Main System Software

Release of Version 1.8

DENSO has updated main system software designed for -D and -E series of robots and WINCAPSII to Version 1.8 that supports new features such as easy teaching facility.

Contents

October 2001 DENSO WAVE INCORPORATED Factory Automation Division

1. Easy Teaching

Easy teaching is a new facility that enables data entry, program edition, and teaching of point coordinates from the teach pendant with ease of operation.

The easy teaching allows you to:

(1) Insert a motion command and its parameters (including destination points) to the desired program by one-touch operation.

(2) Edit motion commands and their parameters by one-touch operation.

(3) Execute and check programs you have edited before compiling them.

1.1 Inserting a motion command

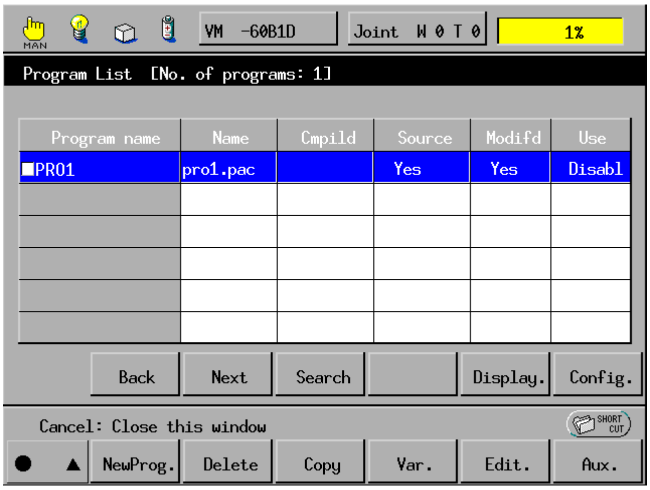
What follows is an operation flow for inserting a motion command by using the easy teaching facility.

■ Possible from the teach pendant only

Step 1 Set the mode selector switch to the MANUAL position.

Step 2 On the top screen, press [F1 Program].

The Program List window appears as shown below.



Program List [No. of programs: 1]

Program name	Name	Cmpild	Source	Modifd	Use
■PRO1	pro1.pac		Yes	Yes	Disabl

Cancel: Close this window

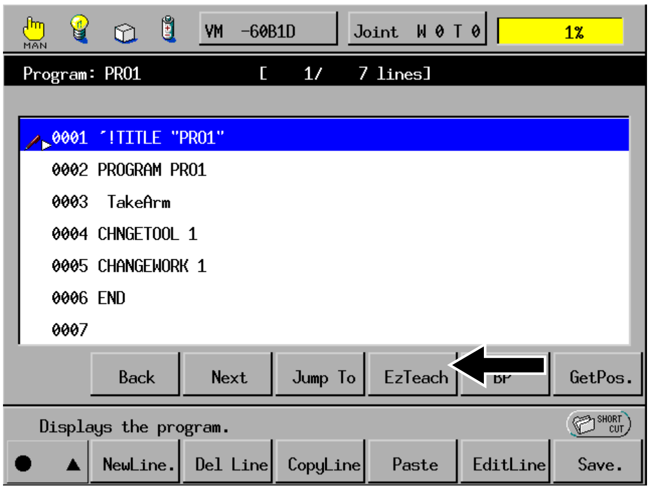
F1 F5

Select a program you want to edit and then press [F5 Edit.] or [Display.] (or [F1 NewProg.].

NOTE: To edit existing programs, use [F5 Edit.]; to check them, use [Display.]. To edit a new program, use [F1 NewProg.].

Step 3

The coding list of the selected program appears as shown below.



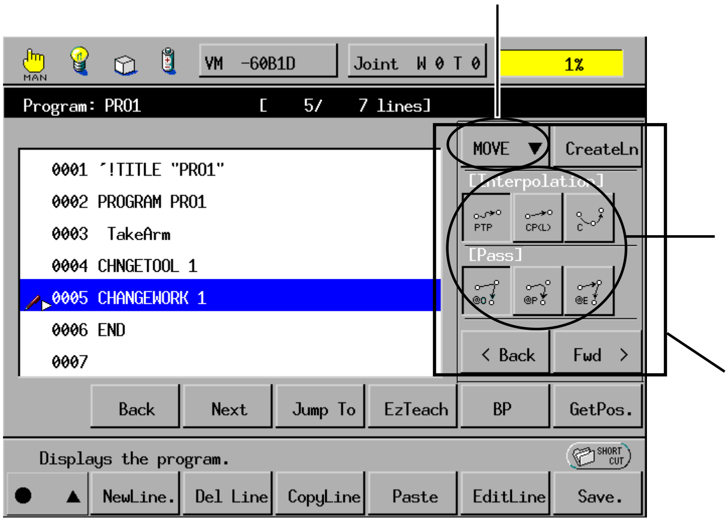
```
Program: PRO1            [   1/    7 lines]
0001 '!TITLE "PRO1"
0002 PROGRAM PRO1
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 END
0007
Displays the program.
```

Press [EzTeach] to call up the easy teaching window.

Step 4

Press parameter icons to set up the motion command parameters.

TIP: The specifications of parameter icons that seem to be depressed will be set up as effective parameter values.



Command select button (reserved for future expansion)

Parameter icons

Easy teaching window

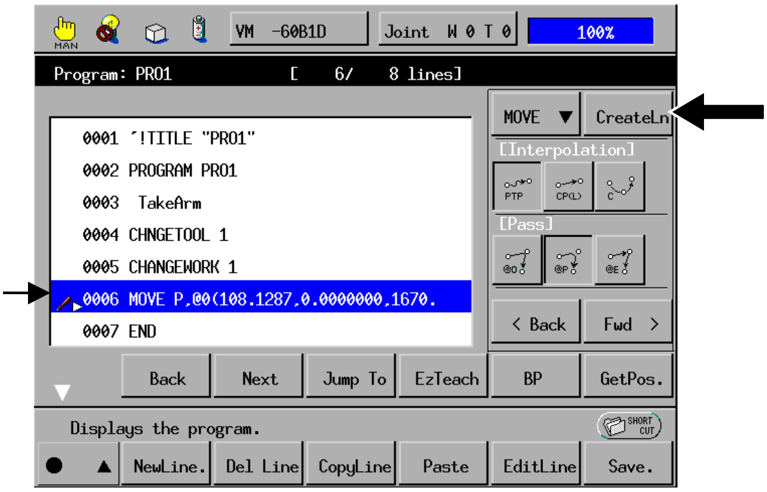
```
Program: PRO1            [   5/    7 lines]
0001 '!TITLE "PRO1"
0002 PROGRAM PRO1
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 END
0007
Displays the program.
```

NOTE: The Command select button is reserved for future expansion. It is currently disabled.

Step 5

Move the cursor to a line immediately preceding the line where you want to insert a motion command, then press [CreateLn].

By calculating the parameter icon information and the current robot position, the controller automatically creates an appropriate motion command. Then it inserts the command to the line next to the selected line.



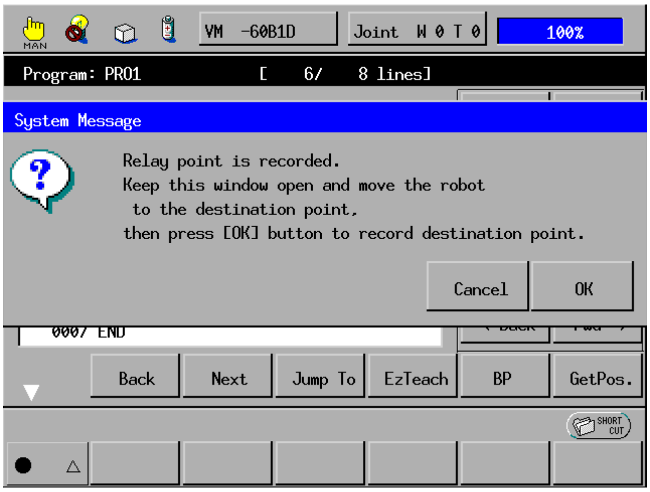
Motion command newly inserted

```
Program: PRO1                [    6/    8 lines]
0001 '!TITLE "PRO1"
0002 PROGRAM PRO1
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 MOVE P,@0(108.1287,0.0000000,1670.
0007 END
```

Displays the program.

