. B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]/P Variable number/P [Array number]

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to be multiplied.	<Data 1>
2	I Variable number/ I [Array number]	Specify the number of the integer type variable to be multiplied.	<Data 1>
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be multiplied.	<Data 1>
4	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be multiplied.	<Data 1>
5	P Variable number/ P [Array number]	Specifies the number of the robot position type variable to be multiplied	<Data 1>

2. Byte type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a B Variable number or B [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
6	Byte type constant	Specify the byte type data to be multiplied.	<Data 2>
7	B Variable number/ B [Array number]	Specify the number of the byte type variable to be multiplied.	<Data 2>
8	I Variable number/ I [Array number]	Specify the number of the integer type variable to be multiplied.	<Data 2>
9	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be multiplied.	<Data 2>
10	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be multiplied.	<Data 2>

3. Integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an I Variable number or I [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
11	Integer type constant	Specify the integer type data to be multiplied.	<Data 2>
12	B Variable number/ B [Array number]	Specify the number of the byte type variable to be multiplied.	<Data 2>
13	I Variable number/ I [Array number]	Specify the number of the integer type variable to be multiplied.	<Data 2>
14	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be multiplied.	<Data 2>
15	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be multiplied.	<Data 2>

4. Double integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a D Variable number or D [Array number is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
16	Double integer type constant	Specify the double integer type data to be multiplied.	<Data 2>
17	B Variable number/ B [Array number]	Specify the number of the byte type variable to be multiplied.	<Data 2>
18	I Variable number/ I [Array number]	Specify the number of the integer type variable to be multiplied.	<Data 2>
19	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be multiplied.	<Data 2>
20	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be multiplied.	<Data 2>

5. Real-number type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an R Variable number or R [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
21	Real-number type constant	Specify the real-number type data to be multiplied.	<Data 2>
22	B Variable number/ B [Array number]	Specify the number of the byte type variable to be multiplied.	<Data 2>
23	I Variable number/ I [Array number]	Specify the number of the integer type variable to be multiplied.	<Data 2>
24	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be multiplied.	<Data 2>
25	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be multiplied.	<Data 2>

6. D Variable number/D [Array number]/ Double-precision type constant

When a P Variable number or P [Array number] is selected from the table in part 1 of this Explanation, choose one of the parameters from the following table:

No.	Parameter	Explanation	Note
26	D Variable number/ D {Array Number]	Specifies the number of the double-precision variable by which the element of position type variable is multiplied.	<Data 2>
27	Double-precision type constant	Specifies the double-precision type data by which the position type variable is multiplied.	<Data 2>

Example

(1) Multiply B000 10
Multiplies B000 by 10, and stores the result in B000.

(2) Multiply I000 I001
Multiplies I000 by I001, and stores the result in I000.

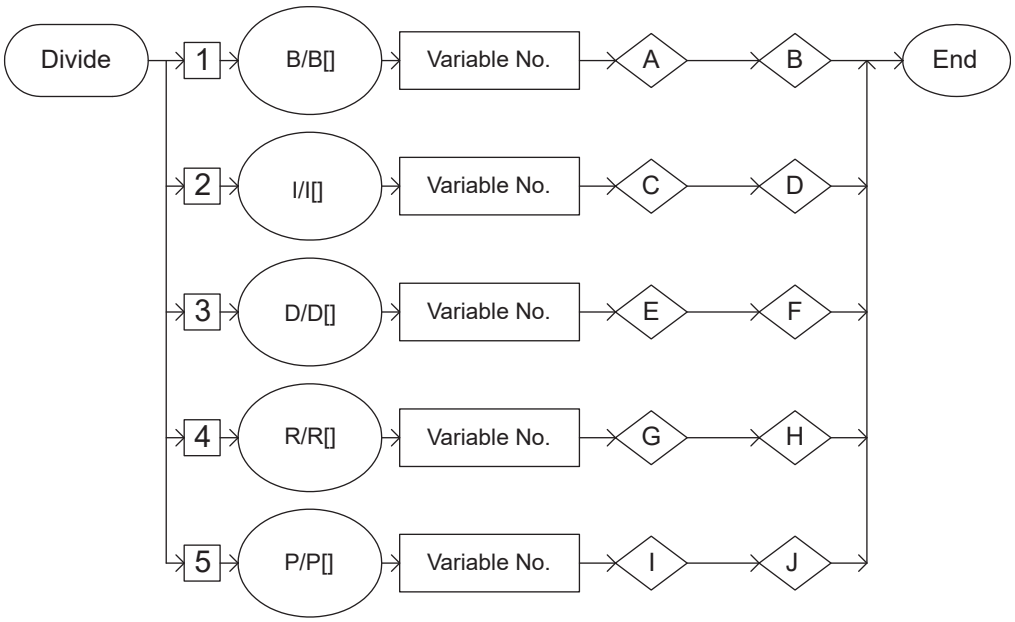
Divide

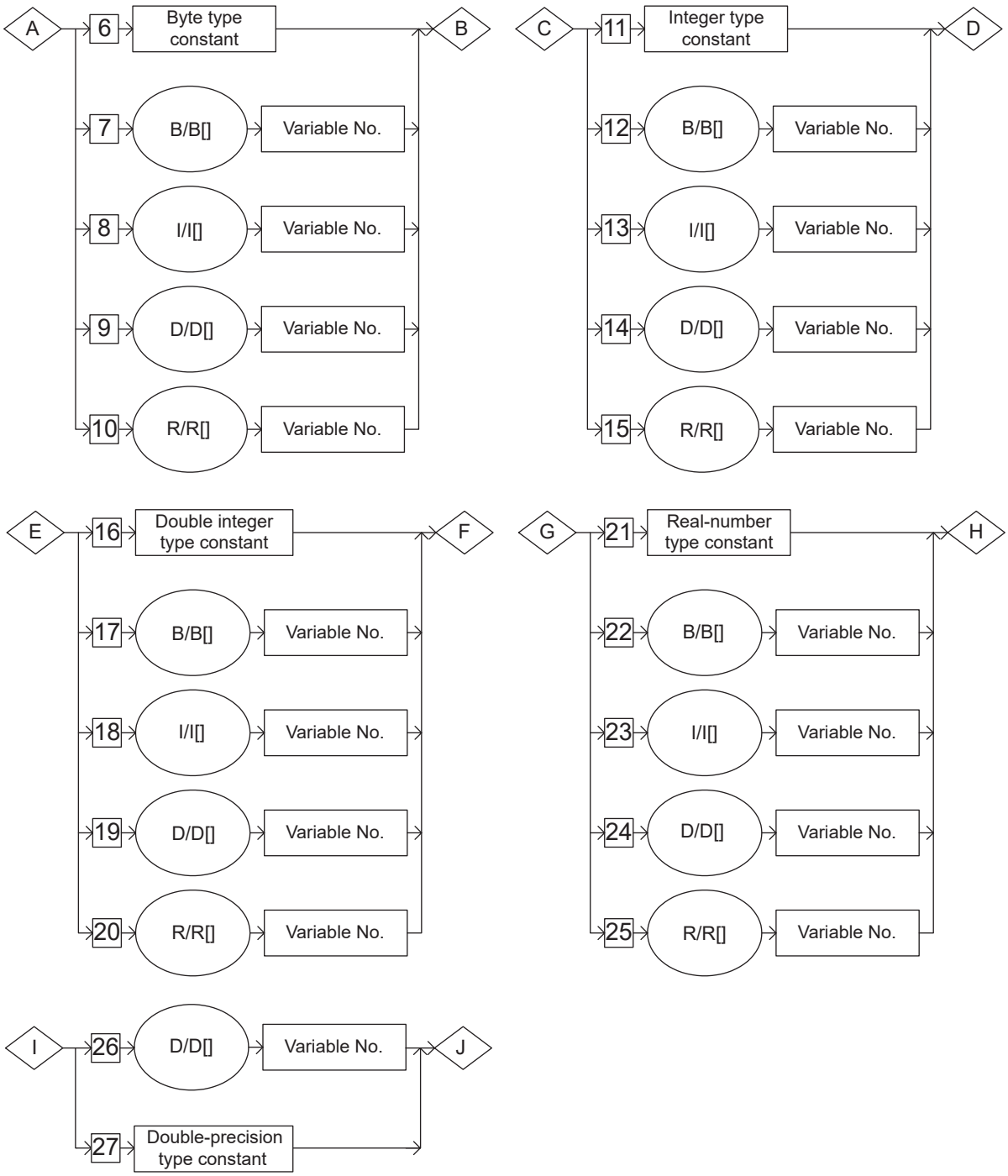
Function

Divides Data 1 by Data 2, and stores the result in Data 1.

Syntax

Divide <Data 1> <Data 2>




A
6
Byte type constant
B
7
B/B[]
Variable No.
8
I/I[]
Variable No.
9
D/D[]
Variable No.
10
R/R[]
Variable No.
C
11
Integer type constant
D
12
B/B[]
Variable No.
13
I/I[]
Variable No.
14
D/D[]
Variable No.
15
R/R[]
Variable No.
E
16
Double integer type constant
F
17
B/B[]
Variable No.
18
I/I[]
Variable No.
19
D/D[]
Variable No.
20
R/R[]
Variable No.
G
21
Real-number type constant
H
22
B/B[]
Variable No.
23
I/I[]
Variable No.
24
D/D[]
Variable No.
25
R/R[]
Variable No.
I
26
D/D[]
Variable No.
J
27
Double-precision type constant

Explanation

1. B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]/P Variable number/P [Array number]

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to be divided.	<Data 1>
2	I Variable number/ I [Array number]	Specify the number of the integer type variable to be divided.	<Data 1>
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be divided.	<Data 1>
4	R Variable number/ R [Array number]	Specify the number of the real-number type variable to be divided.	<Data 1>
5	P Variable number/ P [Array number]	Specifies the number of the robot position type variable to be divided.	<Data 1>

2. Byte type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a B Variable number or B [Array number] is selected from the table in part 1 of this explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
6	Byte type constant	Specify the byte type data by which Data 1 is divided.	<Data 2>
7	B Variable number/ B [Array number]	Specify the number of the byte type variable by which Data 1 is divided.	<Data 2>
8	I Variable number/ I [Array number]	Specify the number of the integer type variable by which Data 1 is divided.	<Data 2>
9	D Variable number/ D [Array number]	Specify the number of the double integer type variable by which Data 1 is divided.	<Data 2>
10	R Variable number/ R [Array number]	Specify the number of the real-number type variable by which Data 1 is divided.	<Data 2>

3. Integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an I Variable number or I [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
11	Integer type constant	Specify the integer type data by which Data 1 is divided.	<Data 2>
12	B Variable number/ B [Array number]	Specify the number of the byte type variable by which Data 1 is divided.	<Data 2>
13	I Variable number/ I [Array number]	Specify the number of the integer type variable by which Data 1 is divided.	<Data 2>
14	D Variable number/ D [Array number]	Specify the number of the double integer type variable by which Data 1 is divided.	<Data 2>
15	R Variable number/ R [Array number]	Specify the number of the real-number type variable by which Data 1 is divided.	<Data 2>

4. Double integer type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When a D Variable number or D [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
16	Double integer type constant	Specify the double integer type data by which Data 1 is divided.	<Data 2>
17	B Variable number/ B [Array number]	Specify the number of the byte type variable by which Data 1 is divided.	<Data 2>
18	I Variable number/ I [Array number]	Specify the number of the integer type variable by which Data 1 is divided.	<Data 2>
19	D Variable number/ D [Array number]	Specify the number of the double integer type variable by which Data 1 is divided.	<Data 2>
20	R Variable number/ R [Array number]	Specify the number of the real-number type variable by which Data 1 is divided.	<Data 2>

5. Real-number type constant/B Variable number/B [Array number]/I Variable number/I [Array number]/D Variable number/D [Array number]/R Variable number/R [Array number]

When an R Variable number or R [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
21	Real-number type constant	Specify the real-number type data by which Data 1 is divided.	<Data 2>
22	B Variable number/ B [Array number]	Specify the number of the byte type variable by which Data 1 is divided.	<Data 2>
23	I Variable number/ I [Array number]	Specify the number of the integer type variable by which Data 1 is divided.	<Data 2>
24	D Variable number/ D [Array number]	Specify the number of the double integer type variable by which Data 1 is divided.	<Data 2>
25	R Variable number/ R [Array number]	Specify the number of the real-number type variable by which Data 1 is divided.	<Data 2>

6. D Variable number/D [Array number]/Double-precision type constant

When a P Variable number or P [Array number] is selected from the table in part 1 of this Explanation, choose one of the parameters from the following table:

No.	Parameter	Explanation	Note
26	P Variable number/ P [Array number]	Specifies the number of the double-precision variable by which the position type variable is divided.	<Data 2>
27	Double-precision type constant	Specifies the double-precision type data by which the position type variable is divided.	<Data 2>

Example

(1) Divide B000 10
Divides B000 by 10, and stores the result in B000.

