COSIMIR® · MELFA-BASIC IV
MELFA-BASIC IV

Overview

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
Introduction on MELFA-BASIC-IV

An explanation of the individual commands for the MELFA-BASIC IV language is provided in the following chapters:

MELFA-BASIC IV command method

With the ample operation functions, a work program more advanced than the Movemaster command method can be created.

Format description method

Parts that have uppercase letters should be written as they are.

The contents of the arrow brackets (<>) should be described by the user. Follow the usage explanations for the individual commands when making the entry.

The square brackets indicate that it may be omitted, if so needed.

The vertical bar (|) represents OR. For example, if <Line no. | label> is written, either the line no. or the label will be written.

Enter all commas (,), periods (.) , semicolons (;), and parentheses (), as they are. Here the various commands used to create a program with the MELFA-BASIC III command method or Movemaster command method will be explained. The features of each command method are as shown in the following chapters.

Commands that are marked with an * cannot be executed alone. They have to be used in connection with other commands.

Commands that are marked with an * * can be used in connection with other commands or stand alone.
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
MELFA-BASIC-IV program control commands

These commands are concerned with conditional branching, repetitive operation, interrupting of signals, starting and stopping, counter operation, etc.

Command	Function
ACT	(Act)
CALLP	(Call P)
COM OFF	(Communication OFF)
COM ON	(Communication ON)
COM STOP	(Communication Stop)
DEF ACT	(Define act)
END	(End)
FOR NEXT	(For Next)
FPRM	(FPRM)
GOSUB	(Go Subroutine)
GOTO	(Go To)
HLT	(Halt)
IF THEN ELSE	(If Then Else)
ON COM GOSUB	(On Communication Go Subroutine)

ON GOSUB	(On Gosub)
ON GOTO	(On Go To)
PRIORITY	(Priority)
RESET ERR	(Reset Error)
RETURN	(Return)
SKIP	(Skip)
WAIT	(Wait)
WHILE WEND	(While End)
XCLR	(X Clear)
XLOAD	(X Load)
XRST	(X Reset)
XRUN	(X Run)
XSTP	(X Stop)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MELFA-BASIC-IV position and motion control commands

These commands are concerned with the definition of positions and coordinates as well as the assignment of interpolation, speed, timer, tool, palette, etc.

Command	Function
ACCEL	(Accelerate)
BASE	(Base)
CMP JNT	(Composition Joint)
CMP POS	(Composure Posture)
CMP TOOL	(Composition Tool)
CMP OFF	(Composition OFF)
CMPG	(Composure Gain)
CNT	(Control)
DEF JNT	(Define Joint)
DEF PLT	(Define pallet)
DEF POS	(Define Position)
DLY	(Delay)
FINE	(Fine)
JOVRD	(J override)
JRC	(Joint Roll Change)
MOV	(Move)
MVA	(Move Arch)
MVC	(Move C)
MVR	(Move R)
MVR2	(Move R2)
MVR3	(Move R3)
MVS	(Move S)
OADL	(Optimum Acceleration/Deceleration)
OVRD	(Override)
PLT	(Pallet)
SERVO	(Servo)
SPD	(Speed)
TOOL	(Tool)
TORQ	(Torque)
WTH	(With)
WTHIF	(With If)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MELFA-BASIC-IV hand control commands

These commands are concerned with opening/closing action of the hand.

Command	Function
HCLOSE	(Hand Close)
HOPEN	(Hand Open)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
MELFA-BASIC I/O-IV control commands

These commands are concerned with input/output control of external I/O. Both single and parallel bits can be handled enabling the logical operation in the internal register.

Command	Function
DEF IO	(Define IO)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
Other commands in MELFA-BASIC-IV

These commands are concerned with setting of parameters, selecting of programs, resetting of alarms, and describing of comments.

Command	Function
CLOSE	(Close)
CLR	(Clear)
DEF ARCH	(Define Arch)
DEF CHAR	(Define Character)
DEF DOUBLE	(Define Double)
DEF FLOAT	(Define Float)
DEF FN	(Define Function)
DEF INTE	(Define Integer)
DIM	(Dim)
INPUT	(Input)
OPEN	(Open)
PRINT	(Print)
REM	(Remarks)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
Alphabetical Overview of MELFA-BASIC-IV commands

Command	Function
ACCEL	(Accelerate)
ACT	(Act)
BASE	(Base)
CALLP	(Call P)
CHRSRCH	(Character search)
CLOSE	(Close)
CLR	(Clear)
CMP JNT	(Composition Joint)
CMP POS	(Composure Posture)
CMP TOOL	(Composition Tool)
CMP OFF	(Composition OFF)
CMPG	(Composure Gain)
CNT	(Control)
COM OFF	(Communication OFF)
COM ON	(Communication ON)
COM STOP	(Communication Stop)
DEF ACT	(Define Act)
DEF ARCH	(Define Arch)
DEF CHAR	(Define Character)
DEF DOUBLE	(Define Double)
DEF FLOAT	(Define Float)
DEF FN	(Define Function)
DEF INTE	(Define Integer)
DEF IO	(Define IO)
DEF JNT	(Define Joint)
DEF PLT	(Define Pallet)
DEF POS	(Define Position)
DIM	(Dim)
DLY	(Delay)
END	(End)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ACCEL (Accelerate)

Function:

Designate the robot's acceleration and deceleration speeds as a percentage (%).

Format

```
ACCEL [<Acceleration rate>],[ <Deceleration rate>]
```

Controller software version G2 or later:

```
```

```
ACCEL  [<Acceleration rate>] [, <Deceleration rate>]
       ,[<Acceleration rate when moving upward>], [<Deceleration rate when moving
upward>]
       ,[<Acceleration rate when moving downward>], [<Deceleration rate when moving
downward>]
```

Terminology

<Acceleration/Deceleration>	1 to 100. Designate the acceleration/deceleration to reach the maximum speed from speed 0 as a percentage. This can be described as a constant or variable. A default value of 100 is set if the argument is omitted. A value of 100 corresponds to the maximum rate of acceleration/deceleration. Unit: mn
<Acceleration/ Deceleration rate when moving upward>	Specify the acceleration/deceleration rate when moving upward in an arch motion due to the MVA instruction. A default value of 100 is set if the argument is omitted. It is possible to specify the argument either by a constant or variable.
<Acceleration/ Deceleration rate when moving downward>	Specify the acceleration/deceleration rate when moving downward in an arch motion due to the MVA instruction. A default value of 100 is set if the argument is omitted. It is possible to specify the argument either by a constant or variable.

Explanation

- The maximum acceleration/deceleration is determined according to the robot being used. Set the corresponding percentage. The system default value is 100,100.
- The acceleration percentage changed with this command is reset to the system default value when the program is reset or the END statement executed.
- The smooth operation when CNT is valid will have a different locus according to the acceleration speed or operation speed. To move smoothly at a constant speed, set the acceleration and deceleration to the same value. CNT is invalid in the default state.
- If the optimum acceleration/deceleration control is active (OADL ON), the setting of the ACCEL instruction is ignored.

Reference Program

```
10 ACCEL 50,100                 'Heavy load designation (when
                                acceleration / deceleration is 0.2
                                seconds, the acceleration will be 0.4,
                                and the deceleration will be 0.2 seconds)
20 MOV P1
30 ACCEL 100,100                'Standard load designation
40 MOV P2
50 DEF ARCH 1,10,10,25,25,1,0,0
60 ACCEL 100,100,20,20,20,20    'Specify the override value to 20 when
                                moving upward or downward due to the MVA
                                instruction
70 MVA P3,1
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ACT (Act)

Function:

Designates the enable/disable status of the interrupt.

Format

```
ACT <Priority No.> = <enable/disable>
```

Terminology

<Priority No.>	0: Either enables or disables the entire interrupt. 1 to 8: Designate the priority No. for the interrupt defined in the DEF ACT statement. When entering the priority No., always leave a space (character) after the ACT command. If described as ACT1, it will be a variable name declaration statement.
<enable/disable>	1: Allows interrupts, 0: Prohibits interrupts

Explanation

- When the program starts, the status of <Priority No.> 0 is "enabled." When <Priority No.> 0 is "disabled," even if <Priority No.> 1 to 8 are set to "enabled," no interrupt will be enabled.
- The statuses of <Priority No.> 1 to 8 are all "disabled" when the program starts.
- An interrupt will occur only when all of the following conditions have been satisfied:
  <Priority No.> 0 is set to "enabled."
  The status of the DEF ACT statement has been defined.
  When the <Priority No.> designated by DEF ACT is made valid by an ACT statement.
- The return from an interrupt process should be done by describing either RETURN 0 or RETURN 1.
- Even if the robot is in the middle of interpolation, an interrupt defined by a DEF ACT statement will be executed.
- During an interrupt process, that <Priority No.> will be executed with the status as "disable."
- A communications interrupt (COM) has a higher priority than an interrupt defined by a DEF ACT statement.
- The relationship of priority rankings is as shown below:
  COM > ACT > WTHIF (WTH) > Pulse substitution

Reference Program

```
10 DEF ACT 1,M_IN(1)=1 GOSUB *INTR   'Assign input signal 1 to the
                                     interrupt 1 condition
20 MOV P1
30 ACT 1=1                           'Enable interrupt 1
40 MOV P2
50 ACT 1=0                           'Disable interrupt 1
:
100 *INTR                            'If input signal 1 changes to ON
                                     (to 1) while the robot is moving
                                     from P1 to P2, it will stop.
110 HLT                              'Stop
120 RETURN 0
```

Reference

DEF ACT, RETURN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

BASE (Base)

Function:

Designates the base conversion data.

Format

BASE <Base conversion data>

Terminology

<Base conversion data>	Designate the base conversion data in terms of a position operation expression.

Explanation

- The X, Y, and Z components of the position data represent the amount of parallel movement from the origin of the world coordinate system to the origin of the base coordinate system. The base conversion data can be changed only with the BASE command. The components A, B, and C of the position data represent the amount that the base coordinate system is tilted in relation to the world coordinate system.
  X ... Distance to move parallel to X axis

Y ... Distance to move parallel to Y axis
Z ... Distance to move parallel to Z axis
A ... Angle to turn toward the X axis
B ... Angle to turn toward the Y axis
C ... Angle to turn toward the Z axis
L 1 .. Movement amount of additional axis 1
L 2 .. Movement amount of additional axis 2

- For A. B and C. the clockwise direction looking from the front of the origin of the coordinate, used as the center of rotation, is the forward rotation direction.
- The contents of the structural flag have no meaning.
- The base coordinate system changed by the BASE command will keep the changes even after the power has been turned OFF.
- The system's default value for this data is P_NBASE=(0,0,0,0,0,0) (0,0). This is calculated with the 6-axis three dimensional regardless of the mechanism structure.

Reference Program

```
10  BASE (50,100,0,0,0,0,0,0)       'Input the conversion data as a
                                     constant
20  MVS P1
30  BASE P2                         'Input the conversion data as a
                                     variable
40  MVS P1
50  BASE P_NBASE                    'Reset the conversion data to the
                                     default values.
```

Reference

TOOL, Robot status variables

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CALLP (Call P)

Function:

Executes the designated program. (Program call version of GOSUB)

Format

CALLP "<Program name>" [, <Argument> [,<Argument>] ...]

Terminology

<Program name>	Designate the program name with a character string constant or character string variable.
<Argument>	Designate the variable to be transferred to the program when the program is called up. Up to 16 variables can be transferred.

Explanation

- If the argument differs from the type of argument defined (FPRM) in the CALLP called program, an error will occur. If the number of arguments differs from the CALLP called program, an error will occur when execution is started. When the program is reset, the control will return to the head of the main program (program in which host program does not exist.)
- The definition statements (DEF ACT, DEF FN, DEF PLT, DIM) executed in the CALLP calling program are invalid in the CALLP called program. These will be validated again when the program returns from CALLP.
- The speed and tool data are all valid.
- Up to seven programs can be called out from one program. This number of programs includes when a program is called from the subprogram.
- A file already opened and executed cannot be called from another slot and used. It can be called repeatedly from the same slot.
- A program cannot call itself out.
- A host program cannot be called.
- The argument is received with the FPRM command by the receiving program.
- The results calculated with the CALLP called program cannot be substituted in the argument variable and used in the CALLP calling program. In this case, use external variables to transfer the value.

Reference Program

```
10 CALLP "P10" ,M1,P1,P2   '"P10" Program side
   program  "p10"
10 FPRM M1, P1, P2
```

Reference

END, FPRM

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CHRSRCH (Character search)

Function:

Searches the character string out of the character array

Format

CHRSRCH <Character string array variable>,<Character string>,<Search result storage destination>

Terminology

<Character string array variable>	Specify the character string array to be searched.
<Character string>	Specify the character string to be searched.
<Search result storage destination>	The number of the element for which the character string to be searched is found, is set

Explanation

- Specified character string is searched from the character string array variables, and the element number of the completely matched character string array is set in <search result storage destination>. Partially matched character strings are not searched. Even if CHRSRCH C1$(1), "ROBO", M1 are described in the above statement example, the matched character string is not searched.
- the character string to be searched is not found, 0 is set in <search result storage destination>.
- Character string search is performed sequentially beginning with element number 1, and the element number found first is set. Even if CHRSRCH C1$(3), "MELFA", M2 are described in the above statement example, 2 is set in M2. (The same character string is set in C1$(2) and C1$(6).)
- The <character string array variable> that can be searched is the one-dimensional array only. If a two-dimensional or higher array is specified as a variable, an error will occur at the time of execution.

Reference Program

```
10  DIM C1$(10)
20  C1$(1)="ABCDEFG"
30  C1$(2)="MELFA"
40  C1$(3)="BCDF"
50  C1$(4)="ABD"
60  C1$(5)="XYZ"
70  C1$(6)="MELFA"
80  C1$(7)="CDF"
90  C1$(8)="ROBOT"
100 C1$(9)="FFF"
110 C1$(10)="BCD"
120 CHRSRCH C1$(1), "ROBOT", M1      ' 8 is set in M1
130  CHRSRCH C1$(1), "MELFA", M2     ' 2 is set in M2
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CLOSE (Close)

Function:

Closes the designated file.

Format

```
CLOSE [[#] <File No.> [, [[#] <File No.> ... ]
```

Terminology

<File No.> Designate the No. of the file to be closed.

Explanation

- If a file has been opened for input/output the CLOSE statement will sweep out the data in the buffer. Consequently, the output processing for the file can be completed properly.
- Executing an END statement will also close a file.

Reference Program

```
10 OPEN "COM1:" AS #1   'Open "COM1:" as file No. 1
20 PRINT #1,M1
:  :
100 INPUT #1,M2
110 CLOSE #1            'Close file No. 1, "COM1:"
:  :
200 CLOSE               'Close all open files
```

Reference

END

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CLR (Clear)

Function:

Clears the general-purpose output signals, local variables in program, and global variables between programs.

Format

```
CLR <Type>
```

Terminology

<Type> 0 All steps 1 to 3 below are executed.
1 The general-purpose output signals are cleared based on the output reset pattern. The output reset pattern is designated with parameters ORST0 to ORST224. (0: OFF, 1: ON, *:Hold)
2 All local numeric variables and numeric array variables used in the program are cleared to zero.
3 All global numeric variables and numeric array variables used between programs are cleared to zero.

Explanation

- Either a constant or variable can be used for <Type>.

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CMP JNT (Composition Joint)

Function:

Start the soft control mode (compliance mode) of the specified axis in the JOINT coordinates system. Note: The available robot type is limited such as RH-nAH.

Format

CMP JNT, <Axis designation>

Terminology

<Axis designation>	Specify the axis to be controlled in a pliable manner with the bit pattern. 1 : Enable, 0 : Disable &B00000000 This corresponds to axis 87654321.

Explanation

- It is possible to control each of the robot's axes in the joint coordinate system in a pliable manner. For exam-ple, if using a horizontal multi-joint robot to insert pins in a workpiece by moving the robot's hand up and down, it is possible to insert the pins more smoothly by employing pliable control of the J1 and J2 axes (see the statement example above).
- The degree of compliance can be specified by the CMPG instruction, which sets the spring constant. If the robot is of the RH-AH type, specify 0.0 for the horizontal axes J1 and J2 to make the robot behave equiv-alently to a servo free system (the spring constant is zero). (Note that the vertical axes cannot be made to behave equivalently to a servo free system even if 0.0 is set for them. Also, be careful not to let these axes reach a position beyond the movement limit or where the amount of diversion becomes too large.) Note that 4) and 5) below do not function if this servo-free equivalent behavior is in use.
- The soft state is maintained even after the robot program execution is stopped. To cancel the soft status, execute the CMP OFF command or turn OFF the power.
- When pressing in the soft state, the robot cannot move to positions that exceed the operation limit of each joint axis.
- If the amount of difference between the original target position and the actual robot position becomes greater than 200 mm by pushing the hand, etc., the robot will not move any further and the operation shifts to the next line of the program.
- It is not possible to use CMP JNT, CMP POS, and CMP TOOL at the same time. In other words, an error occurs if the CMP POS or CMP TOOL instruction is executed while the CMP JNT instruction is being performed. Cancel the CMP JNT instruction once using the CMP OFF instruction to execute these instructions.
- Be aware that the position of the robot may change if the servo status is switched on while this instruction is active.
- It is possible to perform jog operations while the robot is in compliance mode. However, the setting of the com-pliance mode cannot be canceled by the T/B; in order to do so, execute this instruction in a program or exe-cute it directly via the program edit screen of the T/B.
- To change the axis specification, cancel the compliance mode with the CMP OFF instruction first, and then execute the CMP JNT instruction again.

Reference Program

```
10 MOV P1
20 CMPG 0.0,0.0,1.0,1.0, , ,  'Set softness
30 CMP JNT,&B11               'The J1 and J2 axes are put in the state
                              where they are controlled in a playable
                              manner.
40 MOV P2
50 HOPEN 1
60 MOV P1
70 CMP OFF                    'Return to normal state
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CMP OFF (Composition OFF)

Function:

Release the soft control mode (compliance mode).
Note: The available robot type is limited such as RV-4A. Refer to "Available robot type".

Format

```
CMP OFF
```

Reference Program

10	MOV P1	Moves to in front of the part insertion position.
20	CMPG 0.5,0.5,1.0,0.5,0.5,,,	Set softness.
30	CMP TOOL, &B011011	The X, Y, A, and B axes are put in the state where they are controlled in a pliable manner.
40	MVS P2	Moves to the part insertion position.
50	M_OUT(10)=1	Instructs to close the chuck for positioning.
60	DLY 1.0	Waits for the completion of chuck closing.(1 sec.)
70	HOPEN 1	Open the hand.
80	MVS, -100	Retreats 100 mm in the Z direction of the TOOL coordinate system.
90	CMP OFF	Return to normal state.

Explanation

- This instruction cancels the compliance mode started by the CMP TOOL, CMP POS, or CMP JNT instruction.
- In order to cancel jog operations in the compliance mode, either execute this instruction in a program or execute it directly via the program edit screen of the T/B.

Available robot type

RV-20A
RV-4A/5AJ series
RV-1A/2AJ
RV-2A/3AJ series
RH-5AH/10AH/15AH series

Reference
CMPG, CMP JNT, CMP POS, CMP TOOL

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CMP POS (Composure Posture)

Function:

Designates which axis to softly move the robot in the orthogonal coordinate system.

Note: Excluding model RP-*

Format

```
CMP POS, <Axis designation>
```

Terminology

<Axis designation>	Designate axis to move softly with a bit pattern. 00000000 ... from low-order bit X, Y, Z, ... C, L1, L2

Explanation

- The robot can be moved softly with the orthogonal coordinate system.
- For example, when inserting a pin in the vertical direction, if the X, Y, A and B axes are

set to soft operation, the pin can be inserted smoothly.
- The degree of softness can be designated with the CMPG command.
- The soft state is maintained even after the robot program execution is stopped until the CMP OFF is executed or until power is turned ON again after being turned OFF.
- When pressing in the soft state, the robot cannot move to positions that exceed the operation limit of each joint axis.
- If the deviation between the actual robot position and the target position is larger than the parameter designated value NMPCDIST due to pressing, etc., the robot will stop. Note that the robot will not stop while the program is running.
- If the CMP POS command is executed while the CMP TOOL command is functioning, an error will occur. Cancel the CMP OFF command before executing the CMP POS command.
- If the servo turns from OFF to ON while this command is active, the robot position could change.
- Jogging is possible in the soft state.

Reference Program

```
10 MOV P1
20 CMPG 0.5,0.5,1.0,0.5,0.5   'Set softness
30 CMP POS, &B00011011        'Enter soft state
40 MVS P2
50 M_OUT(10)=1
60 MVS P1
70 CMP OFF                    'Return to normal state
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CMP TOOL (Composition Tool)

Function:

Designates which axis to softly move the robot in the tool coordinate system.

Note: Excluding model RP-*

Format

CMP TOOL, <Axis designation>

Terminology

<Axis designation>	Designate axis to move softly with a bit pattern. 00000000 ... from low-order bit X, Y, Z, ... C, L1, L2

Explanation

- The robot can be moved softly with the tool coordinate system.
- For example, when inserting a pin in the tool coordinate Z axis direction, if the X, Y, A and B axes are set to soft operation, the pin can be inserted smoothly.
- The degree of softness can be designated with the CMPG command.
- The soft state is maintained even after the robot program execution is stopped until the CMP OFF is executed or until power is turned ON again after being turned OFF.
- When pressing in the soft state, the robot cannot move to positions that exceed the operation limit of each joint axis.
- If the deviation between the actual robot position and the target position is larger than the parameter designated value NMPCDIST due to pressing, etc., the robot will stop. Note that the robot will not stop while the program is running.
- If the CMP POS command is executed while CMP TOOL command is functioning, an error will occur. Cancel the CMP OFF command before executing the CMP TOOL command.
- If the servo turns from OFF to ON while this command is active, the robot position could change.
- Jogging is possible in the soft state.