Step 6

If you insert a motion command that defines interpolation along an arc (e.g., MOVE C), then pressing [CreateLn] will display the following message. In this case, keep this dialog open and move the robot arm to the destination point, and press [OK]. The command will be inserted.



Program: PRO1 [6/ 8 lines]

System Message

Relay point is recorded.
Keep this window open and move the robot
 to the destination point,
then press [OK] button to record destination point.

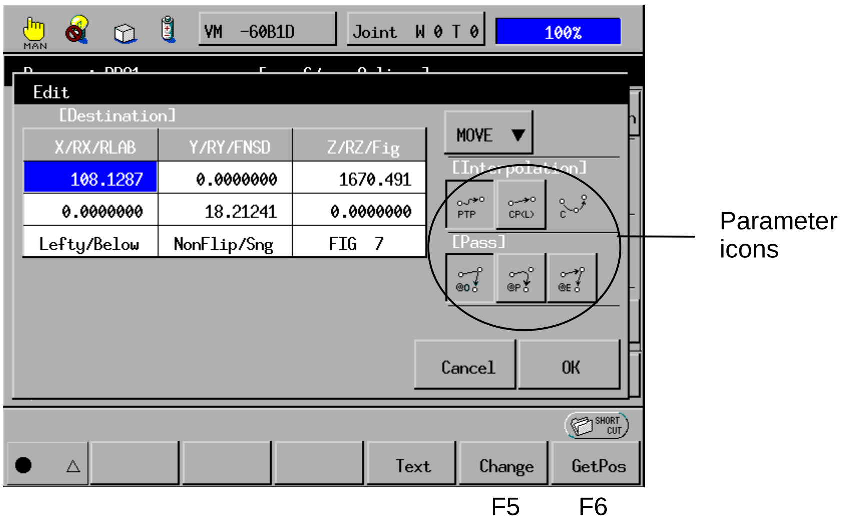
1.2 Editing a motion command

What follows is an operation flow for editing an existing motion command by using the easy teaching facility.

Step 1 Carry out Steps 1 through 3 in Section 1.1 to call up the easy teaching window.

Step 2 Move the blue cursor to a program line you want to edit, then press [F5 EditLine].

If the command on the selected line can be edited in the easy teaching window (refer to Section 1.5), then the parameter information will display according to the parameter icons.



[Destination]

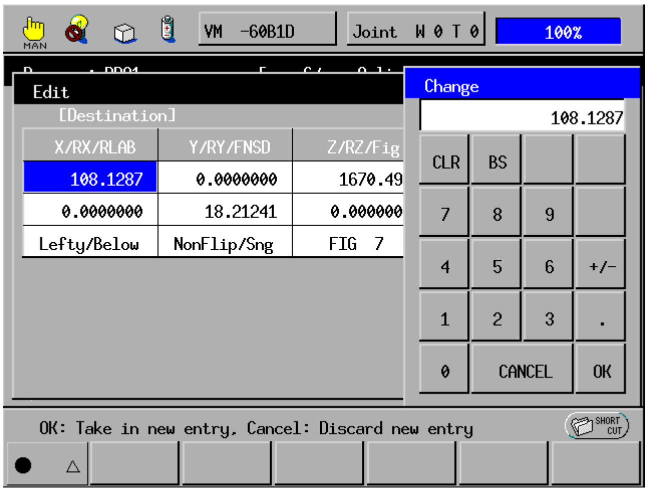
X/RX/RLAB	Y/RY/FNSD	Z/RZ/Fig
108.1287	0.0000000	1670.491
0.0000000	18.21241	0.0000000
Lefty/Below	NonFlip/Sng	FIG 7

When the window is called up, the parameter icons reflect the command information on the selected line.

You may change the parameters by using the parameter icons or change the destination point (or passing points) by pressing [F5 Change] (see Step 3) or [F6 GetPos] (See Step 4). After that, pressing [OK] will make the new settings go into effect.

Step 3 To enter the desired numeric value by using [F5 Change]:

Move the cursor to a field you want to change with the up-, down, left-, and right-arrow keys or jog dial, then press [F5 Change]. The numeric keypad will appear as shown below. Enter the desired numeric value.



[Destination]

X/RX/RLAB	Y/RY/FNSD	Z/RZ/Fig
108.1287	0.0000000	1670.49
0.0000000	18.21241	0.000000
Lefty/Below	NonFlip/Sng	FIG 7

108.1287

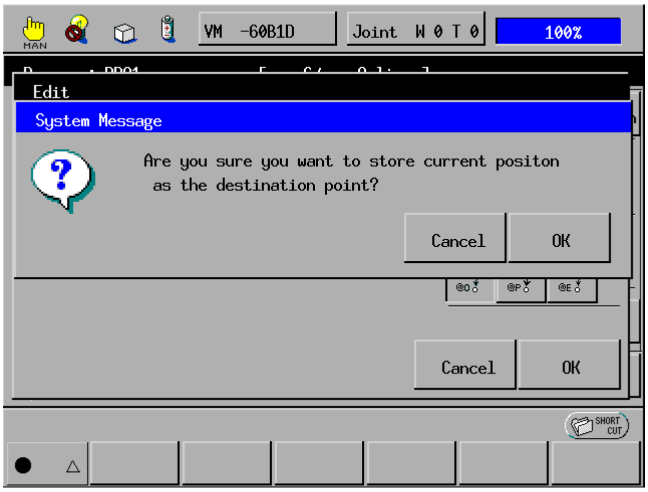
OK: Take in new entry, Cancel: Discard new entry

NOTE: If a motion command you selected defines interpolation along an arc (e.g. MOVE C), you may define the passing points and destination points by using the up- and down-arrow keys or jog dial.

Step 4

To get the current robot position by using [F6 GetPos]:

Press [F6 GetPos], and the controller will get the current robot position into the memory area and overwrite the line you are editing with the gotten data.

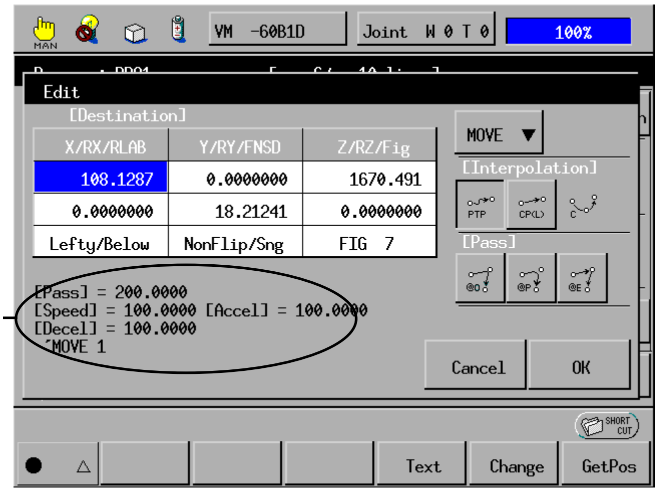


Are you sure you want to store current positon as the destination point?

NOTE: If you are editing a motion command that defines interpolation along an arc (e.g. MOVE C), a window will appear prompting you to choose either a passing point or destination point. Choose either one.

Step 5

Circled below are parameters that cannot be edited in this screen.