(2) Divide I000 I001
Divides I000 by I001, and stores the result in I000.

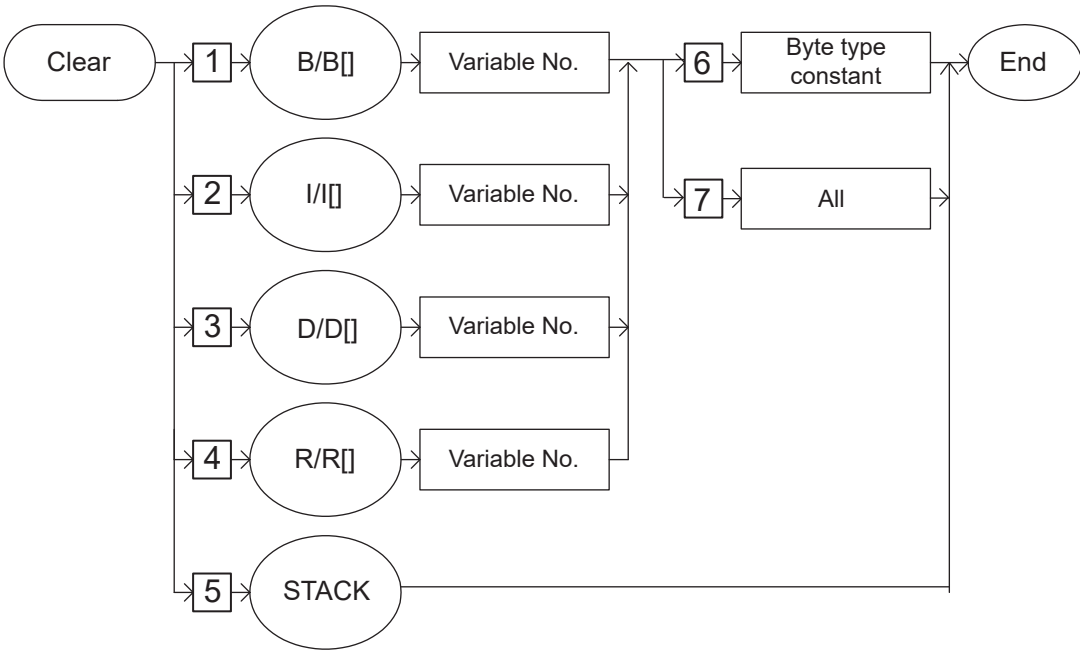
Clear

Function

In Data 1, the variable content from the specified number on, is cleared to 0 only by the amount specified in Data 2.

Syntax

Clear <Data 1> <Data 2>



Explanation

1. B Variable number /B [Array number] /I Variable number /I [Array number] /D Variable number /D [Array number] /R Variable number/R [Array number]/STACK

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to be cleared.	< Data 1 >
2	I Variable number/ I [Array number]	Specify the number of the integer type variable to be cleared.	< Data 1 >

No	Parameter	Explanation	Note
3	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be cleared.	< Data 1 >
4	R Variable number/ R [Array number]	Specify the number of the real type variable to be cleared.	< Data 1 >
5	STACK	Clear all the job call stack. The stack has twelve stack levels in total (the stack level 12) and stores the called position.	< Data 1 >

2. Byte type constant

When a B Variable number, B [Array number], I Variable number, I [Array number], D Variable number, D [Array number], R Variable number, R [Array number] is selected from the table in part 1 of this Explanation, one of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
6	Byte type constant	Specify the number cleared starting from the number of the specified variable.	<Data 2>
7	All	All variables starting from the number of the specified variable are cleared	<Data 2>

Example

(1) Clear B003 10
The content of the variables from B003 to B012 are cleared to 0.

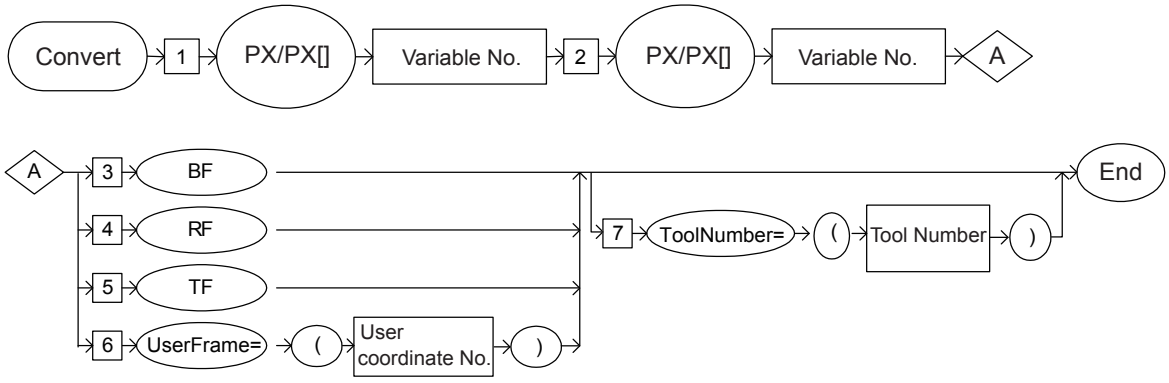
Convert

Function

Converts the joint angle type position variable of Data 2 to the XYZ type position variable in the specified coordinate system, and stores the result in Data 1.

Syntax

Convert <Data 1> <Data 2> Coordinate system designation



Explanation

1. PX Variable number/PX [Array number]

PX Variable is same as P Variable, when only one robot is in the group.

The following parameter must be added.

No.	Parameter	Explanation	Note
1	PX Variable number/ PX [Array number]	Specify the number of the expanded position type variable where the converted data is stored.	<Data 1>

2. PX Variable number/PX [Array number]

PX Variable is same as P Variable, when only one robot is in the group.

The following parameter must be added.

No.	Parameter	Explanation	Note
2	PX Variable number/ PX [Array number]	Specify the number of the expanded position type variable to be converted.	<Data 2>

3. BF/RF/TF/UserFrame= (User coordinate number)

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
3	BF	Specify the conversion in the base coordinate system.	
4	RF	Specify the conversion in the robot coordinate system.	
5	TF	Specify the conversion in the tool coordinate system.	
6	UserFrame= (User coordinate number)	Specify the conversion in the user coordinate system.	No.: 1 to 63 Variable B/I/D can be used.

4. ToolNumber= (Tool Number)

The following parameter can be added or omitted.

No.	Parameter	Explanation	Note
7	ToolNumber= (Tool Number)	Specify the tool number.	No.: 0 to 63 Variable B/I/D can be used.

Example

(1) Convert PX000 PX001 BF

For the job Robot 1, the joint angle type position data of P001 is converted to the XYZ type position data in the base coordinate system and stores the converted data in P000.

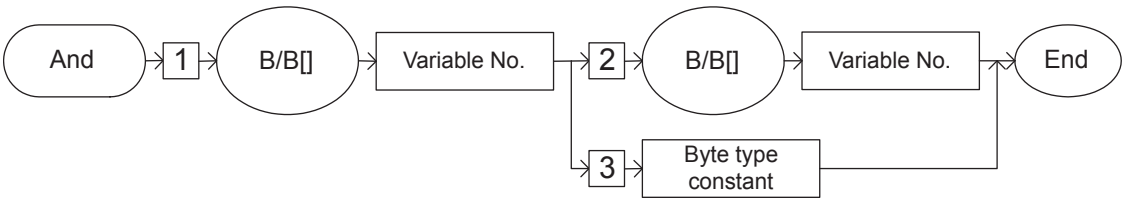
And

Function

Carries out logical multiplication of Data 1 and Data 2, and stores the result in Data 1.

Syntax

And <Data 1> <Data 2>



Explanation

1. B Variable number/B [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the logical multiplication is carried out.	<Data 1>

2. B Variable number/B [Array number]/Byte type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the logical multiplication is carried out.	<Data 2>
3	Byte type constant	Specify the byte type data for which the logical multiplication is carried out.	<Data 2>

Example

(1) Set B000 5
Set B010 1
And B000 B010

Carries out the logical multiplication of B000 (0000 0101) and B010 (0000 0001), and stores the result (0000 0001=1) in B000.

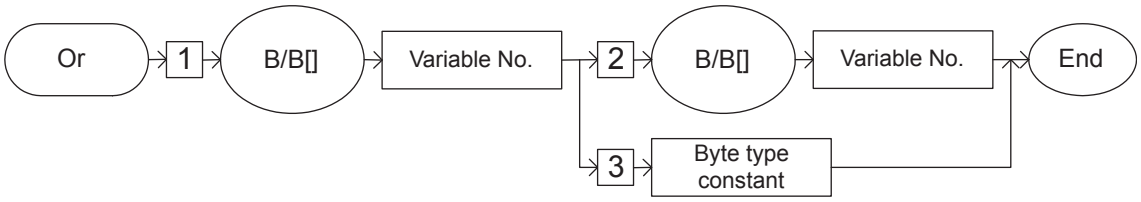
Or

Function

Carries out the logical sum of Data 1 and Data 2, and stores the result in Data 1.

Syntax

Or <Data 1> <Data 2>



Explanation

1. B Variable number/B [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the logical sum is carried out.	<Data 1>

2. B Variable number/B [Array number]/Byte type constant

One of the parameters must be selected from the following table.

No	Parameter	Explanation	Note
2	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the logical sum is carried out.	<Data 2>
3	Byte type constant	Specify the byte type data for which the logical sum is carried out.	<Data 2>

Example

(1) Set B000 5
Set B010 10
Or B000 B010

Carries out the logical sum of B000 (0000 0101) and B010 (0000 1010), and stores the result (0000 1111=15) in B000.

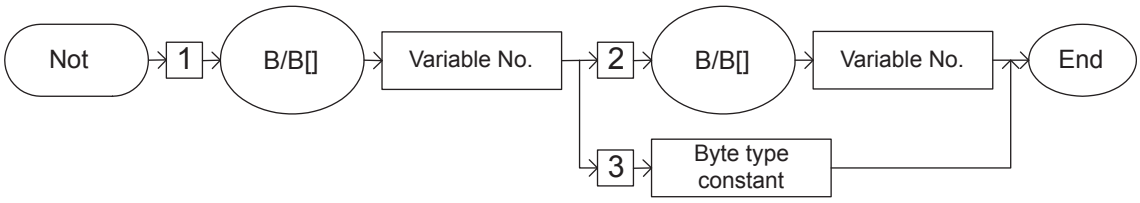
Not

Function

Carries out the logical negation of Data 2, and stores the result in Data 1.