Reference Program

```
10 MOV P1
20 CMPG 0.5,0.5,1.0,0.5,0.5   'Set softness
30 CMP TOOL, &B00011011       'Enter soft state
40 MVS P2
50 M_OUT(10)=1
60 MVS P1
70 CMP OFF                    'Return to normal state
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CMPG (Composition Gain)

Function:

Designates the robots's softness.

Note: Excluding model RP-*

Format

CMP POS, CMP TOOL:

```
CMPG [<X axis gain>], [<Y axis gain>], [<Z axis gain>], [<A axis gain>],[<B axis gain>],[<C axis gain>], [<L1 axis gain>], [<L2 axis gain>]
```

CMP JNT:

```
CMPG [<J1 axis gain>], [<J2 axis gain>], [<J3 axis gain>], [<J4 axis gain>],[<J5 axis gain>],[<J6 axis gain>], [<J7 axis gain>], [<J8 axis gain>]
```

Terminology

<X to L2 axis gain>	The softness can be set for each axis. 1.0 is the normal state, and 0.0 is the softest state. If the value is omitted, the current setting value will be applied.
<J1 to J8 axis gain>	s.o.

Explanation

- The softness can be designated in each axis units.
- The soft state will not be entered unless validated with the CMP POS or CMP TOOL command.
- A spring-like force will be generated in the proportion to the deviation of the command position and actual position. CMPG designates that spring constant.
- The deviation of the command position and actual position can be read with M_CMPDST. The success/failure of pin insertion can be checked using this variable.
- If a small gain is set, and the soft state is entered with the the CMP POS and CMP TOOL commands, the robot position could drop. Set the softness state gradually while checking it.
- If this command is executed in the soft state, the softness can be changed during the operation.

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CNT (Continuous)

Function:

Designates continuous operation control for interpolation operations.

Format

```
CNT <Continuous operation mode/acceleration/deceleration operation mode>][, <Numeric value 1 > ] [Numeric value 2>]
```

Terminology

<valid/invalid>	Designate the continuous operation or acceleration/deceleration operation mode. Designate either 1 or 0. 1 is continuous operation, and 0 is acceleration/deceleration operation.
<Numeric value 1>	Designate the neighborhood distance for starting the orbit change with a mm unit.

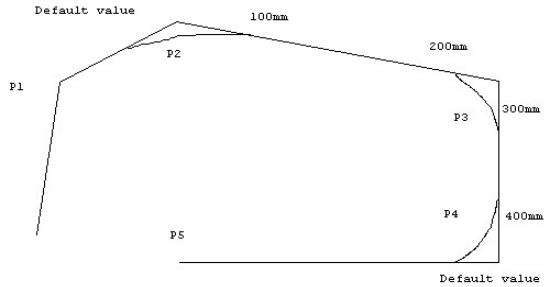
<Numeric value 2> Designate the neighborhood distance for ending the orbit change with a mm unit.

Explanation

- To make the change start position and end position both less than the designated value, one is changed with a distance shorter than the designated value.
- The CNT default value is "CNT0".
- If numeric value 1 and numeric value 2 are omitted, these will be connected with the default setting value.
- If numeric value 2 is omitted, the same value as numeric value 1 will be applied.
- The FINE valid designation is invalid for smooth operation.
- If the numeric value designation is reduced, the operation time may take longer than when CNT is invalid.

Reference Program

```
10 CNT 0              'Invalidate CNT
20 MVS P1             'Operate with acceleration/deceleration
30 CNT 1              'Validate CNT (default setting value)
40 MVS P2             'Execute with smooth operation
50 CNT 1, 100, 200    'Start distance 100mm. end position 200mm
60 MVS P3             'Execute with smooth operation
70 CNT 1, 300         'Start distance 300mm, end position 300mm
80 MOV P4             'Execute with smooth operation
90 CNT 0              'Invalidate CNT
100 MOV P5
```



Reference

Movement related commands, speed related commands, acceleration/deceleration related commands, FINE, OADL

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

COM OFF (Communication OFF)

Function:

Designates "disable" for any interrupts from a communication line.

Format

COM [(<Communication line No.>)] OFF

Terminology

<Communication Line No.> Describe 1, 2 or 3.

Explanation

- When COMMON OFF is executed, even if communications are attempted, the interrupt will not be generated.
- For information on communication line Nos., refer to the page for OPEN.

Reference Program

```
10 COM(1) OFF   'Disable communication interrupt from communication line
                No. 1.
```

Reference

OPEN, COM_ON, COM STOP, ON COM GOSUB

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

COM ON (Communication ON)

Function:

Designates "enable" for interrupts coming from a communication line.

Format

COM [(<Communication line No.>)] ON

Terminology

<Communication Line No.> Describe 1, 2 or 3.

Explanation

- For information on communication line Nos., refer to the page for OPEN.

Reference Program

```
10 COM(1) ON   'Enable communication interrupt from communication line
               No. 1.
```

Reference

OPEN, COM_OFF, COM STOP, ON COM GOSUB

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

COM STOP (Communication Stop)

Function:

Sets "stop" for the interrupts coming from a communication line.

Format

COM [(<Communication line No.>)] STOP

Terminology

<Communication Line No.> Describe 1, 2 or 3.

Explanation

- After COM STOP is executed, even if communication is attempted, the interrupt will not be generated. Note that the receiving data and the fact of the interrupt will be recorded, and be executed the next time the line is reopened.

Reference Program

```
10 COM(1) STOP
                'Stop the communication interrupts for communication
```

```
                line 1.
```

Reference

OPEN, COM_OFF, COM ON, ON COM GOSUB

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF ACT (Define act)

Function:

Defines the status of the interrupt and how it will be processed.

Format

DEF ACT <Priority No.>, <Expression> <Process> [, <Type>]

Terminology

<Priority No.>	This is the priority No. of the interrupt. It can be set with constant Nos. 1 to 8.
<Expression>	For the interrupt status, use the formats described below: (Refer to the syntax diagram) <Numeric type data> <Comparison operator> <Numeric type data> or <Numeric type data> <Logical operator> <Numeric type data> <Numeric type data> refers to the following: <Numeric type constant>| <Numeric variable>|<Numeric array variable>|<Component data>
<Process>	Refers to a GOTO statement or a GOSUB statement used to process an interrupt when it occurs.
<Type>	When omitted: Stop type 1 The robot stops at the stop position, assuming 100% execution of the external override. If the external override is small, the time required for the robot to stop becomes longer, but it will always stop at the same position. S : Stop type 2 (only for software version E3 or later) The robot decelerates and stops in the shortest time and distance possible, independently of the external override. L : Execution complete stop The interrupt processing is performed after the robot has moved to the target position (the line being executed is completed).

Explanation

- The priority level for the interrupts is decided by the <Priority No.>, and the priority level, from the highest ranges from 1 to 8.
- There can be up to 8 settings for the interrupts. Use the <Priority No.> to differentiate them.
- When two interrupts have been defined with the same priority level, the one defined later is validated.
- Since DEF ACT defines only the interrupt, always use the ACT command to designate the enable/disable status of the interrupt.
- The communications interrupt (COM) has a higher priority level than any of the interrupts defined by DEF ACT.
- DEF ACT definitions are valid only in the programs where they are defined. These are invalid when called up in a program by CALL P. If necessary, the data in a subprogram may need to be redefined.
- If an interrupt is generated when a GOTO command is designated by <Process> for a DEF ACT command, during execution of the remaining program, the interrupt in progress will remain, and only interrupts of a higher level will be accepted. The interrupt in progress for a GOTO statement can be canceled with the execution of an END statement.

Reference Program

```
10  DEF ACT 1,M_IN(10)=1 GOSUB 100        'Defines the subroutine at line
                                          100 to be the one to be called
                                          up when the status for the
                                          general purpose input signal
                                          No.17 is ON.
20  DEF ACT 2,MFG1 AND MFG2 GOTO 200      'Defines the line 200 as the
```

```
                                        one to jump to when the logic
                                        operation of AND applied to
                                        MFG1 or MFG2 results in "true".
30  DEF ACT 3,M_TIMER(1)>10.5 GOSUB 300  'When 10.5 seconds pass, the
                                        program jumps to the line 300
                                        subroutine.

100 M_TIMER(1)=0
110 ACT 3=1
```

Reference

ACT, GOSUB, RETURN, Robot status variables

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF ARCH (Define arch)

Function:

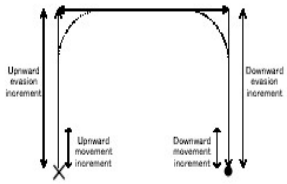
This instruction defines an arch shape for the arch motion movement corresponding to the MVA instruction.

Format

This function is available for controller software version G2 or later.

DEF ARCH <Arch number>,
[<upward movement increment>] [<downward movement increment >] [<Upward evasion increment>], [<downward evasion increment>], [<interpolation type>], [<interpolation type 1>, <interpolation type 2>]

Terminology

<Arch number>	Arch motion movement pattern number. Specify a number from 1 to 4 using a constant or a variable.
<Upward movement increment> <Downward movement increment >	
<Upward evasion increment>	
<Downward evasion increment>	
<Interpolation type>	Interpolation type for upward and downward movements. Linear/joint = 1/0
<Interpolation type 1>	Detour/short cut = 1/0
<Interpolation type 2>	3- axis XYZ/ Equivalent rotation = 1/0

Explanation

- If the MVA instruction is executed without the DEF ARCH instruction, the robot moves according to the arch shape specified by the parameters.
- Used to change the increments in a program, etc. [Related instructions]

Reference Program

```
10 DEF ARCH 1,5,5,20,20
20 MVA P1,1          'Performs the arch motion movement defined in the shape definition
                     in line 10.
30 MVA P2,2          'The robot moves according to the default values specified by the
                     parameters.
```

Reference

ACCEL (Accelerate), OVRD (Override),MVS (Move S)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF CHAR (Define Character)

Function:

Declares a character string variable

Format

DEF CHAR <Character string position variable name> [, <Character string variable name >] ...

Terminology

<Character string variable name>	Designate the variable name.

Explanation

- The variable name can have up to eight characters. Refer to the "Types of characters that can be used" for the characters that can be used.
- When designating multiple variable names, the maximum value (123 characters including command) can be set on one line.
- The declared variable follows the "Character string variable„ rules.
- After declaration, the variable can be used in the same manner as the C variable.

Reference Program

```
10  DEF CHAR MESSAGE      'Declare MESSAGE as a character string
                          variable.
20  MESSAGE = "WORKSET"   'Substitute "WORKSET" in the MESSAGE variable.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF FN (Define Function)

Function:

Defines a function and gives it name.

Format

DEF FN <Name> [(Dummy argument> [, <Dummy argument>] ...)] = <Function definition expression>

Terminology

<Name>	Describe an identifier character + character string.
<Dummy argument>	When a function has been called up, it is transferred to the function. It is possible to describe all the variables, and up to 16 variables can be used.
<Function Definition Expression>	Describe the expression for what operation to use as a function.

Explanation

- FN + <Name> becomes the name of the function. The function name can be up to 8 characters long.
  Example:
  Numeric typeFNMAX Identifier character: M
  Character string typeFNCAME$ Identifier character:
  C (Describe $ at the end of the name)
- A function defined with DEF FN is called a user-defined function. A function as long as one line can be described.
- Built-in functions and user-defined functions that have already been defined can be used in the function definition expression. In this case, up to 16 levels of user-defined functions can be written.
- If the variables used in <Function Definition Expression> are not located in <Dummy

Argument>, then the value that the variable has at that time will be used. Also, an error will occur if during execution, the number or argument type (numeric value or character string) of arguments differs from the number or type declared.
- A user-defined function is valid only in the program where it is defined. It cannot be used by a CALLP designation program.

Reference Program

```
10  DEF FNMAVE(MA,MB)=(MA+MB)/2     'Defines FNMAVE to be used to find the
                                    average of two numeric values.
20  MDATA1=20
30  MDATA2=30
40  MAVE=FNMAVE(MDATA1,MDATA2)      '25, the average of 20 and 30, is
                                    substituted for the numeric variable
                                    MAVE.
```

Reference

Variables, Array variables, Functions

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF INTE/FLOAT/DOUBLE (Define Integer/Float/Double)

Function:

Declares the arithmetic variable.

Format

```
DEF INTE <Arithmetic variable name> [, <Arithmetic variable name>] ...
DEF FLOAT <Arithmetic variable name> [, <Arithmetic variable name>] ...
DEF DOUBLE <Arithmetic variable name> [, <Arithmetic variable name>] ...
```

Terminology

<Arithmetic Variable Name>	Designate the variable name.

Explanation

- The variable name can have up to eight characters. Refer to the "Types of characters that can be used" for the characters that can be used.
- When designating multiple variable names, the maximum value (123 characters including command) can be set on one line.
- The variable declared with INT will be an integer type. (-32768 ~ +32767)
- The variable declared with FLOAT will be a single-precision type. (+/- 1.70141E+38)
- The variable declared with DOUBLE will be a double-precision type. (+/- 1.701411834604692E+308)

Reference Program

```
10 DEF INTE WORK   'Declare WORK as an arithmetic variable name.
20 WORK = 100      'Substitute the value 100 in WORK.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF IO (Define IO)

Function:

Declares an input/output variable.

Format

```
DEF IO Input/Output variable name> = <Type designation>, <Input/Output bit No.> [, <Mask information>]
```

Terminology

<Input/Output Variable Name>	Designate the variable name.
<Type Designation>	Designate BIT, BYTE, WORD or INTEGER.
<Input/Output Bit No.>	Designate the input or output bit No.
<Mask Information>	Designate when only a specific signal is to be validated.

Explanation

- The variable name can have up to eight characters. Refer to the „Types of characters that can be used„ for the characters that can be used.
- When mask information is designated, only the specified signal will be validated.
  Example: For the 2Oth line, mask processing is carried out with hexadecimal 0F, so Nos. 107 to 110 will be validated, and Nos. 111 to 114 will be invalidated (always 0).
  InvalidValid
  0 0 0 0 1 1 1 1
  No.114 No. 107 (Input/output bit No.)

Reference Program

```
10 DEF IO PORT1 = BIT,6          'Assign the input/output variable named
                                 PORT1 to the input/output bit No. 6 with
                                 the BIT type.
20 DEF IO PORT2 = BYTE,8,&H0F    'Assign the input/output variable named
                                 PORT2 to the input/output bit No. 8 with
                                 the BYTE type, and designate the mask
                                 information as hexadecimal 0F.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF JNT (Define Joint)

Function:

Declares a joint variable.

Format

DEF JNT <Joint variable name> [, <Joint variable name>] ...

Terminology

<Joint Variable Name>	Designate the variable name.

Explanation

- The variable name can have up to eight characters. Refer to the "Types of characters that can be used" for the characters that can be used. The declared variable follows the "Joint variable" rules. After declaration, the variable can be used in the same manner as the J variable.

Reference Program

```
10 DEF JNT SAFE   'Declare SAFE as the joint variable
20 MOV SAFE       'Move to SAFE
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF PLT (Define Pallet)

Function:

Defines the pallet. (3-point pallet, 4-point pallet)

Format

DEF PLT <Pallet No.>, <Start point>, <End point A>, <End point B>, [<Diagonal point>], <Quantity A>, <Quantity B>, <Assignment direction>

Terminology

<Pallet No.>	This is the selection No. of the set pallet. (1 to 8)
<Start Point>	Refers to the pallet's start point. Position operation expressions can be described.
<End Point A>	One of the ending points for the pallet. Transit point of arc for arc pallet. Position operation expressions can be described.
<End Point B>	Another ending point for the pallet. Transit point of arc for arc pallet. Position operation expressions can be described.
<Diagonal Point>	The diagonal point from the pallet's start point. Insignificant for arc pallet. Position operation expressions can be described.
<Quantity A>	The No. of workpieces from the pallet's start point to the end point A. The No. of workpieces between the pallet start point and arc end point when using an arc pallet. Numeric operation expressions can be described.
<Quantity B>	The No. of workpieces from the pallet's start point to the end point A. Insignificant for an arc pallet. (0, etc., must be designated.) Numeric operation expressions can be described.
<Assignment Direction>	Describe 1, 2 or 3 for the direction of assignment for when the grid is divided and given numbers. Numeric operation expressions can be used. (Refer to the explanation section for details on the assignment direction.)

Explanation

- A 3-point pallet or a 4-point pallet can be selected.
  - A 3-point pallet offers simplified teaching.
  - A 4-point pallet is effective for better precision.
- The assignment direction is as follows.
  Zigzag = 1 , Same direction = 2, Arc pallet = 3
- The command is valid only within the program being executed. The command is invalid in the program that calls up the command from another program. If necessary, redefine.
- Quantity A and B should be a non-zero positive number, while if 0 or a negative number is assigned, an error will occur.
- If Quantity A x Quantity B exceeds 32,767, an error will occur when operation starts.
- The value of quantity B is insignificant for the arc pallet, but it must not be omitted. The diagonal point will be insignificant even when designated.

Reference Program

```
10 DEF PLT 1,P1,P2,P3, ,4,3,1    '3-point pallet definition
20 DEF PLT 1,P1,P2,P3,P4,4,3,1   '4-point pallet definition
```

Reference

PLT, Movement commands

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DEF POS (Define Position)

Function:

Declares a position variable.

Format

DEF POS <Position variable name> [, <Position variable name>] ...

Terminology

<Position Variable Name>	Designate the variable name.

Explanation

- The variable name can have up to eight characters. Refer to the "Types of characters that can be used" for the characters that can be used.
- When designating multiple variable names, the maximum value (123 characters including command) can be set on one line.

The declared variable follows the "Position variable" rules.
- After declaration, the variable can be used in the same manner as the P variable.

Reference Program

```
10 DEF POS WORKSET                  'Declare WORKSET as the position
                                    variable.
20 MOV P1                           'For XYZ type position variables
                                    starting with P, the definition of
                                    "DEF POS" is not required.
30 WORKSET=(250,460,100,0,0,-90,0,0)(0,0) 'Define WORKSET
40 MOV WORKSET                      'Move to WORKSET
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DIM (Dim)

Function:

Declares the quantity of elements in the array variable. (Arrays up to the third dimension are possible.)

Format

DIM <Variable name> (<Max. value of subscript>, [<Max. value of subscript>]) [,<Variable name> (<Max. value of subscript> [, <Max. value of subscript>])]

Terminology

<Variable Name>	Describe the name of the array variable.
<Max. Value of Subscript>	Describe in terms of constants, the number of elements in an array variable.

Explanation

- A one-dimensional, two-dimensional or three-dimensional array can be used.
- <Max. Value of Subscript> can be described with numeric constants from 1 to 999. Numeric operation expressions cannot be used.
- If a subscript with a value larger than the maximum value defined with DIM is used, an error will occur at the execution.
- When the DIM statement is executed, the array variable does not have a default value, and instead is non-specified.
- An array cannot be used without the DIM statement. If the number of elements is a real number, the decimal numbers will automatically be rounded off to create an integer.
- The command is valid only within the program being executed. The command is invalid in the program that calls up the command from another program.
- This command must be redefined when used in a subprogram. A value (teachable) can be directly set or read from the position array variable data declared with this command.

Reference Program

```
10 DIM PDATA(10)    'Defines the position array variable PDATA having ten
                    elements.
20 DIM MDATA(2,3)   'Defines the two-dimensional numeric array variable
                    MDATA having 2x3 elements.
```

Reference

Array variables

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DLY (Delay)

Function

When used as a single command: At a designated time, it causes a wait.

When used as an additional pulse output: Designates an output time for a pulse

Format

When used as a single command

```
DLY <Time>
```

When used as an additional pulse output

```
<command> DLY <Time>
```

Terminology

<Time>	Describes the waiting time or the output time for the pulse output, in terms of a numeric operation expression. Unit: [Seconds] The minimum value that can be set is 0.05 seconds.

Explanation

- This command is used to create delay times in programs, or to write in the OUT statement and set the pulse output time.
- The pulse output will be executed simultaneously as the next command in the lines that follow.
- Up to 4 pulse outputs can be issued simultaneously. Exceeding this, an error will occur when the program tries to execute it.
- After the designated time has elapsed, the condition that existed just before the pulse output was issued will resume.
- Within the designated time, if an END command, or the last line of the program is executed, or if an emergency stop is made to stop the program, the signal output status will keep its current status.
- The relation of the priority levels for other interrupts is as shown below:
  COM> ACT> WTHIF (WTH) >Pulse output (Time setting ON)

Reference Program

```
10 DLY 30                 'Wait for 30 seconds
20 M_OUT(17)=1 DLY 10.0   'Send the signal output to the general-purpose
                          output signal 17 for 10 seconds.
```

Reference

Robot Status Variables, Substitute

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

END (End)

Function:

Ends the program execution.

Format

```
END
```

Explanation

- Several END statements can be described within one program.
- The END statement does not need to be described at the end of the program. Used in the host program, the command will close all files before closing.
- An END statement in a program called up by a CALLP command will transfer control to the program that issued the CALL P command.
- Used in the host program, the command will close all files before closing.
- At program END; the SPD, ACCEL, OADL, JOVRD, OVRD, FINE, and CNT settings will be initialized.

Reference Program

```
100 END
```

Reference

CALLP, FPRM, GOSUB, RETURN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ERROR (Error)

Function:

Generates an error in the user program.

Format

ERROR <Error no.>

Terminology

<Error no.>	Either a constant or a numeric operation expression can be set. Designate the No. within the range of 9000 to 9299.

Explanation

- It is possible to generate any error in the 9000's number range by executing this instruction.
- If a LOW level or HIGH level error is generated, the program is paused. Lines after the ERROR instruction are not executed. A CAUTION error does not pause a program; the next line and onward are executed. The action of system by error number is shown in the Table below.
- It is possible to create up to 20 error messages using parameters UER1 to UER20.
- A system error occurs if a value outside the error number range shown in Table below is specified.
- It is also possible to variable as an 'error-number'.

Error no.	System behavior
9000 - 9099 (H level error)	The program execution is stopped, and the servo power is shut off. The error state is reset when error reset is input.
9100 - 9199 (L level error)	The program execution is stopped. The error state is reset when error reset is input.
9200 - 9299 (CAUTION)	The program execution is continued. The error output turns OFF with error reset.