[Destination]

X/RX/RLAB	Y/RY/FNSD	Z/RZ/Fig
108.1287	0.0000000	1670.491
0.0000000	18.21241	0.0000000
Lefty/Below	NonFlip/Sng	FIG 7

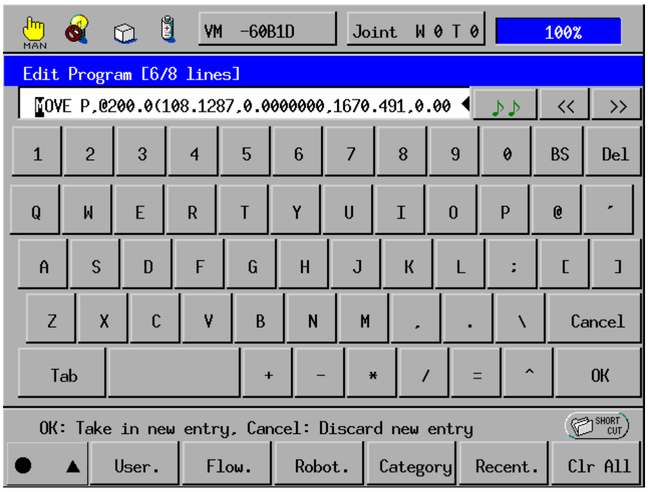
```
[Pass] = 200.0000
[Speed] = 100.0000 [Accel] = 100.0000
[Decel] = 100.0000
`MOVE 1
```

Parameters that cannot be edited here.

F4

To edit them, press [F4 Text]. The alphanumeric key screen will appear as shown in Step 6.

Step 6 Use the alphanumeric key screen shown below to edit parameters in the text mode.



Edit Program [6/8 lines]

```
MOVE P,@200.0(108.1287,0.0000000,1670.491,0.00
```

♪♪
<<
>>
1 2 3 4 5 6 7 8 9 0 BS Del
Q W E R T Y U I O P @ ´
A S D F G H J K L ; [ ]
Z X C V B N M , . \ Cancel
Tab + - * / = ^ OK
SHORT CUT
● ▲ User. Flow. Robot. Category Recent. Clr All

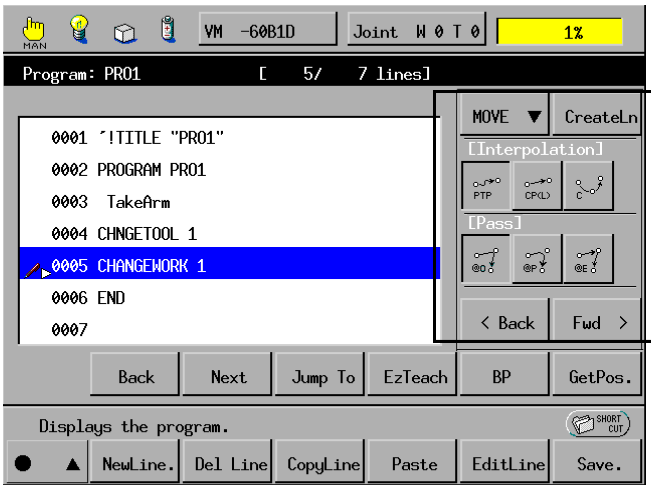
OK: Take in new entry, Cancel: Discard new entry

NOTE: Pressing [Cancel] in the above screen will cancel all settings made in the easy teaching window.

1.3 Executing motion commands

What follows is an operation flow for executing motion commands by using the easy teaching facility.

Step 1 Carry out Steps 1 through 3 in Section 1.1 to call up the easy teaching window.



```
Program: PRO1            [    5/    7 lines]
0001 '!TITLE "PRO01"
0002 PROGRAM PRO1
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 END
0007
```

Displays the program.

Easy teaching window

Move the blue cursor to a program line to be executed, then press [<Back] or [Fwd>].

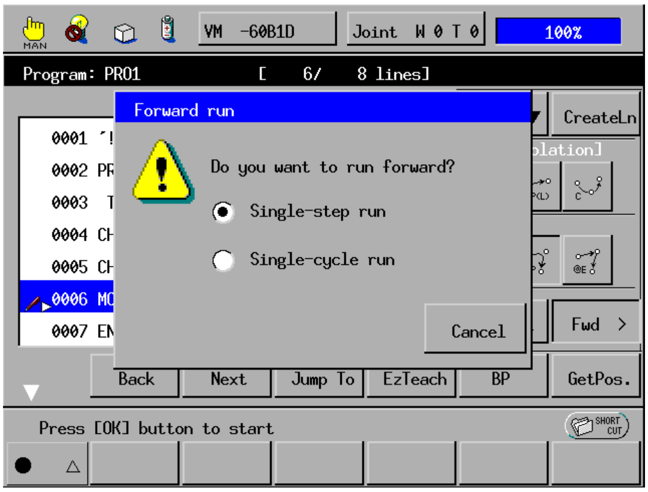
If a command on the selected line can be edited in the easy teaching window (refer to Section 1.5), then the program line will execute

[Fwd>]: Pressing this button will execute the command selected with the blue cursor.

[<Back]: Pressing this button will execute the command selected with the blue cursor if the command does not include a destination point. If the command includes a destination point, it will apply a destination point defined in a motion command preceding the selected command.

Step 2

Pressing [<Back] or [Fwd.>] button will call up the execution confirmation window shown below.



Program: PR01 [6/ 8 lines]

```
0001 '!
0002 PR
0003 T
0004 CH
0005 CH
0006 MO
0007 EN
```

Forward run

Do you want to run forward?

● Single-step run

○ Single-cycle run

Cancel

Press [OK] button to start

Choose "Single-step run" or "Single-cycle run." Then while holding down the deadman switch, press the OK button. The program will execute. Releasing either one of the deadman switch and OK button will stop the running program.

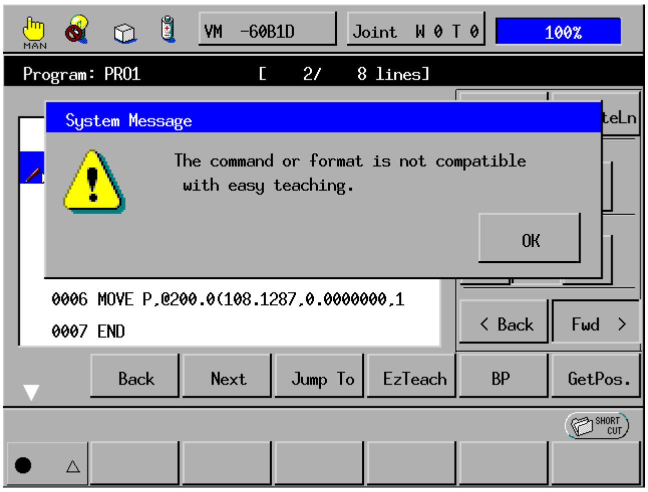
"Single-step run": Runs a single step of the selected program and stops the execution.

"Single-cycle run": Runs the selected program once from the beginning to the end or to any command unexecutable by easy teaching.

In either "Single-step run" or "Single-cycle run," you can stop the program run halfway just by performing any of stop operations (e.g., pressing the STOP key) during execution.

Tip

If you attempt to run any command that is unexecutable by easy teaching or contains a syntax error, then the following warning message will appear.



Program: PR01 [2/ 8 lines]

System Message

The command or format is not compatible with easy teaching.

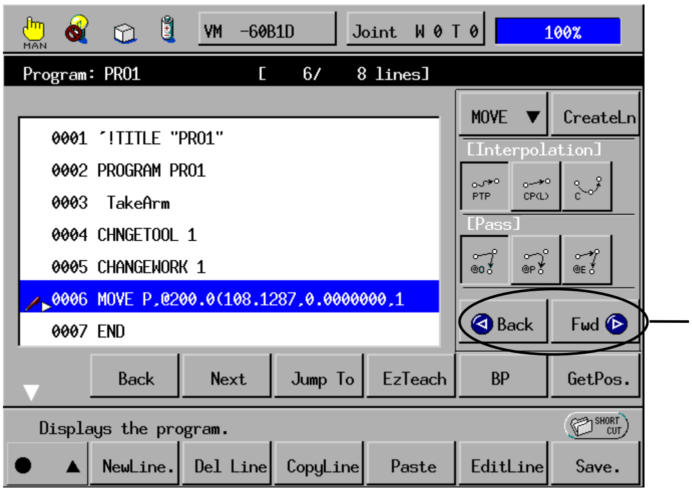
OK

```
0006 MOVE P,@200.0(108.1287,0.0000000,1
0007 END
```

Tip

When the easy teaching window is displayed, pressing the deadman switch will change the functions assigned to the left- and right-arrow cursor keys from the horizontal scrolling on a program line to the same function as the [<Back] and [Fwd>], respectively.