Syntax

Not <Data 1> <Data 2>



Explanation

1. B Variable number/B [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable to store the result of logical negation.	<Data 1>

2. B Variable number/B [Array number]/Byte type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the logical negation is carried out.	<Data 2>
3	Byte type constant	Specify the byte type data for which the logical negation is carried out.	<Data 2>

Example

(1) Set B000 0
Set B010 1
Not B000 B010

Carries out the logical negation of B010 (0000 0001), and stores the result (1111 1110=254) in B000.

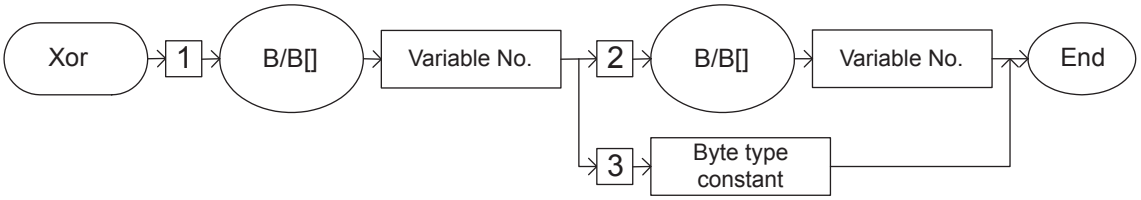
Xor

Function

Carries out the logical exclusive OR of Data 1 and Data 2, and stores the result in Data 1.

Syntax

Xor <Data 1> <Data 2>



Explanation

1. B Variable number/B [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the exclusive OR is carried out.	<Data 1>

2. B Variable number/B [Array number]/Byte type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	B Variable number/ B [Array number]	Specify the number of the byte type variable for which the exclusive OR is carried out.	<Data 2>
3	Byte type constant	Specify the byte type data for which the exclusive OR is carried out.	<Data 2>

Example

(1) Set B000 1
Set B010 5
Xor B000 B010

Carries out the exclusive OR of B000 (0000 0001) and B010 (0000 0101), and stores the result (0000 0100=4) in B000.

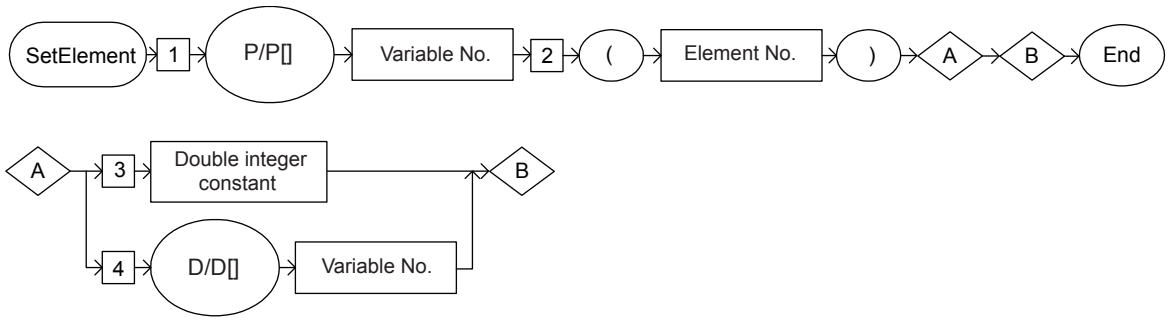
SetElement

Function

Sets Data 2 in the element of position type variable of Data 1.

Syntax

SetElement <Data 1> <Data 2>



Explanation

1. P Variable number/P [Array number]

The following parameter must be added

No.	Parameter	Explanation	Note
1	P Variable number/ P [Array number]	Specify the number of the robot axis position type variable where Data 2 is set as an element.	<Data 1> Unit is 0.001. So when set number is 12345, then the coordinate will be 12.345

2. Element number

The following parameter must be added.

No.	Parameter	Explanation	Note
2	Element number	Specify the element of the position type variable to be set.	<Data 2> Element No.: 1 to 255 Variable B can be used.

Element of position type variable

The element of position type variable differs depending on the type of variable as follows.

• Robot axis position type variable

<Joint angle type>

(1): 1st axis data, (2): 2nd axis data, (3): 3rd axis data, (4) 4th axis data, (5): 5th axis data, (6) 6th axis data

<XYZ type>

(1): X axis data, (2): Y axis data, (3) Z axis data,
(4): Rx axis data (5): Ry axis data, (6): Rz axis data

3. Double integer type constant/D Variable number/D [Array number]

When an element number is selected from the table in part 2 above, one of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
3	Double integer type constant	Specify the double integer type data to be set.	
4	D Variable number/ D [Array number]	Specify the number of the double integer type variable to be set.	

Example

(1) SetElement P000 (3) 2000

2000 is set in the Z axis data of P000.

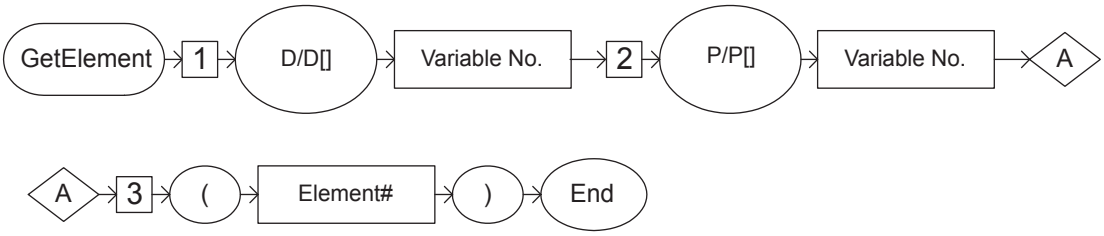
GetElement

Function

Stores the element of position type variable of Data 2 in Data 1.

Syntax

GetElement <Data 1> <Data 2> (Element number)



Explanation

1. D Variable number/D [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	D Variable number/ D [Array number]	Specify the number of the double integer type variable where the element of position type variable is stored.	<Data 1>

2. P Variable number/P [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
2	P Variable number/ P [Array number]	Specify the number of the robot axis position type variable to be set.	<Data 2>

3. (Element number)

The following parameter must be added.

No.	Parameter	Explanation	Note
3	(Element number)	Specify the number of the of the position type variable element to be stored.	Element No.: 1 to 255 Variable B can be used.

Example

(1) GetElement D000 P000 (3)

The Z axis data of P000 is stored in D000.

GetSystemVar

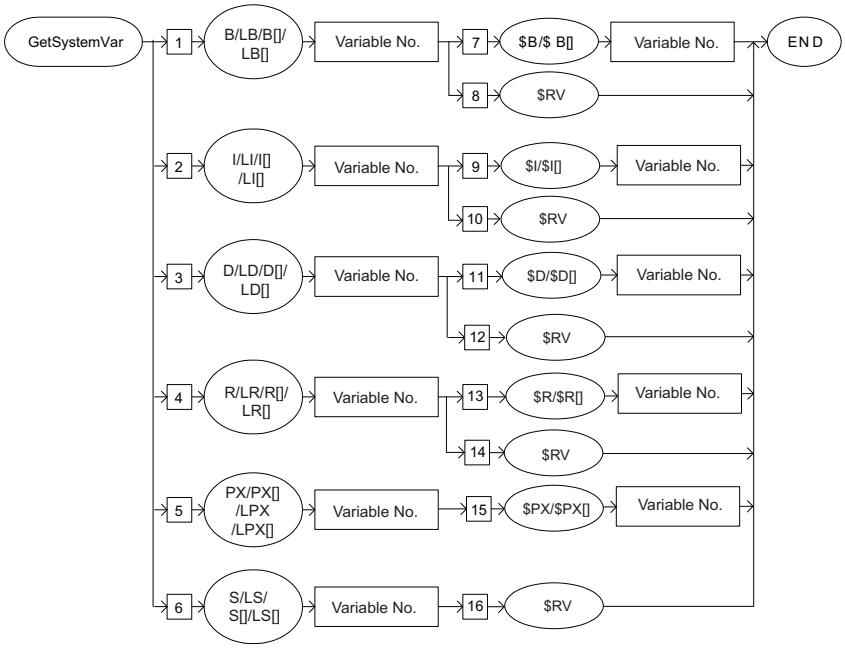
SUBSET	STANDARD	EXPANDED
Not available	Available	Available

Function

Stores the system variable of Data 2 in Data 1.

Construction

GetSystemVar <Data 1> <Data 2>



Explanation

1. B Variable number/LB Variable number/B [Array number]/LB [Array number]/I Variable number/LI Variable number/I [Array number]/LI [Array number]/ D Variable number/LD Variable number/D [Array number]/LD [Array number]/R Variable number/LR Variable number/R [Array number]/LR [Array number]/ PX [Array number]/LPX [Array number]/S Variable number/LS Variable number/S [Array number]/LS [Array number]

Choose one of the parameters from the following table.

No.	Parameter	Explanation	Note
1	B Variable number/ LB Variable number/ B [Array number]/ LB [Array number]	Specifies the number of the byte type variable to store the system variable.	<Data 1>
2	I Variable number/ LI Variable number/ I [Array number]/ LI [Array number]	Specifies the number of the integer type variable to store the system variable.	<Data 1>
3	D Variable number/ LD Variable number/ D [Array number]/ LD [Array number]	Specifies the number of the double-precision type variable to store the system variable.	<Data 1>
4	R Variable number/ LR Variable number/ R [Array number]/ LR [Array number]	Specifies the number of the real-number type variable to store the system variable.	<Data 1>
5	PX Variable number/ LPX Variable number/ PX [Array number]/ LPX [Array number]	Specifies the number of the expanded position type variable to store the system variable.	<Data 1>
6	S Variable number/ LS Variable number/ S [Array number]/ LS [Array number]	Specifies the number of the character string type variable to store the system variable.	<Data 1>

System variable

The system variables are written by the YRC Controller system and can be referred only by a GetSystemVar instruction. The following system variables are available.