Reference Program

Generate the error 9000

```
100 ERROR 9000
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

FINE (Fine)

Function:

Designates the positioning end conditions for interpolation when not in smooth operation control.

Format

FINE <Invalid/No. of pulses> [, <Axis No.>]

Terminology

<No. of pulses>	Designate the positioning pulses. This will be invalid to when set to 0. The default value is 0.
<Axis No.>	Designate the axis No. to which the positioning pulses are to be designated. The positioning pulses will be applied on all axes when omitted.

Explanation

- FINE is invalid in the program until the FINE command is executed. Once FINE is validated, it remains valid until invalidated.
- FINE is invalidated at the end of the program.
- When the CNT valid state (in smooth operation control) is entered, the FINE command will be ignored even if it is valid (i.e., it will be treated as invalid, but the status will be kept).

Reference Program

```
10 FINE 300     'Designate 300 for the positioning pulses.
20 MOV P1
30 FINE 100,2   'Change the 2nd axis positioning pulses to 100.
40 MOV P2
50 FINE 0       'Invalidate the positioning pulse designation.
60 MOV P3
70 FINE 100     'Designate 100 for the positioning pulses.
80 MOV P4
```

Reference

CNT,movement command

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

FOR NEXT (For Next)

Function:

Repeatedly executes the program between the FOR statement and NEXT statement until the end conditions are satisfied.

Format

```
FOR <Counter> = <Default value> TO <End value> [STEP <Increment>]
.
.
NEXT [<Counter 1> [, <Counter2>] ... ]
```

Terminology

<Counter>	Describes the numeric value data that represents the counter for the number of repetitions. Same for <Counter 1> and <Counter 2>.
<Default Value>	Sets default value of the counter for the number of repetitions as a numeric operation expression.
<End Value>	Set the end value of the counter for the number of repeats as a numeric operation expression.
<Increment>	Sets the value of the increments for the counter for the number of repetitions as a numeric operation expression.

Explanation

- Program depth
  - It is possible to describe FOR-NEXT statements between other FOR-NEXT statements.
  - With each FOR-NEXT process, the control structure of the program becomes one level deeper.
  In this system, it is possible for the control structure to have up to 16 levels within a program. Exceeding 16 levels will cause an error during execution.

Reference Program

A program that adds the numbers 1 to 10

```
10 MSUM=0                  'Initialize the total MSUM.
20 FOR M1=1 TO 10 STEP 1   'Increase the counter by 1 from 1 to 10 for
                           the numeric variable M1. If 10 is exceeded, go
                           to line 50. Step 1 can be omitted.
30 MSUM=MSUM+M1            'Add M1 value to numeric variable MSUM.
40 NEXT M1                 'Return to line 20.
50 END                     'End the program.
```

A program that puts the result of a product of two numbers into a 2-dimensional array variable (Using FOR-NEXT as a nesting structure)

```
10 DIM MBOX(10,10)         'Reserve space for a 10 x 10 array.
20 FOR M1=1 TO 10 STEP 1   'Increase the counter by 1 from 1 to 10 for
                           the numeric variable M1. If 10 is exceeded,
                           transfer control to line 70. Step 1 can be
                           omitted.
30 FOR M2=1 TO 10 STEP 1   'Increase the counter by 1 from 1 to 10 for
                           the numeric variable M2. If 10 is exceeded,
                           transfer control to line 60. Step 1 can be
                           omitted.
40 MBOX(M1,M2)=M1*M2       'Substitute the value of M1*M2 for the array
                           variable MBOX (M1, M2).
50 NEXT M2                 'Return to line 30.
60 NEXT M1                 'Return to line 20.
70 END                     'End the program.
```

Cases in which the repeated control is not executed are shown below.

- The counter's <Default Value> is greater than <End Value> and <Increment> is a positive number.

- The counter's <Default Value> is smaller than <End Value>, and <Increment> is a negative number.

If the FOR statement and NEXT statement contradict each other, an error will occur.

Note that when the FOR and NEXT statements are in a nesting structure and have the same end value, the statement can be described with one NEXT statement. For example, line 50 and 60 in the example can be combined to be written "NEXT M2, M1" in one line.

When the NEXT statement corresponds to the closest FOR statement, the variable name in the NEXT statement can be omitted. In the example, "M2" in line 50 and "M1" in line 60 can be omitted.

Reference

GOSUB, WHILE, END, Numeric operation expressions (Syntax diagram)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

FPRM (FPRM)

Function:

Defines the order of the arguments, the type, and number for the main program that uses arguments in a subprogram (i.e., when the host program uses another program with CALLP).

Format

FPRM <Dummy argument> [, <Dummy argument> ...]

Terminology

<Dummy Argument>	The variable in the subprogram that is transferred to the main statement when executed. All variables can be used. Up to 16 variables may be used.

Explanation

- FPRM is unnecessary if there are no arguments in the subprogram that is called up.

- If a variable used in a program is not in <Dummy Argument> the variable will use the value it currently holds.
- Depths between programs
  - The calling up of programs allows the level of the control structure between programs to go deeper.
  - One calling up of a program results in the program structure becoming one level deeper.
- It is possible for the control structure in a program to go as deep as 7 levels. If 7 levels are exceeded, an error will occur during execution.
- The calculation results in the subprogram cannot be transferred to the host program using temporary arguments. In this case, transfer the results using external variables.

Reference Program

```
10 FPRM M1,P1,P2
```

Reference

CALLP, Variables, Array variables, Functions

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

GETM (Get Mechanism)

Function:

Acquires the mechanism resource for the designated mechanism No.

Format

GETM <Mechanism no.>

Terminology

<Mechanism no.>	1 to 3, Specify this argument by using a numeric value or variable. The standard system,s robot arm is assigned to mechanism 1.

Explanation

- If the mechanism,s resource is not acquired with GETM, the movement commands of motor power ON/OFF commands cannot be executed..
- GETM cannot be commanded to a mechanism for which GETM has already been executed.
- If the argument is omitted from the system status variable requiring the mechanism designation, the currently acquired mechanism will be designated.
- Main slot (slot 1) acquires mechanism 1 in the default state.
- If the program is stopped, RELM will be executed automatically by the system. When the program is restarted, GETM will be executed automatically.

Reference Program

```
10 GETM 1      'Acquire mechanism 1 resource
20 SERVO ON    'Turn mechanism 1 servo ON
30 MOV P1
40 MVS P2
45 P3=P_CURR   'Substitute P3 in mechanism 1 current position.
50 SERVO OFF   'Turn mechanism 1 servo OFF.
60 RELM        'Release mechanism 1 resource.
70 END
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

GOSUB (Go Subroutine)

Function:

Calls up the subroutine at the designated line No. or line label.

Format

GOSUB <Call destination>

Terminology

<Call Destination> Describes the line No. or label name.

Explanation

- When calling a subroutine in the base program, use a label name starting with "L_". Note that if the same label is found in the local program, the local program's routine will be executed.

Reference Program

For a line number

```
100  GOSUB 1000
110  END
1000 MOV P1
1010 RETURN        ' Be sure to use the RETURN instruction to return
```

For a label

```
100  GOSUB *LBL
110  END
1000 * LBL
1010 MOV P1
1010 RETURN        ' Be sure to use the RETURN instruction to return
```

Reference

RETURN, END, GOTO, DEF ACT, ACT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

GOTO (Go To)

Function:

Unconditionally branches to a designated line No. or label.

Format

GOTO <Branch destination>

Terminology

<Branch Destination> Describes the line No. or label name.

Explanation

- GOTO cannot be commanded to a label in the base program. Thus, even if the designated label name is in the base program, the no skip destination error will occur.

Reference Program

```
100 GOTO 500       'Branch to line 500
:
500 MOV P1         'Move to Position P1
:
700 GOTO *LABLE    'Branch to the Subprogram LABLE
:
900 *LABLE         'Define „LABLE" as a branchmark
```

Reference

IF THEN ELSE, GOSUB, DEF ACT, ACT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

HLT (Halt)

Function:

Interrupts the execution of the program and movement of the robot, and stops.

Format

```
HLT
```

Explanation

- Interrupts the execution of a program and decelerates the robot to a stop. The system will enter the waiting state.
- To restart, start the O/P or issue the start signal from an external source. The program will be restarted at the next line after the HLT statement.
- Note that if the HLT statement is an appended statement, the operation will restart from the same line of the program where it was interrupted.

Reference

WTHIF, WTH, SKIP

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

HOPEN/HCLOSE (Hand Open/Close)

Function:

Commands the hand to open or close.

Format

```
HOPEN <Hand No.> [, <Starting grasp force>, <Holding grasp force>, <Starting grasp force holding time>]
HCLOSE <Hand No.>
```

Terminology

<Hand No.>	Select a numeric value between 1 and 8.
<Starting grasp forcer>	This parameter is valid for the motorized hand, and invalid for any other type of hand. Set the required grasping force for starting the hand open/close. Set the grasping force as a step between 0 and 63 (63 = 3.5kgf). The default value is 63. When omitted, the previous setting value will be applied.
<Holding grasp force>	This parameter is valid for the motorized hand, and invalid for any other type of hand. Set the required grasping force for holding the hand open / close. Set the grasping force as a step between 0 and 63 (63 = 3.5kgf). The default value is 63. When omitted. the previous setting value will be applied.

<Starting grasp force holding timer>	This parameter is valid for the motorized hand. Set the time to hold the starting grasping force as a value from 0.00 (sec.). The default value is 0.3 sec.

Explanation

- The operation (single/double) of each hand is set with parameter HANDTYPE.
- The status of the hand output signal when the power is turned ON is set with parameter HANDINIT.
- The hand input signal can be confirmed with the robot status variable M-HNDCQ ("Hand input number"). The signal can also be confirmed with the input signals No.900 to 907 (when there is one mechanism).
  10 HCLOSE 1
  20 IF M_HNDCQ(1)<>1 THEN GOTO 20
  30 MOV P1
- There are related parameters. Refer to "1.6 Functions set with parameters" on page 33 of this manual.
-

Reference Program

```
10   HOPEN 1              'Open hand 1.
20   HCLOSE 2             'Close hand 2.
```

Reference

Robot Status Variable
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

IF THEN ELSE (If Then Else)

Function:

A process is selected and executed according to the results of an expression.

Format

IF <Expression> THEN <Process> [ELSE <Process>]

Terminology

<Expression>	Describe the expression targeted for comparison as a comparison operation expression or logic operation expression.
<Process>	Describe the process following THEN for when the comparison results are true, and the process following ELSE for when the comparison results are false.

Reference Program

```
100 IF M1>10 THEN 1000
110 IF M1>10 THEN GOTO 1000 ELSE GOTO 2000
```

Reference

ON GOSUB, ON GOTO, Numeric operation expressions (syntax diagram)

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

INPUT # (Input)

Function:

Inputs data from a file (input device). All data uses the ASCII format.

Format

INPUT # <File No.>, <Input data name> [, <Input data name>] ...

Terminology

<File No.>	Describe a number between 1 and 8. This corresponds to the file No. assigned with the OPEN command.
<Input Data Name>	Describe the variable name for saving the input data. All variables can be described.

Explanation

- Data is input from file (input device) having the file No. opened with the OPEN statement, and is substituted in the variable. If the OPEN statement has not been executed, an error will occur.
- The type of data input and the type of variable that is substituting it must be the same.
- When describing multiple variable names, use a comma (,) between variable names as delimiters.
- When the INPUT statement is executed, the status will be "standby for input". The input data will be substituted for the variables at the same time as the carriage return (CR and LF) are input.
- If multiple elements are input due to the number of arguments in the INPUT statement, reading will continue to the next INPUT statement. When the END or CLOSE statement is executed, the data saved in the buffer will be erased.
  Example: To input both a character string, numeric value and position.
  10 INPUT# 1,C1$,M1,P1
  PRN MELFA,125.75,(130.5,-117.2,55.1,16.2,0,0)(1,0)
  MELFA is substituted in C1$, 125.75 in M1, and (130.5, -117.2,55.1,16.2,0,0)(1,0) in P1.

Reference Program

```
10 OPEN "COM1:" AS #1   'Assign RS-232-C for file No. 1.
20 INPUT# 1,M1          'When data is input from the keyboard, that value
                        will be input in the numeric variable M1
```

Reference

OPEN, PRINT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

JOVRD (J Override)

Function:

Designates the override that is valid only during the robot,s joint movements.

Format

JOVRD <Designated override>

Terminology

<Designated override>	Describes the override as a real number. A numeric operation expression can also be described. Unit: [%] (Recommended range: 1 to 100.0)

Explanation

- The JOVRD command is valid only during joint interpolation.
- The actual override is = (Operation panel (T/B) override setting value) x (Program override (OVRD command)) x (Joint override (JOVRD command)). The JOVRD command changes only the override for the joint interpolation movement.
- The 100% <Designate override> is the maximum capacity of the robot. Normally. the system default value (M_NOVRD) is set to 100%. The value is reset to the default value when the END statement is executed or the program is reset.

Reference Program

```
10  JOVRD 50
20  MOV P1
30  JOVRD M_NJOVRD   'Set the default value
```

Reference

OVRD, SPD, Movement commands, Robot status variable
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

JRC (Joint Roll Change)

Function:

Robot arm's J6 axis: The current position is rewritten to a position obtained by adding ± 360 degrees to the current joint position of the J6 axis.

User-defined axis (additional axis, user-defined mechanism): The current position is rewritten to a position obtained by adding or subtracting a value designated in the parameters to the current joint position of the designated axis. Either a joint axis or linear axis can be used.
The origin can also be reset at the current position.

Format

```
JRC <[+]1/-1/0>[,<Axis No.>] ...
```

Terminology

<+1>	The current joint angle of the designated axis is incremented by the amount designated in parameter JRCQTT. (The sign can be omitted) For the robot arm's J6 axis, the value is fixed to 360 degrees.
<-1>	The current joint angle of the designated axis is decremented by the amount designated in parameter JRCQTT. For the robot arm's J6 axis, the value is fixed to 360 degrees.
<0>	The origin for the designated axis is reset at the value designated in parameter JRCORG. This can be used only for the user-defined axis.
<Axis No.>	The target axis is specified with the number. The default axis is J6 of robot arm.

Related parameters

JRCEXE	Set whether to enable/disable the JRC execution Execution disabled = 0 (default value) / execution enabled = 1
JRCQTT	Designate the amount to move (1 deg./1mm unit) when incrementing or decrementing with the JRC command. in additional axis or user-defined mechanism. The value is fixed to 360 degrees for the JRC target axis on the robot arm
JRCORG	Designate the origin for executing JRC 0. in additional axis or user-defined mechanism.

Explanation

- The current joint angle of the designated axis is incremented/decremented by the designated amount with the JRC 1/-1 command.
  The origin for the designated axis is reset with the JRC 0 command.
  The joint angle will change, but the robot will not move.
- When using this command, change the movement range of the target axis beforehand so that it does not leave the movement range when the command is executed. The range can be changed by changing the - side and + side value of the corresponding axis in the joint movement range parameter "MEJAR". Set the movement range for the rotating axis of -2340 deg. to 2340 deg.
- If the designated axis is omitted, the priority axis will be the target. The priority axis is the rotating axis (J6 axis) at the end robot.
- If the designated axis is omitted when a priority axis does not exist (robot incapable of JRC), or id the designated axis is not target for JRC, an error will occur when the command is executed.
- If the origin is not set, an error will occur when the command is executed.
- The robot is stopped while the JRC command is executed. Even if CNT is validated, the interpolation connection will not be continuous when this command is executed.
- The following parameter must be set before using the JRC command.
  - Set JRCEXE to 1. (JRC execution enabled)
  - Change the movement range of the target axis with MEJAR
  - Set the position change amount during the JRC 1/-1 execution with JRCQTT. (Only for the additional axis or user-defined mechanism)
  - Set the origin position for executing JRC 0 with JRCORG. (Only for the additional axis or user-defined mechanism)
- When parameter JRCEXE is set to 0, no process will take place even if JRc command is executed.

- If the movement amount designated with parameter JRCQTT is not within the pulse data 0 to MAX., an error will occur during the initialization. Here, MAX is 2^(Number of encoder bits + 15) - 1. For example, with a 13-bit encoder (8192 pulses), this will be MAX. = 2^(13 15) - 1 = 0x0fffffff, and for a 14-bit encoder (16384 pulse), this will be MAX. = 2^(14 + 15) - 1 = 0x1fffffff.
  The movement amount to pulse data conversion is as follows:
  For rotating axis:
  Pulse data = movement amount (deg.)/360 * gear ratio denominator/gear ratio numerator * Number of encoder pulses
  For linear axis:
  Pulse data = movement amount (mm)/360 * gear ratio denominator/gear ratio numerator * Number of encoder pulses
- The origin data will change when JRC is executed, so the default origin data will be unusable. If the controller needs to be initialized due to a version upgrade, etc., the parameters must be backed up beforehand in the original state.
- Step return operation is not possible with the JRC command.

Reference Program

```
10  Mov P1      'Moves to P1
20  JRC 1       'Increases J6 axis by 360°
30  MOV P1      'Moves to P1
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

* (Label)

Function:

This indicates the jump site.

Format

* <Label Name>

Terminology

<Label Name>	Describe a character string that starts with an alphabetic character. Up to 8 characters can be used. (Up to 9 characters including *.)

Explanation

- An error will not occur even if this is not referred to during the program.
- If the same label is defined several times in the same program, an error will occur at the execution.

Reference Program

```
100 * SUB1
```

Reference

GOTO, GOSUB, Label

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

LOADSET (Load Set)

Function:

Designates the hand and workpiece conditions for carrying out optimum acceleration/deceleration.

Format

LOADSET <Hand condition No.>, <Workpiece condition No.>

Terminology

<Hand condition No.>	Designate the hand condition (HNDDAT 1 to 8) No. for which the weight and size are designated.
<Workpiece condition No.>	Designate the workpiece condition (WRKDAT 1 to 8) No. for which the weight and size are designated.

Explanation

- Set the hand conditions and workpiece conditions used for optimum acceleration / deceleration. This is used when setting the optimum acceleration / deceleration for workpiece types having different weights.
- The maximum load is set for the hand when the program execution starts.
- Set the weight, size (X, Y, Z) and center of gravity position (X, Y, Z) as the hand conditions in parameter (HNDDAT 1 to 8).
- Set the weight, size (X, Y, Z) and center of gravity position (X, Y, Z) as the workpiece conditions in parameter (WRKDAT 1 to 8).
- The hand conditions and workpiece conditions changed when this command is executed are reset to the system default value when the program is reset and when the END statement is executed.
- As the system default values, the hand conditions are set to the rated load, and the workpiece conditions are set to none (0kg).
- Refer to "OADL (Optimal Acceleration)" on page 130 for details on the optimum acceleration / deceleration.

Reference Program

```
5  OADL ON
10 LOADSET 1,1   'Hand 1 and workpiece 1 conditions
20 MOV P1
30 MOV P2
40 LOADSET 1,2   'Hand 1 and workpiece 2 conditions
50 MOV P1
60 MOV P2
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MOV (Move)

Function:

Using joint interpolation operation, moves from the current position to the destination position.

Format

MOV <Target Position> [,<Close Distance>][Type <Constants_1>,<Constants_2>] [<Appended conditions>]

Terminology

<Movement Target Position>	This is the final position for interpolation operation. Describe a movement position statement.
<Close Distance>	If this value is designated, the actual movement target position will be a position separated by the designated distance in the tool coordinate system Z axis direction (+/- direction).
<Constants_1>	1/0: Detour/short cut.The default Value is 1(detour).
<Constants_2>	Invalid (Specify 0).
<Appended conditions>	The WTH and WTHIF statements can be used.

Explanation

- The joint angle differences of each axis are evenly interpolated at the starting point and endpoint positions. This means that the path of the tip cannot be guaranteed.
- By using the WTH and WTHIF statement, the signal output timing and motion can be synchronized.

- The numeric constant 1 for the TYPE designates the posture interpolation amount.
- Detour refers to the operating exactly according to the teaching posture. Short cut operation may take place depending on the teaching posture.
- Short cut operation refers to posture interpolation between the start point and end point in the direction with less motion.
- The detour/short cut designation is significant when the posture axis has a motion range of 180 deg. or more.
- Even if short cut is designated, if the target position is outside the motion range, the axis may move with the detour in the reverse direction.
- The TYPE numeric constant 2 setting is insignificant for joint interpolation.
- This instruction cannot be used in a constantly executed program.

Reference Program

```
10 MOV P1 TYPE 1,0
20 MOV J1
30 MOV (PLT 1,10),100.0 WTH M_OUT(17)=1
40 MOV P4+P5,50.0 TYPE 0.0 WTHIF M_IN(18)=1,M_OUT(20)=1
```

Reference

Movement commands, WTH, WTHIF, FINE, CNT
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MVA (Move Arch)

Function:

This instruction moves the robot from the current position to the target position with an arch movement (arch interpolation).

Format

This function is available for controller software version G2 or later

```
MVA <Target Position>[,<Arch Number>]
```

Terminology

<Target Position>	Final position of interpolation movement. This position may be specified using a position type variable and constant, or a joint variable.
<Arch Number>	A number defined by the DEF ARCH instruction (1 to 4). If the argument is omitted, 1 is set as the default value.

Explanation

- The robot moves upward along the Z-axis direction from the current position, then moves to a position above the target position, and finally moves downward, reaching the target position. This so-called arch motion movement is performed with one instruction.
- If the MVA instruction is executed without the DEF ARCH instruction, the robot moves with the arch shape configuration set in the parameters. Refer to DEF ARCH for a detailed description about the parameters.
- The interpolation form, type and other items are also defined by the DEF ARCH instruction.
- This instruction cannot be used in a constantly executed program.
- If paused during execution of a MVA instruction and restarted after jog feed, the robot returns to the interrupted position and restarts the MVA instruction.