The [<Back] and [Fwd>] icons will be also changed as shown below.



```
Program: PRO1                [    6/    8 lines]
0001 '!TITLE "PRO1"
0002 PROGRAM PRO1
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 MOVE P,@200.0(108.1287,0.0000000,1
0007 END
```

Displays the program.

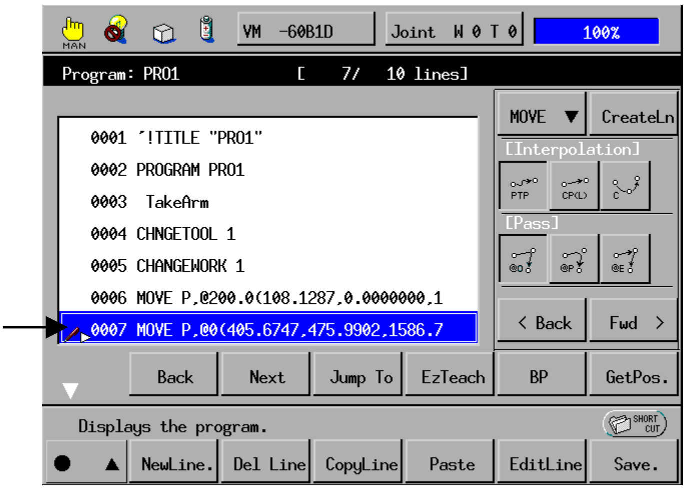
Changed like this.

1.4 Additional explanation about easy teaching

[1] Box frame on a program line

If you run a motion command by using the easy teaching window, a box frame may appear on a program line in addition to a blue cursor. The box frame indicates that the boxed line includes a destination point obtained by a command executed most recently.

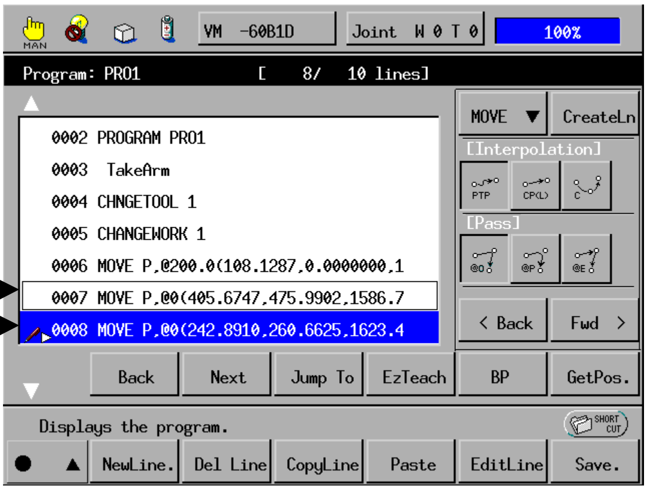
In the case of [Fwd>]: If you temporarily stop the following program at line 0007 being executed by [Fwd>], for example, then a box frame comes on line 0007 since the destination point obtained by the most recently executed motion command is written in line 0007.



```
Program: PR01              [    7/    10 lines]
0001 '!TITLE "PR01"
0002 PROGRAM PR01
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 MOVE P,@200.0(108.1287,0.0000000,1
0007 MOVE P,@0(405.6747,475.9902,1586.7
Displays the program.
```

Box frame and blue cursor are overlapped with each other.

If line 0007 is executed again with [Fwd>] by a single step, the blue cursor moves to line 0008 but the box frame stays on 0007. This is because the destination point obtained by the most recently executed motion command is written in line 0007.

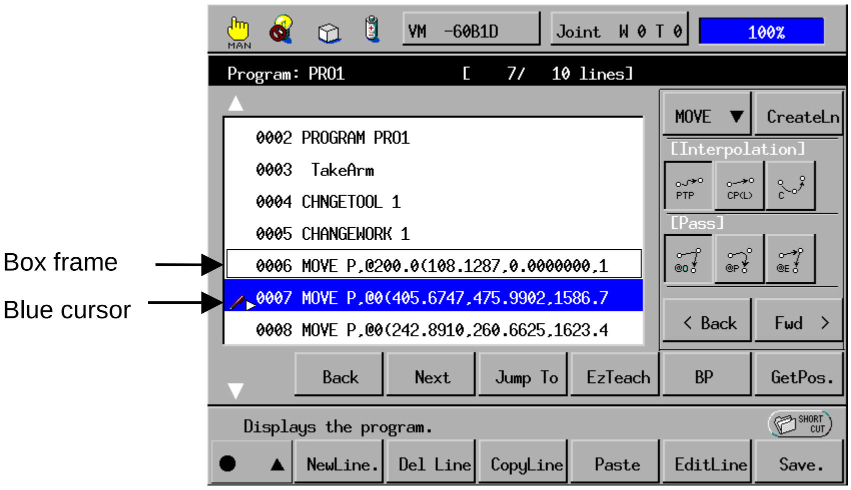


```
Program: PR01              [    8/    10 lines]
0002 PROGRAM PR01
0003  TakeArm
0004 CHNGETOOL 1
0005 CHANGEWORK 1
0006 MOVE P,@200.0(108.1287,0.0000000,1
0007 MOVE P,@0(405.6747,475.9902,1586.7
0008 MOVE P,@0(242.8910,260.6625,1623.4
Displays the program.
```

Box frame

Blue cursor

In the case of [<Back]: If line 0007 is executed with [<Back] by a single step (or the execution is stopped temporarily), then the blue cursor stays on line 0007 but the box frame comes on line 0006. This is because the destination point is still in the motion command immediately preceding the line marked with the blue cursor.



```
Program: PRO1                [   7/   10 lines]
0002  PROGRAM PRO1
0003   TakeArm
0004  CHNGETOOL 1
0005  CHANGEWORK 1
0006  MOVE P,@200.0(108.1287,0.0000000,1
0007  MOVE P,@0(405.6747,475.9902,1586.7
0008  MOVE P,@0(242.8910,260.6625,1623.4
```

Displays the program.

[2]   About the GetPos function

You may use the GetPos function when a coding list of the selected program or the easy teaching window is displayed.

The GetPos function may get the current robot position even expressed in immediate operand format*, provided that the easy teaching window is displayed.

*Immediate operand format refers to writing command operands in numeric values as shown below, not in variables.

```
MOVE P,@0(1025.721,-354.7859,1026.708,-179.9987,65.01270,160.9215,5)
```

Immediate operands

1.5 Applicability of parameters in easy teaching

The table given below shows whether parameters of motion commands and coordinate system change commands are applicable to editing, displaying of the easy teaching window, and execution in easy teaching.

Parameters of MOVE Command

	Interpolation			Initial passing start position				
	P (PTP)	L (CP)	C (Arc)	@0	@P	@E	@n	None
Editing	Yes	Yes	Yes	Yes	Yes	Yes	Yes/No	Yes/No
Displaying of the easy teaching window	Yes	Yes	Yes	Yes				
Execution in the easy teaching window	Yes	Yes	Yes	Yes				
Remarks	Unchangeable to C.		Unchange-able to P or L.	—			Changeable to @0, @P, or @E. Shown in a text format in the easy teaching window. (In the case of "None," nothing displays.)	
				If you edit any arc motion, your change in the initial passing start position defined as a destination point will be reflected on parameter icons.				

	Destination point (passing point)		Speed setting			NEXT option	Comments
	Immediate operand	Others	SPEED	ACCEL	DECEL		
Editing	Yes	No	No	No	No	No	No
Displaying of the easy teaching window	Yes	No	Yes	Yes	Yes	Yes	Yes
Execution in the easy teaching window	Yes	No	Yes	Yes	Yes	Yes	Yes
Remarks	—		Shown in a text format in the easy teaching window.			Shown in a text format in the easy teaching window.	