System Variable	Type	No	Explanation
$PX type variable	Expanded position type	$PX040	Path correction amount (available only with the optional COMARC function)
		$PX041	Base coordinate shift amount
		$PX042	Robot coordinate shift amount
		$PX043	Tool coordinate shift amount
		$PX044	User coordinate shift amount
		$PX045	3D shift amount
		$PX050	Each axis torque command position
		$PX100 to $PX149	RIN#(1)'s detected position by the optional NSRCH instruction (pulse type)
		$PX150 to $PX199	RIN#(1)'s detected position by the optional NSRCH instruction (XYZ type)
		$PX200 to $PX249	RIN#(2)'s detected position by the optional NSRCH instruction (pulse type)
		$PX250 to $PX299	RIN#(2)'s detected position by the optional NSRCH instruction (XYZ type)
		$PX300 to $PX349	RIN#(3)'s detected position by the optional NSRCH instruction (pulse type)
		$PX350 to $PX399	RIN#(3)'s detected position by the optional NSRCH instruction (XYZ type)
		$PX400 to $PX449	RIN#(4)'s detected position by the optional NSRCH instruction (pulse type)
		$PX450 to $PX499	RIN#(4)'s detected position by the optional NSRCH instruction (XYZ type)
		$PX500 to $PX549	RIN#(5)'s detected position by the optional NSRCH instruction (pulse type)
		$PX550 to $PX599	RIN#(5)'s detected position by the optional NSRCH instruction (XYZ type)
		$PX600 to $PX649	RIN#(6)'s detected position by the optional NSRCH instruction (pulse type)
		$PX650 to $PX699	RIN#(6)'s detected position by the optional NSRCH instruction (XYZ type)

Continued

SUPPLE -MENT

Continued

System Variable	Type	No	Explanation
$B type variable	Byte type	$B001	Execution series number (0 to 17)
		$B002	Detected /Not detected by the optional SRCH/ NSRCH instruction 0: Not detected, 1: Detected
		$B008	Result of the optional SYSTART instruction 1: Normal termination, 0: Abnormal termination
		$B009	Result of the SETFILE/GETFILE instruction 0: Normal termination, Other than 0: Abnormal termination
		$B014	Result of the optional HSEN instruction 1: Setting status completed, 0: Others
		$B016	The number of RIN#(1)s detected by the optional NSRCH instruction
		$B017	The number of RIN#(2)s detected by the optional NSRCH instruction
		$B018	The number of RIN#(3)s detected by the optional NSRCH instruction
		$B019	The number of RIN#(4)s detected by the optional NSRCH instruction
		$B020	The number of RIN#(5)s detected by the optional NSRCH instruction
		$B021	The number of RIN#(6)s detected by the optional NSRCH instruction
$PX type variable	Expanded position type	$PX000	Current value (pulse type)
		$PX001	Current value (XYZ type)
		$PX002	Position detected by the optional SRCH instruction (pulse type)
		$PX003	Position detected by the optional STCH instruction (XYZ type)
		$PX004	Current value excluding the shift amount (XYZ type)
		$PX005	Teaching position (pulse type)
		$PX006	Operation target position (pulse type)
		$PX007	Current position excluding the shift amount and profiling amount (XYZ type)
		$PX008	F/B pulse → current value (XYZ type)
		$PX009	The unit vector (XYZ type) of the travel direction excluding the real time correction amount
		$PX010	F/B pulse

2. $B Variable number/$B [Array number]/$RV

When a B Variable number, LB Variable number, B [Array number], or LB [Array number] is selected from the table in part 1 of this Explanation, add the following parameter.

No.	Parameter	Explanation	Note
7	$B Variable number/ $B [Array number]	Specifies the number of the byte type system variable to be stored.	<Data 2>
8	$RV	Receives the returned value of the job	<Data 2>

3. $I Variable number/$I [Array number]/$RV

When an I Variable number, LI Variable number, I [Array number], or LI [Array number] is selected from the table in part 1 of this Explanation, add the following parameter.

No.	Parameter	Explanation	Note
9	$I Variable number/ $I [Array number]	Specifies the number of the integer type system variable to be stored.	<Data 2>
10	$RV	Receives the returned value of the job	<Data 2>

4. $D Variable number/$D [Array number]/$RV

When a D Variable number, LD Variable number, D [Array number], or LD [Array number] is selected from the table in part 1 of this Explanation, add the following parameter.

No.	Parameter	Explanation	Note
11	$D Variable number/ $D [Array number]	Specifies the number of the double-precision type system variable to be stored.	<Data 2>
12	$RV	Receives the returned value of the job	<Data 2>

5. $R Variable number/$R [Array number]/$RV

When an R Variable number, LR Variable number, R [Array number], or LR [Array number] is selected from the table in part 1 of this Explanation, add the following parameter.

No.	Parameter	Explanation	Note
13	$R Variable number/ $R [Array number]	Specifies the number of the real-number type system variable to be stored.	<Data 2>
14	$RV	Receives the returned value of the job	<Data 2>

6. $PX Variable number/$PX [Array number]

When a PX Variable number, LPX Variable number, PX [Array number], or LPX [Array number] is selected from the table in part 1 of this Explanation, add the following parameter.

No.	Parameter	Explanation	Note
15	$PX Variable number/ $PX [Array number]	Specifies the number of the expanded position type system variable to be stored.	<Data 2>

7. $RV

When a S Variable number, LS Variable number, S [Array number], or LS [Array number] is selected from the table in part 1 of this Explanation, add the following parameter.

No.	Parameter	Explanation	Note
16	$RV	Receives the returned value of the job	<Data 2>

Example

(1) GetSystemVar B000 $B002
The result of the SRCH instruction is stored in B000.

(2) GetSystemVar PX000 $PX000
For the job R1, the pulse type current value is stored in P000.

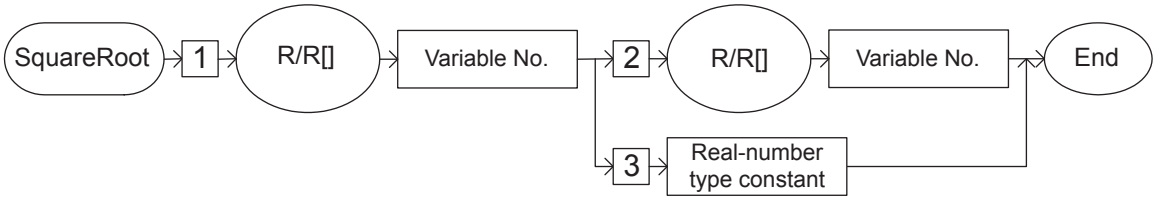
SquareRoot

Function

Calculates the square root of Data 2, and stores the result in Data 1.

Syntax

SquareRoot <Data 1> <Data 2>



Explanation

1. R Variable number/R [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	R Variable number/ R [Array number]	Specify the number of the real-number type variable to store the result.	<Data 1>

2. R Variable number/R [Array number]/Real-number type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	R Variable number/ R [Array number]	Specify the number of the real-number type variable whose square root is calculated.	<Data 2>
3	Real-number type constant	Specify the real-number type data whose square root is calculated.	<Data 2>

Example

(1) SquareRoot R000 2

1.414214E + 00 is stored in R000.

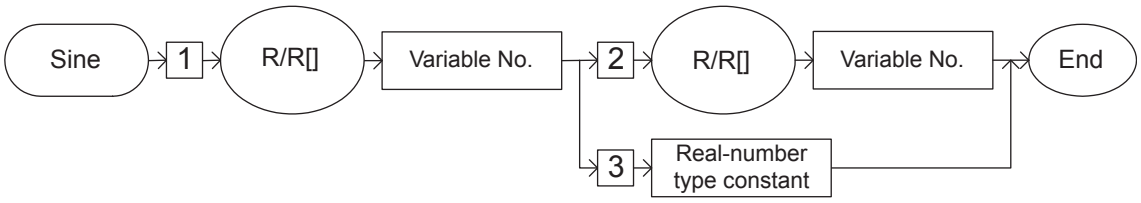
Sine

Function

Calculates the sine of Data 2(Units: deg), and stores the result in Data 1.

Syntax

Sine <Data 1> <Data 2>



Explanation

1. R Variable number/R [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	R Variable number/ R [Array number]	Specify the number of the real-number type variable to store the result.	<Data 1>

2. R Variable number/R [Array number]/Real-number type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	R Variable number/ R [Array number]	Specify the number of the real-number type variable whose sine is calculated.	<Data 2>
3	Real-number type constant	Specify the real-number type data whose sine is calculated.	<Data 2>

Example

(1) Sine R000 60

8.660254E - 01 is stored in R000.

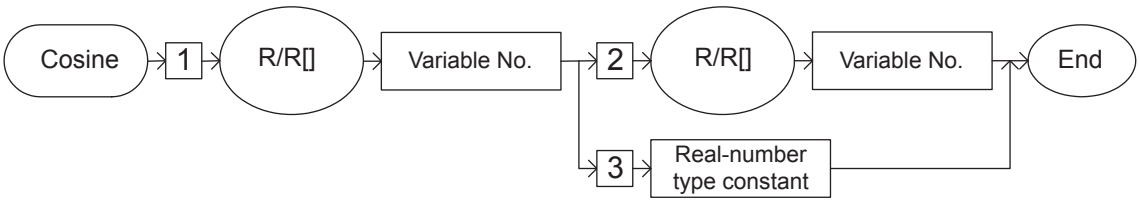
Cosine

Function

Calculates the cosine of Data 2(Units: deg), and stores the result in Data 1.

Syntax

Cosine <Data 1> <Data 2>



Explanation

1. R Variable number/R [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	R Variable number/ R [Array number]	Specify the number of the real-number type variable to store the result.	<Data 1>

2. R Variable number/R [Array number]/Real-number type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	R Variable number/ R [Array number]	Specify the number of the real-number type variable whose cosine is calculated.	<Data 2>
3	Real-number type constant	Specify the real-number type data whose cosine is calculated.	<Data 2>

Example

(1) Cosine R000 60

5.000000E - 01 is stored in R000.

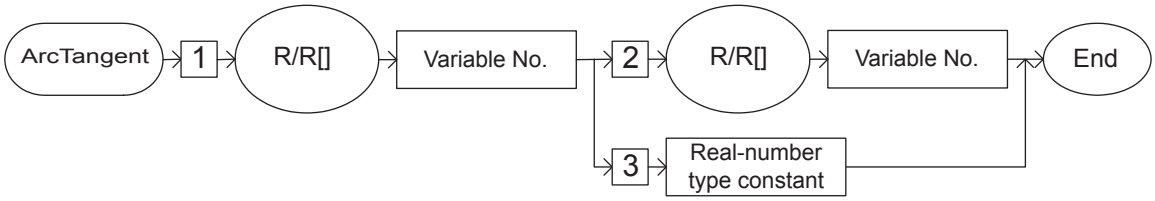
ArcTangent

Function

Calculates the arc tangent of Data 2(Units: deg), and stores the result in Data 1.

Syntax

ArcTangent <Data 1> <Data 2>



Explanation

1. R Variable number/R [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	R Variable number/ R [Array number]	Specify the number of the real-number type variable to store the result.	<Data 1>

2. R Variable number/R [Array number]/Real-number type constant

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
2	R Variable number/ R [Array number]	Specify the number of the real-number type variable whose arc tangent is calculated.	<Data 2>
3	Real-number type constant	Specify the real-number type data whose arc tangent is calculated.	<Data 2>

Example

(1) ArcTangent R000 60

8.904516E + 01 is stored in R000.

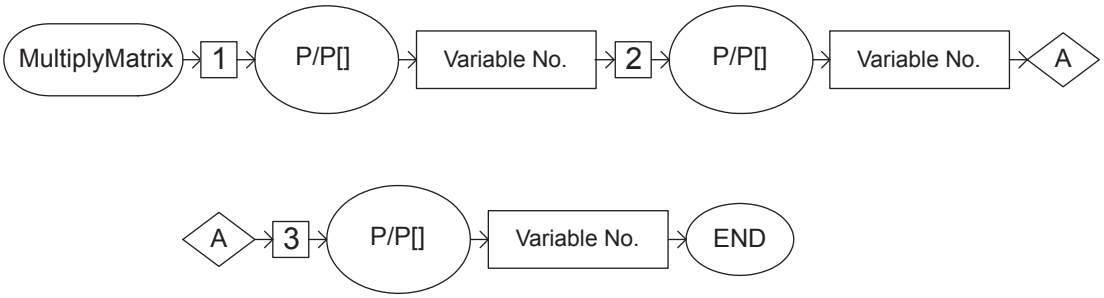
MultiplyMatrix

Function

Calculates the matrix multiplication of Data 2 and Data 3, and stores the result in Data 1.

Syntax

MultiplyMatrix <Data 1> <Data 2> <Data 3>



Explanation

1. P Variable number/P [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	P Variable number/ P [Array number]	Specify the number of the position type variable to store the result.	<Data 1>

2. P Variable number/P [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
2	P Variable number/ P [Array number]	Specify the number of the position type variable for which the matrix multiplication is calculated.	<Data 2>

3. P Variable number/P [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
3	P Variable number/ P [Array number]	Specify the number of the position type variable for which the matrix multiplication is calculated.	<Data 3>

Example

MultiplyMatrix and InvertMatrix instructions

The amount of shift for a three-dimensional shift can be obtained by using the MultiplyMatrix and InvertMatrix instructions.