Reference Program

```
10  DEF ARCH 1,5,5,20,20            'Defines the arch shape
                                    configuration
20  OVRD 100,20,20                  'Specifies override
30  ACCEL 100,100,50,50,50,50       'Specifies acceleration/
                                    deceleration rate.
40  MVA P1,1                        'Performs the arch motion movement
                                    according to the defined in line
                                    10
40  MVA P2,2                        'Moves the robot according to the
                                    default values registers
```

Reference

DEF ARCH (Define Arche), ACCEL (Accelerate), OVRD (Override), Movement commands
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MVC (Move C)

Function:

Carries out 3D circular interpolation in the order of start point, transit point 1, transit point 2 and start point.

Format

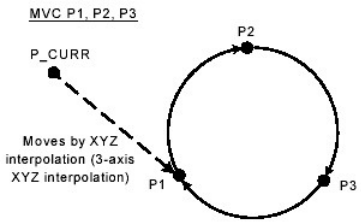
```
MVC <Start point>, <Transit point 1>, <Transit point 2> [<Additional condition>]
```

Terminology

<Start point>	The start point and end point for a circle. Describe a position operation expression or joint operation expression.
<Transit point 1>	Transit point 1 for a circular arc. Describe a position operation expression or joint operation expression.
<Transit point 2>	Transit point 2 for a circular arc. Describe a position operation expression or joint operation expression.
<Additional condition>	Describe a WTH conjunction or a WTHIF conjunction.

Explanation

- In circular interpolation motion, a circle is formed with the 3 given points, and the circumference is moved. (360 degrees)
- The posture during circular interpolation does not change.
- If the current position does not agree with the start point, the robot will automatically move to start point by linear interpolation.
- Once the execution has been interrupted and once jog operations have been restarted, it will start moving in relation to the position where it was interrupted, by linear interpolation.
- This instruction cannot be used in a constantly executed program.



Reference Program

```
10 MVC P1,P2,P3
20 MVC P1,J2,P3
30 MVC P1,P2,P3 WTH M_OUT(17)=1
40 MVC P3,(PLT 1,5),P4 WTHIF M_IN(20)=1,M_OUT(21)=1
```

Reference

Movement commands, WTH, WTHIF, FINE, CNT
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MVR (Move R)

Function:

Carries out 3-dimensional circular interpolation movement from the start point to the end point via transit points.

Format

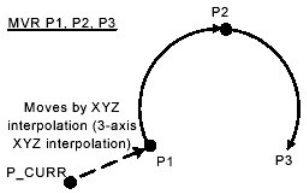
```
MVR <Start point>, <Transit point>, <End point> [ TYPE <Constants_1>,<Constants_2>]
[<Appended Condition>]
```

Terminology

<Start Point>	Start point for the circular arc. Describe a position operation expression or joint operation expression.
<Transit Point>	Transit point for the circular arc. Describe a position operation expression or joint operation expression.
<End Point>	End point for the circular arc. Describe a position operation expression or joint operation expression.
<Constants_1>	Detour/short cut = 1/0.The default value is 0.
<Constants_2>	3-axis XYZ/Equivalent rotation = 1/0. The default value is 0.
<Appended Condition>	The WTH and WTHIF conjunctions.

Explanation

- In circular interpolation motion, a circle is formed with three given points, and robot moves along the circumference.
- The posture is interpolation from the start point to the end point; the transit point posture has no effect.
- If the current position and start point do not match, the robot will automatically move with linear interpolation (3-axis orthogonal interpolation) to the start point.
- If the execution is stopped, the robot is moved with jog, and then the execution is restarted, the robot will move with joint interpolation to the position where execution was stopped, and then the remaining circular interpolation will restart.
- If the start point and end point structure flags differ for an interpolation method other than 3-axis orthogonal interpolation, an error will occur at the execution.
- Of the three designated points, if any points coincide with the other, or if three points are on a straight line, linear interpolation will take place from the start point to the end point. An error will not occur.
- If 3-axis orthogonal is designated for the numeric constant 2, the numeric constant 1 will be invalidated, and the robot will move with the taught posture.
- Numeric constant 2 designates the posture interpolation type. 3-axis orthogonal is used when carrying out interpolation on the (X, Y, Z, J4, J5, J6) coordinate system, and the robot is to move near a particular point.
- This instruction cannot be used in a constantly executed program.



Reference Program

```
10 MVR P1,P2,P3
20 MVR P1,J2,P3
30 MVR P1,P2,P3 WTH M_OUT(17)=1
40 MVR P3,(PLT 1,5),P4 WTHIF M_IN(20)=1,M_OUT(21)=1
```

Reference

Movement commands WTH, WTHIF, FINE, CNT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MVR2 (Move R2)

Function:

Carries out 3-dimensional circular interpolation motion from the start point to the end point on the arc composed of the start point, end point, and reference points.The direction of movement is in a direction that does not pass through the reference points.

Format

```
MVR2 <Start point>, <End point>, <Reference point> [ TYPE  <Constants_1>,
```

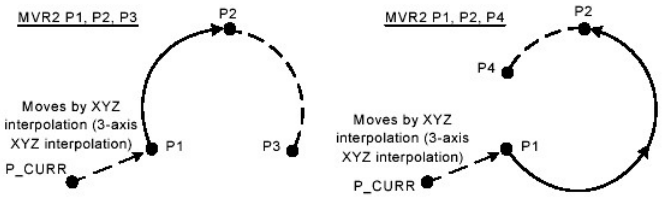
<Constants_2>] [<Appended Condition>]

Terminology

<Start Point>	Start point for the circular arc. Describe a position operation expression or joint operation expression.
<End Point>	End point for the circular arc. Describe a position operation expression or joint operation expression.
<Reference Point>	Reference point for a circular arc. Describe a position operation expression or joint operation expression.
<Constants_1>	Detour/short cut = 1/0, The default value is 0.
<Constants_2>	3-axis XYZ/Equivalent rotation = 1/0, The default value is 0.
<Appended Condition>	The WTH and WTHIF statements can be used.

Explanation

- In circular interpolation motion, a circle is formed with three given points, and robot moves along the circumference.
- The posture is interpolation from the start point to the end point; the transit point posture has no effect.
- If the current position and start point do not match, the robot will automatically move with linear interpolation (3-axis orthogonal interpolation) to the start point.
- If the execution is stopped, the robot is moved with jog, and then the execution is restarted, the robot will move with joint interpolation to the position where execution was stopped, and then the remaining circular interpolation will restart.
- The direction of movement is in a direction that does not pass through the reference points.
- If the start point and end point structure flags differ for an interpolation method other than 3-axis orthogonal interpolation, an error will occur at the execution.
- Of the three designated points, if any points coincide with the other, or if three points are on a straight line, linear interpolation will take place from the start point to the end point. An error will not occur.
- If 3-axis orthogonal is designated for the numeric constant 2, the numeric constant 1 will be invalidated, and the robot will move with the taught posture.
- Numeric constant 2 designates the posture interpolation type. 3-axis orthogonal is used when carrying out interpolation on the (X, Y, Z, J4, J5, J6) coordinate system, and the robot is to move near a particular point.



Reference Program

```
10 MVR2 P1,P2,P3
20 MVR2 P1,J2,P3
30 MVR2 P1,P2,P3 WTH M_OUT(17)=1
40 MVR2 P3,(PLT 1,5),P4 WTHIF M_IN(20)=1,M_OUT(21)=1
```

Reference

Movement commands, WTH, WTHIF, FINE, CNT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MVR3 (Move R3)

Function:

Carries out 3-dimensional circular interpolation movement from the start point to the end point on the arc composed of the center point, start point and end point.

Format

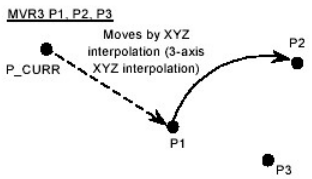
MVR3 <Start point>, <End point>, <Center point> ,[TYPE <Constants_1>, <Constants_2>] [<Appended Condition>]

Terminology

<Start Point>	Start point for the arc. Describe a position operation expression or joint operation expression.
<End Point>	End point for the circular arc. Describe a position operation expression or joint operation expression.
<Center Point>	Center point for the arc. Describe a position operation expression or joint operation expression.
<Constants_1>	Detour/short cut = 1/0, The default value is 0.
<Constants_2>	3-axis XYZ/Equivalent rotation = 1/0, The default value is 0.
<Appended Condition>	The WTH and WTHIF conjunctions.

Explanation

- In circular interpolation motion, a circle is formed with three given points, and robot moves along the circumference.
- The posture is interpolation from the start point to the end point; the transit point posture has no effect.
- If the current position and start point do not match, the robot will automatically move with linear interpolation (3-axis orthogonal interpolation) to the start point.
- If the execution is stopped, the robot is moved with jog, and then the execution is restarted, the robot will move with joint interpolation to the position where execution was stopped, and then the remaining circular interpolation will restart.
- If the start point and end point structure flags differ for an interpolation method other than 3-axis orthogonal interpolation, an error will occur at the execution.
- Of the three designated points, if any points coincide with the other, or if three points are on a straight line, linear interpolation will take place from the start point to the end point. An error will not occur.
- If 3-axis orthogonal is designated for the numeric constant 2, the numeric constant 1 will be invalidated, and the robot will move with the taught posture.
- The fan angle from the start point to the end point is 0 < fan angle < 180 deg.
- Designate the positions so that the difference from the center point to the end point and the center point to the distance is within 0.01 mm.
- If the three points are on the same line, or if the start point and center point, or end point and center point are the same, an error will occur.
- If the start point and end point are the same, or if the three points are the same, the robot will move with linear interpolation from the start point to the end point.
- This instruction cannot be used in a constantly executed program.



Reference Program

```
10 MVR3 P1,P2,P3
20 MVR3 P1,J2,P3
30 MVR3 P1,P2,P3 WTH M_OUT(17)=1
40 MVR3 P3,(PLT 1,5),P4 WTHIF M_IN(20)=1,M_OUT(21)=1
```

Reference

Movement commands, WTH, WTHIF, FINE, CNT
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MVS (Move S)

Function:

Carries out linear interpolation movement from the current position to the movement target position.

Format 1

MVS <Movement Target Position> [, <Close distance>] [<Interpolation Type>] [<Appended Condition>]

Format 2

MVS <Separation Distance> [<Interpolation Type>]

Terminology

<Movement Target Position>	The final position for the linear interpolation. Describe a movement position statement
<Close Distance>	If this value is designated, the actual movement target position will be a position separated by the designated distance in the tool coordinate system Z axis direction {+/- direction).
<Interpolation Type>	Designate the interpolation type. TYPE <Numeric constant 1>, <Numeric constant 2> Numeric constant 1 ... Detour/short cut = 1/0 Numeric constant 2 ... 3-axis orthogonal/Equivalent rotation = 1/0 The default value is 0, 0 {detour, equivalent rotation).
<Appended conditions>	The WTH and WTHIF statements can be used.
<Separation Distance>	If this value is designated, the robot will move from the current position to a position separated by the designated amount in the tool coordinate system Z axis direction. When a positive value is set, the movement will take place in the direction that the mechanical interface {flange surface) is facing, and when a negative value is set. the movement will take place in the opposite direction {hand retract).

Explanation

- Linear interpolation motion is a type of movement where the robot moves from its current position to the movement target position so that the locus of the control points is in a straight line.
- The posture is interpolation from the start point to the end point.
- If the execution is stopped, the robot is moved with jog, and then the execution is restarted, the robot will move with joint interpolation to the position where execution was stopped, and then will move to the target position.
- Note that when resuming a command for the target position, such as MVS, 100, the robot will not move the remaining distance.
- If the start point and end point structure flags differ for an interpolation method other than 3-axis orthogonal interpolation, an error will occur at the execution.
- Of the three designated points, if any points coincide with the other, or if three points are on a straight line, linear interpolation will take place from the start point to the end point. An error will not occur.
- If 3-axis orthogonal is designated for the numeric constant 2, the numeric constant 1 will be invalidated, and the robot will move with the taught posture.

Reference Program

```
10 MVS PLT1,10,100.0 WTH M_OUT(17)=1
20 MVS P4+P5,50.0 WTHIF M_IN(18)=1,M_OUT(20)=1
30 MVS ,50
```

Reference

Movement commands, WTH, WTHIF, FINE, CNT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

OADL (Optimal Acceleration)

Function:

Automatically sets the optimum acceleration/deceleration time according to the designated robot hand's load state.

Format

OADL <Switch>

Terminology

<Switch>	Set whether to turn optimization ON or OFF.

Explanation

- The robot moves with the optimum acceleration/deceleration according to the hand conditions and workpiece conditions designated with the LOADSET command.
- The workpiece grasp/not grasp for when the hand is opened or closed is set with parameter HNDHOLD 1 to 8.
- The ACCEL command is invalid when OADL is ON.
- OADL is set to OFF as the default.
- Once OADL is ON, it is valid until OADL OFF is executed or until the program END is executed.

Reference Program

```
10 OADL ON
20 MOV P1         'Move with maximum load.
30 LOADSET 1,1    'Set hand 1 and workpiece 1.
40 MOV P2         'Move with hand 1 + workpiece 1 load.
50 HOPEN 1
60 MOV P3         'Move with hand 1 load.
70 HCLOSE 1
70 HOPEN P4       'Move with hand 1 + workpiece 1 load.
80 LOADSET 1,2    'Set hand 1 and workpiece 2.
90 MOV P5         'Move with hand 1 load.
100 HCLOSE 1
110 MOV P6        'Move with hand 1 + workpiece 2 load
```

*When parameter HNDHOLD1 is set to 0, 1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ON COM GOSUB (On Communication Go Subroutine)

Function:

Defines the starting line of a branching subroutine when an interrupt is generated from a designated communication line.

Format

ON COM [(<File No.>)] GOSUB <Call destination>

Terminology

<File No.>	Describe a number between 1 and 3 assigned to the communication line.
<Call Destination>	Describe the line No. or label name.

Explanation

- If the file No. is omitted, 1 will be used as the file No.
- The file No. with the smallest No. have the order of priority for the interrupt.
- If a communication interrupt is generated while the robot is moving. the robot will stop.
- It is possible to use COM STOP to stop the interrupt. and prevent the robot from stopping.
- In the default state, the interrupt is disabled. Execute the COM ON command after this command to validate the interrupt.

Reference Program

```
10 OPEN "COM1:" AS #1
20 OPEN "COM3:" AS #2
30 ON COM GOSUB 300          'If an interrupt is generated from the file
                             No.1 communication line (COM1:), carry out
                             line 300 process.
40 ON COM(2) GOSUB *RECV     'If an interrupt is generated from the
                             communication line No.1 communication line,
                             carry out the label RECV process.
50 COM(1) ON                 'Enable interrupt from file No.1 communication
                             line.
60 COM(2) ON                 'Enable interrupt from file No.2 communication
                             line.
70 COM OFF                   'Disable interrupt from file No.1
                             communication line.
80 COM(2) OFF                'Disable interrupt from file No.2
                             communication line.
```

Reference

COM ON, COM OFF, COM STOP, OPEN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ON GOSUB (On Gosub)

Function:

Calls up the subroutine at the line No. or label corresponding to the value.

Format

ON <Terminology> GOSUB [<Expression>] [, [<Call destination>]] ...

Terminology

<Terminology>	Designate the line No. or label on the nth line to branch to with a numeric operation expression.
<Call Destination>	Describe the line No. or the label No.

Explanation

- The value of <Expression> determines which line No. or label subroutine to call. For example, if the value of <Expression> is 2, the line No. or label described for the second value is called.
- If the value of <expression> is larger than the number of <destinations called up>, the program control jumps to the next line. For example, the program control jumps to the next line if the value of <expression> is 5 and there are only three <destinations called up>.
- When a line No. or label that is called up does not exist, or when there are two definitions, an error will occur.
- Make sure to return from a subroutine using the RETURN instruction. An error occurs if the GOTO instruction is used to return, because the free memory available for control structure (stack memory) decreases and eventually becomes insufficient.

Value of <Expression>	Process <Control>
Real number	Value is converted to an Integer by rounding it off, and then branching is executed.
When 0, or when the value exceeds the number of line Nos. or labels	Control proceeds the next line
Negative number or 32767 is exceeded	Execution Error
Line number or label is omitted	Execution Error

Reference Program

```
100 ON M1 GOSUB 1000, *SUBPR   'Call line 1000 when numeric variable MI
                               value is 1, and the label *SUBPR line when
                               2.
```

Reference

RETURN, END, ON GOTO, GOSUB, IF THEN ELSE

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ON GOTO (On go to)

Function:

Branches to the line with the line No. or label that corresponds to a designated value.

Format

ON <Expression> GOTO [<Branch destination>] [, [<Branch destination>]] ...

Terminology

<Expression>	Designate the line No. or label on the nth line to branch to with a numeric operation expression.
<Call Destination>	Describe the line No. or the label No.

Explanation

- This is the GOTO version of ON GOSUB.
- When a line No. or label that is called up does not exist, or when there are two definitions, an error will occur.

Value of <Expression>	Process <Control>
Real number	Value is converted to an Integer by rounding it off, and then branching is executed.
When 0, or when the value exceeds the number of line Nos. or labels	Control proceeds the next line
Negative number or 32767 is exceeded	Execution Error
Line number or label is omitted	Execution Error

Reference Program

```
100 ON M1 GOTO 1000,*SUBPR   'Branch to line 1000 when numeric variable
                             M1 value is 1; and the label *SUBPR line
                             when 2.
```

Reference

GOTO, ON GOSUB, IF THEN ELSE

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

OPEN (Open)

Function:

Open a file or communication line

Format

OPEN "<File descriptor>" [FOR <Mode>] AS [#] <File No.>

Terminology

<File Descriptor>
Describe a file name (including communication lines).
To open a communication line, set "<Communication Line File Name>:"
When not using a communications line, set "<File Name>"

<Mode>
Designate the method to access a file.
Omitted = random mode
This can be omitted when using a communication line.
INPUT = input mode
Inputs from an existing file.
OUTPUT = output mode (new file)
Creates a new file and outputs it there.
APPEND = Output mode (existing file)
Appends output to the end of an existing file.

File Descriptor	File name	Access method
File	Describe with 16 characters or less	INPUT, PRINT, APPEND
Communication line file	COM 1:, COM 2:, COM 3:	Omitted = random mode only

Describe a constant within the following range.

<File No.>	To interrupt from communication line: 1 to 3. To not interrupt from communication line: 1 to 8.

Explanation

- Designate the designated file No. of the file described in <File Descriptor>, and open. Use this file No. when reading from or writing to the file.
- A communication line is handled as a file.

Communicationfile	Hardware device name
COM 1	Standard RS232-C
COM 2	(Reserved)
COM 3	(Reserved)

Reference Program

Communication line

```
10 OPEN "COM1:" AS #1    'Open standard RS-232-C line as file No. 1.
20 MOV P_01              'Move to position P_01
30 PRINT #1,P_CURR       'Output current position to external source.
                         "(100.00,200.00,300.00,400.00)(7.)" format
40 INPUT #1,M1,M2,M3     'Send from external source with
                         "101.00,202.00,303.00" ASCII format.
50 P_01.X=M1
60 P_01.Y=M2
70 P_01.C=M3             'Copy to global data.
80 CLOSE                 'Close all opened files.
90 END
```

File operation

```
10 OPEN "temp.txt" FOR APPEND AS #1 'Create the file "temp.txt" to the controller
                                    an write "abc"
20 PRINT #1, "abc"
30 CLOSE #1
```

Reference

INPUT, PRINT, COM ON, COM OFF, COM STOP, ON COM GOSUB

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

OVRD (Override)

Function:

This instruction specifies the speed of the robot movement as a value in the range from 1 to 100%. This is the override applied to the entire program.

Format

```
OVRD <Designated override>
```

This function is available for controller software version G2 or later.

```
OVRD <Designated override>[<Override when moving upward>] [,<Override when moving downward>]
```

Terminology

<Designated override>	Designate the override with a real number. Unit: [%] (Recommended range: 1 to 100.0) A numeric operation expression can also be described. If 0 or a value over 100 is set, an error will occur.
<Override when moving upward/downward>	Sets the override value when moving upward/downward by the arch motion instruction (MVA)

Explanation

- The OVRD command is valid regardless of the interpolation type.
- The actual override is as follows:
  During joint interpolation: (Operation panel (T/B) override setting value) x (Program override (OVRD command)) x (Joint override (JOVRD command)).
  During Linear interpolation: (Operation panel (T/B) override setting value) x (Program override (OVRD command)) x (Linear designated speed (SPD command)).
- The OVRD command changes only the program override. 100% is the maximum capacity of the robot. Normally, the system default value (M_NOVRD) is set to 100%. The designated override is the system default value until the OVRD command is executed in the program.
- Once the OVRD command has been executed, the designated override is applied until the next OVRD command is executed, the program END is executed or until the program is reset. The value will return to the default value when the END statement is executed or the program is reset.

Reference Program

```
10 OVRD 50
20 MOV P1
30 MVS P2
40 OVRD M_NOVRD              'Set the default value
```

Also possible with controller Software G2

```
10 OVRD 30,10,10             'Sets the override when moving upward/downward
20 MVA P3,3                  'moving by the arch motion instruction
```

Reference

JOVRD, SPD, Movement commands, Robot status variable

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

PLT (Pallet)

Function:

Calculates the position of grid in the pallet.

Format

PLT <Pallet No.> <Numeric operation expression>

Terminology

<Pallet No.>	Select a pallet No. between 1 and 8 that has already been defined with a DEF PLT command.
<Numeric Operation Expression>	Sets the grid No. for the selected pallet.

Explanation

- The position of grid of a pallet defined by the DEF PLT statement is operated.
- The pallet Nos. are from 1 to 8, and up to 8 can be defined at once.
- Note that the position of the grid may vary because of the designated direction in the pallet definition.
- If a grid No. is designated that exceeds the largest grid No. defined in the pallet definition statement, an error will occur during execution.
- When using the pallet grid point as the target position of the movement command, an error will occur if the point is not enclosed in parentheses as shown above.