Motion commands applicable to [Fwd>] only in easy teaching

CHANGETOOL , CHANGEWORK
SPEED, ACCEL, DECEL
JSPEED, JACCEL, JDECELL

NOTES

(1) If a command contains any parameter not listed in these tables, then no easy teaching window will display. Neither is such a command executable.

(2) When an easy teaching window is displayed:

- Pressing the OK key will eliminate unnecessary space codes in parameters.
- A very small value entered to a destination point or initial passing start position may be automatically expressed in floating point or double-precision floating point.

1.6 Notes for easy teaching

(1) At the start of a pass motion, the robot motion in easy teaching may be slightly different from that in Teach check, Auto, or External mode.

(2) If a figure flag is set to -1, the final robot figure may be different between [Fwd>] and [<Back] run.

(3) If you stop the robot temporarily during arc motion, you may restart it from the stopped position only in the same direction ([Fwd>] or [<Back] run) as before the temporary stop. Restart in the opposite direction is not allowed.

Once you attempt to restart the robot in the opposite direction, Error 27BE ("Arc motion not allowed from the current position") will occur. After that, restart is no longer possible even in the same direction.

(4) No arc motion cannot start unless the current robot position is the same as a destination point defined in the immediately preceding motion command. If you attempt to do so, Error 27BE ("Arc motion not allowed from the current position") will occur.

(5) If any motion command is executed in easy teaching, then arm semaphore needed for that motion will be automatically obtained.

The arm semaphore will be released when:

- The robot arm has reached the final destination point after a single-step run or single-cycle run.
- The coding list of the selected program is closed.
- In the Shortcut Menu, the [F8 ProgRst.] has been performed.
- In Teach check, Auto, or External mode, the [F7 ProgRst.] has been performed, regardless of whether "This program only" or "All programs" is selected.
- A Level 3 error has occurred except errors caused when you are using TP/MP/OP.
- A Level 4 error or higher one has occurred.
- *Program reset* signal is received via I/O line in Standard mode (only when negative values are set in the data area).
- *Program reset* & *Operation preparation start* signals are received via I/O line in Compatible mode.

IMPORTANT

Once you temporarily stop the robot halfway through a single-step run or a single-cycle run, then manual operation or operation mode switching (between Joint mode, X-Y mode, and Tool mode) is no longer possible. This is because the currently running program has held an arm semaphore.

To make manual operation or operation mode switching possible, release the arm semaphore by carrying out [F8 ProgRst.] in the Shortcut Menu.

1.7 Addition to the configuration list

The following item has been added to the configuration list in the User Preferences window given in the SETTING-UP MANUAL, Chapter 2, Section 2.11.

No.	Items	Factory default	Powering-on default	Description	Comments
199	Arc motion allowance in easy teaching	100	100	Positional error allowance for arc motion in easy teaching	Do not change this setting if not necessary.

2. Enhanced Commands

2.1 Parameter <Storage variable> added to WAIT command

Refer to the PROGRAMMER’S MANUAL, page 12-61.

Parameter <Storage variable> has been added to the WAIT command.

WAIT (Statement) [Conforms to SLIM]

Function

Stops program processing conditionally.

Syntax

```
WAIT <Conditional expression> [,<Timeout period>[,<Storage variable>]]
```

Description

This statement stops program processing until <Conditional expression> is satisfied.

If <Timeout period> is set, control stops the execution of a WAIT statement after the designated time elapses and proceeds to the next command. Infinite stoppage can be avoided by using this.

<Timeout period> is expressed in ms.

The reevaluation interval for monitoring <Conditional expression> or <Timeout period> depends on the priority of the task.

In Version 1.8 or later, when <Storage variable> is set, the WAIT command will assign TRUE (1) or FALSE (0) to the designated variable if control passes out of the WAIT statement by the satisfied <Conditional expression> or by timeout, respectively.

Related Terms

DELAY

Example

```
DEFINT li1, li2, li3, li4, li5
WAIT li1 = 1        'Wait until li1 = 1 is satisfied.
WAIT li2 = 0, 2000  'Wait until li2 = 0 is satisfied. Even if it is not satisfied
                    'after 2 seconds, the system proceeds to the next statement.
WAIT li3 = li4, li5 'Wait until li3 = li4 is satisfied. Even if it is not
                    'satisfied after the time of li5,
                    'the system proceeds to the next statement.
WAIT IO[10] = ON    'Wait until the 10th IO comes ON.
```

[Version 1.8 or later]

```
WAIT li3 = li4, li5, li6  'Wait until li3 = li4. If the conditional expression
                          'is not satisfied within li5 period, pass control to
                          'the next statement and assign FALSE to li6.
                          'If satisfied within li5 period, pass control to the
                          'next statement and assign TRUE to li6.
```

Notes

If you have set <Timeout period>, be careful with the fact that the timer will be running during temporary stop caused by an instantaneous stop in command execution.

2.2 Parameter <Error detect condition> and other new enhancements added to AREA command

Refer to the PROGRAMMER'S MANUAL, page 9-2.

Parameter `<Error detect condition>` and other new enhancements have been added to the `AREA` command.

AREA (Statement)

Function

Declares the area where an interference check should be performed.

Syntax

```
AREA <Area number>, <Position>, <Vector>, <I/O number>, <P variable number
for interference position> [,<Error output>]
```

Description

This statement declares an interference check area.

The number of areas that can be declared with `<Area number>` is 8, from 0 to 7.

`<Position>` is the center position and angle of an interference check area.

`<Vector>` designates the interference check area zone.

The side length of an area becomes twice each component of `<Vector>`.

`<I/O number>` is an I/O signal number that will be set when interference occurs. The I/O status will be retained until `RESETAREA` is executed or the I/O is RESET. **In Version 1.8 or later, `<I/O number>` can be expressed in variable such as IO104 or IO[104]. Also in Version 1.8 or later, setting -1 to `<I/O number>` prohibits output to the I/O line.**

`<P variable number for interference position>` is a position variable number in which you want to save the coordinates where area interference occurs. **In Version 1.8 or later, `<P variable number for interference position>` can be expressed in position variable such as P55 or P[55]. Also in Version 1.8 or later, setting -1 to `<P variable number for interference position>` prohibits assignment to the position variable.**

In Version 1.8 or later, `<Error output>` is any of the following numbers.

`<Error output>`	The system will detect it as an error when:	Error signal output
0	**The robot arm invades the defined area.**	**No**
1		**Yes**
2		**Yes (You may switch to Manual mode and operate the robot manually for recovery.)**
3	**The robot arm exits from the defined area.**	**No**
4		**Yes**
5		**Yes (You may switch to Manual mode and operate the robot manually for recovery.)**

To check interference, the system compares the cube defined as an interference check area with the origin of the currently active tool coordinates. If the origin of the tool coordinates is inside the interference check area, then the system determines it as interference.

If area interface is detected and the I/O signal is set to active, then the system will get the origin of the current tool coordinates in the user coordinates into a position variable number specified by `<P variable number for interference position>`. Usually, the origin of the tool coordinates lies on the surface of the cube; however, if the origin lies inside the cube at execution of SETAREA, the system gets that position into `<P variable number for interference position>`.

These parameters above may be set from the teach pendant of in WINCAPSII.

2.3 Parameters added to <ParameterType> of set_button command

Refer to the PROGRAMMER'S MANUAL, page 13-27.

Variable type parameters have been added to `<ParameterType>` of `set_button` command that is one of the TP operation screen programming commands.

set_button

Function

Sets button parameters.

Syntax

```
set_button <ButtonNumber>,<ParameterType>,<NewValue>
```

`<ButtonNumber>` Number indicating the button location in all button arrangement on a TP operation panel.

`<ParameterType>` Button attributes including color, position and others. (See the table below.)