The instructions to obtain the shift amount are used for the optional three-dimensional shift function. However, the amount of the three-dimensional shift can be also obtained by using the standard instructions, MultiplyMatrix and InvertMatrix.

The target value for a three-dimensional shift can be calculated by the following equation.

$$Pnew=P3d \times Pold$$

Where Pnew: Target position after a three-dimensional shift

P3d: Three-dimensional shift amount

Pold: Taught position

The amount of a three-dimensional shift can be obtained as follows:

$$P3d=Pnew \times Pold^{-1}$$

(1)
```
LinearMove P010 Speed=500.0(mm/sec)
LinearMove P011 Speed=500.0(mm/sec)
GetPosition PX020 STEP#(1)
InvertMatrix P021 P011
MultiplyMatrix P023 P020 P021
```

Stores taught position 1 (XYZ type) value in P020.
Calculates the inverse matrix of the taught position 2.
The amount of the three-dimensional

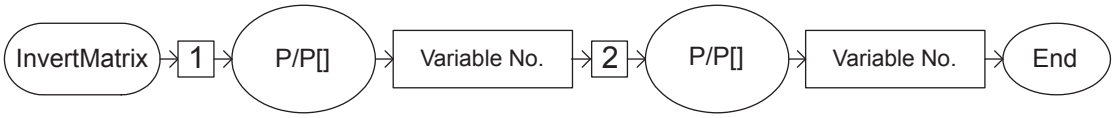
InvertMatrix

Function

Calculates the inverse matrix of Data 2, and stores the result in Data 1.

Syntax

InvertMatrix <Data 1> <Data 2>



Explanation

1. P Variable number/P [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
1	P Variable number/ P [Array number]	Specify the number of the position type variable to store the result.	<Data 1>

2. P Variable number/P [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
2	P Variable number/ P [Array number]	Specify the number of the position type variable for which the inverse matrix is calculated.	<Data 2>

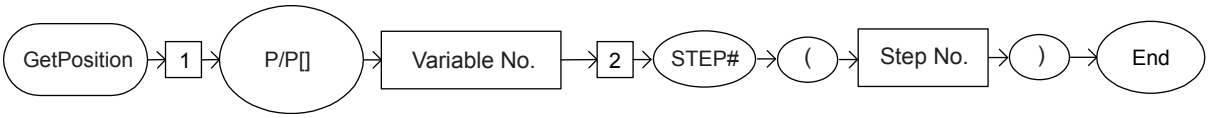
GetPosition

Function

Stores the position data of the data 2 (step number) in Data 1.

Syntax

GetPosition <Data 1> <Data 2>



Explanation

1. P Variable number/P [Array number]

PX Variable is same as P Variable, when only one robot is in the group.

The following parameter must be added.

No.	Parameter	Explanation	Note
1	PX Variable number/ PX [Array number]	Specify the expanded position type variable to store the teaching position data.	<Data 1>

2. STEP# (Step number)

The following parameter must be added.

No.	Parameter	Explanation	Note
2	Step# (Step number)	Specify the step number of the teaching position data to store.	<Data 2> No.: 1 to 999 Variable B/I/D can be used.

Example

(1) GetPosition PX 000 Step# (1)

Stores the teaching position data of the first step in P000.

MakeFrame

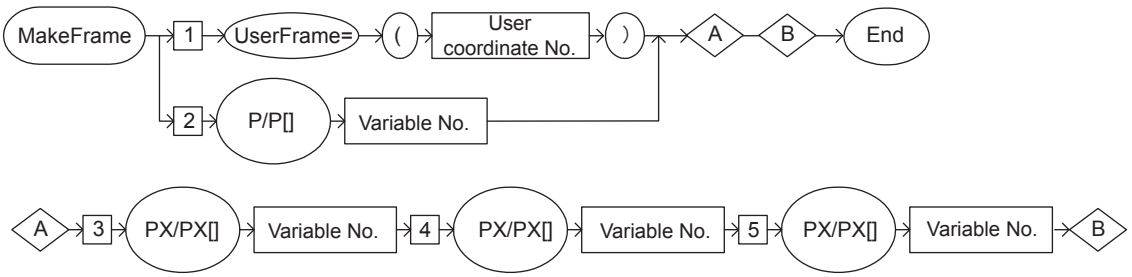
Function

By using this instruction, the user coordinates are generated from the position data from three position data: Data 1, Data 2 and Data 3.

- Data 1 shows the position data of the defined point ORIGIN
- Data 2 shows the position data of the defined point XX
- Data 3 shows the position data of the defined point XY

Syntax

MakeFrame the user coordinates specification <DATA 1> <DATA 2> <DATA 3>



Explanation

1. UserFrame= (User coordinate number)/P Variable number/P [Array number]

One of the parameters must be selected from the following table.

No.	Parameter	Explanation	Note
1	UserFrame= (User coordinate number)	Allocates the number for the user coordinate to be created.	No.: 1 to 63 Variable B/I/D can be used.
2	P Variable number/ P [Array number]	Specify the number of the position type variable where the coordinate values of the user coordinate to be created is stored.	

2. PX Variable number/PX [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
3	PX Variable number/ PX [Array number]/	Specify the number of the expanded position type variable where the position data of the user coordinates defined point ORG has been stored.	<Data 1>

3. PX Variable number/PX [Array number]

The following parameter must be added.

No.	Parameter	Explanation	Note
4	PX Variable number/ PX [Array number]/	Specify the number of the expanded position type variable where the position data of the user coordinates defined point XX has been stored.	<Data 2>

4. PX Variable number/PX [Array number]

PX Variable is same as P Variable, when only one robot is in the group.

The following parameter must be added.

No	Parameter	Explanation	Note
5	PX Variable number/ PX [Array number]	Specify the number of the expanded position type variable where the position data of the user coordinates defined point XY has been stored.	<Data 3>

Example

(1) MakeFrame UserFrame=1 PX000 PX001 PX002

The user coordinate number 1 is created according to three types of position data; P000, P001, and P002 of the user coordinate system.

StoreStrPos

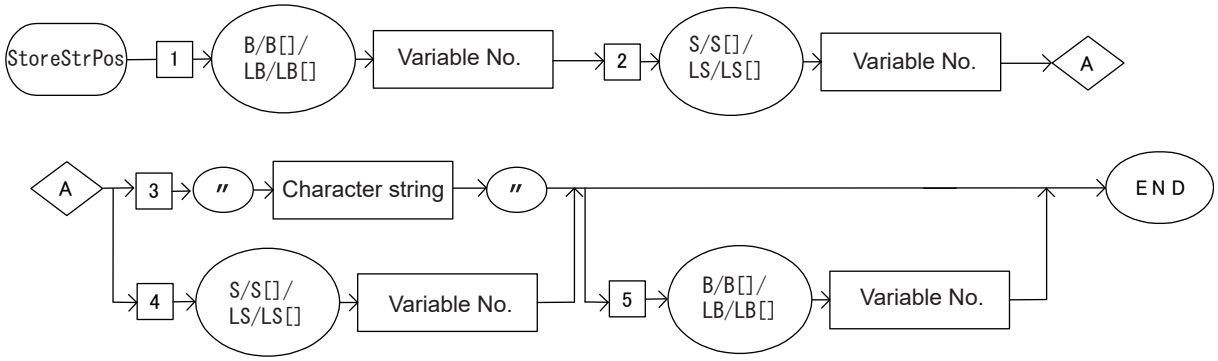
SUBSET	STANDARD	EXPANDED
Not available	Available	Available

Function

Usage: StoreStrPos {data1} {data2} {data3}. The index of the first occurrence of the substring in {data3} that is found in {data2} will be stored into {data1}.

Construction

StoreStrPos <Data 1> <Data 2> <Data 3>



Explanation

1. B Variable number/LB Variable number/B [Array number]/LB [Array number]

Add the following parameter.

No.	Parameter	Explanation	Note
1	B Variable number/ LB Variable number/ B [Array number]/ LB [Array number]	Specifies the number of byte type variable to store the appearance position.	<Data 1>

2. S Variable number/LS Variable number/S [Array number]/LS [Array number]

Add the following parameter.

No.	Parameter	Explanation	Note
2	S Variable number/ LS Variable number/ S [Array number]/ LS [Array number]	Specifies the number of the character type variable where the character string to search the character string.	<Data 2>

3. Character string/S Variable number/LS Variable number/S [Array number]/LS [Array number]

Choose one of the parameters from the following table.

No.	Parameter	Explanation	Note
3	Character string	Specifies the character strings to search.	<Data 3>
4	S Variable number/ LS Variable number/ S [Array number]/ LS [Array number]	Specifies the number of the character type variable stored the character string to search.	<Data 3>

4. B Variable number/LB Variable number/B [Array number]/LB [Array number]

The following parameter can be added or omitted.

No.	Parameter	Explanation	Note
5	B Variable number/ LB Variable number/ B [Array number]/ LB [Array number]	Specifies the number of byte type variable to store the search start position data of the search character string <data2> .	

Example

(1) StoreStrPos B000 S000 “DEF”

Stores “4” into B000 when “ABCDEFGHI” is stored in S000.

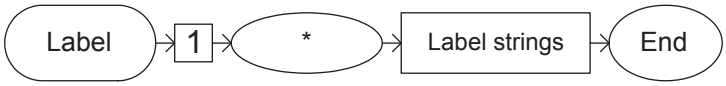
2.5 Control Instruction

Label *(Label)

Function

Specify the label for the jump.

Syntax



Explanation

1. *Label strings

The following parameter must be added.

No	Parameter	Explanation	Note
1	*Label strings	Specify the label strings.	String: 8 characters

Example

(1) Label *1
LinearMove Speed=100.0(mm/sec)
Jump *1
End Job

NOTE The label is effective only in the same job. It does not jump to the same label in other jobs.

Jump

Function

Jumps to specified label or job.

Syntax

Explanation

1. *Label character string

The following parameter must be added.

No	Parameter	Explanation	Note
1	*Label strings	Specify the label string.	String: 8 characters

Example

(1) Jump *1
Jumps to *1.

IfThen

Function

Determines true or false of the added conditional expression, and then execute the next instruction.

This instruction is a structured language.

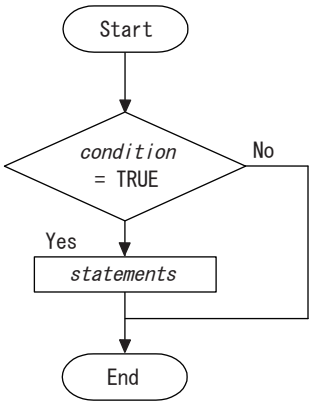
Syntax

If (*condition*)Then

[*statements*]

EndIf

Specified item	Contents
condition	The conditional expression to determine true or false. Able to specify up to six conditional expressions by using the “And”, “Or” and brackets “()”.
statements	The series of processing to be executed when the argument *condition* is true.



ElseIf

SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Executes the new conditional expression if the conditional expression of the IFThen instruction or the ElseIf instruction is false.

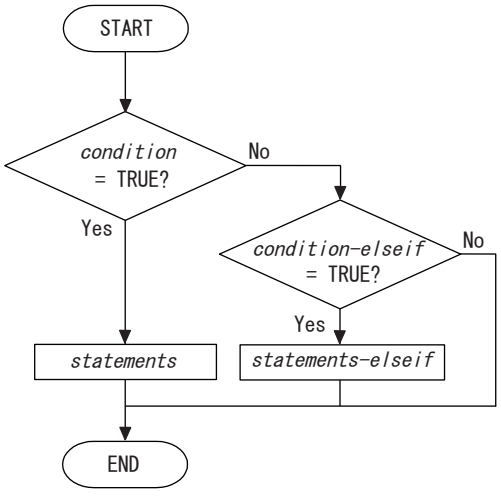
This instruction can be registered only between IfThen and EndIf. The multiple ElseIf instructions can be registered within a range.