Reference Program

```
100   DEF PLT 1,P10,P11,P12,P13,8,8,1     'The definition of the four-
                                          point pallet. (P1,P2,P3,P4)
110
120   M1 = 1                              'Initialize the counter M1.
130   *LOOP                               'Setzen eines Labels
140   MOV PICK,50                         'Moves 50 mm above the work
                                          unload position.
150   OVRD 50
160   MVS PICK
```

```
170   HCLOSE 1                        'Close the hand
180   DLY 0.5                         'Wait for the hand to close
                                      securely (0.5 sec.)
190   OVRD 100
200   MVS,50                          'Moves 50 mm above the
                                      current position.
210   PLACE=PLT 1,M1                  'Calculates the M1th position
220   MOV PLACE,50                    'Moves 50 mm above the pallet
                                      top mount position.
230   OVRD 50
240   MVS PLACE
250   HOPEN 1                         'Open the hand.
260   DLY 0.5
270   OVRD 100
280   MVS,50                          'Moves 50 mm above the
                                      current position.
290   M1=M1+1                         'Add the counter.
300   IF M1<=12 THEN *LOOP            'If the counter is within the
                                      limits, repeats from *LOOP..
310   MOV PICK,50
320   END
```

Reference

DEF PLT, Movement Command

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

PREC (Precision)

Function:

This instruction is used to improve the motion path tracking. It switches between enabling and disabling the high accuracy mode. This is only available for certain types of robots (RV-1A/2AJ, RV-4A/5AJ/3AL/4AJL, and RV-20A).

Format

This function is available for controller software version D1 or later.

```
PREC <Switch>
```

Terminology

<Switch>	ON : When enabling the high accuracy mode. OFF : When disabling the high accuracy mode.

Explanation

- The high accuracy mode is enabled using the PREC ON instruction if it is desired to perform interpolation movement with increased path accuracy.
- When this instruction is used, the path accuracy is improved but the program execution time (tact time) may become longer because the acceleration/deceleration times are changed internally.
- The enabling/disabling of the high accuracy mode is activated from the first interpolation instruction after the execution of this instruction.
- The high accuracy mode is disabled if the PREC OFF or END instruction is executed, or a program reset operation is performed.
- The high accuracy mode is disabled immediately after turning the power on.

- The high accuracy mode is always disabled in jog movement.

Reference Program

```
10 PREC ON                    ' Enables the high accuracy mode
20 MVS P1
30 MVS P2
40 PREC OFF                   ' Disables the high accuracy mode
50 MOV P1
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

PRINT (Print)

Function:

Outputs data into a file (including communication lines). All data uses the ASCII format.

Format

PRINT # <File No.> [, [<Expression>;] ... [<Expression> [;]]]

Terminology

<File No.>	Described with numbers 1 to 8. Corresponds to the control No. assigned by the OPEN command.
<Expression>	Describes numeric operation expressions, position operation expressions and character string expressions.

Explanation

- If <Expression> is not described, then a carriage return will be output.
- If <File No.> is omitted, the default value of 1 will be used.
- Output format of data (reference)
  The output space for the value for <Expression> and for the character string is in units of 10 characters. When outputting multiple values, use a comma between each <Expression> as a delimiter.
  If a semicolon (;) is used at the head of each space unit, it will output after the item that was last displayed.
- The carriage will always be returned after the OPEN statement.
  Example
  10 M1=123.5
  20 P1=(130.5,-117.2,55.1,16.2,0.0,0.0)(1,0)
  - 30 PRINT# 1,"OUTPUT TEST",M1,P1 is described,
    OUTPUT TEST 123.5 (130.5,-117.2,55.1,16.2,0.0)(1,0) is output.
  - 30 PRINT# 1,"OUTPUT TEST";M1' P1 is described,
    OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2,0.0)(1,0) is output.
    If a comma or semicolon is inserted after a <Expression>, the carriage return will not be issued, and instead, printing will continue on the same line.
  - 30 PRINT# 1,"OUTPUT TEST",
    40 PRINT# 1,M1,
    50 PRINT# 1,P1 is described,
    OUTPUT TEST 123.5(130.5,-117.2,55.1,16.2)(1,0) is output.

Reference Program

```
10 OPEN "COM1:" AS #1                'Open .standard RS-232-C line as
                                     file No. 1.20 MOV P_01
20 MDATA=150                         'Substitute 150 for the numeric
                                     variable MDATA.
30 PRINT #1,"***PRINT TEST***"       'Outputs the character string
                                     "***PRINT TEST****".
40 PRINT #1                          'Issue a carriage return
50 PRINT #1, "MDATA=",MDATA          'Output the character string
                                     "MDATA" followed by the value of
                                     MDATA, (150).
60 PRINT #1                          'Issue a carriage return
40 PRINT #1,"**********************"  'Outputs the character string
                                     "**********************".
50 END                               'End the program.
```

The output result is shown below.

PRINT TEST

MDATA=150

****

Reference

OPEN, CLOSE ,INPUT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

PRIORITY (Priority)

Function:

In multitask program operation, multiple program lines are executed in sequence (one by one line according to the default setting). This instruction specifies the priority (number of lines executed in priority) when programs are executed in multitask operation.

Format

This function is available for controller software C2 or later.

PRIORITY <Number of executed lines>[,<Slot number>]

Terminology

<Number of executed lines>	Specify the number of lines executed at once . Use a numerical value from 1 to 31.
<Slot number>	1 to 32. If this argument is omitted, the current slot number is set.

Explanation

- Programs of other slots are not executed until the specified number of lines is executed. For example, as in the statement example above, if PRIORITY 3 is set for slot 1's program and PRIORITY 4 is set for slot 2's program, three lines of the slot 1 program are executed first, then four lines of the slot 2 program are executed. After-ward, this cycle is repeated.
- The default value is 1 for all the slots. In other words, the execution moves to the next slot every time one line has been executed.
- An error occurs if there is no program corresponding to the specified task slot.
- It is possible to change the priority even while the program of the specified task slot

Reference Program

Slot 1

```
 10      PRIORITY 3                  ' Sets the number of executed lines for the current slot
Slot 2

 10      PRIORITY 4                  ' Sets the number of executed lines for this slot to 4
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RELM (Release Mechanism)

Function:

Releases the mechanism resource.

Format

```
RELM
```

Explanation

- Releases the currently acquired mechanism resource.
- If an interrupt is applied while the mechanism is acquired and the program execution is stopped, the acquired mechanism resource will be automatically released.
- This instruction cannot be used in a constantly executed program.

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

REM (Remarks)

Function:

Uses the following character strings as comments.

Format

```
REM [<Comment>]
```

Terminology

<Comment>	Describe a user-selected character string. Descriptions can be made in the range of position lines.

Explanation

- REM can be abbreviated to be a single quotation mark (').
- This can also be after another command on the same line.

Reference Program

```
10  REM ***MAIN PROGRAM***
20  ' ***MAIN PROGRAM***
30  MOV P1                    ' Move to P1
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RESET ERR (Reset Error)

Function:

This instruction resets an error generated in the robot controller. It is not allowed to use this instruction in the initial status. If an error other than warnings occurs, normal programs other than constantly executed programs can-not be operated. This instruction is effective if used in constantly executed programs.

Format

This function is available for controller software version B1 or later

```
RESET ERR
```

Explanation

- This instruction is used in a program whose start condition is set to constant execution (ALWAYS) by the "SLT*" parameter when it is desired to reset system errors of the robot.
- It becomes enabled when the controller's power is turned on again after changing the value of the "ALWENA" parameter from 0 to 7.

Reference Program

Example of execution in a constantly executed program

```
10   IF M_ERR=1 THEN RESET ERR        ' Resets an error when an error occurs in the controller
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RETURN (Return)

Function:

When returning from a normal subroutine returns to the next line after the GOSUB.When returning from an interrupt processing subroutine, returns either to the line where the interrupt was generated, or to the next line.

Format

When returning from a normal subroutine:

```
RETURN
```

When returning from an interrupt processing subroutine:

```
RETURN <Return designation No.>
```

Terminology

<Return Designation No.>	Designate the line where control will return to after an interrupt has been generated and processed. 0 ... Return control to the line where the interrupt was generated. 1 ... Return control to the next line after the line where the interrupt was issued.

Explanation

- When there is a RETURN command in a normal subroutine with a return-to designation number, and when there is a RETURN command in an interrupt-processing subroutine with no return-to destination number, an error will occur.
- Writes the RETURN instruction at the end of the jump destination processing called up by the GOSUB instruction.
- An error occurs if the RETURN instruction is executed without being called by the GOSUB instruction.
- Always use the RETURN instruction to return from a subroutine when called by the GOSUB instruction. An error occurs if the GOTO instruction is used to return, because the free memory available for control structure (stack memory) decreases and eventually becomes insufficient.

Reference

GOSUB, ON GOSUB, ON COM GOSUB , DEF ACT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SELECT CASE (Select Case)

Function:

Executes one of multiple statement blocks according to the condition expression value.

Format

```
SELECT <Condition>
CASE <Expression>
[<Process>]
BREAK
CASE <Expression>
[<Process>]
BREAK
...
CASE <Expression>
[<Process>]
BREAK
DEFAULT
[<Process>]
BREAK
END SELECT
```

Terminology

<Condition>	Describe a numeric operation expression.
<Expression>	Describe a numeric operation expression. The type must be the same as the condition expression.
<Process>	Describe a command statement (excluding a branch condition statement or repeated control statement) provided with MELFA-BASICI V, a line No. or a label name.

Explanation

- If the condition matches one of the CASE items, the process will be executed until the next CASE, DEFAULT or ENDSELECT.
- If the case does not match with any of the CASE items but DEFAULT is described, that block will be executed.
- If there is no DEFAULT, the program will jump to the line after ENDSELECT without processing.
- The SELECT CASE and END SELECT statements must always correspond.
- If an END SELECT statement that does not correspond to SELECT CASE is executed, an execution error will occur. Note that if the END statement is executed midway, or if the last line of the program is executed, the program execution will stop at that line.
- Another SELECT CASE cannot be described in the SELECT CASE. However, WHILE WEND or FOR NEXT can be described.
- Use "CASE IS", when using the comparison operators (<, =, >, etc.) for the "<Expression>".

Reference Program

```
10  SELECT MCNT
20  M1=10           'This line is not executed
30  CASE IS <= 10   'MCNT <= 10
40  MOV P1
50  BREAK
60  CASE 11         'MCNT = 11
70  MOV P2
80  BREAK
90  CASE 12         'MCNT = 12
100 MOV P3
110 BREAK
120 CASE 13 TO 18   '13 <= MCNT <= 18
130 MOV P4
140 DEFAULT         'Other than the above
150 M_OUT(10)=1
160 BREAK
170 END SELECT
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SERVO (Servo)

Function:

Controls the ON and OFF of the servo.

Format

The usual program

```
SERVO <Switch>
```

The program of always (ALWAYS) execution

```
SERVO <Switch>[,<Mechanism No.>]
```

Terminology

<Switch>	ON: When turning the servo motor power on. OFF: When turning the servo motor power off.
<Mechanism No.>	This is valid only within the program of always execution. The range of the value is 1 to 3, and describe by constant

Explanation

- SERVO controls the servo power for all axes
- If additional axes are attached, the servo power supply for the additional axes is also affected.

Reference Program

```
10 SERVO ON                    'Servo ON
20 IF M_SVO<>0 THEN GOTO 20    'Wait for servo ON
30 SPD M_NSPD
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SKIP (Skip)

Function:

Transfers control of the program to the next line.

Format

```
SKIP
```

Explanation

- This command can be described with the WTH or with WTHIF statements. In this case, the execution of that line is interrupted, and control is automatically transferred to the next line. Execution of skip can be seen with the M_SKIPCQ information.

Reference Program

```
10 MOV P1 WTHIF M_IN(17)=0,SKIP          'If the input signal (M_IN
                                         (17)) turns ON while moving
                                         with joint interpolation to
                                         the position indicated with
                                         position variable P1, stop
                                         the robot interpolation
                                         motion, and stop execution
                                         of this command.
20  IF M_SKIPCQ=1 THEN HLT               ' Pauses the program if the
                                         execution is skipped.
```

Reference

HLT, WTH, WTHIF

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SPD (Speed)

Function:

Designates the speed for the robot's linear and circular movements.

Format

```
SPD <Designated speed>
```

Terminology

<Designated Speed>	Designate the speed as a real number. Unit: [mm/s]

Explanation

- The SPD command is valid only for the robot's linear and circular movements.
- The actual designated override is (Operation panel (T/B) override setting value) x (Program override (OVRDcommand)) x (Linear designated speed (SPD command)).
- The SPD command changes only the linear designated speed.
- When M_NSPD is designated for the designated speed, the robot will always move at the maximum possible speed, so the line speed will not be constant.
- The system default value is applied for the designated speed until the SPD command is executed in the program. Once the SPD command is executed, that designated speed is held until the next SPD command.
- The designated speed will return to the system default value when the program END statement is executed.

Reference Program

```
10 SPD 100
20 MVS P1
30 SPD M_NSPD   'Set the default value
```

Reference

ACCEL, CNT, OVRD, JOVRD, Movement command, Robot status variable

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

Substitute

Function:

The results of an operation are substituted in a variable or array variable.

Format 1

```
<Variable Name> = <Expression 1>
```

Format 2 (For pulse substitution)

```
<Variable Name> = <Expression 1> DLY <Expression 2>
```

Terminology

<Variable Name>	Describe the name of the variable where the value is to be substituted. (Refer to the syntax diagram for the types of variables.)
<Expression 1>	Substitution value. Describe an arithmetic operation expression.
<Expression 2>	Pulse timer. Describe an arithmetic operation expression.

Explanation

- When using this additionally for the pulse output, the pulse will be executed in parallel with the execution of the commands on the following lines.
- After the designated time has passed, the state will return to that before the pulse

output.
- If the END command or program's last line is executed during the designated time, or if the program execution is stopped due to an emergency stop, etc., the output state will be held. (The output state during the interrupt can be selected with the system parameters if it is to be return to the state before the pulse output in the same manner as after the designated time has passed.)

Reference Program

```
10 P100=P1+P2*2
20 M_OUT(17)=1 DLY 10.0
```

Reference

DLY, Syntax diagram

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

TOOL (Tool)

Function:

Designates the tool conversion data.

Format

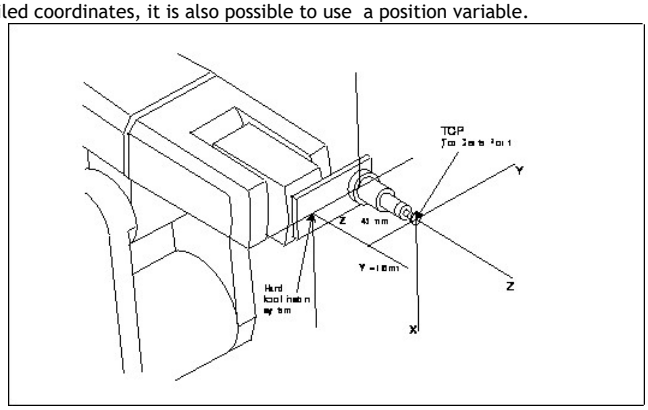
TOOL <Tool conversion data>

Terminology

<Tool conversation data>	Specifies the tool conversation data using the position operation expression.

Explanation

- The TOOL instruction is used to specify the control points at the tip of each hand in a system using double hands. If both hands are of the same type, the control point should be set by the "MEXTL" parameter instead of by the TOOL instruction.
- The system default value (P_NTOOL) is applied until the TOOL command is executed. Once the TOOL command is executed, the designated tool conversion data is applied until the next TOOL command is executed. This is operated with 6-axis three-dimension regardless of the mechanism structure.
- The tool conversion data changed with the TOOL command is saved in parameter MEXTL, and is saved even after the controller power is turned OFF.
- If different tool conversion data are used at teaching and automatic operation, the robot may move to an unexpected position. Make sure that the settings at operation and teaching match. The valid axis element of tool conversion data is different depending on the type of robot.
- Instead of detailed coordinates, it is also possible to use a position variable.



Reference Program

```
10 TOOL (100,100,0,0,0,0)
20 MVS P1
30 TOOL P_NTOOL
```

Reference

P_TOOL, P_NTOOL

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

TORQ (Torque)

Function:

Designates the torque limit for each axis.

Format

TOOL <Axis no.>, <Numeric value>

Terminology

<Axis no.>	Designate the axis No., with a numeric constant. (1 to 8)
<Numeric value>	Designate the limit of the force generated from the axis as a percentage. (1 to 100)

Explanation

- The torque limit value of the designated axis is limited. The movement is limited so that a torque exceeding the designation is not applied. Designate a percentage in respect to the standard torque limit.
- This is used when the grasping force is to be limited for the servo hand, etc.
- The available rate of torque limitation is changed by robot type.

Reference Program

```
10 TORQ 4,80
20 MVS P1
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

WAIT (Wait)

Function:

Waits for the variable to reach the designated value.

Format

WAIT <Numeric value> = <Numeric constant>

Terminology

<Numeric value>	Designate an M variable.
<Numeric constant>	Designate a numeric constant.

Explanation

- This command is used as the interlock during signal input wait and during multitask execution.
- The WAIT instruction allows the program control to continue to the next line once the specified condition is met.
- In case the WAIT instruction is executed in several tasks at one time in the multitask execution status, the processing time (tact time) may become longer and affect the system. In such cases, use the IF-THEN instruction instead of the WAIT instruction.

Reference Program

```
10 WAIT M_IN(1) = 1    'Signal state
20 WAIT M_IN(3) = 0
30 WAIT M_RUN(2) = 1   'Task slot state
40 WAIT M_01 = 100     'Variable state
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

WHILE WEND (While End)

Function:

The program between the WHILE statement and WEND statement is repeated until the loop conditions are satisfied.

Format

```
WHILE<Loop condition>
.
.
WEND
```

Terminology

<Loop Condition>	Describe a numeric operation expression. (Refer to the syntax diagram)

Explanation

- The program between the WHILE statement and WEND statement is repeated.
- If the result of <Expression> is true (not 0), then the control moves to the line following the WHILE statement and the process is repeated.
- If the result of <Expression> is false (is 0), then the control moves to the line following the WEND statement.
- If a GOTO instruction forces the program to jump out from between a WHILE statement and a WEND state-ment, the free memory available for control structure (stack memory) decreases. Thus, if a program is executed continuously, an error will eventually occur. Write a program in such a way that the loop exits when the condition of the WHILE statement is met.

Reference Program

```
20 WHILE (M1>=-5) AND (M1<=5)   'Repeat the process while the numeric
                                variable M1 value is between -5 and +5,
                                and transfer control to line after WEND
                                statement if range is exceeded.
30 M1=-(M1+1)                   'Add 1 to M1, and reverse the sign
40 PRINT# 1, M1                 'Output the M1 value
50 WEND                         'Return to the WHILE statement (line 20)
60 END                          'End the program
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

WTH (With)

Function:

A process is added to the interpolation motion.

Format

```
<Expression> WTH <Process>
```

Terminology

<Expression>	For example a movement command like 'MOV P1'
<Process>	Describe the process to be added. The commands that can be described are as follow. 1. <Numeric type data B> <Substitution operator><Numeric type data A> [Substitute, signal modifier command (refer to syntax diagram)] 2. HLT statement 3. SKIP statement

Explanation

- This command can only be used to describe the additional command for the movement command.
- An error will occur if the WTH command is used alone.
- The process will be executed simultaneously with the start of motion.
- The relationship between the interrupts regarding the priority order is shown below.
  COM > ACT > WTHIF (WTH) > Pulse substitution

Reference Program

```
10      MOV P1 WTH M_OUT(17)=1 DLY M1+2          'Move to position P1 and set bit 17 for the duration of (M1+2) Seconds
```

Reference

Movement commands, WTHIF, Substitute

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

WTHIF (With If)

Function:

A process is conditionally added to the interpolation motion command.

Format

WTHIF <Additional command>, <Process>

Terminology

<Additional Command>	Describe the condition for adding the process. (Same as ACT condition expression)
<Process>	Describe the process to be added with the additional conditions are established. (Same as WTH) The commands that can be described as a process are as follow. (Refer to syntax diagram.) 1. <Numeric type data B> <Substitution operator><Numeric type data A> 2. HLT statement 3. SKIP statement

Explanation

- This command can only be used to describe the additional command with conditions for the movement command.
- Monitoring of the condition will start simultaneously with the start of movement.
- It is not allowed to write the DLY instruction at the processing part.

Reference Program

```
10  MOV P1 WTHIF M_IN(17)=1, HLT                 'If the input signal 17 turns on, the robot will stop
20  MVS P2 WTHIF M_RSPD>200,M_OUT(17)=1 DLY M1+2 'If the current command speed exceeds 200mm/s, turn on the output for M1+2 seconds.
30  MVS P3 WTHIF M_MRATIO>15, M_OUT(1)=1         'If the rate of arrival exceeds 15% during movement to P3, turn on the output.
```

Reference

Movement commands, WTH, Substitute, Syntax Diagram, Substitute
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

XCLR (X Clear)

Function:

This instruction cancels the program selection status of the specified task slot from within a program. It is used during multitask operation.

Format

```
XCLR <Slot No.>
```

Terminology

<Slot No.>	Designate the slot number

Explanation

- An error occurs at execution if the specified slot does not select the program.
- If the designated program is being operating, an error will occur at execution.
- If the designated program is being pausing, an error will occur at execution.
- If this instruction is used within a constantly executed program, it becomes enabled by changing the value of the "ALWENA" parameter from 0 to 7 and turning the controller's power off and on again.

Reference Program

```
10          XRUN 2,"1"              'Executes the first program in task slot
                ...
100         XSTP 2                  'Pauses the program of task slot 2.
110         WAIT M_WAI(2)=1         'Waits until the program of task slot 2 pauses.
120         XRST 2                  'Cancels the pause status of the program of task slot 2.
                ...
200         XCLR 2                  'Cancels the program selection status of the program
210         END
```

Reference

XLOAD , XRST , XRUN, XSTP

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

XLOAD (X Load)

Function:

This instruction commands the specified program to be loaded into the specified task slot from within a program. It is used during multitask operation.

Format