`<NewValue>` Parameter value for making new settings (See the table below.)

`<Parameter Type>`	Explanation	`<NewValue>` **(Note 1)**
1	Upper left X coordinate	0 to 640 in dots
2	Upper left Y coordinate	0 to 350 in dots
3	Lower right X coordinate	0 to 640 in dots
4	Lower right Y coordinate	0 to 350 in dots
5	Button type	0: None 1: Label 2: Line 17: 2D button (Change variable) 18: 3D button (Change variable) 19: 3D button (Change variable) 20: Circle (Change variable) 33: 2D LED (lamp) 34: Circle LED (lamp) 35: 3D button (Change IO)
6	Button status	**0: Center characters** **1: Left-justify characters** **2: Right-justify characters**

`<Parameter Type>`	Explanation	`<NewValue>` **(Note 1)**
7	Background color	0: Black 1: Blue 2: Green 3: Cyan 4: Red 5: Magenta 6: Brown 7: Light gray 8: Gray 9: Light blue 10: Light green 11: Light cyan 12: Light red 13: Light magenta 14: Yellow 15: White
8	Text color	0: Black 1: Blue 2: Green 3: Cyan 4: Red 5: Magenta 6: Brown 7: Light gray 8: Gray 9: Light blue 10: Light green 11: Light cyan 12: Light red 13: Light magenta 14: Yellow 15: White
9	Usable state	0: Disable 1: Enable
10	Visible/invisible state	0: Invisible 1: Visible
11	Variable type **(Note 2)**	**1: Integer** **2: Floating point** **3: Double-precision floating point** **4: Character string (up to 32 characters)**
12	Variable number	Variable number that may be changed by the change variable button.
13	I/O number	Variable number that may be changed by the change I/O button. (128 to 511)
14	Display page number	Page number in which the buttons are displayed.

(Note 1) Always enter integers to `<ParameterType>`. Any other value will cause a "data tag error."

(Note 2) A floating-point or double-precision floating-point number occupies 13- or 22-character space, respectively. Take it into account and reserve suitable spaces when programming a numeric entry button using floating point or double-precision variable.

2.4 Newly added commands

NORMTRN and SetCPSpdMode have been added to the Pose data transformation functions and Arm movement PAC library, respectively.

NORMTRN (Function)

Refer to the PROGRAMMER'S MANUAL, page 15-28.

Function

Normalizes homogeneous-transformation data.

Syntax

```
NORMTRN (<H-TransData>)
```

Description

This function normalizes data designated by <H-TransData>.

"Normalize" refers to transforming an orient vector to orthogonal vector against the base approach vector or transforming an approach or orient vector to its norm vector.

Example

```
DEFTRN 1t1,1t2
1t1=NORMTRN((350,0,450,0,1,0,0,0,-1))

1t1=NORMTRN(1t2)
```

SetCPSpdMode (Library)

Refer to the PROGRAMMER'S MANUAL, page 22-50.

Function

Keeps the TCP speed (Tool end speed in CP motion) constant or restores it to the default.

Syntax

```
SetCPSpdMode<SetValue>
```

Description

If you specify a CP motion involving the rotation of the robot hand, then the system will automatically decrease the TCP speed according to the rotation angle by default. Setting 1 to <SetValue> may keep the TCP speed constant as long as the rotation angle is less than the specified value.

Example

```
CALL SetCPSpdMode(0)    'Restore to the default.

CALL SetCPSpdMode(1)    'Set the TCP speed to"1."
```

Notes

If <SetValue> is set to 1, the rotation speed will vary depending upon the robot motion in order to keep the TCP speed constant. If you specify such a motion that will exceed the rotation speed limit, then the system will issue a warning message and run the robot while decreasing the TCP speed.

3. Error Level Setting for Logging Errors

Refer to the SETTING-UP MANUAL, page 5-132.

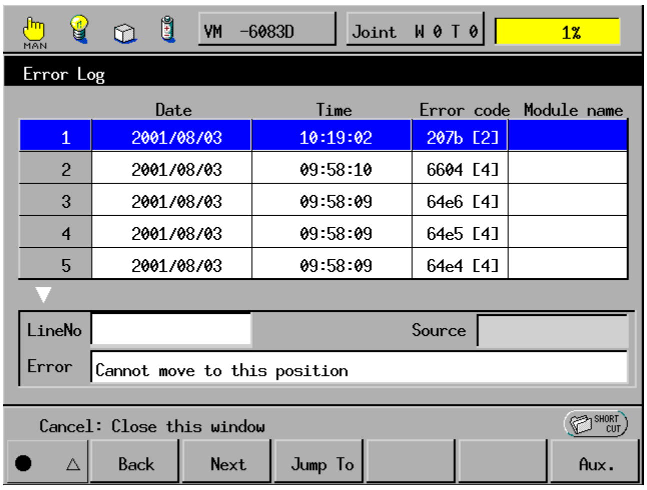
You may choose which level errors should be logged.

Displaying the error log

Access: [F6 Set]—[F2 Log.]

Displays the error log.

(1) Press [F2 Log.] in the Settings (Main) window, and the Log window will appear as shown below.



Error Log

	Date	Time	Error code	Module name
1	2001/08/03	10:19:02	207b [2]	
2	2001/08/03	09:58:10	6604 [4]	
3	2001/08/03	09:58:09	64e6 [4]	
4	2001/08/03	09:58:09	64e5 [4]	
5	2001/08/03	09:58:09	64e4 [4]	

LineNo: Source:

Error: Cannot move to this position

Cancel: Close this window

F1 F2 F3 **F6**

In the bottom line of the log list appears the error details of the selected error number.

The LineNo and Source areas show nothing.

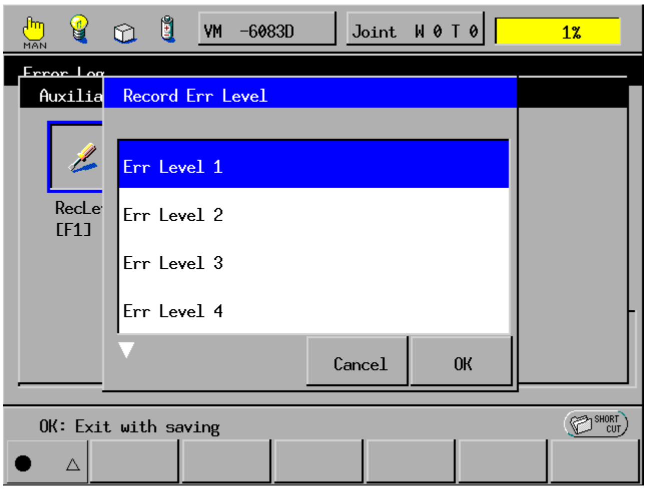
(2) To check older log, scroll the log list with the cursor keys, jog dial, [F1 BACK], or [F2 NEXT]. Or, press [F3 Jump To] to call up the numeric keypad where you enter the line number of the desired error log. Doing so will directly call up the target log.

(3) In Version 1.8 or later, you may choose which level errors should be logged. In the Error Log window shown on the previous page, press [F6 Aux.] and [F1 ErrLvl] to call up the Record Err Level window shown below.

Choose the desired error level. The system will log errors at the level you have chosen or higher errors.

NOTE: Errors at levels lower than the specified here will not be logged but it does not mean that such errors will no longer occur.

NOTE: The "Emergency stop ON 600C (Error level 0)" will be always logged regardless of the error level setting.



Record Err Level

Err Level 1

Err Level 2

Err Level 3

Err Level 4

OK: Exit with saving

4. Error Conditions Added for Area Interference Detection

Refer to the SETTING-UP MANUAL, page 4-45.

In earlier versions of the main system software, only when the origin of the tool coordinates invades the interference check area, the system interprets it as an error. In Version 1.8 or later, the system may detect also when the origin exits from the interference check area. You may choose either "invading the interference check area" or "exiting from it." This feature allows you to use the AREA statement for setting the motion area.