This instruction is a structured language.

Construction

```
If (condition)Then
[statements]
ElseIf(condition-elseif)Then
[statements-elseif]
EndIf
```

Specified Item	Contents
condition *condition-elseif*	The conditional expression to determine true or false. Able to specify up to six conditional expressions by using the “AND”, “OR” and brackets “()”.
statements	The series of processing to be executed when the argument condition is true.
statements-elseif	The series of processing to be executed when the argument condition-elseif is true.



Else

SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

This instruction determines a location to execute the instruction when the all instructions do not corresponded to the any branches of the IfThen.

This instruction can be registered only between IfThen and EndIf.

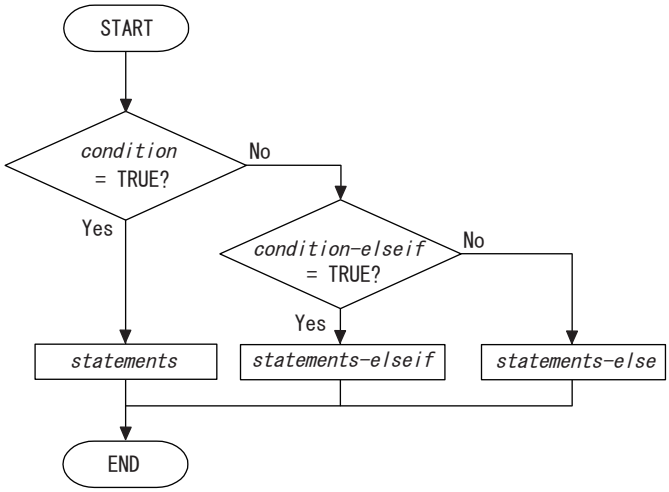
Able to register only one instruction within a range.

This instruction is a structured language.

Construction

```
If (condition)Then
[statements]
ElseIf(condition-elseif)Then
[statements-elseif]
Else
[statements-else]
EndIf
```

Specified item	Contents
condition *condition-elseif*	The conditional expression to determine true or false. Able to specify up to six conditional expressions by using the “AND”, “OR” and brackets “()”.
statements	The series of processing to be executed when the argument *condition* is true.
statements-elseif	The series of processing to be executed when the argument *condition-elseif* is true.
statements-else	The series of processing to be executed when all of the defined conditional expressions before Else instruction are false.


START
condition
= TRUE?
No
Yes
condition-elseif
= TRUE?
No
Yes
statements
statements-elseif
statements-else
END

While

SUBSET	STANDARD	EXPANDED
Available	Available	Available

Function

Determines true or false of the added conditional expression, and execute the series of instructions within a range when it is true.

Repeats the instruction as long as the conditional expression is true.

This instruction is a structured language.

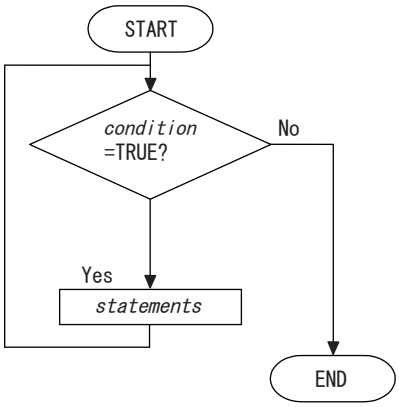
Syntax

While (*condition*)

[*statements*]

EndWhile

Specified item	Contents
condition	The conditional expression to determine true or false. Able to specify up to six conditional expressions by using the “AND”, “OR” and brackets “()”.
statements	The series of processing to be executed when *variable* value is equal to *expresstionlist.*



For

Function

Executes the repetition of the series of the instructions for specified number of times within a range.

This instruction is a structured language.

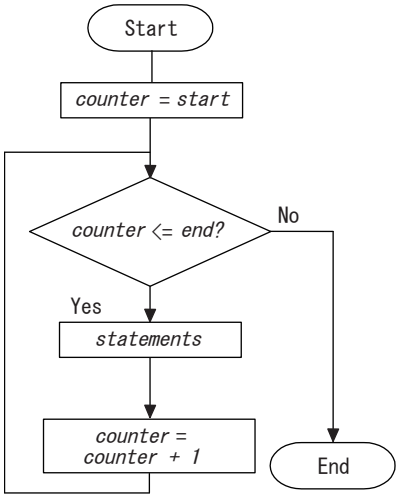
Syntax

For counter =start to end

[statements]

Next counter

Specified item	Contents
counter	Specify the variable to use the loop counter. Able to specify the I variable
start	Specify the initial value of the argument counter.
end	Specify the end value of the argument counter.
statements	The series of processing to be executed within the loop.



Example

(2) Set B000 1

For I000 = 1 to 10

Increment B000

Next I000

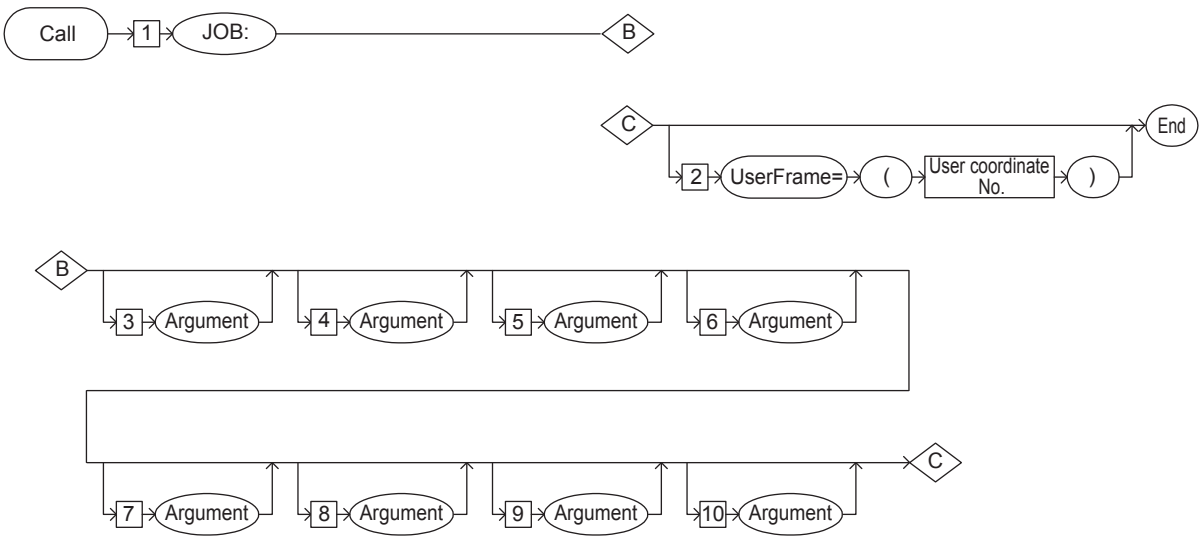
The loop repeat for 10 times, and B000 becomes 11.

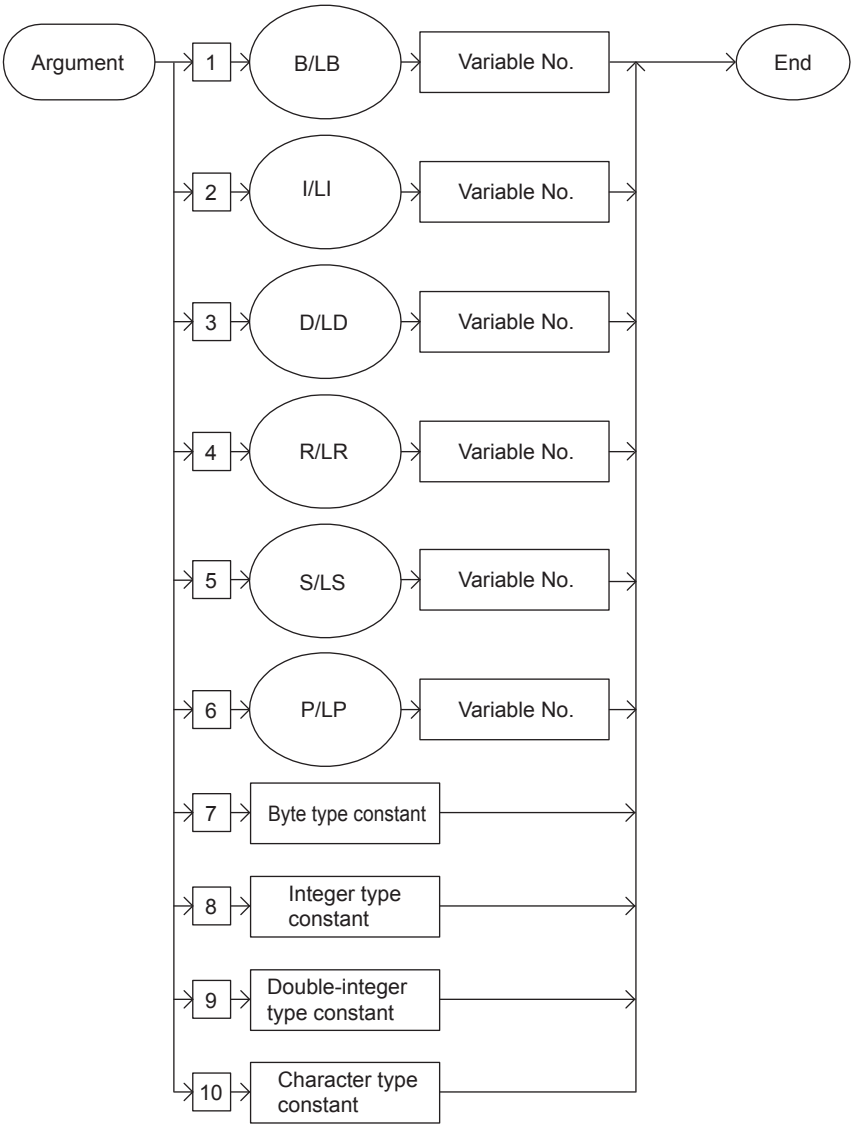
Call

Function

Calls the specified job.

Syntax




Argument
1
B/LB
Variable No.
End
2
I/LI
Variable No.
3
D/LD
Variable No.
4
R/LR
Variable No.
5
S/LS
Variable No.
6
P/LP
Variable No.
7
Byte type constant
8
Integer type constant
9
Double-integer type constant
10
Character type constant

Explanation

1. JOB:

The following tag must be added.

No	Tag	Explanation	Note
1	JOB:	Specify a job to be called.	

2. UserFrame= (User coordinate number)

The following tag can be added or omitted.

No	Tag	Explanation	Note
2	UserFrame= (User coordinate number)	Specify the user coordinate system of the job to be called.	Available in the relative job function.

3. Argument

The following tag can be added or omitted.

No	Tag	Explanation	Note
3 to 10	Argument (all available types of variables and constants)	Eight arguments (from 3 to 10) can be passed to the job to be called. All types of variables and constants can be used as arguments.	

Example

(1) Call JOB:TEST1
The job named TEST1 is called.

Outline (Call Instruction with Argument)

The Call can pass 8 arguments to the called job. This enables the transfer of information without using the global variable to pass information. Thus, it makes the management of information easier, the number of lines of the calling job fewer, and the process simpler.