```
XLOAD <Slot no.>, "<Program name>"
```

Terminology

<Slot no.>	Designate the slot no.
<Program name>	Designate the program name.

Explanation

- If the designated program is already selected for another slot, an error will occur at execution.
- An error occurs at execution if the specified program does not exist.
- If the designated program is being edited, an error will occur at execution.
- Designate the program name in double quotations. " ".
- If the designated program is being executed, an error will occur at execution.
- If used in a program that is executed constantly, this instruction is enabled by changing the value of the "ALWENA" parameter from 0 to 7 and then turning the controller's power on again.

Reference Program

```
10 XLOAD 2, "10"     'Select program 10 for slot 2
20 XRUN 2            'Start slot 2
30 WAIT M_RUN(2)=1   'Wait to confirm starting of slot 2
```

Reference

XCLR, XRST , XRUN, XSTP

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

XRST (X Reset)

Function:

The execution line of the program in the designated slot is returned to the head line.

Format

XRST <Slot no.>

Terminology

<Slot no.> Designate the slot no.

Explanation

- This is valid only when the slot is in the stopped state.
- If used in a program that is executed constantly, this instruction is enabled by changing the value of the "ALWENA" parameter from 0 to 7 and then turning the controller's power on again.

Reference Program

```
10  XRUN 2             'Start
20  WAIT M_RUN(1)=1    'Wait for start to complete
:
100 XSTP 2             'Stop
110 WAIT M_WAIT(1)=1   'Wait for stop to complete
:
150 XRST 2             'Set program execution start line to head line.
160 WAIT M_PSA(1)=1    'Wait for program reset to complete
:
200 XRUN 2             'Restart
210 WAIT M_RUN(1)=1    'Wait for restart to complete
```

Reference

XRUN, XSTP, XCLR, XLOAD

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

XRUN (X Run)

Function:

Runs the designated program in parallel.

Format

```
XRUN <Slot no.>, <Program Name> [,<Operation Mode>]
```

Terminology

<Slot no.>	Designate the slot no.
<Program Name>	Designate the program name.
<Operation Mode>	0 = Continuous operation 1 = Cycle stop operation

Explanation

- If the designated slot no. Is already in use, an error will occur at execution.
- An error occurs at execution if the specified program does not exist.
- If a program has not been loaded into a task slot, this instruction will load it. It is thus possible to operate the program without executing the XLOAD instruction.
- If XRUN is executed in the "waiting" state with the program stopped midway, continuous execution will start.
- Designate the program name in double quotations. " ".
- If the operation mode is omitted, the current operation mode will be used.
- If used in a program that is executed constantly, this instruction is enabled by changing the value of the "ALWENA" parameter from 0 to 7 and then turning the controller's power on again.

Reference Program

Example 1

```
10 XRUN 1,"1"          'Run program 1 with slot 1.
20 XRUN 3,"2",1        'Restart program 2 with slot 3 in the cycle
                       operation mode.
```

Example 2

```
10 XLOAD 2,"1"          'Select the program 1 as slot 2
20 DLY 0.5              'Wait for the 0.5 seconds till the completion of
                        the program load.
30 XRUN 2               'Start the slot 2
20 WAIT M_RUN(2)=1      'Wait to have started
```

Example 3

```
10 XLOAD 3,"2"          'Select the program 1 as slot 2
20 DLY 0.5              'Wait for the 0.5 seconds till the completion of
                        the program load.
30 XRUN 3,,1            'Start the program 1 with cycle operation
20 WAIT M_RUN(3)=1      'Wait to have started
```

Reference

XCLR , XLOAD, XRST , XSTP

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

XSTP (X Stop)

Function:

Operation of the program in the designated slot is stopped.

Format

```
XSTP <Slot no.>
```

Terminology

<Slot no.> Designate the slot no.

Explanation

- If the program is already stopped, an error will not occur.
- XSTP can also stop the constant execution attribute program.

Reference Program

```
10 XRUN 2
:
100 XSTP 2            'Stop
110 WAIT M_WAI(1)=1   'Wait for stop to complete
:
200 XRUN 2            'Restart
```

Reference

XCLR , XLOAD, XRST ,XRUN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
Overview of the MELFA-BASIC IV Built-in Functions

Alphabetical overview of the MELFA-BASIC IV Built-in Functions

Alphabetical overview of the MELFA-BASIC IV Built-in Numeric Functions

Alphabetical overview of the MELFA-BASIC IV Built-in Trigonometric Functions

Alphabetical overview of the MELFA-BASIC IV Built-in Character String Functions

Alphabetical overview of the MELFA-BASIC IV Built-in Position Functions

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
List of Numeric Functions

Function name	Function
ABS	Produces the absolute value
ASC	Provides a character code for the first character of the character string in the expression
CINT	Rounds off the decimal value and converts into an integer
CKSUM	Calculates the checksum of the string.
CVD	Converts an 8-byte character string into a double-precision real number
CVI	Converts a 2-byte character string into integers
CVS	Converts a 4-byte character string into a single-precision real number
DEG	Converts the angle unit from radian (rad) to degree (deg)
EXP	Calculates the value of the expression,s exponential function
FIX	Produces an integer section
INT	Produces the largest integer that does not exceed the value in the expression
LEN	Produces the length of the character string

LN	Produces the natural logarithm
LOG	Produces the common logarithm.
MAX	Obtains the max. value from a random number of arguments
MIN	Obtains the min. value from a random number of arguments
RAD	Converts the angle unit from degree (deg) to radian (rad)
RND	Rounds off the number
SGN	Checks the sign of the number in the expression
SQR	Calculates the square root
STRPOS	Obtains the 2nd argument character string position in the 1st argument character string
VAL	Converts a character string into a numeric value

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

List of Built-in Position Functions

Function name	Function
CALARC	Provides information regarding the arc that contains the three specified points.
DIST	Obtains the distance between two points
FRAM	Calculates the coordinate system designated with three points. Position 1 is the plane origin, position 2 is the point on the +X axis, and position 3 is the point on the +Y axis direction plane. The plane origin point and posture are obtained from the XYZ coordinates of the three positions, and is returned with a return value (position). This is operated with 6-axis three dimensions regardless of the mechanism structure
RDFL1	Returns the structure flag of the designated position as character data. Argument (<numeric value>) 0=R/L, 1=A/B, 2=F/N is returned
SETFL1	Changes the structure flag of the designated position. The data to be changed is designated with characters (R/L/A/B/F/N)
RDFL2	Returns the multi-rotation data of the designated position as a numeric value (-2 to 1). The argument <numeric expression> returns the axis No. (1 to 8)
SETFL2	Changes the multi-rotation data of the designated position as a numeric value (-2 to 1). The left side of the expression is the axis No. to be changed; the right side is the value to be set
ALIGN	Returns the value of the orthogonal position (0,+/-90,+/-180) closest to the position 1 posture axis (A;B;C)
INV	Obtains the reserve line.
POSCQ	Checks whether the given position is within the movement range.
POSMID	Obtain the middle position data when a linear interpolation is performed between two given points.
PTOJ	Converts the position data into joint data.
JTOP	Converts the joint data into position data.
ZONE	Checks whether position 1 is within the space created by the position 2

	and position 3 points. Outside range = 0, within range = 1
ZONE2	Checks if the specified position is within the specified area (Cylindrical area defined by two points).

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
List of Built-in Character String Functions

Function name	Function
BIN$	Converts numeric expression value into binary character string
CHR$	Provides character having numeric expression value character code
HEX$	Converts numeric expression value into hexadecimal character string
LEFT$	Obtains character string having length designated with 2nd argument from left side of 1st argument character string.
MID$	Obtains character string having length designated with 3rd argument from the position designated with the 2nd argument in the 1st argument character string
MIRROR$	Mirror reversal of the character string binary bit is carried out
MKI$	Converts numeric expression value into 2-byte character string
MKS$	Converts numeric expression value into 4-byte character string
MKD$	Converts numeric expression value into 8-byte character string
RIGHT$	Obtains character string having length designated with 2nd argument from right side of 1st argument character string
STR$	Converts numeric expression value into a decimal character string

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
List of Built-in Trigonometric Functions

Function name	Function
ATN	Calculates the arc tangent. Unit: radian Definition range: Numeric value, Value range: -Pi/2 to +Pi/2
ATN2	Calculates the arc tangent. Unit: radian Teta=ATN2(y,x) Definition range: Numeric value of y or x that is not 0 Value range: -Pi to +Pi
COS	Calculates the cosine. Unit: radian Definition range: Numeric value, Value range: -1 to +1
SIN	Calculates the sine. Unit: radian Definition range: Numeric value, Value range: -1 to +1
TAN	Calculates the tangent. Unit: radian Definition range: Numeric value, Value range: Range of numeric value

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

Alphabetical Overview of MELFA-BASIC-IV Built-In Functions

Function name	Function
ABS	Produces the absolute value
ALIGN	Returns the value of the orthogonal position (0,+/-90,+/-180) closest to the position 1 posture axis (A;B;C)
ASC	Provides a character code for the first character of the character string in the expression
ATN	Calculates the arc tangent. Unit: radian Definition range: Numeric value, Value range: -Pi/2 to +Pi/2
ATN2	Calculates the arc tangent. Unit: radian Teta=ATN2(y,x) Definition range: Numeric value of y or x that is not 0 Value range: -Pi to +Pi
BIN$	Converts numeric expression value into binary character string
CALARC	Provides information regarding the arc that contains the three specified points.
CHR$	Provides character having numeric expression value character code
CINT	Rounds off the decimal value and converts into an integer
CKSUM	Calculates the checksum of the string.
COS	Calculates the cosine. Unit: radian Definition range: Numeric value, Value range: -1 to +1
CVD	Converts an 8-byte character string into a double-precision real number
CVI	Converts a 2-byte character string into integers
CVS	Converts a 4-byte character string into a single-precision real number
DIST	Obtains the distance between two points
EXP	Calculates the value of the expression,s exponential function
FIX	Produces an integer section
FRAM	Calculates the coordinate system designated with three points. Position 1 is the plane origin, position 2 is the point on the +X axis, and position 3 is the point on the +Y axis direction plane. The plane origin point and posture are obtained from the XYZ coordinates of the three positions, and is returned with a return value (position). This is operated with 6-axis three dimensions regardless of the mechanism structure
HEX$	Converts numeric expression value into hexadecimal character string
INT	Produces the largest integer that does not exceed the value in the expression
INV	Obtains the reserve line.
JTOP	Converts the joint data into position data.
LEFT$	Obtains character string having length designated with 2nd argument from left side of 1st argument character string.
LEN	Produces the length of the character string
LN	Produces the natural logarithm
LOG	Produces the common logarithm.
MAX	Obtains the max. value from a random number of arguments

MID$	Obtains character string having length designated with 3rd argument from the position designated with the 2nd argument in the 1st argument character string
MIN	Obtains the min. value from a random number of arguments
MIRROR$	Mirror reversal of the character string binary bit is carried out
MKI$	Converts numeric expression value into 2-byte character string
MKD$	Converts numeric expression value into 8-byte character string
MKS$	Converts numeric expression value into 4-byte character string
POSCQ	Checks whether the given position is within the movement range.
POSMID	Obtain the middle position data when a linear interpolation is performed between two given points.
PTOJ	Converts the position data into joint data.
RAD	Converts the angle unit from degree (deg) to radian (rad)
RDFL1	Returns the structure flag of the designated position as character data. Argument (<numeric value>) 0=R/L, 1=A/B, 2=F/N is returned
RDFL2	Returns the multi-rotation data of the designated position as a numeric value (-2 to 1). The argument <numeric expression> returns the axis No. (1 to 8)
RIGHT$	Obtains character string having length designated with 2nd argument from right side of 1st argument character string
RND	Rounds off the number
SETFL1	Changes the structure flag of the designated position. The data to be changed is designated with characters (R/L/A/B/F/N)
SETFL2	Changes the multi-rotation data of the designated position as a numeric value (-2 to 1). The left side of the expression is the axis No. to be changed; the right side is the value to be set
SGN	Checks the sign of the number in the expression
SIN	Calculates the sine. Unit: radian Definition range: Numeric value, Value range: -1 to +1
SQR	Calculates the square root
STR$	Converts numeric expression value into a decimal character string
STRPOS	Obtains the 2nd argument character string position in the 1st argument character string
TAN	Calculates the tangent. Unit: radian Definition range: Numeric value, Value range: Range of numeric value
VAL	Converts a character string into a numeric value
ZONE	Checks whether position 1 is within the space created by the position 2 and position 3 points. Outside range = 0, within range = 1
ZONE2	Checks if the specified position is within the specified area (Cylindrical area defined by two points).

©·2005·EFR·GERMANY

Function

Returns the absolute value of a given value.

Format

```
<Numeric Variable>= ABS (<Numerical Expression>)
```

Terminology

<Numeric Variable>	Specifies a numeric variable to assign.
<Numerical Expression>	Specifies a numerical value (variable)

Explanation

- Returns the absolute value (Value with the positive sign) of a given value.

Reference Program

```
10      P2.C=ABS(P1.C)              'P2.C will contain the value of P1.C
                                    without the sign.
```

Reference

SGN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ALIGN

Function

Positional posture axes (A, B, and C axes) are converted to the closest XYZ postures (0, ±90, and ±180). ALIGN outputs numerical values only. The actual operation will involve movement instructions such as the MOV instruction.

Format

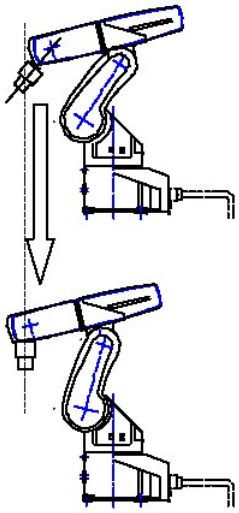
```
<Position Variable>= ALIGN (<Position>)
```

Terminology

<Position Variable>	Specifies a position variable to assign.
<Position>	Specifies a position (value or variable)

Explanation

- Converts the A, B, and C components of the position data to the closest XYZ postures (0, ±90, and ±180).
- Since the return value is of position data type, an error will be generated if the left-hand side is of joint variable type.
- This function cannot be used in vertical multi-joint 5-axes robot.



Reference Program

```
10      P1=P_CURR
20      P2=ALIGN(P1)
10      MOV P2
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ASC

Function

Returns the character code of the first character in the string.

Format

<Numeric Variable>=ASC ("<Character String Expression>")

Terminology

<Numeric Variable>	Specifies a numeric variable to assign.
<Character String Expression>	Specifies a string (explicit or a variable)

Explanation

- Returns the character code of the first character in the string.
- An error will be generated if the string is a null string.

Reference Program

```
10      M1=ASC("A")                    '&H41 is assigned to M1.
```

Reference

CVS, CVI, CVD, CHR$, VAL

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ATN

Function

Calculates the arc tangent.

Format

<Numeric Variable>=ATN (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign.
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Calculates the arc tangent of a given numerical expression. Unit is in radians.
- The range of the returned value for ATN is -pi/2 < ATN < pi/2.

Reference Program

```
10      M1=ATN(100/100)                 'pi/4 is assigned to M1.
```

Reference

SIN, COS, ATN2, TAN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ATN2

Function

Calculates the arc tangent.

Format

<Numeric Variable>=ATN2 (<Numerical Expression_1>, <Numerical Expression_2>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign.
<Numerical Expression_1>	Specifies numerical value or variable
<Numerical Expression_2>	Specifies numerical value or variable

Explanation

- Calculates the arc tangent of a given numerical expression. Unit is in radians.
- The range of the returned value for ATN2 is -pi < ATN < pi.
- If <Numerical Expression_2> evaluates to 0, ATN2 will return pi/2 when <Numerical Expression_1> evaluates to a positive value and -pi/2 when <Numerical Expression 1> evaluates to a negative value.
- In the case of ATN2, it is not possible to describe a function that contains an argument in <numerical expression_1> and <numerical expression_2>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      M1=ATN2(-100/100)               'pi/4 is assigned to M1.
```

Reference

SIN, COS, ATN, TAN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

BIN$

Function

Numerical value is converted to a binary string.

Format

<Character String Variable> = BIN$ (<Numerical Expression>)

Terminology

<Character String Variable>	Specifies a character string variable to assign
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Value is converted to a binary string.
- If the equation does not evaluate to an integer, the integral value obtained by rounding the fraction will be con
- converted to a binary string.
- VAL is a command that performs the opposite of this function.

Reference Program

```
10      M1=&B11111111
20      C1$=BIN$(M1)                    'C1$ will contain the character
                                        string of "11111111".
```

Reference

VAL, STR$, HEX$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CALARC

Function

Provides information regarding the arc that contains the three specified points.

Format

<Numeric Variable 4> = CALARC (<Position 1>, <Position 2>, <Position 3>, <Numeric Variable 1>, <Numeric Variable 2>, <Numeric Variable 3>, <Position Variables 1>)

Terminology

<Position 1>	Specifies the starting point of the arc.
<Position 2>	Specifies the passing point of the arc. Same as the three points in the MVR instruction.
<Position 3>	Specifies the endpoint of the arc.
<Numeric Variable 1>	Radius of the specified arc (in mm) will be calculated and returned.
<Numeric Variable 2>	Central angle of the specified arc (in radians)

	will be calculated and returned.
<Numeric Variable 3>	Length of the specified arc (in mm) will be calculated and returned.
<Position Variables 1>	The center coordinates of the specified arc (in mm) will be calculated and returned (as a position data type, ABC are all zeros).
<Numeric Variable 4>	Return value 1 : Calculation was performed normally. -1 : Of positions 1, 2, and 3, either two points had the exact same position or all three points were on a straight line. -2 : All three points are at approximately the same position.

Explanation

- Provides information regarding the arc that is determined by the three specified points, position 1, position 2 and position 3.
- If the arc generation and calculation of various values succeeded, 1 will be returned as the return value.
- If some points have the exact same position or if all three points are on a straight line, -1 will be returned as the return value. In such cases, the distance between the starting point and the endpoint will be returned as the arc length, -1 as the radius, 0 as the central angle, and (0, 0, 0) as the center point.
- If circular arc generation fails, -2 will be returned as the return value. If a circular arc cannot be generated, -1, 0, 0 and (0, 0, 0) are returned as the radius, central angle, arc length and center point, respectively.
- It is not possible to describe a function that contains an argument in <position 1>, <position 2>, <position 3>, <numeric variable 1>, <numerical variable 2>, <numeric variable 3> and <position variable 1>. If such a function is described, an error will be generated during execution.

Reference Program

```
10    M1=CALARC
      (P1,P2,P3,M10,M20,M30,P10)
20    IF M1<>1 THEN END              'Ends if an error occurs.
30    MR=M10                         'Radius.
40    MRD=M20                        'Circular arc angle.
50    MARCLEN=M30                    'Circular arc length.
60    PC=P10                         'Coordinates of the center point.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CHR$

Function

Returns the character that has the character code obtained from the specified numerical expression.

Format

<Character String Variable> = CHR$ (<Numerical Expression>)

Terminology

<Character String Variable>	Specifies a character string variable to assign
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Returns the character that has the character code obtained from the specified numerical expression.
- If the numerical expression does not evaluate to an integer, the character will be returned whose character code corresponds to the integral value obtained by rounding the fraction.

Reference Program

```
10      M1=&H40
20      C1$=CHR$(M1+1)          'C1$ will contain the character
                                "11111111".
```

Reference

ASC

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CINT

Function

Rounds the fractional part of an numerical expression to convert the value into an integer.

Format

<Numeric Variable>=CINT (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Returns the value obtained by rounding the fractional part of an equation.

Reference Program

```
10      M1=CINT(1.5)            '2 is assigned to M1
20      M2=CINT(1.4)            '1 is assigned to M2
30      M3=CINT(-1.4)           '-1 is assigned to M3
20      M4=CINT(-1.5)           '-2 is assigned to M4
```

Reference

INT, FIX

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CKSUM

Function

Calculates the checksum of the string.

Format

<Numeric Variable>=CKSUM (<Character String>, <Numerical Expression 1>, <Numerical

Expression 2>)

Terminology

<Character String>	Specifies the string from which the checksum should be calculated.
<Numerical Expression 1>	Specifies the first character position from where the checksum calculation starts.
<Numerical Expression 2>	Specifies the first character position from where the checksum calculation ends.

Explanation

- Adds the character codes of all characters in the string from the starting position to the end position and returns a value between 0 and 255.
- If the starting position is outside the range of the string, an error will be generated.
- If the end position exceeds the end of the string, checksum from the starting position to the last character in the string will be calculated.
- If the result of addition exceeds 255, a degenerated value of 255 or less will be returned.
- It is not possible to describe a function that contains an argument in <Character String>, <Numerical Expression 1> and <Numerical Expression 2>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      M1=CKSUM("ABCDEFG",1,3)          '&H41("A")+&H42("B")+&H43("C")=&HC6
                                         is assigned to M1.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

COS

Function

Gives the cosine.

Format

<Numeric Variable>=COS (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Calculates the cosine of the numerical expression.
- The range of arguments will be the entire range of values that are allowed.
- The range of the return value will be from -1 to 1.
- The unit of arguments is in radians.

Reference Program

```
10      M1=COS(RAD(60))    'The COS of 60Radiant is assigned to M1
```

Reference

SIN, ATN2, ATN, TAN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CVD

Function

Converts the character codes of the first eight characters of a string into a double precision real number.

Format

<Numeric Variable>=CVD (<Character String Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String Expression>)	Specifies a string

Explanation

- Converts the character codes of the first eight characters of a string into a double precision real number.
- An error will be generated if the string consists of seven character or less.
- MKD$ can be used to convert numerical values into a string.

Reference Program

```
10      M1=CVD("FFFFFFFF")'     '+3.52954E+30 is assigned to M1.
```

Reference

ASC, CVI, CVS, MKI$, MKS$, MKD$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CVI

Function

Converts the character codes of the first two characters of a string into an integer.

Format

<Numeric Variable>=CVI (<Character String Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String Expression>)	Specifies a string

Explanation

- Converts the character codes of the first two characters of a string into an integer.
- An error will be generated if the string consists of one character or less.
- MKI$ can be used to convert numerical values into a string.
- This can be used to reduce the amount of communication data when transmitting numerical data with external devices.