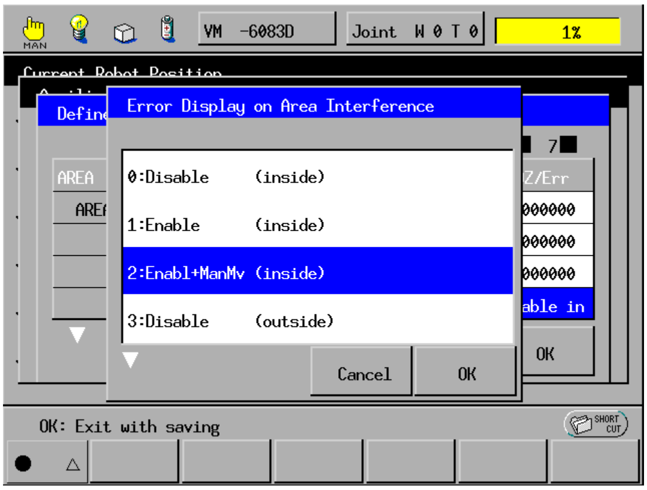
Interpreting a detected area interference as an error

The moment the origin of the tool coordinates interferes with the active interference area, the system may detect it as an error and cut the motor power off to prevent the arm from proceeding into the area further.

In Version 1.8 or later, you may choose any of the following items on the Error Display on Area Interference window.

Items	The system will detect it as an error when:	Error signal output
0: Disable (inside)	The robot arm invades the defined area.	No
1: Enable (inside)		Yes
2: Enable+ManMv (inside)		Yes (You may switch to Manual mode and operate the robot manually for recovery.)
3: Disable (outside)	The robot arm exits from the defined area.	No
4: Enable (outside)		Yes
5: Enable+ManMv (outside)		Yes (You may switch to Manual mode and operate the robot manually for recovery.)

Access: [F2 Arm]—[F6 Aux.]—[F6 Area.]



Error Display on Area Interference

0:Disable (inside)
1:Enable (inside)
2:Enabl+ManMv (inside)
3:Disable (outside)

OK: Exit with saving

NOTE: Once an area interference error occurs, the system cuts off the motor power so that the origin of the tool coordinates remains in the area. If you attempt to turn the motor power on under this state, an error will occur again and the motor power will be cut off. You need to disable the error detection, turn the motor power on, move the origin of the tool coordinates out of the area by manual operation, and then enable the error detection again.

In Version 1.8 or later, if you set any of items 3 through 5 (Detection when the robot arm exits from the defined area) for two check areas or more concurrently, then the system will interpret the overlapped area as a motion area. If no overlapped area exists, the robot cannot run, so you need to modify the settings.

For the setting procedure to enable or disable the error detection, refer to the SETTING-UP MANUAL, Chapter 4, Section 4.1.2 "[8] Interpreting a detected area interference as an error."

5. Parameters Added to the Configuration List in the User Preferences Window

Refer to the SETTING-UP MANUAL, page 2-56.

The "TCP speed (Tool end speed in CP motion)" and "TOOL/WORK resume" have been added to the configuration list that may be displayed in the User Preferences window of the teach pendant.

Access: Top screen—[F2 Arm]—[F6 Aux.]—[F7 Config.]

5.1 TCP speed (Parameter No. 197)

If you specify a CP motion involving the rotation of the robot hand, then the system will automatically decrease the TCP speed (Tool end speed in CP motion) according to the rotation angle by default. This causes some problems that the robot does not run in the specified speed or at constant speed.

In Version 1.8 or later, it is possible to keep the TCP speed constant by setting Parameter No. 197 to 1. If you specify such a motion that will exceed the rotation speed limit, then the system will issue a warning message and run the robot while increasing/decreasing the TCP speed.

Settings	0: Conventional speed control (Factory default)
	1: Constant TCP speed

5.2 Resume function for TOOL/WORK coordinate settings (Parameter No. 198)

In earlier versions, the system will not retain TOOL/WORK coordinates declared or defined in programs if the power is turned off. Accordingly, to make a same job, you need to set up the same operation environments every time when restarting the robot.

Setting Parameter No. 198 to 1 will make the system retain those operation environments. When the power is on, they will be resumed.

Settings	0: No resume (Factory default)
	1: Resume

6. Enhancements for WINCAPSII Version 1.8

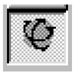
Updating from Version 1.7 to 1.8 will bring the following enhancements in WINCAPSII.

6.1 "Viewpoint Ctrl" button added to the arm manager screen

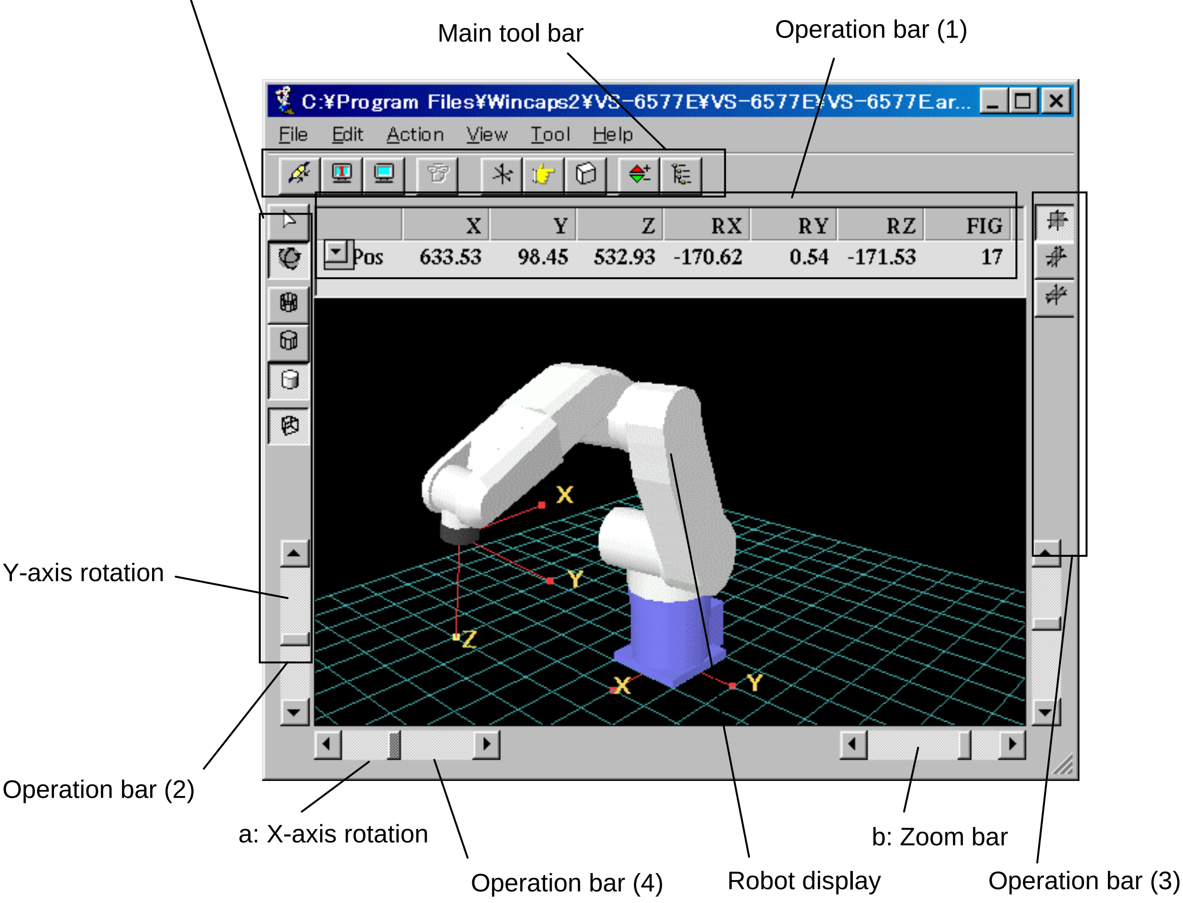
Refer to the WINCAPSII GUIDE, page 8-1.