(1) Call Instruction with Argument

For example, when creating the job which only waits for incoming in order to wait for the incoming of I/Os, the input number's incoming to be waited for had to be set to the global variable, and the information had to be passed from the calling job to the called job. For example, the case when the input of specified three points are Input#(1), Input#(5), and Input#(7) is as follows:

<Calling Job>

Start Job

Set B000 1

Set B001 5

Set B002 7

Call JOB: WAIT_INPUT

:

<Called Job>

Job name: WAIT_INPUT

Start Job

Wait Input#(B000) ON

Wait Input#(B001) ON

Wait Input#(B002) ON

End Job

This makes the calling job program long, and the global variable will be used only for this information transfer.

By using the Call instruction with argument, the above can be described as follows:

<Calling Job>

Start Job

Call JOB: WAIT_INPUT (1, 5, 7)

:

<Called Job>

Job name: WAIT_INPUT

Start Job

GetArgument LB000, InputArgument#(1)// INPUT NUMBER 1 (receiving the argument 1)

GetArgument LB001, InputArgument#(2)// INPUT NUMBER 2 (receiving the argument 2)

GetArgument LB002, InputArgument#(3)// INPUT NUMBER 3 (receiving the argument 3)

Wait Input#(LB000) ON

Wait Input#(LB001) ON

Wait Input#(LB002) ON

Return

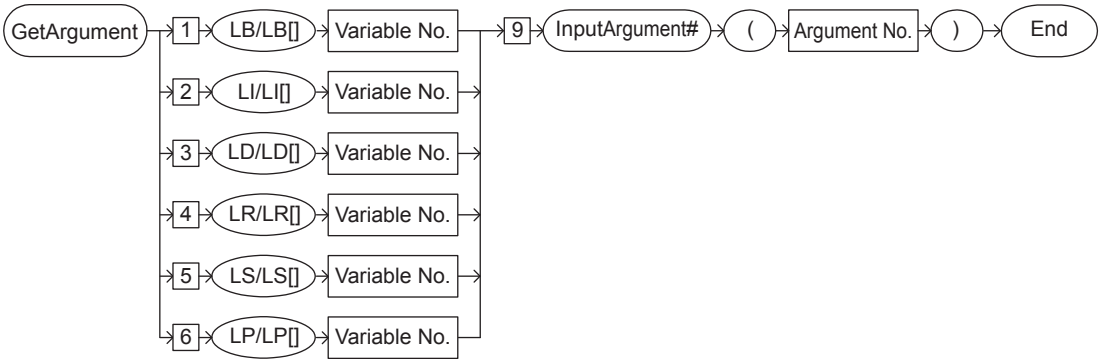
This makes the calling job program only one-line long. Also, the called job can be made by using only the local variable.

GetArgument

Function

This is the instruction to receive arguments for the Call instruction and macro instruction. When the instruction is executed, the argument data added to the Call instruction or macro instruction is retrieved, then stored in the local variable specified to be used in the Call job or macro job.

Syntax



Explanation

1. LB Variable number/LB [Array number]/ LI Variable number/LI [Array number]/ LD Variable number/LD [Array number]/ LR Variable number/LR [Array number]/ LS Variable number/LS [Array number]/LP Variable number/LP [Array number]/

One of the tags must be selected from the following table.

No	Tag	Explanation	Note
1	LB Variable number LB [Array number]	Specify the local byte type variable which stores the argument.	
2	LI Variable number LI [Array number]	Specify the local integer type variable which stores the argument.	
3	LD Variable number LD [Array number]	Specify the local double integer type variable which stores the argument.	
4	LR Variable number LR [Array number]	Specify the local real type variable which stores the argument.	
5	LS Variable number LS [Array number]	Specify the local character type variable which stores the argument.	
6	LP Variable number LP [Array number]	Specify the local robot axis position type variable which stores the argument.	

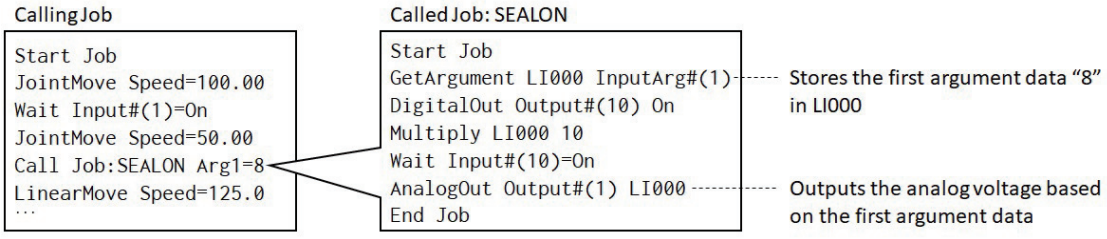
2. InputArgument# (Argument No.)

The following tag must be added.

No.	Tag	Explanation	Note
7	InputArgument# (Argument No.)	Specify the argument number to be stored in the local variable.	Variable B/I/D/LB/LI/LD can be used.

Example

(1) An example is shown below.

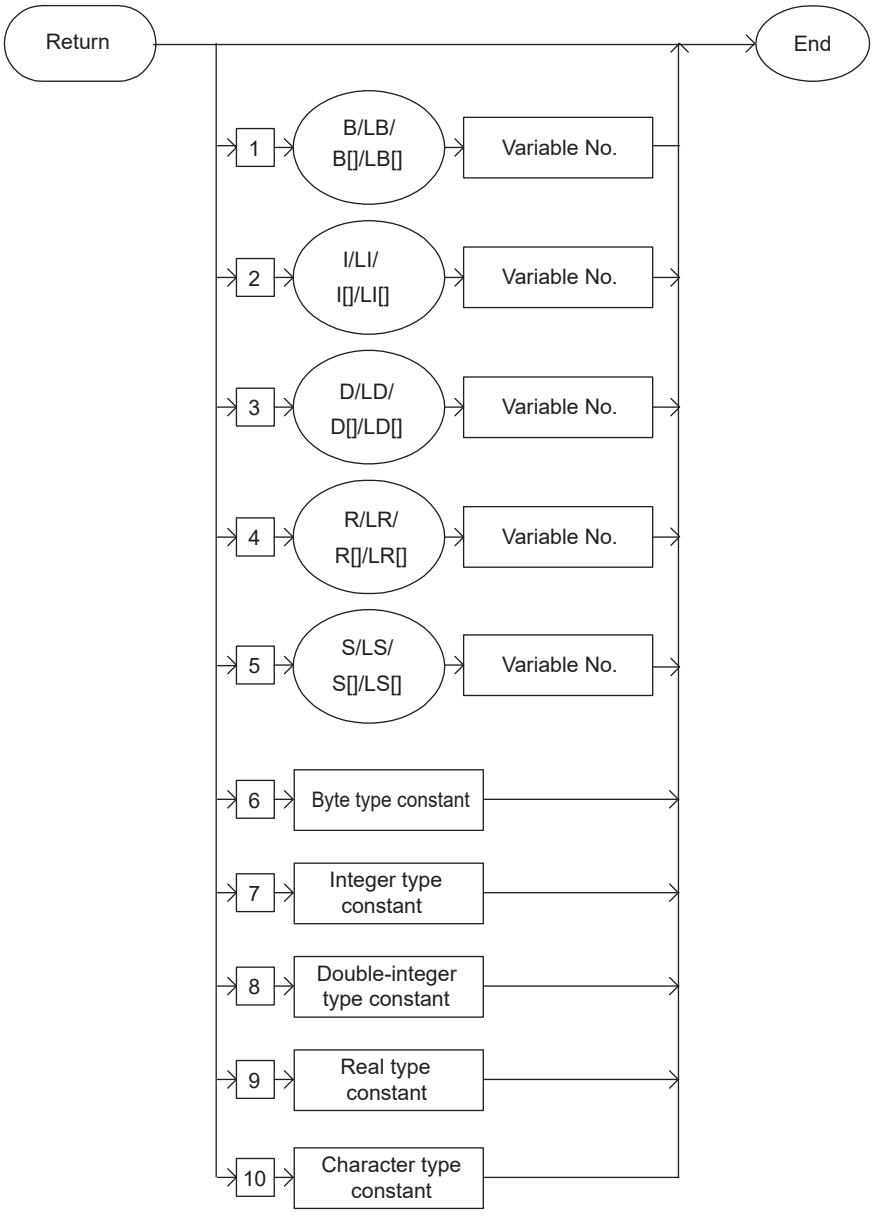


Return

Function

Returns from the called job to the calling job.

Syntax



1. Return Value

The following tag can be added or omitted.

No	Tag	Explanation	Note
1 to 10	Variables other than position type variables and constants	Return the return value to the call-origin job.	

Arguments can be added to the Call instruction, and the return value can be returned with the Return instruction. Variables other than the position type variables (P variable, and PX variable) and constants can be returned as the return value.

At the call-origin job, the return value is received by the GetSystemVar instruction. For details, refer to “5. Argument” in the explanation of the Call instruction.

Example

The following example is the case when the value returned from Job: SAMPLE is received by I000.

Call Job: SAMPLE

GetSystemVar: I000 $RV

In this case, at Job: SAMPLE, the argument must be added to Return as follows:

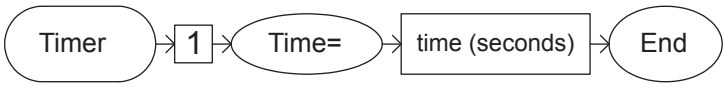
```
Start Job
        :
Return   LI000
End Job
```

Timer

Function

Stops for the specified time.

Syntax



Explanation

1. Time=time

The following parameter must be added.

No	Parameter	Explanation	Note
1	Time=time	Specify the stopping time.	For the YRC1000 Timer : 0.01 to 655.35 seconds Variable I/I[] can be used. Units : 0.01 seconds For the YRC1000micro Timer : 0.001 to 65.535 seconds Variable I/I[] can be used. Units : 0.001 seconds

Example

(1) Timer Time=12.50(seconds)
Stops for 12.5 seconds.

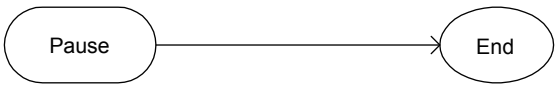
(2) Set I002 5
Timer Time=I002(seconds)
For the YRC1000 : Stops for 0.05 seconds.
For the YRC1000micro : Stops for 0.005 seconds.

Pause

Function

Stops the job temporarily.

Syntax



Example

(1) Pause
Stops the job temporarily.

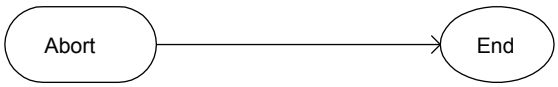
Abort

Function

Suspends playback.

After the playback is suspended by Abort instruction, it will not restart until the job is reselected on the job list.

Syntax



Example

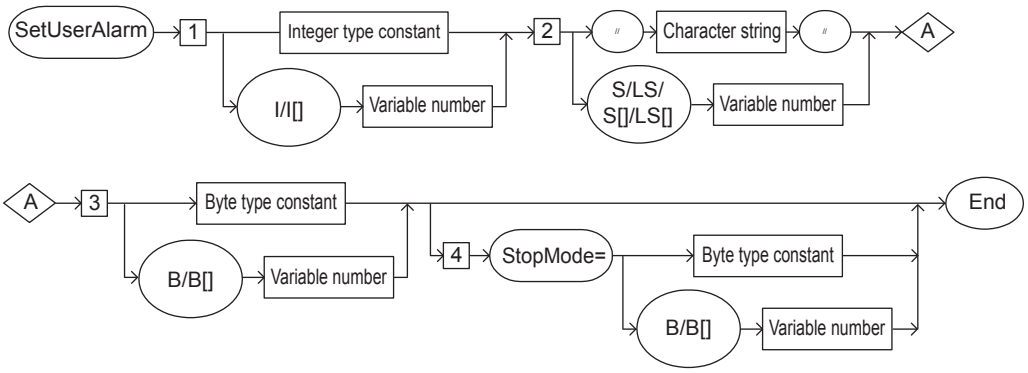
(1) Abort
Stops the playback of job.