Reference Program

```
10      M1=CVI("10ABC")'        '&H3031 is assigned to M1.
```

Reference

ASC, CVD, CVS, MKI$, MKS$, MKD$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
CVS

Function

Converts the character codes of the first four characters of a string into a single precision real

Format

<Numeric Variable>=CVS (<Character String Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String Expression>)	Specifies a string

Explanation

- Converts the character codes of the first four characters of a string into an single-precision real number.
- An error will be generated if the string consists of three character or less.
- MKS$ can be used to convert numerical values into a string.

Reference Program

```
10      M1=CVS("FFFF")'         '12689.6 is assigned to M1.
```

Reference

ASC, CVD, CVI, MKI$, MKS$, MKD$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
DEG

Function

Converts the unit of angle measurement from radians (rad) into degrees (deg).

Format

<Numeric Variable>=DEG (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Converts the radian value of an numerical expression into degree value.
- When the posture angles of the position data are to be displayed using positional constants, the unit used for ((500, 0, 600, 180, 0, 180) (7, 0)) is DEG. As in the case of P1.C, the unit used will be in radians (rad) when the rotational element of the positional variable is to be referenced directly. Value of P1.C can be handled in DEG. In such case, set parameter "PRGMDEG" to 1.

Reference Program

```
10      P1=P_CURR
20      IF DEG(P1.C) < 170 OR DEG(P1.C) > -150 THEN      'Calculate if skipping the
        *NOERR                                           error or not
30      ERROR(9100)
40      *NOERR
```

Reference

RAD

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

DIST

Function

Calculates the distance between two points (position variables).

Format

<Numeric Variable>=DIST (<Position 1>, <Position 2>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Position 1>	Specifies a position (variable or value)
<Position 2>	Specifies a position (variable or value)

Explanation

- Returns the distance between positions 1 and 2 (in mm).
- Posture angles of the position data will be ignored; only the X, Y, and Z data will be used for calculation.
- The joint variables cannot be used. Trying to use it will result in an error during execution.
- It is not possible to describe a function that contains an argument in <position 1> and <position 2>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      M1=DIST (P1,P2)      'M1 will contain the distance between positions 1 and
                             2.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

EXP

Function

Calculates exponential functions. (an numerical expression that uses "e" as the base.)

Format

<Numeric Variable>=EXP (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numerical value or variable

Explanation

- Returns the exponential function value of the numerical expression.

Reference Program

```
10      M1=EXP(2)              'e^2 is assigned to M1
```

Reference

LN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

FIX

Function

Returns the integral portion of the numerical expression.

Format

<Numeric Variable>=FIX (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numerical value or variable

Explanation

- Returns the integral portion of the numerical expression value.
- If the numerical expression evaluates to a positive value, the same number as INT will be returned.
- If the numerical expression evaluates to a negative value, then for instance FIX(-2.3) = -2.0 will be observed.

Reference Program

```
10      M1=FIX(5.5)        '5 is assigned to M1.
```

Reference

CINT, INT

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

FRAM

Function

Calculates the position data that indicates a coordinate system (plane) specified by three position data. Normally, use DEF PLT and PLT instructions for pallet calculation.

Format

<Numeric Variable 4>=FRAM (<Numeric Variable 1>, <Numeric Variable 2>, <Numeric Variable

3>)

Terminology

<Numeric Variable 1>	This will be the origin of X, Y, and Z of the plane to be specified by three positions. A variable or a constant.
<Numeric Variable 2>	A point on the X axis of the plane to be specified by three positions. A variable or a constant.
<Numeric Variable 3>	A point in the positive Y direction of the X-Y plane on the plane to be specified by three positions. A variable or a constant.
<Numeric Variable 4>	Variable to which the result is assigned.

Explanation

- This can be used to define the base coordinate system.
- This creates a plane from the three coordinates X, Y, and Z for the three positions to calculate the position of the origin and the inclination of the plane, and returns the result as a position variable. The X, Y, and Z coordinates of the position data will be identical to that of position variable 1, while A, B, and C will be the inclination of the plane to be specified by the three positions.
- Joint variables cannot be used as the argument. When a joint variable is used, an error will be generated.
- Since the return value is a position data, an error will be generated if a joint variable is used in the left-hand side.
- It is not possible to describe <Numeric Variable X> as a function. If you do so, an error will be generated during execution.

Reference Program

```
10      P100=FRAM(P1,P2,P3)          'Create P100 coordinate system based on P1, P2 and P3 positions.
20      BASE P100                    'Position of P100 will be used as the origin for robot.

10      M1=FIX(5.5)                  '5 is assigned to M1.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

HEX$

Function

Converts the value of an numerical expression (Between -32768 to 32767) into hexadecimal string.

Format

<Character String Variable >=HEX$ (<Numerical Expression> [, <Number of output characters>])

Terminology

<Character String Variable>	Specifies a character string variable to assign
<Numerical Expression>	Specifies numerical value or variable
<Number of output characters>	Number of characters, starting form the least significant byte

Explanation

- Converts the value of an numerical expression into hexadecimal string.
- If <Number of output characters> is specified, the right most part of the converted string is output for the specified length.
- If the numerical value is not an integer, the integer value obtained by rounding the fraction will be converted into hexadecimal string.
- VAL is a command that performs this procedure in reverse.
- If <number of output characters> is specified, it is not possible to describe a function that contains an argument in <numerical expression>. If such a function is described, an error

will be generated during execution.
NG example C1$=HEX$(ASC("a"),1)

Reference Program

```
10      C1$=HEX$(&H41FF)        '"41FF" is assigned to C1$.
20      C2$=HEX$(&H41FF,2)      '"FF" is assigned to C2$.
```

Reference

BIN$, STR$, VAL

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

INT

Function

Returns the largest integer that does not exceed the value of the numerical expression.

Format

<Numeric Variable>=INT (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numerical value or variable

Explanation

- Returns the largest integer that does not exceed the value of the numerical expression.
- If the numerical expression evaluates to a positive value, the same number as FIX will be returned.
- If the numerical expression evaluates to a negative value, then for instance FIX(-2.3) = -3.0 will be observed.

Reference Program

```
10      M1=INT(3.3))        '3 is assigned to M1.
```

Reference

CINT. FIX

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

INV

Function

Obtains the position data of the inverse matrix of the position variable. This is used to perform relative calculation of the positions.

Format

<Position Variables>=INV (<Position Variables>)

Terminology

<Position Variables>	Specifies a Position variable to assign
<Position Variables>	specifies a Position variable to assign

Explanation

- Obtains the position data of the inverse matrix of the position variable.
- Joint variables cannot be used as the argument. When a joint variable is used, an error will be generated.
- Since the return value is a position data, an error will be generated if a joint variable is used in the left-hand side.

Reference Program

```
10      P1=INV(P2)                  'P1 will contain the inverse matrix of
                                    P2.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

JTOP

Function

Given joint data will be converted into position data.

Format

<Position Variables>=JTOP (<Joint Variable>)

Terminology

<Position Variables>	Specifies a position variable to assign
<Joint Variable>	Specifies a joint variable to assign

Explanation

- Converts the joint data into the position data.
- Position variables cannot be used as the argument. When a position variable is used, an error will be generated.
- Since the return value is a position data, an error will be generated if a joint variable is used in the left-hand side.

Reference Program

```
10      P1=JTOP(J1)                 'The position that expresses the J1
                                    (joint type) position using the XYZ
                                    type will be assigned to P1.
```

Reference

PTOJ

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

LEFT$

Function

Obtains a string of the specified length starting from the left end.

Format

<Character String Variable >=LEFT$ ("<Character String>", <Numerical Expression>)

Terminology

<Character String Variable>	Specifies a character string variable to assign
<Character String>	Specifies a string (variable or expression)
<Numerical Expression>	Specifies the length of the assigned String.

Explanation

- Obtains a string of the specified length starting from the left end.
- An error will be generated if the value is a negative value or is longer than the string.
- It is not possible to describe a function that contains an argument in <Character String> and <Numerical Expression>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      C1$=LEFT$("ABC",2)          '"AB" is assigned to C1$.
```

Reference

MID$, RIGHT$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

LEN

Function

Returns the length of the string.

Format

<Numeric Variable>=LEN ("<Character String>")

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String>	Specifies a string (variable or expression)

Explanation

- Returns the length of the argument string.

Reference Program

```
10      M1=LEN("ABCDEFG")          '7 is assigned to M1.
```

Reference

MID$, RIGHT$, LEFT$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

LN

Function

Returns the natural logarithm. (Base e.)

Format

<Numeric Variable>=LN (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numeric expression or variable

Explanation

- Returns the natural logarithm of the value of the numerical expression.
- An error will be generated if the numerical expression evaluates to a zero or a negative value.

Reference Program

```
10      M1=LN(2)                '0.693147 is assigned to M1
```

Reference

EXP, LOG

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

LOG

Function

Returns the common logarithm. (Base 10.)

Format

<Numeric Variable>=LOG (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numeric expression or variable

Explanation

- Returns the common logarithm of the value of the numerical expression.
- An error will be generated if the numerical expression evaluates to a zero or a negative value.

Reference Program

```
10      M1=LOG(2)               '0.301030 is assigned to M1
```

Reference

EXP, LN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MAX

Function

Obtains the maximum value.

Format

<Numeric Variable>=MAX (<Numerical Expression 1>, <Numerical Expression 2> [, <Numerical Expression 3>...])

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression n>	Specifies a numeric expression or variable

Explanation

- Returns the maximum value among the arbitrary number of arguments.
- The length of this instruction can be up to the number of characters allowed in a single line (123 characters).
- It is not possible to describe <Numerical Expression X> as a function. If you do so, an error will be generated during execution.

Reference Program

```
10      M1=MAX(2,1,3,4,10,100)      '100 is assigned to M1
```

Reference

MIN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MID$

Returns a string of the specified length starting from the specified position of the string.

Format

<Character String Variable >=MID$ (<Character String>, <Numerical Expression 2>, <Numerical Expression 3>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String>	Specifies the string
<Numerical Expression 1>	Specifies the starting position, starting for the left
<Numerical Expression 2>	Specifies the number of elements to assign

Explanation

- A string of the length specified by argument 3 is extracted from the string specified by the first argument starting from the position specified by argument 2 and returned.
- An error will be generated if numerical expression 2 or 3 evaluates to a zero or a negative value.
- An error is generated if the position of the last character to be extracted is larger than the length of the string specified by the first argument.
- It is not possible to describe a function that contains an argument in <Character String>,

<Numerical Expression 2> and <Numerical Expression 3>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      C1$=MID$("ABCDEFG",3,2)      '"CD" is assigned to C1$.
```

Reference

LEFT$, RIGHT$, LEN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MIN

Function

Obtains the minimum value.

Format

<Numeric Variable>=MIN (<Numerical Expression 1>, <Numerical Expression 2> [, <Numerical Expression 3>...])

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression X>	Specifies a numeric expression or variable

Explanation

- Returns the minimum value among the arbitrary number of arguments.
- The length of this instruction can be up to the number of characters allowed in a single line (123 characters).
- It is not possible to describe <Numerical Expression X> as a function. If you do so, an error will be generated during execution.

Reference Program

```
10      M1=MIN(2,1,3,4,10,100)      '1 is assigned to M1
```

Reference

MIN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MIRROR$

Function

Inverts the bit string representing each character code of the string in binary, and obtains the character-coded string.

Format

<Character String Variable >=MIRROR$ (<Character String Expression>)

Terminology

<Character String Variable >	Specifies a string variable to assign
<Character String Expression>	Specifies a Character String Expression or variable

Explanation

- Inverts the bit string representing each character code of the string in binary, and obtains the character-coded string.

Reference Program

```
10      C1$=MIRROR$("BJ")           '"RB" is assigned to C1$.
                                    '"BJ" =&H42,&H4A=&B01000010,&B01001010.
                                    'Inverted
                                    '=&H52,&H42=&B01010010,&B01000010. 'Output
                                    ="RB". [
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MKD$

Converts the value of an numerical expression (double-precision real number) into a eight-byte string.

Format

<Character String Variable >=MKD$ (<Numerical Expression>)

Terminology

<Character String Variable >	Specifies a string variable to assign
<Numerical Expression>	Specifies a numerical variable or expression

Explanation

- Converts the lowest eight bytes of the value of an numerical expression (single-precision real number) into the strings.
- Use CVD to convert the string to a value.
- This can be used to reduce the amount of communication data when transmitting numerical data to external devices.

Reference Program

```
10      C1$=MKD$(10000.1)
20      M1=CVD(C1$)'                '10000.1 is assigned to M1.
```

Reference

ASC, CVI, CVS, MKI$, MKS$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MKI$

Function

Converts the value of an numerical expression (integer) into a two-byte string.

Format

<Character String Variable >=MKI$ (<Numerical Expression>)

Terminology

<Character String Variable >	Specifies a string variable to assign
<Numerical Expression>	Specifies a numerical variable or expression

Explanation

- Converts the lowest two bytes of the value of an numerical expression (integer) into a strings.
- Use CVI to convert the string to a value.
- This can be used to reduce the amount of communication data when transmitting numerical data to external devices.

Reference Program

```
10      C1$=MKI$(123)           '
20      M1=CVD(C1$)'            '123 is assigned to M1.
```

Reference

ASC, CVI, CVS, MKD$, MKS$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MKS$

Function

Converts the value of an numerical expression (single-precision real number) into a four-byte string.

Format

<Character String Variable >=MKS$ (<Numerical Expression>)

Terminology

<Character String Variable >	Specifies a string variable to assign
<Numerical Expression>	Specifies a numerical variable or expression

Explanation

- Converts the lowest four bytes of the value of an numerical expression (single-precision real number) into the strings.
- Use CVS to convert the string to a value.
- This can be used to reduce the amount of communication data when transmitting numerical data to external devices.

Reference Program

```
10      C1$=MKI$(100.1)         '
20      M1=CVD(C1$)'            '100.1 is assigned to M1.
```

Reference

ASC, CVI, CVS, MKD$, MKI$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

POSCQ

Function

Checks whether the given position is within the movement range.

Format

<Numeric Variable>=POSCQ (<Position Variable>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Position Variables>	Specifies a joint variable to assign

Explanation

- Check whether the position data given by an argument is within the movement range of the robot. Value 1 will be returned if it is within the movement range of the robot; value 0 will be returned if it is outside the movement range of the robot.
- Arguments must give either the position data type or joint data type.

Reference Program

```
10      M1=POSCQ(P1)      'M1 will contain 1 if the position P1 is within
                          the movement range.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

POSMID

Function

Obtain the middle position data when a linear interpolation is performed between two given points.

Format

<Position Variables>=POSMID (<Position Variable 1>, <Position Variables 2>,<Numerical Constant 1>, <Numerical Constant 2>)

Terminology

<Position Variable>	Specifies a position variable to assign
<Position Variable 1/2>	Specifies a position variable to assign
<Numerical Constant 1>	Detour/short cut = 1/0
<Numerical Constant 2>	3-axis orthogonal/Equivalent rotation = 1/0

Explanation

- Obtain the position data of the middle point when a linear interpolation is performed between two position data.
- The first argument gives the starting point of the linear interpolation, while the second argument gives the end-point of the linear interpolation.
- The third and fourth arguments correspond to the two TYPE arguments of the MVS command.
- The arguments for the starting and end points must be positions that allow linear interpolation with the specified interpolation type. For instance, an error will be generated if the structure flags of the starting and end points are different.
- It is not possible to describe a function that contains an argument in <Position Variables

1>, <Position Variables 2>,<Numerical Expression 1> and <Numerical Expression 2>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      P1=POSMID(P2,P3,0,0)          'The position data (including
                                      posture) of the middle point
                                      between P2 and P3 will be assigned
                                      to P1.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

PTOJ

Function

Converts the given position data into a joint data.

Format

<Joint Variable>=PTOJ (<Position Variables>)

Terminology

<Joint Variable>	Specifies a joint variable to assign
<Position Variables>	Specifies a joint variable to assign

Explanation

- Converts the position data into the joint data.
- Joint variable (J variable) cannot be used as the argument. When a joint variable is used, an error will be generated.
- Since the return value is a joint data, an error will be generated if a position variable is used in the left-hand side.

Reference Program

```
10      J1=PTOJ(P1)       'J1 will contain the value of P1 (XYZ position
                          variable) that has been converted into joint data
                          type.
```

Reference

JTOP

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RAD

Function

Converts the unit of angle measurement from degrees (deg) into radians (rad).

Format

<Numeric Variable>=RAD (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies numerical value or variable

Explanation

- Converts the degree value of an numerical expression into radian value.
- This can be used to assign values to the posture components (ABC) of a position variable or to execute trigonometric functions.

Reference Program

```
10      P1=P_CURR                       '
20      P1.C=RAD(90)                    '
30      MOV P1                          'Moves to P1, which is obtained by
                                        changing the C axis of the current
                                        position to 90 degrees.
```

Reference

DEG

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RDFL1

Function

Returns the structure flag of the specified position using character data "R"/"L", "A"/"B", and "N"/"F".

Format

<Character String Variable >=RDFL1 (<Position Variables>, <Numerical Expression>)

Terminology

<Position Variable>	Specifies a position variable to assign
<Numerical Expression>	Specifies which structure flag is to be extracted. 0 = "R" / "L", 1 = "A" / "B", 2 = "N" / "F"

Explanation

- Of the structure flags in the position data specified by argument 1, the flag specified by argument 2 will be extracted.
- This function extracts information from the FL1 element of position data.
- It is not possible to describe a function that contains an argument in <Position Variables> and <Numerical Expression>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      P1=(100,0,100,180,0,180)(7,0)   'Since the structure flag 7 (&B111)
                                        is RAN
20      C1$=RDFL1(P1,1)                 'C1$ will contain "A".
```

Reference

RDFL2, SETFL1, SETFL2

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RDFL2

Function

Returns the multiple rotation information of the specified joint axis.

Format

<Numeric Variable>=RDFL2 (<Position Variables>, <Numerical Expression>)

Terminology

<Position Variable>	Specifies the position variable from which the multiple rotation information is to be extracted.
<Numerical Expression>	Specifies the value for the joint axis from which the multiple rotation information is to be extracted. (1 through 8)

Explanation

- Of the multiple rotation information of the position data specified by argument 1, the value for the joint axis specified by argument 2 is extracted.
- The range of the return value is between -8 and 7.
- This function extracts information from the FL2 element of position data.
- Structure flag 2 (multiple rotation information) has a 32-bit structure, which contains 4 bits of information per axis for 8 axes.
- It is not possible to describe a function that contains an argument in <Position Variables> and <Numerical Expression>. If such a function is described, an error will be generated during execution.
- When displaying in T/B and the multiple rotation is a negative value, value of -1 to -8 is converted into F to 8 (4-bit signed hexadecimal notation) and displayed.

<Sample display of multiple rotation information in TB>	87654321 axis	<Relationship between display and number of multiple rotations per axis>
When multiple rotation of axis J6 is +1:	FL2=00100000	.....-2 -1 0 +1 +2.....
When multiple rotation of axis J6 is -1:	FL2=00F00000	..... E F 0 1 2 .....

Reference Program

```
10      P1=(100,0,100,180,0,180)                ,
        (7,&H00100000)
20      M1=RDFL2(P1,6)                          '1 is assigned to M1.
```

Reference

RDFL1, SETFL1, SETFL2

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RIGHT$

Function

Obtains a string of the specified length starting from the right end.

Format

<Character String Variable >=RIGHT$ (<Character String Expression>, <Numerical Expression>)

Terminology

<Character String Variable >	Specifies a string variable to assign
<Character String Expression>	Specifies a Character String Expression or variable
<Numerical Expression>	Specifies a numerical expression or variable

Explanation

- Obtains a string of the specified length starting from the right end.
- An error will be generated if the value of the second argument is a negative value or is longer than the first string.
- It is not possible to describe a function that contains an argument in <Character String> and <Numerical Expression>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      C1$=RIGHT$("ABCDEFG",3)          '"EFG" is assigned to C1$.
```

Reference

LEFT$, MID$, LEN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RND

Function

Generates a random number.

Format

<Numeric Variable>=RND (<Numerical Expression>)

Terminology

<Numeric Variable>	A value in the range of 0.0 to 1.0 will be returned.
<Numerical Expression>	A value in the range of 0.0 to 1.0 will be returned.

Explanation

- Initializes random numbers using the value provided by the argument and extracts a random number.
- If the numerical expression provided as the argument evaluates to 0, initialization of random numbers will not take place and the next random number will be extracted.
- When the same value is used to perform initialization of random numbers, identical random number sequence will be obtained.