To improve the arm manager operability, Version 1.8 or later has added "Viewpoint Ctrl" button to operation bar (2). This allows you to move your viewpoint of 3D images on the screen and zoom in/out the image by using your mouse.

"Viewpoint Ctrl" button



- Left-click & drag: Moves your viewpoint.
- Right-click & drag: Zoom in/out
- Left click with SHIFT held down: Choose an object.



C:¥Program Files¥Wincaps2¥VS-6577E¥VS-6577E¥VS-6577Ear...

File Edit Action View Tool Help

	X	Y	Z	RX	RY	RZ	FIG
Pos	633.53	98.45	532.93	-170.62	0.54	-171.53	17

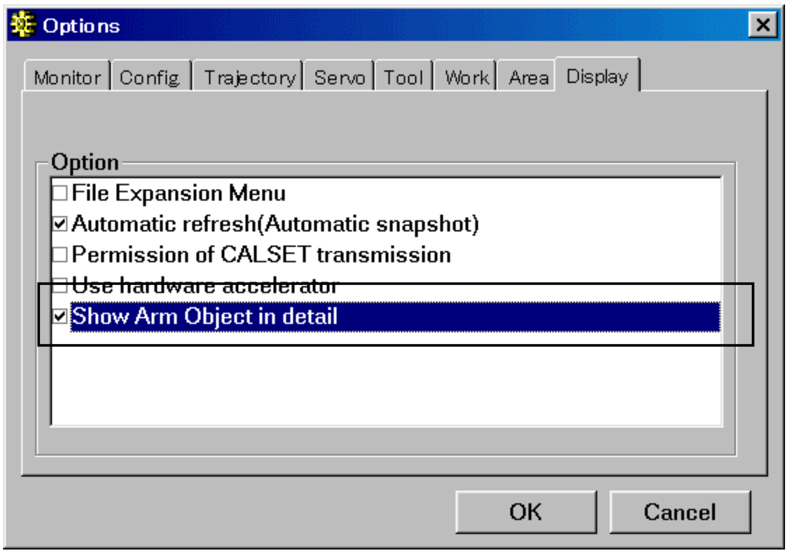
Fig. 8-1 Arm Manager Window

6.2 Detailed arm object added to the Options menu

Refer to the WINCAPSII GUIDE, pages 8-22 to 24.

The new item "Show Arm Object in Details" has been added to the Display tab of the Options menu that may be called up by [Arm Manager]—[Tool]—[Options]—[Display].

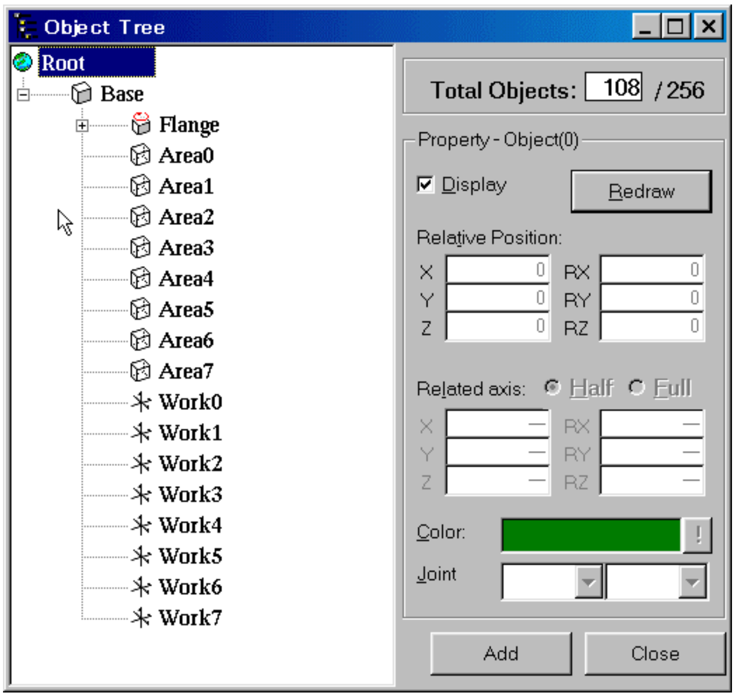
If you keep "Show Arm Object in Details" active, then Arm Manager will show all robot objects from the base to flange in the object tree through the entire hierarchy level. This setting will go into effect when WINCAPSII restarts.



Option

□ File Expansion Menu
☑ Automatic refresh(Automatic snapshot)
□ Permission of CALSET transmission
□ Use hardware accelerator
☑ Show Arm Object in detail

Fig. 8-24 Display Tab (Options Dialog Box)



- Root
  - Base
    - Flange
    - Area0
    - Area1
    - Area2
    - Area3
    - Area4
    - Area5
    - Area6
    - Area7
    - Work0
    - Work1
    - Work2
    - Work3
    - Work4
    - Work5
    - Work6
    - Work7

Total Objects: 108 / 256

Property - Object(0)

☑ Display

Relative Position:

X	0	RX	0
Y	0	RY	0
Z	0	RZ	0

Related axis: ● Half ○ Full

X	—	RX	—
Y	—	RY	—
Z	—	RZ	—

Color: 🟩

Joint

Fig. 8-26 Object Tree Dialog Box

7. Added/Updated Error Code Tables

Refer to the ERROR CODE TABLES.

(1) Listed below are error codes added or updated in Version 1.8.

Code	Message	Level	Description	Remedy
27BE	Can't start arc motion from here.	2	To start an arc motion in easy teaching, the current robot position should be the same as a destination point defined in the immediately preceding motion command.	Before staring an arc motion, move the robot position to a destination point defined in the immediately preceding motion command.
609F	Power module failure	5	An error for an unknown axis occurs in the power module.	This error may cause a serious damage in the controller. Be sure to follow the steps below. 1) Check that the controller is installed correctly as specified in the INSTALLATION & MAINTENANCE GUIDE. 2) Check that all power modules are inserted correctly into the specified slots. 3) Check the current ambient temperature. 4) Clean up the fan filters.
60DB	Decreased TCP speed	1	The specified rotation is so much that the robot cannot run at the specified TCP speed. The TCP speed has been decreased.	Set the smaller rotation angle. If it is impossible, decrease the whole robot speed.
6449	J1 encoder counter error 2	5	J1 encoder counter error (Coarse data)	Restart the controller and CALSET J1.
644A	J2 encoder counter error 2	5	J2 encoder counter error (Coarse data)	Restart the controller and CALSET J2.
644B	J3 encoder counter error 2	5	J3 encoder counter error (Coarse data)	Restart the controller and CALSET J3.
644C	J4 encoder counter error 2	5	J4 encoder counter error (Coarse data)	Restart the controller and CALSET J4.
644D	J5 encoder counter error 2	5	J5 encoder counter error (Coarse data)	Restart the controller and CALSET J5.
644E	J6 encoder counter error 2	5	J6 encoder counter error (Coarse data)	Restart the controller and CALSET J6.
644F	J7 encoder counter error 2	5	J7 encoder counter error (Coarse data)	Restart the controller and CALSET J7.

Code	Message	Level	Description	Remedy
6450	J8 encoder counter error 2	5	J8 encoder counter error (Coarse data)	Restart the controller and CALSET J8.
77F8	TSR start canceled	2	The power was turned on with the deadman switch being held down, so no supervisory task has not started. NOTE: This message will appear even when the supervisory task function is disabled if you turn the power on with the deadman switch being held down.	To start a supervisory task when turning the power on, do not press the deadman switch. If this message appears when the supervisory task function is disabled, then delete it and proceed to your task.

(2) Addition to *2 given in the ERROR CODE TABLES, on page 1.

*2 If any of errors 6071 to 607B and 6671 to 667B (software motion limit over, out of motion space, or singular point), 607F (figure mismatch**), 6081 to 6088 (Jx command speed limit over), and 6AF3 (Interference area detected by J1, 2, 3)** occurs when the robot is in operation in Manual mode, then the robot controller will not cut off the motor power.