SetUserAlarm

Function

Generates an alarm with any given number, name and sub code.

Syntax



Explanation

1. Alarm Code

The following parameter must be added.

No	Parameter	Explanation	Note
1	I Variable number/ I [Array number]/ [Integer type constant]	Specify the integer type variable number.	Number: 8000 to 8999

2. Alarm Name

The following parameter must be added.

No	Parameter	Explanation	Note
2	S Variable number/ S [Array number]/ [Character string]	Specify the alarm name.	32 characters

3. Subcode

The following parameter must be added.

No	Parameter	Explanation	Note
3	B Variable number/ B [Array number]/ [Byte type constant]	Specify the byte type variable number.	Number: 0 to 255

4. StopMode

The following parameter can be added or omitted.

No	Parameter	Explanation	Note
4	B Variable number/ B [Array number]/ [Byte type constant]	Specify StopMode (operation stop mode).	Number: 0: All Tasks Job Stop 1: No Job Stop 2: Each Task Job Stop

Example

(1) SetUserAlarm 8000 S000 0 StopMode=1
The alarm with the alarm number 8000, alarm name is stored in S000, and subcode 0 occurs.
If StopMode is 1, the job does not stop.

By this instruction, the execution of job is stopped according to StopMode after the alarm occurs.

It does not guarantee that the execution of job stops immediately after the instruction is executed.

Switch

Function

This instruction refers to the specified variable, and determines the next instruction to perform.

Combining the Case instruction or the Default instruction enables to branch the more than three directions.

This instruction is a structured language.

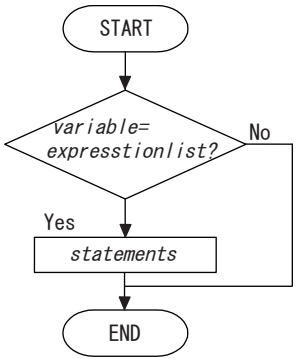
Syntax

Switch *variable* Case *expressionlist*

[*statements*]

EndSwitch

Specified item	Contents
variable	Indicates the variable to use the conditional branch. I variable or LI variable can be specified.
expresstionlist	The branch destination of *variable.* Must be specified.
statements	The series of processing to be executed when *variable* value is equal to *expresstionlist.*



Case

Function

This instruction adds a branch of the next instruction for the Switch instruction.

This instruction can be registered between Switch and EndSwitch. The multiple Case instructions can be registered within range.

This instruction is a structured language.

Syntax

Switch *variable* Case *expressionlist*

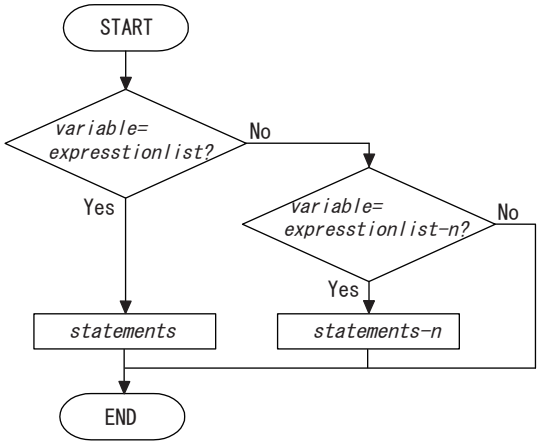
[*statements*]

Case *expressionlist-n*

[*statements-n*]

EndSwitch

Specified item	Contents
variable	Indicates the variable to use the conditional branch. I variable or LI variable can be specified.
expresstionlist	This is the first branch destination of *variable.* Must be specified.
expresstionlist-n	This is a branch destination of *variable after second branch.*
statements	The series of processing to be executed when *variable* value is equal to *expresstionlist.*
statements-n	The series of processing to be executed when *variable* value is equal to *expresstionlist-n.*



Default

Function

This instruction determines a location to execute the instruction when the instruction does not corresponded to the any branches of the Switch.

This instruction can be only registered between Switch and EndSwitch.

Able to register only one instruction within a range.

This instruction is a structured language.

Syntax

Switch *variable* Case *expresstionlist*

[*statements*]

Case *expresstionlist-n*

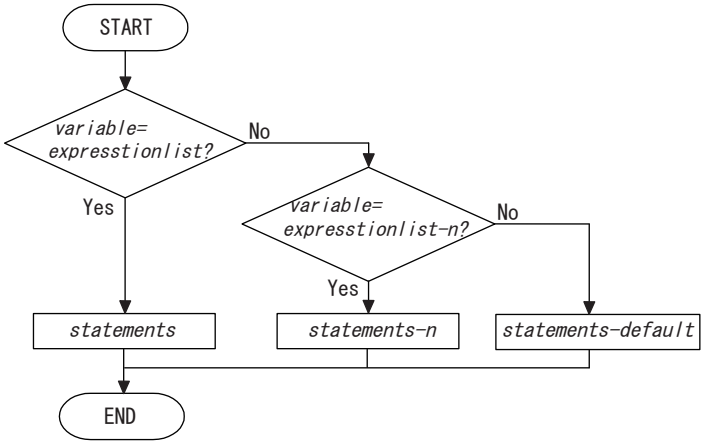
[*statements-n*]

Default

[*statements-default*]

EndSwitch

Specified item	Contents
variable	Indicates the variable to use the conditional branch. I variable or LI variable can be specified.
expresstionlist	This is the first branch destination of *variable.* Must be specified.
expresstionlist-n	This is a branch destination of *variable after second branch.*
statements	The series of processing to be executed when *variable* value is equal to *expresstionlist.*
statements-n	The series of processing to be executed when *variable* value is equal to *expresstionlist-n.*
statements-default	The series of processing to be executed when *variable* value and *expresstionlist* or *expressionlist-n* does not correspond.

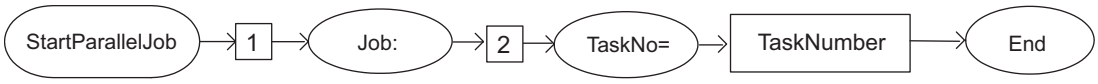

START
variable=
expresstionlist?
No
Yes
variable=
expresstionlist-n?
No
Yes
statements
statements-n
statements-default
END

StartParallelJob

Function

Start a job in a Parallel Task. This job runs in the background while the calling job continues execution.

Syntax



Explanation

1. Job:

Add the following tag.

No	Parameter	Explanation	Note
1	Job:	Specify the job to be started.	

2. Task Number

Add the following tag.

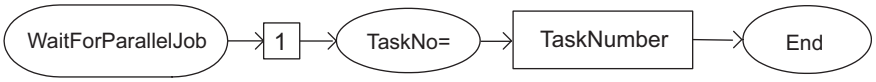
No	Parameter	Explanation	Note
2	TaskNo=Task Number	Specify the Task Number to use for job.	Constant value:1 to max number of tasks defined for Controller.

WaitForParallelJob

Function

Wait for a Parallel Task (i.e. a job started by StartParallelJob/PSTART) to complete

Syntax



Explanation

1. Task Number:

Add the following tag.

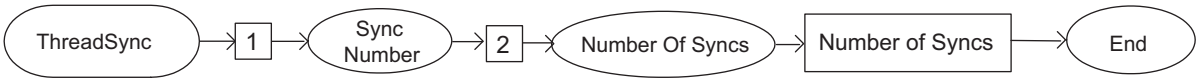
No	Parameter	Explanation	Note
1	TaskNo=Task Number	Specify the Task Number to use for job.	Constant value:1 to max number of tasks defined for Controller.

ThreadSync

Function

Synchronize execution between multiple parallel jobs/tasks.

Syntax



Explanation

1. Sync Number:

Add the following tag.

No	Parameter	Explanation	Note
1	Sync Number	A user defined synchronization number.	Constant value:1 to 32

2. Number Of Syncs:

Add the following tag.

No	Parameter	Explanation	Note
2	NumberOfSyncs= Number of Syncs	The number of synchronization numbers to wait for.	Constant value: 2 to max number of tasks defined for Controller plus one

2.6 Instruction Which Adheres to an Instruction

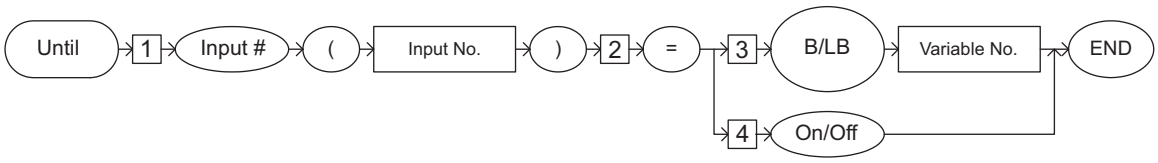
Until

SUBSET	STANDARD	EXPANDED
Not available	Available	Available

Function

During the execution of move instruction to which Until instruction is added, the specified condition is evaluated. If the condition is satisfied, the ongoing move instruction is forcibly completed and the next instruction is executed. This instruction is added after other instructions for processing.

Construction



Explanation

1. Input# (Input number)

Add the following tag.

No.	Tag	Explanation	Note
1	Input# (Input number)	Specifies the number of the general-purpose input signal which becomes a input condition.	No.: 1 to 4096 Variable B/I/D/LB/LI/LD can be used.

2. =

Add the following tag.

No.	Tag	Explanation	Note
2	=	It is equal.	

3. B Variable number/LB Variable number/on/off.

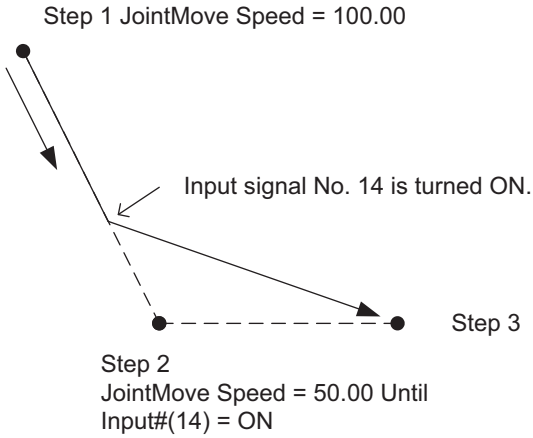
Choose one of the tags from the following table after "=" is selected from the table in part 2 of this Explanation.

No.	Tag	Explanation	Note
3	B Variable number/ LB Variable number	Specifies the byte type variable to be the condition of execution.	Least significant bit: 0: OFF 1: ON
4	on/off	Specifies the condition as ON or OFF.	

Example

(1) Step 1 JointMove Speed= 100.00
Step 2 JointMove Speed= 50.00 Until Input#(14)=On
Step 3 JointMove Speed=25.00

The axis moves toward Step 2 until input signal No. 14 is turned ON. When input signal No. 14 is turned ON, the axis moves toward Step 3.



YRC1000/YRC1000micro

SUPPLEMENTAL INSTRUCTIONS FOR Smart Pendant

(JZRCR-APP30-1)
(INSTRUCTIONS FOR INFORM LANGUAGE)

©2018 YASKAWA ELECTRIC CORPORATION
Published by YASKAWA

YASKAWA

YASKAWA ELECTRIC CORPORATION

August 2021 -08-30

MANUAL NO.
HW1485511 ⑦