

MotoSim Commands & Syntax

Introduction

This document reviews some basic INFORM commands that can be used for robot jobs as well as some unique commands that can be used to control the function of your simulation. These MotoSim “MDL” commands appear as comments in the robot job so they will not affect the operation of your program when run on a real controller. However, they can improve the realism of your simulation by showing or hiding models and moving models.

MotoSim Syntax

- | | |
|----------------------------------|---|
| 1. WAIT IN#(1)=ON | Waiting for input #1 to be turned on |
| 2. DOUT OT#(1) ON | Turns output#1 on |
| 3. DOUT OT#(1) OFF | Turns output#1 off |
| 4. CALL JOB:OUTPUT | Calls job names "OUTPUT" |
| 5. CALL (variable) | Call a job based on the value of a variable. If the command was entered as CALL B010, and B010 had a value stored as 12, a job named 12 would be executed |
| 6. RET | Returns to the job that called it (put this at the end of any & all called jobs) |
| 7. RET IF IN#(2) ON | Returns like above but only if #2 is on |
| 8. 'MDL:SEE WHEEL2 | Turns on visibility of model names "WHEEL2" |
| 9. 'MDL:HID WHEEL3 | Turns off visibility of model names "WHEEL3" |
| 10. 'MDL:SET M1 M2 | Sets the parent (NOT move parent) of M1 to M2 |
| 11. 'MDL:MOV M1 M2 | Changes M1 position by "move parent" M1 coordinates relative to its parent stay the same, but parent is changed to be M2 |
| 12. 'MDL:ADDX6 M1=1000,0,0,0,0,0 | Moves model named "M1" 1000mm in the positive X direction, (as if you opened the position dialogue box of the model that is being moved, M1 in this case, and changed the values) If this does not work on rotations, then put a dummy model as its parent to try and solidify the parental frame reference |
| 13. TIMER T=1 | Starts a timer of 1 second |
| 14. JUMP *1 IF IN#(3) ON | Jumps to *1 flag/line in the program if input# 3 is on |
| 15. JUMP *TOP IF B00<4 | Jumps to *TOP flag/line in the program if Binary variable is less than 4 |
| 16. *TOP | A flag/target line in the program to jump to |

This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.

17. JUMP LABEL: I01
Jumps to a label whose value is equal to the variable value (Variables available are I, D, R, B, P--for value ranges of types of variables see page 60 in Basic Programming manual except P is a position variable on pages 42-45 in Advanced manual)
18. SET I015 1
Sets Integer variable in address slot 015
19. SET B000 0
Sets Binary variable in address slot 000 to value 0
20. SET I006 I039+5
Sets value of Integer variable in address slot 006 to value in I006 plus 5
21. SET R004 D007/8
Sets value of Real variable in address slot 004 to value in D006 divided by 8
22. SET D032 2000*B005
Sets value of Double Precision variable in address 032 to B005 multiplied by 2000
23. INC B000
Increments the value of Binary variable B000 by 1 so that B000 now = B000 + 1
24. DEC B000
Decrements the value of Binary variable B000 by 1 so that B000 now = B000 - 1
25. ADD R015 -3424
Adds constant (i.e. -3424) to the Real variable value stored in address slot 015
26. ADD P000 P001
Adds P001 (X, Y, Z, Rx, Ry, Rz) values to P000 values (X, Y, Z, Rx, Ry, Rz)
27. SFTON P0000 UF#1
Using UF#1 for what X, Y, Z, Rx, Ry, Rz will be this shift all positions for programmed points by the values recorded in P0000 in the header of the *.jbi file
28. SFTON P0000 RF
Using the Robot Frame for what X, Y, Z, Rx, Ry, Rz will be (...same as above).
29. SFTOF
Turns positional shifting OFF
30. MOVJ P[I000] VJ=50.00
Position variable array addressing