Reference Program

```
10      DIM MRND(10)                     '
20      C1=RIGHT$(C_TIME,2)              'Initializes random
                                         numbers using the clock.
30      MRNDBS=CVI(C1))                  'In order to obtain
                                         different sequence of
                                         numbers.
40      MRND(1)=RND(MRNDBS)              'Sets the initial value
                                         of random numbers and
                                         extracts the first
                                         random number.
50      FOR M1=2 TO 10                   'Obtain other nine
                                         random numbers.
60      MRND(M1)=RND(0)
70      NEXT M1
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SETFL1

Function

Changes the structure flag of the specified position using a string (such as "RAN").

Format

<Position Variable>=SETFL1 (<Position Variable>, <Character String>)

Terminology

<Position Variables>	Specifies the position variable from which the multiple rotation information is to be extracted.
<Character String>	Specifies the structure flag to be changed. Multiple flags can be specified. "R" or "L": Right/Left setting. "A" or "B": Above/Below setting. "N" or "F": Nonflip/Flip setting.

Explanation

- Returns the position data obtained by changing the structure flags in the position data specified by argument 1 to flag values specified by argument 2.
- This function changes information from the FL1 element of position data. The content of the position data given by the argument will remain unchanged.
- The structure flag will be specified starting from the last character in the string. Therefore, for instance, if the string "LR" is specified, the resulting structure flag will be "L".
- If the flags are changed using a numerical value, set P1.FL1=7.
- Structure flags may have different meanings depending on the robot model. For details, please refer to "ROBOT ARM SETUP & MAINTENANCE" for each robot.
- The structure flag corresponds to 7 in the position constant (100, 0, 300, 180, 0, 180) (7, 0). The actual position is a bit pattern.

Reference Program

```
10      MOV P1
20      P2=SETFL1(P1,"LBF")
30      MOV P2
```

Reference

RDFL1, RDFL2, SETFL2

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SETFL2

Function

Changes the multiple rotation data of the specified position.

Format

<Position Variables>=SETFL2 (<Position Variables>, <Numerical Expression 1>, <Numerical Expression 2>)

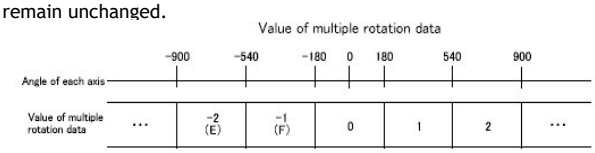
Terminology

<Position Variable>	Specifies the position variable whose multiple rotation data are to be changed.
<Numerical Expression 1>	Specifies the axis number for which the multiple

	rotation data are to be changed. (1 through 8).
<Numerical Expression 2>	Specifies the multiple rotation data value to be changed (-8 through 7).

Explanation

- Returns the position data obtained by changing the position data's multiple rotation information of the joint axis specified by numerical expression 1 to the value specified by numerical expression 2.
- This function changes information from the FL2 element of position data.
- The content of the position of position variables given by the argument (X, Y, Z, A, B, C, and FL1) will remain unchanged.



- It is not possible to describe a function that contains an argument in <Position Variables>, <Numerical Expression 1> and <Numerical Expression 2>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      MOV P1
20      P2=SETFL2(P1,6,1)
30      MOV P2
```

Reference

RDFL1, RDFL2, SETFL1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SGN

Function

Checks the sign of the numerical expression.

Format

<Numeric Variable>=SGN (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numeric expression or variable

Explanation

- Checks the sign of the numerical expression and returns the following value.
  - Positive value 1
  - Negative value -1

Reference Program

```
10      M1=-12                  '
20      M2=SGN(M1)              '-1 is assigned to M2.
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SIN

Function

Calculates the sine

Format

<Numeric Variable>=SIN (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numeric expression or variable

Explanation

- Calculates the sine to which the given numerical expression evaluates.
- The range of values will be the entire range that numerical values can take.
- The range of the return value will be from -1 to 1.
- The unit of arguments is in radians.

Reference Program

```
10      M1=SIN(RAD(60))                 '0.866025 is assigned to M1.
```

Reference

COS, ATN, ATN2, TAN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SQR

Function

Calculates the square root of an numerical expression value.

Format

<Numeric Variable>=SQR (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numeric expression or variable

Explanation

- Calculates the square root of the value to which the given numerical expression evaluates.
- An error will be generated if the numerical expression given by the argument evaluates to a negative value.

Reference Program

```
10      M1=SQR(2)                       '1,414214 is assigned to M1
```

Reference

COS, ATN, ATN2, TAN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
STR$

Function

Converts the value of the numerical expression into a decimal string.

Format

<Character String Variable >=STR$ (<Numerical Expression>)

Terminology

<Character String Variable >	Specifies a string variable to assign
<Numerical Expression>	Specifies a numerical variable or expression

Explanation

- Converts the value of the numerical expression into a decimal string.
- VAL is a function that performs this procedure in reverse.

Reference Program

```
10      C1$=STR$(123)           '"123" is assigned to C1$.
```

Reference

BIN$, HEX$, VAL

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
STRPOS

Function

Searches for a specified string in a string.

Format

<Numeric Variable>=STRPOS(<Character String 1>, <Character String 2>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String 1/2>	Specifies a the character strings, that should be compared

Explanation

- Returns the position of the first occurrence of the string specified by argument 2 from the string specified by argument 1.
- An error will be generated if the length of the argument 2 is 0.
- For instance, if argument 1 is "ABCDEFG" and argument 2 is "DEF", 4 will be returned.
- If the search string could not be found, 0 will be returned.
- It is not possible to describe a function that contains an argument in <Character String 1> and <Character String 2>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      M1=STRPOS("ABCDEFG","DEF")      '
```

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

TAN

Function

Calculates the tangent.

Format

<Numeric Variable>=TAN (<Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Numerical Expression>	Specifies a numeric expression or variable

Explanation

- Returns the tangent of the value to which the numerical expression evaluates.
- The range of arguments will be the entire range of values that are allowed.
- The range of return values will be the entire range that numerical values can take.
- The unit of arguments is in radians.

Reference Program

```
10      M1=TAN(RAD(60))                 '1.732051 is assigned to M1.
```

Reference

COS, ATN, ATN2, SIN

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

VAL

Function

Converts the value in the string into a numerical value.

Format

<Numeric Variable>=VAL (<Character String Expression>)

Terminology

<Numeric Variable>	Specifies a numeric variable to assign
<Character String Expression>	Specifies an Character expression or variable

Explanation

- Converts the given character string expression string into a numerical value.
- Binary (&B), decimal, and hexadecimal (&H) notations can be used for the string.
- In the example above, M1, M2 and M3 evaluate to the same value (15).

Reference Program

```
10      M1=VAL("15")
20      M2=VAL("&B1111")
30      M3=VAL("&HF")
```

Reference

BIN$, HEX$, STR$

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ZONE

Function

Checks if the specified position is within the specified area (a rectangular solid defined by two points).

Format

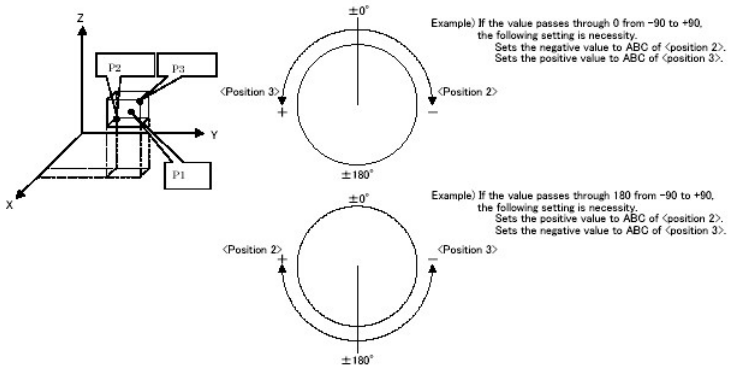
<Numeric Variable>=ZONE (<Position 1>, <Position 2>, <Position 3>)

Terminology

<Numeric Variable>	Specifies a numerical variable to assign
<Position 1>	The position to be checked.
<Position 2>	The position of the first point that specifies the area.
<Position 3>	The position of the second point that specifies the area. (diagonal point)

Explanation

- This will check if position 1 is inside the rectangular solid defined by the two points, position 2 and position 3. (The two points will become the diagonal points of the rectangular solid.) If the point is inside the rectangular solid, 1 is returned; otherwise, 0 is returned.
- To check whether position 1 is inside that area, each element of position 1 (X, Y, Z, A, B, and C) will be checked if it is between the values for position 2 and position 3.
- As for the posture angles (A, B, and C), they are checked by rotating in the positive direction from the angle in position 2 to position 3 and by seeing if the target value is inside the swiped range. Example) If P2.A is -100 and P3.A is +100, if P1.A is 50, the value is within the range. Similar checking will be performed for B and C axes. (Refer to diagram below.)
- For components that are not checked or do not exist, if the unit is in degrees, position 2 will be set to -360 and position 3 will be set to 360. If the unit is in millimeters, position 2 will be set to -10000 and position 3 will be set to 10000.
- It is not possible to describe a function that contains an argument in <Position 1>, <Position 2> and <Position 3>. If such a function is described, an error will be generated during execution.



Reference Program

```
10      M1=ZONE2(P1,P2,P3)
20      IF M1=1 THEN MOV P_SAFE ELSE END
```

Reference

ZONE2

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ZONE2

Function

Checks if the specified position is within the specified area (Cylindrical area defined by two points).

Format

<Numeric Variable>=ZONE2 (<Position 1>, <Position 2>, <Position 3>, <Numerical Expression>)

Terminology

<Numeric Variable>	Specifies a numerical variable to assign
<Position 1>	The position to be checked.
<Position 2>	The position of the first point that specifies the area.
<Position 3>	The position of the second point that specifies the area.
<Numerical Expression>	Radius of the hemisphere on both ends

Explanation

- This will check if position 1 is inside the cylindrical area (Refer to diagram below) defined by the two points, position 2 and position 3, and the radius represented by the numerical expression. If the point is inside the space, 1 is returned; otherwise, 0 is returned.
- This function checks whether the check position (X, Y, and Z coordinates) is within the specified area, but does not take the posture components into consideration.

- It is not possible to describe a function that contains an argument in <Position 1>, <Position 2>, <Position 3> and <Numerical Expression>. If such a function is described, an error will be generated during execution.

Reference Program

```
10      M1=ZONE2(P1,P2,P3,50)
20      IF M1=1 THEN MOV P_SAFE ELSE
        END
```

Reference

ZONE

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

Overview of the Parameters of MELFA-BASIC IV

Alphabetical overview of the MELFA-BASIC IV parameters

Alphabetical overview of the MELFA-BASIC IV parameters for dedicated I/O

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

Alphabetical Overview of MELFA-BASIC-IV parameters

Parameter	Function
AREA*AT	User area
AREA*ME	User area
AREA*P1	User area
AREA*P2	User area
BZR	Buzzer ON/OFF
CBAU232	Communication setting: Baud rate
CPRC232	Communication setting: Communication method
CPRTY232	Communication setting: Parity
CSTOP232	Communication setting: Stop bit
CTERM232	Communication setting: End code
HANDINIT	Hand initial state
HANDTYPE	Hand type
HNDDAT*	Maximum acceleration/deceleration setting
HNDHOLD*	Maximum acceleration/deceleration setting
INB	Stop input B contact designation
JOGJSP	Jog setting
JOGPSP	Jog setting
JOGSPMX	Jog speed value
MEJAR	Joint movement range
MEPAR	Orthogonal movement range
MEXBS	Tool base coordinates
MEXTL	Standard tool coordinates
OPPSL	Program selection rights
ORST*	Output signal pattern
PRGUSR	User base program
PRSTNA	Program reset operation rights
RETPATH	Automatic return setting
RMTPSL	Program selection rights
SFC*AT	Free plane limit
SFC*ME	Free plane limit
SFC*P	Free plane limit
SLRSTIO	Output reset at reset
SLT*	Slot table
SLOTON	Program selection save
TASKMAX	No. of multitasks
USRORG	User-designated origin
WRKDAT*	Maximum acceleration/deceleration setting

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

AREA*AT

User area

* is 1 to 8.

No. of Arrays

Integer1

Explanation

Designate an area (rectangle defined with two orthogonal coordinate points.)

A signal will be output if that area is outside the movement area (interference). or if the robot,s current position is within that area. Up to eight limits can be set using the parameters AREA*P1, AREA*P2, AREA*ME and AREA*AT.

Designate the user-defined area type. (Invalid/zone/interference = 0/1/2)

Zone: The dedicated output signal USRAREA will turn ON.

Interference: An error will occur.

Factory setting

0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

AREA*ME

User area

* is 1 to 8.

No. of Arrays

Integer1

Explanation

Designate an area (rectangle defined with two orthogonal coordinate points.)

A signal will be output if that area is outside the movement area (interference). or if the robot,s current position is within that area. Up to eight limits can be set using the parameters AREA*P1, AREA*P2, AREA*ME and AREA*AT.

Designate the mechanism No. for which the user-defined area is to be validated.

The mechanism No. is 1 to 4. but normally 1 is set.

Factory setting

0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

AREA*P1

User area

* is 1 to 8.

No. of Arrays

Real value8.

Explanation

Designate an area (rectangle defined with two orthogonal coordinate points.)

A signal will be output if that area is outside the movement area (interference). or if the robot,s current position is within that area. Up to eight limits can be set using the parameters AREA*P1, AREA*P2, AREA*ME and AREA*AT.

Designate the first point of the area.

(X,Y,Z, A, B, C, L1, L2) Unit: mm or deg.

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

AREA*P2

User area

* is 1 to 8.

No. of Arrays

Real value8.

Explanation

Designate an area (rectangle defined with two orthogonal coordinate points.)

A signal will be output if that area is outside the movement area (interference). or if the robot,s current position is within that area. Up to eight limits can be set using the parameters AREA*P1, AREA*P2, AREA*ME and AREA*AT.

Designate the second point of the area.

(X,Y,Z, A, B, C, L1, L2) Unit: mm or deg.

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
BZR

Buzzer ON/OFF

No. of Arrays

Integer1

Explanation

Designate whether the buzzer sound is ON or OFF.

(OFF/ON=0/1)

Factory setting

1
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
CBAU232

Communication setting

No. of Arrays

Integer1

Explanation

Baud rate

Factory setting

9600
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
CPRC232

Communication setting

No. of Arrays

Integer1

Explanation

Communication method (0: Non-procedural, 1:Procedural, 2: Data link)

Factory setting

0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
CPRTY232

Communication setting

No. of Arrays

Integer1

Explanation

Parity (0: None. 1: Odd. 2: Even)

Factory setting

2
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
CSTOP232

Communication setting

No. of Arrays

Integer1

Explanation

Stop bit (1, 2)

Factory setting

2
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
CTERM232

Communication setting

No. of Arrays

Integer1

Explanation

End code(O:CR1:CR+LF)

Factory setting

0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
HANDINIT

Hand initial state

No. of Arrays

Integer8

Explanation

Set the pneumatic hand I/F output for when the power is turned ON.

Factory setting

1, 0, 1, 0, 1, 0, 1, 0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
HANDTYPE

Hand type

No. of Arrays

Character string8.

Explanation

Set the single/double solenoid hand type and output signal No.

Set the signal No. after the hand type.

When D900 is set the signal No.900 and 901 will be output..

Factory setting

D900, D902, D904, D906, D908, D910, D912, D914
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV
INB

Stop input B contact designation

No. of Arrays

Integer1

Explanation

Chance the dedicated input (stop) between the A contact and B contact.

(A contact/B contact = 0/1)

Factory setting

0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

JOGJSP

Jog setting

No. of Arrays

Real value3.

Explanation

Designate the joint jog and step operation speed.

(Inching H, inching L, maximum override.)

Inching H: Feed amount when jog speed is set to High. Unit: deg.

Inching L: Feed amount when jog speed is set to Low. Unit: deg.

Maximum override: Operates at OP override I maximum override.

Factory setting

Setting value for each mechanism

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

JOGPSP

Jog setting

No. of Arrays

Real value3.

Explanation

Designate the orthogonal jog and step operation speed.

(Inching H, inching L, maximum override.)

Inching H: Feed amount when jog speed is set to High. Unit: deg.

Inching L: Feed amount when jog speed is set to Low. Unit: deg.

Maximum override: Operates at OP override I maximum override.

Factory setting

Setting value for each mechanism

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

JOGSPMX

Jog speed limit value

No. of Arrays

Real value1.

Explanation

Limit the robot movement speed during the teach mode. Unit mm/ s.

Even if a value larger than 250 is set, the maximum value will be limited to 250.

Factory setting

250.0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MEJAR

Joint movement range

No. of Arrays

Real value16.

Explanation

Set the overrun limit value for the joint coordinate system. Set the minus and plus directions.

(-J1, +J1,-J2, +J2, ... , -J8, +J8) Unit: mm

Factory setting

Setting value for each mechanism
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MEPAR

Orthogonal movement range

No. of Arrays

Real value6.

Explanation

Set the overrun limit value for the orthogonal coordinate system. Set the minus and plus directions.

(-X, +X, -Y, +Y, -Z, +Z) Unit: mm.

Factory setting

-10000, 10000, -10000, 10000, -10000, 10000,
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MEXBS

Tool base coordinates

No. of Arrays

Real value6.

Explanation

Set the default value for the relation of the orthogonal coordinate system and robot coordinate system.

(X,Y,Z, A, B, C) Unit: mm or deg.

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

MEXTL

Standard tool coordinates

No. of Arrays

Real value6.

Explanation

Set the default value for the relation of the tool coordinate value and flange.

(X,Y,Z, A, B, C) Unit: mm or deg.

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

PRGUSR

User base program

No. of Arrays

Character string 1

Explanation

Set the user base program.

Factory setting

Non

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

RETPATH

Automatic return setting

No. of Arrays

Integer1

Explanation

When the program is halted by stopping and the robot is moved with jog operation. etc. If the operation is to be resumed. set this to resume the program after returning to the stop position.

(Function valid/function invalid = 1/0)

Factory setting

1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SFC*AT

Free plane limit

* is 1 to 8.

No. of Arrays

Integer1

Explanation

This is the overrun limit set on a free plane.

Create a plane with three coordinate points, and set the area that does not include the origin as the outside-movement area. Up to eight limits can be set using the following three types of parameter SFC*P, SFC*ME and SFC*AT.

Designate the validity of the set free plane limit.

(Valid/Invalid = 1/0)

Factory setting

0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SFC*ME

Free plane limit

* is 1 to 8.

No. of Arrays

Integer1

Explanation

This is the overrun limit set on a free plane.

Create a plane with three coordinate points, and set the area that does not include the origin as the outside-movement area.

Up to eight limits can be set using the following three types of parameter SFC*P, SFC*ME and SFC*AT.

Designate the mechanism No. for which the free plane limit is to be validated.

The mechanism No. is 1 to 4. but normally 1 is set.

Factory setting

0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SFC*P

Free plane limit

* is 1 to 8.

No. of Arrays

Real value9.

Explanation

This is the overrun limit set on a free plane.

Create a plane with three coordinate points, and set the area that does not include the origin as the outside-movement area.
Up to eight limits can be set using the following three types of parameter SFC*P, SFC*ME and SFC*AT.

Designate three points for creating the plane.

(X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3) Unit: mm .

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SLOTON

Program selection save

No. of Arrays

Integer1

Explanation

Designate whether to save the selected program.

(Invalid/valid = 0/1)

If invalidated. the program selection will be canceled when the program is reset. The OP display will be .„P0000„.

Factory setting

1
©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

SLT*

Slot table

* is 1 to 32

No. of Arrays

Character string 4

Explanation

Set the operation conditions for each task. These are set when the

program is reset.

Designate the (program name, operation mode, starting conditions, and order of priority).

Program name: Selected program name

Operation mode: Continuous/1 cycle = REP/CYC

Starting conditions:

Normal/Error/Always = START/ERROR/ALWAYS

Order of priority: 1 to 31 (31 is the maximum)

Factory setting

1, REP, START, 1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

USRORG

User-designated origin

No. of Arrays

Real value8

Explanation

Designate the user-designated origin position.

(J1, J2, J3, J4, J5, J6, J7, J8) Unit: deg

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

WRKDAT*

Maximum acceleration/deceleration setting

* is 1 to 8

No. of Arrays

Real value7

Explanation

Set the hand conditions and work conditions for when OADL is set to ON within the program.

Up to eight conditions can be set. The condition combination is selected with the LOADSET command.

Set the work conditions. (Designate with the tool coordinate system.)

(Weight, size X, size Y, size Z, center of gravity X, center of gravity Y, center of gravity Z)

Unit: kg, mm

Factory setting

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

Alphabetical Overview of MELFA-BASIC-IV dedicated input/output parameters

Parameter

AIRERR1 .. AIRERR5
ATEXTMD
ATTOPMD
AUTOENA
BATERR
CLVLERR
CYCLE
HLVLERR
EMGERR
ERROUT
ERRRESET
HNDCNTL1 .. HNDCNTL5
HNDERR1 .. HNDERR5
HNDSTS1 .. HNDSTS5
IODATA
IOENA
JOGENA

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

AIRERR1 .. AIRERR5

Input:Mechanism 1 .. 5 pneumatic pressure error input signal

Function:

Request the pneumatic pressure error occurrence.

Output: Mechanism 1 .. 5 pneumatic error output signal

Function:

Outputs that a pneumatic pressure error is occurring.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ATEXTMD

Input:-

Function:-

Output: Teaching mode output

Function:

Outputs that the remote mode is entered.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

ATTOPMD

Input:-

Function:-

Output: Automatic mode output

Function:

Outputs that the automatic mode is entered.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

AUTOENA

Input:Automatic operation enable input

Function:

Disables automatic operation when inactive. It this signal is inactive. and the AUTO mode is selected, E5(

Output: Automatic operation enable output

Function:

Outputs the automatic operation enabled state.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

BATERR

Input:-

Function:-

Output: Battery voltage drop

Function:

Outputs that the battery voltage has dropped.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CLVLERR

Input:-

Function:-

Output: Warning level error output signal

Function:

Outputs that a warning level error is occurring

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

CYCLE

Input:Cycle stop input signal

Function:

Starts the cycle stop.

Output: In cycle stop operation output signal

Function:

Outputs that the cycle stop is operating.

Turns OFF when the cycle stop is completed.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC IV

EMGERR

Input:-

Function:-

Output: Emergency stop output signal

Function:

Outputs that an emergency stop is occurring.

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-BASIC

ERROUT

Input:Error No. output request

Function:

Outputs the error No. to the numeric value outpı

Output: Error No. output signal

Function:

Outputs that the error No. is being output to the

Factory setting

-1, -1

©·2005·EFR·GERMANY

COSIMIR® · MELFA-E

ERRRESET

Input:Error reset input signal

Function:

Releases the error state.

Output: Error occurring output sign

Function:

Outputs that an error has occurred.

Factory setting

2, 2

©·2005·EFR·GERMANY

COSIMIR® · MI

HLVLERR

Input:-

Function:-

Output: High level error

Function:

Outputs that a high level

Factory setting

-1, -1

©·2005·EFR·GE

COSIMI
HNDCNTL1 ..

Input:-

Function:-

Output: Mecha

Function:

Outputs the ge

...

Outputs the ge

Factory settin

-1, -1

©·2

C
HN

Max

* is

No.

Rea

Exp

Set

Up t

Set

(We

Unit

Fac

Star

©·2

C
HN

Inpu

Fun

Rec

Out

Fur

Ou

Fa

-1,